A comprehensive guide to the physics and usage of Pythia 8.3

Christian Bierlich[1], Smita Chakraborty[1], Nishita Desai[2], Leif Gellersen[1],
Ilkka Helenius[3,4], Philip Ilten[5], Leif Lönnblad[1], Stephen Mrenna[6], Stefan Prestel[1],
Christian T. Preuss[7,8], Torbjörn Sjöstrand[1], Peter Skands[7], Marius Utheim[1,3]
and Rob Verheyen[9]

1 Dept. of Astronomy and Theoretical Physics, Lund University,
Sölvegatan 14A, S-223 62 Lund, Sweden
2 Tata Institute of Fundamental Research, Homi Bhabha Road, Mumbai 400005, India
3 University of Jyvaskyla, Department of Physics, P.O. Box 35,
FI-40014 University of Jyvaskyla, Finland
4 Helsinki Institute of Physics, P.O. Box 64, FI-00014 University of Helsinki, Finland
5 Dept. of Physics, University of Cincinnati, Cincinnati, OH 45221, USA
6 Fermilab, Batavia, Illinois, USA
7 School of Physics and Astronomy, Monash University, Wellington Rd,
Clayton VIC-3800, Australia
8 Institute for Theoretical Physics, ETH, CH-8093 Zürich, Switzerland
9 Dept. of Physics and Astronomy, UCL, Gower St, Bloomsbury,
London WC1E 6BT, United Kingdom

Abstract

This manual describes the Pythia 8.3 event generator, the most recent version of an evolving physics tool used to answer fundamental questions in particle physics. The program is most often used to generate high-energy-physics collision "events", *i.e.* sets of particles produced in association with the collision of two incoming high-energy particles, but has several uses beyond that. The guiding philosophy is to produce and reproduce properties of experimentally obtained collisions as accurately as possible. The program includes a wide ranges of reactions within and beyond the Standard Model, and extending to heavy ion physics. Emphasis is put on phenomena where strong interactions play a major role. The manual contains both pedagogical and practical components. All included physics models are described in enough detail to allow the user to obtain a cursory overview of used assumptions and approximations, enabling an informed evaluation of the program output. A number of the most central algorithms are described in enough detail that the main results of the program can be reproduced independently, allowing further development of existing models or the addition of new ones. Finally, a chapter dedicated fully to the user is included towards the end, providing pedagogical examples of standard use cases, and a detailed description of a number of external interfaces. The program code, the online manual, and the latest version of this print manual can be found on the Pythia web page:

https://www.pythia.org/

Contents

Part I

Introduction

This manual is organized into three major parts. This first part contains introductory material about event generators in general and the basic technical details of event generation. The second part presents a more detailed description of the physics implemented inside of PYTHIA. The physics is divided according to how it appears in the program flow itself, though the lines drawn can be fuzzy: the hard process (including external calculations); parton showering; multiparton interactions; beam remnants; and hadronization. There are also dedicated sections on the DIRE and VINCIA parton showers, as well as the treatment of heavy-ion collisions. Some of the details have not been thoroughly documented before, while others have appeared in prior publications. The third part is about how the user interacts with PYTHIA. In many applications, PYTHIA is part of a code stack or work flow, with other programs calling into PYTHIA or *vice versa*. This part describes both basic standalone usage and documents typical interfaces in detail.

1 Preliminaries

PYTHIA 8.3 [1] is a scientific code library that is widely used for the generation of events in high-energy collisions between particles, where effects of the strong nuclear force, governed by Quantum Chromodynamics (QCD), are of high importance. It is written mainly in C++ and interweaves a comprehensive set of detailed physics models for the evolution from a few-body hard-scattering process to a complex multi-particle final state. Parts of the physics have been rigorously derived from theory, while other parts are based on phenomenological models, with parameters to be determined from data. Currently, the largest user community comes from the Large Hadron Collider (LHC) experimental collaborations, but the program is also used for a multitude of other phenomenological or experimental studies in astro-, nuclear, and particle physics. Main tasks performed by the program include investigations of experimental consequences of theoretical hypotheses, interpretation of experimental data — including estimation of systematic uncertainties and unfolding — development of search strategies, and detector design and performance studies. It also plays an important role as a versatile vessel for exploring new theoretical ideas and new algorithmic approaches, ranging from minor user modifications to full-fledged developments of novel physics models.

1.1 What is an "event generator" ?

In particle physics, the outcome of a collision between two incoming particles, or of the isolated decay of a particle, is called an "event". At the most basic level, an event therefore consists of a number of outgoing particles such as might be recorded in a snapshot taken by an idealized detector, with conservation laws implying that the total summed energies and momenta of the final-state particles should match those of the initial state, as should any discrete quantum numbers that are conserved by the physics process(es) in question.

Due to the randomness of quantum processes, the number of outgoing particles and their properties vary from event to event. The probability distributions for these properties can be inferred by studying an ensemble of events in data. Conversely, given a set of theoretically calculated (or modelled) probability distributions, it is possible to produce ensembles of simulated events to compare to data.

A numerical algorithm that can produce (or "generate") random sequences of such simulated events, one after the other, is called an "event generator". The simulations can be based on known or hypothetical laws of nature. This allows for the exploration and comparison of competing paradigms, and studies of the sensitivity of proposed physical observables to the differences. Only rarely do the algorithms represent exact solutions however, so a common issue is to consider whether ansätze and approximations made, and the level of detail offered by a given modelling, are adequate for the problem at hand. The detailed physics descriptions contained in the main parts of this report are intended to assist with this task.

Returning to the structure of a high-energy physics event, in its crudest form, it is a list of the sub-atomic particles produced in a collision along with a measure of the probability for that event to occur. In PYTHIA, the list is referred to as the "event record", and it includes the four-momentum, production point, and many other properties of each particle, *cf.* section 9.4 for details. It typically also includes quite a bit of history information showing intermediate stages of the event modelling. The measure of the relative probability of a given event within a sample is given by the weight of that event relative to the sum of weights for the sample. For the typical case of unweighted events, this is just the inverse of the total number of events in the sample; cases that give rise to weighted events are summarized in section 9.8. The total cross section for the sample is also computed, allowing for the conversion of relative probabilities into cross sections.

Note that, although the starting point is often a relatively simple cross section computed in fixed-order perturbation theory, the total probability distribution for simulated events, fully differentially in all relevant phase-space variables and quantum numbers of the produced set of final-state particles, can typically not be expressed analytically. Instead, it is evaluated directly, using numerical methods, with Markov Chain Monte Carlo (MCMC) algorithms based on pseudo-random number generators as the main ingredient. The mathematical basis of the main ones used in PYTHIA is covered in section 2.2.

The aim of the event generator is ambitious: to predict all of the observable properties of a high-energy collision or decay process. The full properties of an event, however, cannot currently be calculated from first principles alone. Many different, complex phenomena, which are likely related, are described by a proliferation of models that each focus on a limited dynamical range. As a result, the predictions of an event generator like PYTHIA 8.3 depend upon $\mathcal{O}(100)$ parameters. The values of these parameters are inferred from comparisons to data. A collection of such parameter values is referred to as a *tune*.

Event-generator predictions are useful, because they serve as a proxy for what an event would look like before interacting with any measurement devices. As such, it can be used to investigate the consequences of new and old phenomena, and study the loss, mismeasurement, and misidentification of particles in experiments. Thus, it is an important tool for interpreting collider data. Event generators are realized as computer codes. In modern times, most of the larger projects are developed in the C++ programming language.

1.2 The structure of a simulated event

The main goal of PYTHIA is to simulate particle production in high-energy collisions over the full range of energy scales accessible to experiments, in as much detail as possible. However, hadron collisions and hadroproduction in particular are exceedingly complex, and no comprehensive theory exists currently that can predict event properties over this full range. For practical purposes, the wide range of phenomena are factored into a number of components. A natural division for these components is a time-ordering or, equivalently, an energy or transverse momentum ordering, where the best understood physics is calculated at the shortest time scales and largest energies, and the least understood physics is modelled at the longest time scales and lowest energies. This division is well motivated and often underpinned by factorization theorems, but it is not entirely unambiguous and sometimes is open to corrections.

The ordering in time is not completely intuitive, at least not in a directional sense from past to future. We should rather speak of time windows centred on a hard collision that then expand forwards and backwards in time, introducing successive phenomena, until we are left with a pair of incoming protons from accelerator beams, for example, and a number of outgoing particles. In momentum space, we normally speak of the "hardness" scale that characterizes each (sub)process, and often use a measure of transverse momentum $p_\perp$ to quantify this.

For simplicity, we will here concentrate on the sufficiently complex case of hadron-hadron collisions, with an explicit schematic of a fully simulated $pp \to t\bar{t}$ event given in fig. 1. The radial coordinate illustrates hardness scales, starting with the hardest subprocess near the centre (labelled $d\hat{\sigma}_0$), and ending with stable final-state particles and the incoming beam particles at the periphery.

In our hardness- or time-ordered picture, the components of a high-energy collision are:

1. A hard scattering of two partons, one from each incoming hadron, into a few outgoing particles. The initial partons are selected using parton distribution functions for the incoming hadrons, and the kinematics of the outgoing particles are based on matrix elements calculated in perturbation theory. Such calculations introduce a factorization scale and a renormalization scale. Partons with momenta below these scales are not
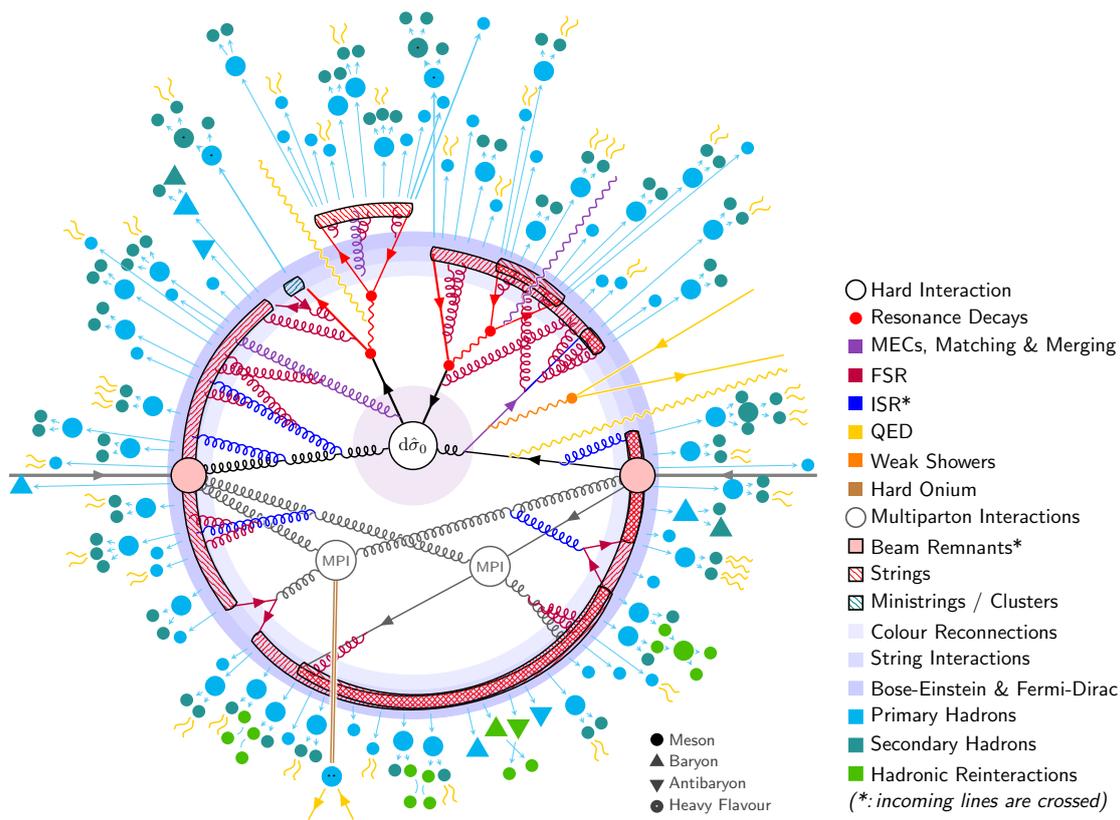
Figure 1: Schematic of the structure of a pp → t̄t event, as modelled by PYTHIA. To keep the layout relatively clean, a few minor simplifications have been made: 1) shower branchings and final-state hadrons are slightly less numerous than in real PYTHIA events, 2) recoil effects are not depicted accurately, 3) weak decays of light-flavour hadrons are not included (thus, *e.g.* a K_S^0 meson would be depicted as stable in this figure), and 4) incoming momenta are depicted as crossed ($p \to -p$). The latter means that the beam remnants and the pre- and post-branching incoming lines for ISR branchings should be interpreted with "reversed" momentum, directed outwards towards the periphery of the figure; this avoids beam remnants and outgoing ISR emissions having to criss-cross the central part of the diagram.

included in the hard scattering, but will be introduced by other stages of the event generation. In the current usage of PYTHIA, it is common to import the results of parton-level calculations from external packages, though a number of simple processes are calculated internally. Hard-scattering predictions depend on a few, universal input parameters that are determined from data, such as the value of the strong coupling at the Z boson mass and parton distribution functions.

2. The hard process may produce a set of short-lived resonances, such as Z or $W^\pm$ gauge bosons or top quarks, whose decay to normal particles has to be considered in close association with the hard process itself.

3. Fixed-order radiative corrections may be incorporated via (combinations of) matrix-element corrections, matching, and/or merging strategies, *cf.* section 5. In fig. 1, the violet shaded region surrounding the hard process represents the range of scales covered by a (generic) matrix-element merging strategy active above some given $p_{\perp\mathrm{min}}$ scale.

4. Initial–State Radiation (ISR) of additional particles (partons, photons, and others) starting from the scattering initiators using numerical resummation of soft and collinear gluon emission. This (together with its final-state equivalent below) is commonly referred to as the parton shower.

5. Final–State Radiation (FSR) of additional particles from the hard scattering itself and also from any resonance decays.

6. In competition with ISR and FSR, further scattering processes between additional partons from the incoming beams may take place, in a phenomenon known as Multiple Parton Interactions (MPI). This is not to be confused with "pileup", which generally refers to several distinct hadron-hadron collisions recorded in the same detector snapshot.

7. At some stage after the MPIs and perhaps before resonance decays, strings begin to form, as the non-perturbative limit of colour dipoles. These dipoles, however, are typically defined by *colour connections* that are assigned in the $N_c \to \infty$ limit, and are not unique for $N_c = 3$. As discussed further in section 7.2, the associated colour-space ambiguities can be modelled via Colour Reconnection (CR). It is also possible that long-range dynamical interactions could physically alter the colour flow and/or change the configuration of the expanding strings before they fragment. Depending on the characteristic timescales involved (often not specified explicitly in simple CR models), such effects may also be referred to as colour reconnections, but could also come under the rubric of string interactions.

8. The strong interaction now results in the confinement of QCD partons into colour-singlet subsystems known as strings or, in small-mass limiting cases, clusters. What is currently left of the incoming hadron constituents are combined into beam remnants. In fig. 1, the transition between the partonic and hadronic stages of the event generation is highlighted by the concentric annuli shaded blue.

9. The strings fragment into hadrons based on the Lund string model. Optionally, effects of overlapping strings may be taken into account, *e.g.* by collecting them into so-called "ropes" and/or allowing interactions between them.

10. Identical particles that are close in phase space may exhibit Bose-Einstein enhancements (for integer-spin particles) or Fermi-Dirac suppressions (for half-integer-spin particles).

11. Unstable hadrons produced in the fragmentation process decay into other particles until only stable particles remain (with some user flexibility to define what is stable).

12. In densely populated regions of phase space, the produced particles may rescatter, reannihilate, and/or recombine with one another.

The introduction of heavy-ion beams introduces an additional layer of complexity wrapped around this picture. Lepton-lepton collisions are much simpler, since they do not involve many of the complications arising from hadron beams.

1.3 To what types of problems can PYTHIA be applied ?

PYTHIA can be applied to a large set of phenomenological problems in particle physics, and to related problems in astro-particle, nuclear, and neutrino physics. Historically, the core of PYTHIA is the Lund string model of hadronization. This model is most appropriate when the invariant masses of the hadronizing systems are above 10 GeV or so. For lower-mass systems, the model is less firmly reliable. Low-mass systems may still occur in PYTHIA, typically then

as subsystems within a larger event, *e.g.*, produced by heavy-flavour decays, colour reconnections, and/or hadronic rescattering. For the very lowest-mass systems, which produce just one or two hadrons, a simple cluster-style model, called ministrings, is implemented, otherwise the normal string fragmentation is applied. In addition to string hadronization, PYTHIA of course also incorporates state-of-the-art models for a wide range of other particle-physics phenomena. Here, we provide a non-exclusive list of various applications of the PYTHIA machinery.

We emphasize that the majority of these models are based on dedicated original work done by authors, students, and sometimes external contributors, representing a significant and sustained intellectual effort. When quoting results obtained with PYTHIA, we therefore ask that users make an effort to cite, alongside this manual, such original works as would be deemed directly relevant to the study at hand, *i.e.* without whose implementation in PYTHIA the study could not have been done. Appropriate references can be found throughout the manual.

- Lepton-lepton, lepton-hadron, and hadron-hadron collisions with configurable beam properties, such as beam energies and crossing angles, to simulate one or many Standard-Model processes encoded in PYTHIA. This is the standard application of PYTHIA, but not the only one.

- The same as above, except using parton-level configurations for the hard process input from an external source.

- Ordinary particle decays, where the particles are produced by another physics program. This includes the limiting case of a particle gun (*i.e.* a single particle with user-defined momentum).

- Beyond Standard Model (BSM) particle decays, including decay chains.

- Resonance decays including the effects of final-state parton showering and hadronization.

- Hadronization of (colour-singlet) partonic configurations, as may arise from ordinary or exotic particle decays.

- Generation of Les Houches Event (LHE) formatted files from the internal hard processes for other physics studies.

- Ion-ion collisions for ion geometries well described with a Woods-Saxon potential (non-deformed, $A > 16$) for $\sqrt{s_{\mathrm{NN}}} > 10$ GeV.

- Astro-particle phenomena like dark-matter annihilation into Standard-Model particles.

- User-inspired modifications of standard PYTHIA modules as allowed by the `UserHooks` methods and those for semi-internal processes and/or semi-internal resonances.

As always, *caveat emptor*.

1.4 Historical evolution of the PYTHIA program

To bring some of the main development lines into context, we here provide a brief summary of the historical evolution of the PYTHIA program and its ancestor, JETSET. Detailed descriptions of the various physics components will be found in subsequent sections, including relevant references; a more elaborate review of the historical evolution of PYTHIA can be found in [2].

In the late seventies the Lund group began to study strong interactions, and notably the hadronization subsequent to a collision process. A linear confinement potential was assumed

to be realized by a string stretched out between a pulled-apart colour–anticolour pair, as a simple one-dimensional representation of a three-dimensional flux tube or vortex line. In order to allow detailed studies, two PhD students were entrusted to code up this model, and also include effects such as particle decays. This program was given the name JETSET. The model and code were gradually extended to encompass more physics, in particular with reference to e$^+$e$^-$ physics. The key addition was a model for e$^+$e$^- \rightarrow$ q$\bar{\text{q}}$g, wherein the colour field was assumed to stretch as one string piece from the q end to the g and then as a second piece on from the g to the $\bar{\text{q}}$ end, with no direct connection between the q and $\bar{\text{q}}$. This model received experimental support at PETRA in 1980 [3], thereby starting the success story of the Lund event generators. The idea of subdividing the full colour topology into a set of colour–anticolour dipoles rapidly prompted extensions also to other collision processes, notably to pp ones, with the PYTHIA generator built on top of JETSET. Later, it also came to develop into the dipole picture of parton showers, and to foreshadow related techniques for higher-order matrix-element calculations.

In part, the continued evolution was driven by interactions with the experimental communities and their priorities. An early involvement in SSC studies led to an extension of the scope of PYTHIA from QCD physics to encompass a wide selection of Standard Model (SM) processes, notably those related to Higgs-boson signatures. At the same time, QCD processes needed to be modelled better, which led to the development of new concepts, such as backwards evolution to handle initial-state radiation, and multiparton interactions and colour reconnection to describe underlying events and minimum-bias physics. When LHC physics studies began in 1990, these capabilities helped PYTHIA play a prominent role in benchmarking the evolving design of the LHC detectors, and additionally many Beyond-the-Standard-Model scenarios were included to cater to the demands of the community.

The Large Electron–Positron Collider (LEP) became the first operating collider where JETSET had been used from the early days of detector design, and the program came to play a key role in most physics analyses carried out there. QCD phenomena were a primary focus of experimental studies, and this led to an emphasis on issues such as parton-shower algorithms and matrix-element corrections to them. The ARIADNE dipole shower [4] offered a successful alternative to the more traditional internal JETSET one. With LEP 2, the emphasis shifted from QCD towards electroweak processes such as W$^+$W$^-$ pair production, which had already been incorporated into PYTHIA. This led, naturally, to the integration of the JETSET capabilities into PYTHIA, with PYTHIA maintaining the project name and legacy.

Also at HERA, the Lund-based programs came to play a prominent role from the onset, with codes such as LEPTO [5], ARIADNE and LDC [6] built on top of JETSET. Photon physics was introduced into PYTHIA to handle γp at HERA and $\gamma\gamma$ at LEP 2.

A further area of study is heavy-ion collisions, where early on the FRITIOF [7] program came to be widely used. Some of these ideas have been revived, updated, and implemented in the PYTHIA 8.3/ANGANTYR model. It is worth noting, also, that many heavy-ion collision models, used notably at the Relativistic Heavy Ion Collider (RHIC), have been based on PYTHIA.

The separation above, by collider, gives one way of describing the evolution of the code(s). Underlying it is a belief in universality, that many aspects of particle collisions are the same, independent of the beam type. Therefore, physics developments made in one context can also be applied to others. This is why one single code has found such widespread use.

The early codes were all written in FORTRAN 77. With the CERN decision to replace that language by C++ for LHC applications, PYTHIA underwent a similar transformation in 2004 – 2008. A new organizational structure was put in place for the new PYTHIA 8, in an attempt to clean up blemishes incurred during the years of rapid expansion, but deep down most of the physics algorithms survived in a new shape.

One area where the evolution has overtaken PYTHIA is that of matrix elements. Before it

was possible for most users to perform matrix-element calculations on computers, such expressions were published in articles and hard-coded from these. Now, with the physics demand for higher final-state multiplicities and higher-order perturbative accuracy, that is no longer feasible. For all but the simplest processes, we therefore rely on separate, external matrix-element codes to provide the hard interactions themselves, *e.g.* via the Les Houches interfaces, to which we then can add parton showers, underlying events, and hadronization. Also parton distribution functions are obtained externally, even if a few of the more commonly used ones are distributed with the code.

The program has continued to expand also after the transition to C++. Some developments are done from the onset within the PYTHIA code, such as the machinery for matching and merging between matrix elements and parton showers, or the PYTHIA 8.3/ANGANTYR framework for heavy-ion collisions, or the space–time picture of hadronization and hadronic rescattering. Other have come by the integration of externally developed packages, such as the VINCIA and DIRE alternatives to the existing simpler parton showers already in place.

In total, the JETSET/PYTHIA manuals have more than 35 000 citations by now, attesting to its widespread use. That use also includes possible future projects such as ILC, FCC, CLIC and EIC. The counting of code citations does not include the numerous articles describing the development and application of the physics content in the programs. This is harder to count, with many borderline cases, but the order of magnitude is comparable with the one for the code itself.

2   Program structure and basic algorithms

The PYTHIA 8.3 general-purpose Monte-Carlo event generator's structure reflects the different physics descriptions and models needed to generate fully exclusive final states as they can be detected at collider experiments. The first part of this section gives a brief overview of the program structure, while the latter parts describe basics of Monte Carlo (MC) techniques and process generation employed by PYTHIA 8.3.

2.1   Program structure and overview

Internally, PYTHIA 8.3 is structurally divided into three main parts: process level, parton level, and hadron level. This reflects the components of an event as introduced in section 1.1.

The process level represents the hard-scattering process, including the production of short-lived resonances. The hard process is typically described perturbatively, with a limited number of particles, typically at high-energy scales.

The parton level includes initial- and final-state radiation, where various shower models are available. Multiparton interactions are also included at this stage, along with the treatment of beam remnants and the possibility of the colour-reconnection phenomenon. At the end of the parton-level evolution, the event represents a realistic partonic structure, including jets and the description of the underlying event.

The hadron level then takes care of QCD confinement of partons into colour-singlet systems. In PYTHIA 8.3, the hadronization is described by QCD strings fragmenting into hadrons. Furthermore, other aspects like the decay of unstable hadrons and hadron rescattering are dealt with at the hadron level. The physics models of hadronization are typically non-perturbative, and thus require modelling and the tuning of parameters. The output of the hadron level is then a realistic event as it can be observed in a detector.

On top of this general structure, a significant number of shared objects and cross talk is passed between these levels: PDFs are relevant in both the process level and ISR, the matching and merging machinery works on the interface between parton showers and process level, and
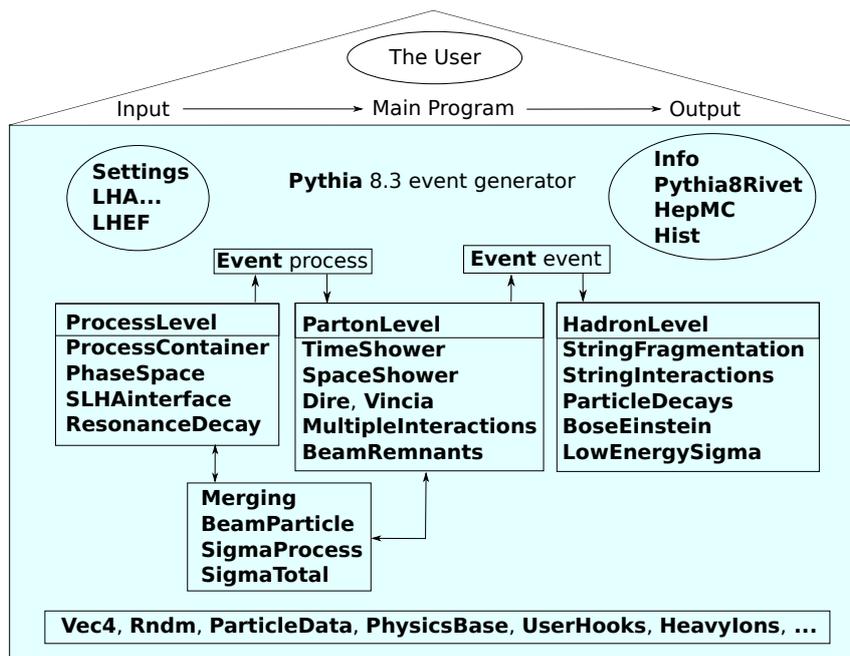
Figure 2: Simplified picture of the PYTHIA 8.3 structure, showing some of the important classes in bold. The main program itself creates one or more `Pythia` objects, and provides input in terms of `Settings` and potentially-perturbative event input. The main physics components are grouped into `ProcessLevel`, `PartonLevel`, and `HadronLevel`, with additional structure to complement and interconnect them.

the `Info` object is used throughout all levels to store and access central information. Under certain circumstances, like the analysis of heavy-ion collisions using the PYTHIA 8.3/ANGANTYR model, multiple parton-level objects can be used for separate subcollisions, which are then combined for hadronization.

From the user's perspective, PYTHIA 8.3 is a C++ library. The actual executable is implemented by the user, based on the requirements regarding input, output, features, and analysis, and many examples come with the PYTHIA 8.3 package. For detailed information on how to install and use PYTHIA 8.3, both standalone and with external interfaces, see part III. Figure 2 gives a rough overview of the PYTHIA 8.3 program structure.

2.2 Monte-Carlo techniques

Real events observed in particle colliders are stochastic. To emulate this, event generators sample from probability distributions using pseudo-random numbers. Naively, a pseudo-random number (between 0 and 1) is compared to a cumulative distribution function to determine an effect, *e.g.* the angle of a particle in a decay, the type of particle produced in hadronization, *etc.* Since real cases are rarely this simple, we use this section to describe some of the technical details of how pseudo-random numbers are used within the program.

2.2.1 Random-number generation

At the core of all Monte-Carlo methods lies the access to a random number generator. Truly random numbers require special equipment and are difficult to obtain at the required pace, so in practice pseudo-random numbers are used, where deterministic computer algorithms are used to emulate a random behaviour. This also allows a user of the code to reproduce a given event sample, simply by setting the same random-number seed. Nevertheless, the numbers

must appear to be random, *e.g.* evenly distributed between 0 and 1, have no detectable correlations, and have a long period before they start to repeat. Many pseudo-random number generators once thought to exhibit no internal correlations, have later been revealed to have flaws, so care is needed.

A review of several current generators is found in [8]. Common for them is that they can be viewed as having an N-dimensional state vector x, living in a N-dimensional hypercube with periodic boundary conditions such that each number is in the range between 0 and 1. A new state is obtained by a matrix multiplication $x_{i+1} = A \times x_i$, where A is a $N \times N$ matrix of integers. There is some sophisticated theory involved in the choice of A, involving concepts such as Kolmogorov–Anosov mixing and the Lyapunov exponent. Some of the key results are that A should have determinant unity, with complex eigenvalues away from the unit circle, and additionally that multiplication with it should require a minimal amount of operations so as to keep speed up.

The RANMAR default in PYTHIA is based on the Marsaglia–Zaman algorithm [9], but implemented in double precision with $N = 97$. There remains some tiny correlations 97 numbers apart, which could be fixed by multiplication by A several times between each set of 97 random numbers actually used [10], but this is not a necessity for event generators, where typically one is in a completely different part of the code 97 random numbers later. The RANMAR algorithm can be initialized to run one of more than 900 000 000 different sequences, each with a period of more than 10^{43}. By default, the same sequence is always run, which is useful for checks and debug purposes.

The MIXMAX alternative [11] is also provided as an option, and additionally there is an interface allowing the user to link in an external algorithm of choice.

2.2.2 Some standard techniques

The Monte Carlo techniques employed in PYTHIA can be generally categorized into two types. Both use pseudo-random numbers to select or reject a value for some variable based on a probability distribution. In the methods described here, the only consequence of rejecting a variable is its effect on the efficiency for generating the desired distribution. In the veto algorithm, discussed in the following section, the rejection step is a key element.

The task considered here is to select a value for a variable x from a known distribution function $f(x)$ such that the probability to select x in a small interval dx around a given x is $\propto f(x)\,dx$. We require $f(x)$ to be non-negative in the desired x range $x_{\min} \le x \le x_{\max}$.

In the simplest case, a primitive function $F(x)$ can be found with an inverse $F^{-1}(x)$ that can be calculated analytically. In this case, x can be sampled using the following algorithm:

$$\int_{x_{\min}}^{x} f(x)\,dx = R \int_{x_{\min}}^{x_{\max}} f(x)\,dx\,,$$
$$\Longrightarrow x = F^{-1}(F(x_{\min}) + R(F(x_{\max}) - F(x_{\min})))\,. \tag{1}$$

Here, and in the following, R denotes a (pseudo-)random number, evenly distributed between 0 and 1, obtained from the random number generator (*cf.* section 2.2.1). If a primitive function and its inverse can be obtained, using this method is desirable, as it is the most efficient generation method possible.

In most cases encountered, a primitive function or its inverse cannot be found. In this case, the simplest sampling method is known as hit-or-miss. It requires knowledge of a maximum $f_{\max}$ of $f(x)$ in the considered x range. The hit-or-miss method requires two numbers, x and

y, to be chosen according to:

$$x = x_{\min} + R_1 (x_{\max} - x_{\min}),$$
$$y = R_2 f_{\max}. \tag{2}$$

The process is repeated until a y fulfilling $y < f(x)$ (where x is the selected x) is selected. Equivalently, the selected x can be accepted with probability $f(x)/f_{\max}$, without the explicit construction of a y. The accepted x value is then distributed uniformly in the area below $f(x)$. On average, the acceptance probability for x (called the *efficiency*) is

$$\frac{\int f(x)\,\mathrm{d}x}{f_{\max}(x_{\max} - x_{\min})}. \tag{3}$$

This method can also be used to estimate the definite integral of $f(x)$ in the desired range:

$$\int_{x_{\min}}^{x_{\max}} f(x)\,\mathrm{d}x \approx (x_{\max} - x_{\min}) \frac{1}{n_{\mathrm{try}}} \sum_{i=1}^{n_{\mathrm{try}}} f(x_i), \tag{4}$$

where x_i runs over all x values tried, whether accepted or not. This is the principle of Monte Carlo integration. Note that there is no restriction on the dimensionality of x, so that multi-dimensional integrals can be handled. The error on the integral decreases like $1/\sqrt{n_{\mathrm{try}}}$. Other numerical integration methods usually converge faster in one dimension, but slower in higher dimensions. A variable transformation of x can be applied to increase the efficiency of the algorithm when an integrand has peaks or, in some cases, even to find an $f_{\max}$ to allow for the use of simple hit-or-miss.

The importance sampling method is related to the idea of variable transformation. This method is beneficial when a function $g(x)$ can be found that overestimates $f(x)$ ($f(x) \leq g(x)$) over the considered x range and has a primitive function $G(x)$ with an inverse $G^{-1}(x)$. Then the methods above can be combined:

$$x = G^{-1}(G(x_{\min}) + R_1 (G(x_{\max}) - G(x_{\min}))),$$
$$y = R_2 g(x). \tag{5}$$

This is repeated until a $y < f(x)$ is selected. Note that the first step selects (x, y) uniformly in the area below $g(x)$, whereas the second half is to accept those that also are below $f(x)$. Using an acceptance probability $f(x)/g(x)$ again removes the need to introduce an intermediate y.

In some cases, it is not possible to find a $g(x)$ that both covers all spikes of $f(x)$, and also has an invertible primitive function. In such cases, a multichannel sampling method [12] can be applied. Rather than relying on a single $g(x)$, the overestimating function is picked as a sum of non-negative functions with invertible primitives $g(x) = \sum_i g_i(x)$, where still $f(x) \leq g(x)$ over the considered interval. Multichannel sampling then extends on the importance-sampling prescription, by using the relative size of the integrals $I_i = \int g_i(x)\,\mathrm{d}x$ to each time pick a new g_i for the x selection in eq. (5). The y selection and the accept/reject works as before, since it is easy to see that the weighted usage of the different $g_i(x)$ adds up to $g(x)$.

In addition to the methods presented above, it should be noted that it is sometimes possible to apply a particularly ingenious transformation, to allow selection from an otherwise tricky distribution. The textbook example is the Box–Muller transformation, allowing sampling from a Gaussian. While a single Gaussian $\exp(-x^2)$ does not have a simple primitive or inverse, the product of two do, by transforming to plane-polar coordinates:

$$e^{-(x^2+y^2)}\,\mathrm{d}x\,\mathrm{d}y = e^{-r^2} r\,\mathrm{d}r\,\mathrm{d}\varphi \propto e^{-r^2}\,\mathrm{d}r^2\,\mathrm{d}\varphi, \tag{6}$$

which gives

$$x = \sqrt{-\ln R_1} \, \cos(2\pi R_2),$$
$$y = \sqrt{-\ln R_1} \, \sin(2\pi R_2), \tag{7}$$

i.e. two Gaussian-distributed numbers are obtained from two random ones. Another such trick is that a judicious choice of convolutions can be used to show that $f(x) = x^{n-1} e^{-x}/(n-1)!$ can be obtained by $x = -\sum_{i=1}^{n} \ln R_i = -\ln\left(\prod_{i=1}^{n} R_i\right)$.

2.2.3 The veto algorithm

A broad class of stochastic evolution algorithms, including ones describing radioactive decays, parton showers, and also PYTHIA's modelling of MPI, involve the generation of *ordered* sequences of state changes (transitions), where the ordering parameter is typically a measure of time and/or resolution scale.

For probability distributions (and/or domains) that are complicated to handle analytically, the veto algorithm offers a convenient and mathematically exact approach by which simple overestimates can be used instead of the original functions that are then reimprinted via a veto step. This circumvents the need for costly and delicate numerical integrations and root finding, and the overestimating functions and domains can be tailored to the problem at hand for maximum efficiency.

Before describing the algorithm itself, however, let us first clear up a point of semantics. In the context of parton showers, the veto algorithm is the main way by which Sudakov form factors (see below) and related quantities are calculated. One therefore occasionally sees the phrase "Sudakov veto algorithm", but this risks giving the mistaken impression that Sudakov invented the veto algorithm. To avoid this, the terms "veto algorithm" and "Sudakov (form) factor" are kept separate in this work, with the former referring to the broad numerical sampling method described in this section and the latter being an (important) example of a physical quantity that can be calculated with it.

Consider a stochastic process that is ordered in some measure of evolution scale. *E.g.* for nuclear decay, the ordering measure could be time (in the rest frame of the decaying nucleus), while for PYTHIA's evolution algorithms, which are formulated in momentum space, the ordering is normally done in a measure of transverse momentum, from high to low. This ensures that infrared and collinear divergences of the corresponding transition amplitudes are associated with vanishing resolution scales, or equivalently with asymptotically late times in the algorithmic sense.

Starting from a given initial value, u, for the evolution scale, the probability for the *next* transition (*e.g.* a nuclear decay, or a shower branching) to happen at a lower scale $t < u$, is given by

$$p(t|u) = f(t)\Pi(u,t), \tag{8}$$

where $f(t)$ is the probability (sometimes called the "naive" probability) for a transition to occur at the scale t under the implicit condition that the state still exists at t. The latter is made explicit by the survival probability, $\Pi(u,t) \in [0,1]$, which represents the probability that the state remains unchanged over the interval $[u,t]$. Analogously to nuclear decay, $\Pi(u,t)$ is given by a simple exponential of the integrated naive transition rate,

$$\Pi(u,t) = \exp\left(-\int_t^u \mathrm{d}\tau f(\tau)\right), \tag{9}$$

such that

$$p(t|u) = \frac{\partial \Pi(u,\tau)}{\partial \tau}\Big|_{\tau=t}. \tag{10}$$
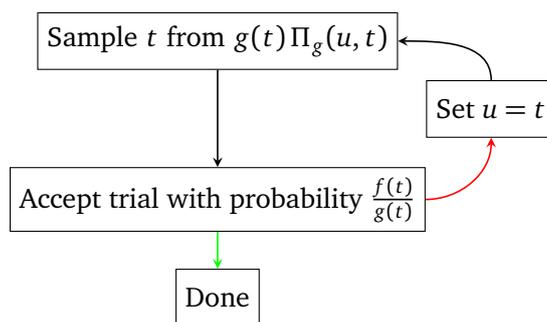
Figure 3: Flowchart representation of the veto algorithm. The red and green arrows refer to rejection and acceptance of the trial scale t respectively.

The survival probability $\Pi(u, t)$ is often referred to as the Sudakov form factor. We note that the two are only strictly identical for final-state showers, while for initial-state showers they are related via ratios of parton distribution functions, and in the context of MPI one can really only talk about a Sudakov-like factor. In this section, only the survival probability itself, which we denote by $\Pi(u, t)$, will be of interest.

It is worth pointing out that the ordered probability density $p(t|u)$ remains well-behaved and bounded by unity even if the integrated naive transition rate exceeds unity. In fact, due to the aforementioned collinear and soft singularities, $f(t)$ typically diverges for $t \to 0$, in which case the total probability for at least one-state change becomes

$$\int_0^u d\tau \, p(\tau|u) = 1 - \exp(F(0) - F(u)) \to 1. \tag{11}$$

That is, since $F(0) \to -\infty$ for a divergent kernel, the probability for at least one-state change simply saturates at unity. This reflects the unitarity of the shower algorithm, which is also manifest in eq. (10). If the naive probability does not diverge, or if the evolution is stopped at a finite cutoff $t_{\text{cut}} > 0$, then there is a non-zero probability, given by $\Pi(u, t_{\text{cut}})$, to have no state change at all.

Starting from eq. (8), probabilities for two or more ordered transitions can easily be constructed as well, *e.g.* for two successive branchings with $t < u < v$:

$$P(t|u|v) = f_2(t) f_1(u) \Pi_2(v, u) \Pi_1(u, t), \tag{12}$$

where $f_1(u)$ is the naive transition rate at the scale of the "first" transition, and that of the "second" transition is $f_2(t)$. Note that we do not assume $f_1 = f_2$ since the state undergoes a change at the intermediate scale $\tau = u$ (and the phase space is generally also different). This is also emphasized by the presence of two separate survival probabilities with different subscripts instead of a single combined $\Pi(v, t)$.

We now turn to how to actually sample from eq. (8). The branching kernel $f(t)$ is typically not simple enough to allow for the use of inversion sampling as described in the previous section. Fortunately, the veto algorithm [13–17] (and its antecedents, see the "thinning algorithm" [18, 19]) enables sampling from eq. (8) in a quite efficient and flexible manner. This algorithm relies on the existence of an overestimating "trial" function $g(t) \geq f(t)$ that is simple enough for samples to be drawn from eq. (8) directly, with f replaced by g. A flowchart representation of the veto algorithm in its simplest form is shown in fig. 3.

To confirm that this algorithm produces eq. (8), we follow along and write out its probability distribution $q(t|u)$ to find

$$q(t|u) = \int_0^u dt' \, g(t') \Pi_g(u, t) \left[\frac{f(t')}{g(t')} \delta(t - t') + \left(1 - \frac{f(t')}{g(t')}\right) p(t|t') \right], \tag{13}$$

where the first term describes the probability to accept the proposed trial scale t', and the second term gives the probability to reject the trial scale. Note that eq. (13) explicitly displays the Markovian nature of the veto algorithm, with every recursive step only depending on the previous one. Equation (13) may be solved by considering the differential equation

$$\frac{\partial}{\partial u} q(t|u) = f(u)\delta(t-u) - f(u)q(t|u), \tag{14}$$

which is found by application of Leibniz's rule for differentiation to eq. (13). We can find a solution by using an ansatz $q(t|u) = \hat{q}(t|u)e^{-F(u)}$, which after integration leads to

$$q(t|u) = f(t)\Pi_f(u,t)\Theta(t-\sigma) + q_0(t,\sigma). \tag{15}$$

The scale σ in the step function $\Theta(t-\sigma)$ and the function q_0 appear because information is lost in converting eq. (13) to eq. (14), but they are easily understood by reconsidering the structure of the algorithm. Mathematically, no other scale σ was introduced at any point, so eq. (15) cannot depend on it. As a result, σ must equal zero and the function q_0 must vanish, recovering eq. (8). In practice, however, the infrared cutoff on the shower evolution *does* introduce a scale σ. In that case, the algorithm shown in fig. 3 is stopped whenever t drops below σ. The function q_0 then represents the superfluous probability of sampling a scale below the cutoff, which is not associated with any change of state.

Many extensions of the veto algorithm are possible and are often used, of which we only discuss a few. Further details may be found in [13, 15–17, 20].

One can replace the acceptance probability $f(t)/g(t)$ by some other $r(t) \in [0,1]$ and compensate by modifying the event weight by a multiplicative factor $f(t)/g(t)r(t)$ in case the scale is accepted and $(1-f(t)/g(t))/(1-r(t))$ in case it is rejected. Writing out the probability distribution again, we find

$$q(t|u) = \int_0^u dt' g(t')\Pi_g(u,t')\left[r(t')\delta\left(t-t'\right)\frac{f(t')}{g(t')r(t')} + \left(1-r(t')\right)p\left(t|t'\right)\frac{1-f(t')/g(t')}{1-r(t')} \right], \tag{16}$$

where the weights appear as multiplicative factors. It is then straightforward to see that eq. (16) reduces to eq. (13). This modification enables sampling from eq. (8) in cases where it is difficult to find a $g(t) \geq f(t)$, or even in cases where $f(t)$ may be negative. However, in both cases events with negative weights will appear.

Applying eq. (16), shower uncertainties can be efficiently incorporated as event weights. In that case, $r(t)$ represents the *baseline* acceptance probability, while $f(t)$ is a modified branching kernel that parameterizes the uncertainties through variations of the renormalization scale, its non-singular components, or choice of parton distribution function for initial state showers. If $g(t)$ overestimates both the baseline and the modified branching kernels, the event weights stay positive. This is also the basis for generating biased emissions of rare splittings. More details can be found in section 4.1.5.

We complete this section by discussing some variations of the veto algorithm in the context of competition between channels. In most cases, multiple branching kernels $f_i(t)$ contribute to the total parton-shower probability distribution, which may then be written as

$$\tilde{p}(t|u) = \tilde{f}(t)\Pi_{\tilde{f}}(t,u) \text{ where } \tilde{f}(t) = \sum_{i=1}^{n} f_i(t). \tag{17}$$

One way to handle competition is to apply the veto algorithm to all channels individually, and then select the channel with the highest scale t_i. A flowchart representation of this procedure is shown in fig. 4.
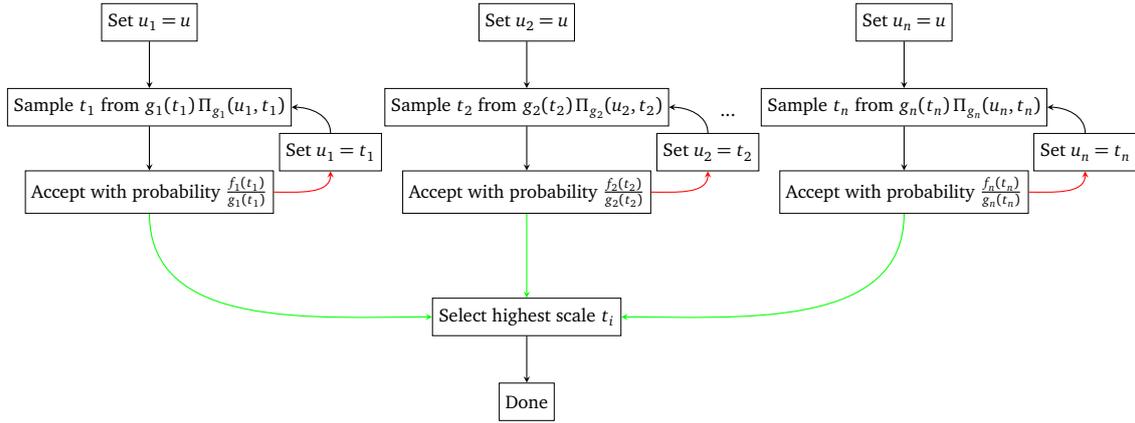
Figure 4: Flowchart representation of the first competition veto algorithm.
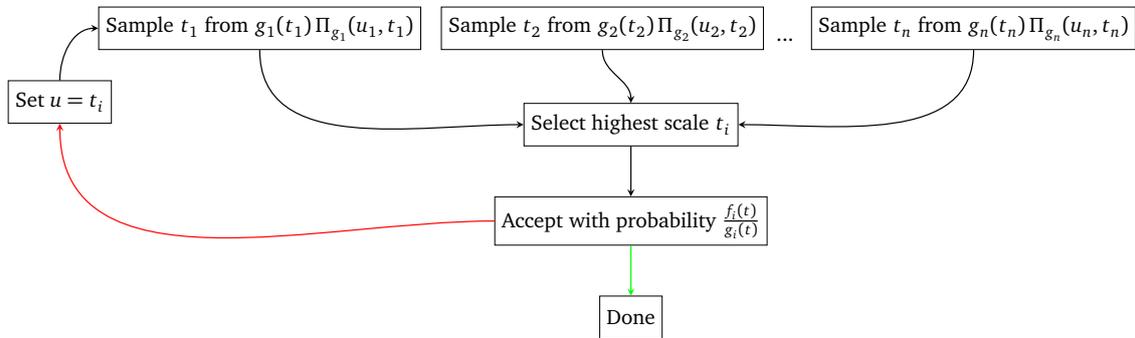
Figure 5: Flowchart representation of the second-competition veto algorithm. This algorithm is used to to interleave initial-state and final-state radiation with multiple parton interactions, and is more efficient when branching-kernel evaluation is expensive, such as for matrix-element corrections.

It may be shown to yield eq. (17) as follows:

$$
\left[\prod_{i=1}^{n} \int_{0}^{u} dt_i f_i(t_i) \Pi_{f_i}(u, t_i)\right] \sum_{j=1}^{n} \left[\prod_{k \neq j} \Theta\left(t_j - t_k\right)\right] \delta\left(t - t_j\right)
$$

$$
= \sum_{i=1}^{n} \left[\prod_{j \neq i} \int_{0}^{t_i} dt_j f(t_j) \Pi_{f_j}(u, t_j)\right] \int_{0}^{u} dt_i f_i(t_i) \Pi_{f_i}(u, t_i) \delta\left(t - t_i\right)
$$

$$
= \sum_{i=1}^{n} f_i(t) \Pi_{f_i}(u, t) \prod_{j \neq i} \Pi_{f_j}(u, t) = \tilde{p}(t|u). \tag{18}
$$

Equivalently, the result of eq. (18) may be used with overestimates $g_i(t_i)$ in place of $f_i(t_i)$, then selecting the highest scale before proceeding to the acceptance step. A flowchart representation of this procedure is shown in fig. 5, and produces the same result. This algorithm is used to interleave initial-state and final-state radiation with multiple parton interactions. Furthermore, it is more efficient when branching-kernel evaluation is expensive, such as for matrix-element corrections. A third option is available, where instead a single scale is drawn according to the sum of overestimates $\tilde{g}(t)$ and a channel is selected with probability $g_i(t)/\tilde{g}(t)$ for the acceptance step. A flowchart representation of this procedure is shown in fig. 6 and again produces eq. (17). This procedure is used in a few specific places, in particular when the overestimates of several channels are very similar. Examples include quark-flavour
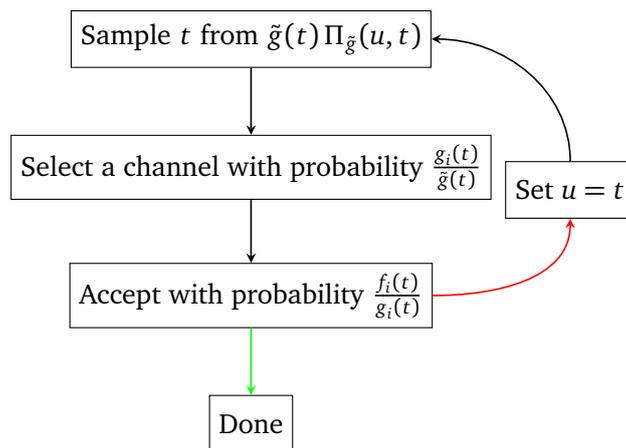
Figure 6: Flowchart representation of the third-competition veto algorithm. Useful in situations where multiple channels have similar overestimates, like for quark-flavour selection in $g \to q\bar{q}$ splittings and the efficient sampling of the large number of branchings in VINCIA's electroweak shower.

selection in $g \to q\bar{q}$ splittings and the efficient sampling of the large number of branchings in VINCIA's Electroweak (EW) shower. It is important to note that these algorithms may also be combined, for instance by grouping several channels for use with the algorithm depicted in fig. 6, and further combining them with the algorithms depicted in fig. 4 or fig. 5. In fact, the different shower models available in PYTHIA often use different procedures to optimize code structure and performance.

2.2.4 Phase space (M-generator and RAMBO)

One standard task is to distribute the momenta of final-state particles uniformly according to Lorentz Invariant Phase Space (LIPS), on top of which then later dynamical aspects can be added. (Non-uniform sampling methods, used *e.g.* when resonances are present, are discussed separately, in section 2.3.) The relevant phase-space density is

$$\mathrm{d}\Phi_n(P; p_1, p_2, \ldots, p_n) = (2\pi)^4 \, \delta^{(4)}\left(P - \sum_{i=1}^{n} p_i \right) \prod_{i=1}^{n} \frac{\mathrm{d}^3 p_i}{(2\pi)^3 \, 2p_i^0} \, , \tag{19}$$

where P is the total four-momentum, and $p_i, i \geq 1$ are the n different outgoing four-momenta. Usually the process is initially considered in the rest frame of the system, $P = (M; \mathbf{0})$, and later boosted to the relevant frame of the whole event.

A common special case is two-body final states, where the Centre of Mass (CM)-frame expression reduces to

$$\mathrm{d}\Phi_2 = \frac{|\mathbf{p}|}{16\pi^2 M} \, \mathrm{d}\Omega = \frac{|\mathbf{p}|}{16\pi^2 M} \, \mathrm{d}\cos\theta \, \mathrm{d}\varphi \, . \tag{20}$$

That is, the direction of one of the outgoing particles has to be picked uniformly on the unit sphere, with the other moving out in the opposite direction. The three-momentum length is

$$|\mathbf{p}| = |\mathbf{p}_1| = |\mathbf{p}_2| = \frac{\sqrt{\lambda\left(M^2, m_1^2, m_2^2\right)}}{2M} \, , \tag{21}$$

where the Källén λ function can be written in a number of equivalent ways

$$
\begin{aligned}
\lambda(a^2, b^2, c^2) &= a^4 + b^4 + c^4 - 2a^2 b^2 - 2a^2 c^2 - 2b^2 c^2 \\
&= (a^2 - b^2 - c^2)^2 - 4b^2 c^2 \\
&= (a^2 - (b+c)^2)(a^2 - (b-c)^2) \\
&= (a+b+c)(a-b-c)(a-b+c)(a+b-c).
\end{aligned}
\tag{22}
$$

The energies are given by

$$
\begin{aligned}
p_1^0 &= \sqrt{m_1^2 + \mathbf{p}^2} = \frac{M^2 + m_1^2 - m_2^2}{2M}, \\
p_2^0 &= \sqrt{m_2^2 + \mathbf{p}^2} = \frac{M^2 + m_2^2 - m_1^2}{2M}.
\end{aligned}
\tag{23}
$$

For three or more final particles, PYTHIA implements two different generic methods, the older M-generator [21] and the newer RAMBO [22] one, but approaches tailor made for the specific situation are also common, *e.g.* in parton showers. RAMBO is the best choice for the case of massless products, whereas the situation is less obvious once the masses constitute a significant fraction of the full energy.

The basic idea of the M-generator strategy is to view the full event as arising from a sequence of fictitious two-body decays. Thus a four-body decay $0 \to 1+2+3+4$, as an example, is viewed as a sequence $0 \to 123+4 \to 12+3+4 \to 1+2+3+4$, where 123 and 12 represent intermediate states. By a suitable insertion of a unit factor

$$
1 = \delta^{(4)}(p_{12} - p_1 - p_2)\, d^4 p_{12}\, \delta(m_{12}^2 - p_{12}^2)\, dm_{12}^2 = \delta^{(4)}(p_{12} - p_1 - p_2)\, \frac{d^3 p_{12}}{2p_{12}^0}\, dm_{12}^2,
\tag{24}
$$

and a similar one for p_{123}, the four-body phase space can be reformulated as

$$
d\Phi_4\left(P; p_1, p_2, p_3, p_4\right) \propto d\Phi_2\left(P; p_{123}, p_4\right) dm_{123}^2\, d\Phi_2\left(p_{123}; p_{12}, p_3\right) dm_{12}^2\, d\Phi_2\left(p_{12}; p_1, p_2\right).
\tag{25}
$$

The mass-dependent parts can be collected and simplified to

$$
\frac{\sqrt{\lambda(m_0^2, m_{123}^2, m_4^2)}}{m_0}\, dm_{123}\, \frac{\sqrt{\lambda(m_{123}^2, m_{12}^2, m_3^2)}}{m_{123}}\, dm_{12}\, \frac{\sqrt{\lambda(m_{12}^2, m_1^2, m_2^2)}}{m_{12}}.
\tag{26}
$$

The (m_{123}, m_{12}) phase space can easily be sampled within allowed borders, but the rest of the expression then becomes a weight that has to be taken into account by hit-and-miss Monte Carlo. This is where the algorithm can be slow. Once the intermediate masses have been selected, two-particle kinematics are constructed in a sequence of rest frames for $1+2$, $12+3$ and $123+4$, interleaved with Lorentz boosts between them.

The RAMBO algorithm provides an alternative sampling of n-body phase space, which by construction has constant (uniform) weight for arbitrary n in the massless limit. The starting point is the following identity for massless four-vectors,

$$
\int d^4 q\, \delta(q^2)\, \exp(-q^0) = \int_0^\infty \frac{q^0}{2}\, \exp(-q^0) \int d\Omega = 2\pi.
\tag{27}
$$

A four-momentum q distributed according to the integrand of the left-hand side of this identity can be generated via the steps

$$
q^0 = -\log(R_1 R_2), \quad \cos\theta = 2R_3 - 1, \quad \varphi = 2\pi R_4,
\tag{28}
$$

$$\implies q = (q^0, q^0 \sin\theta \sin\varphi, q^0 \sin\theta \cos\varphi, q^0 \cos\theta). \tag{29}$$

RAMBO repeats this process n times to produce a set of momenta q_i that initially have $\sum_i q_i^\mu \equiv Q^\mu$. The final momenta p_i are then constructed by applying a boost Λ^μ_ν to the CM frame of Q and scaling by an overall factor $x = M/\sqrt{Q^2}$, so that $P^\mu \equiv \sum p_i^\mu = x(\Lambda^\mu_\nu Q^\nu) = (M, \mathbf{0})$.

To illustrate that this leads to momenta distributed according to eq. (19), we may start from n multiples of eq. (27) and unitarily transform the momenta as

$$(2\pi)^n = \prod_{i=1}^n \int d^4 q_i\, \delta(q_i^2) \exp(-q_i^0)$$

$$\times d^4 Q\, \delta^4\left(Q - \sum_{i=1}^n q_i\right) dx\, \delta\left(x - \frac{M}{\sqrt{Q^2}}\right)$$

$$\times \prod_{i=1}^n d^4 p_i\, \delta^4(p_i - x(\Lambda q_i)). \tag{30}$$

Identifying the phase-space measure eq. (19) in eq. (30) and integrating over the other variables then leads to

$$\int d\Phi_n(P; p_1, p_2, \ldots, p_n) = \left(\frac{\pi}{2}\right)^{n-1} \frac{M^{2n-4}}{(n-1)!(n-2)!}, \tag{31}$$

which is indeed the n-body massless phase-space volume [22]. RAMBO thus samples the massless phase space isotropically, with constant weight given by eq. (31).

For massive particles, no equivalent general expression for the phase-space volume exists. However, the massless RAMBO algorithm may be adapted to the massive case at the cost of introducing variable event weights, which translates to a reduced efficiency at the unweighted level. Starting from the massless momenta p_i, massive momenta k_i are obtained through

$$\mathbf{k}_i = y\mathbf{p}_i, \, k_i^0 = \sqrt{|\mathbf{k}_i|^2 + m_i^2}. \tag{32}$$

The momenta k_i are on-shell and preserve momentum conservation as long as the rescaling parameter y is given by the solution of the equation

$$\sum_{i=1}^n \sqrt{y^2 |\mathbf{p}_i|^2 + m_i^2} = M. \tag{33}$$

Since eq. (33) is a monotonic function of y with a solution $0 \le y \le 1$, the value of y may be determined easily using the Newton–Raphson method. Through a similar procedure as the one followed in eq. (30), the event weight may be determined to be

$$w = \left(\frac{\pi}{2}\right)^{n-1} \frac{1}{(n-1)!(n-2)!} \left(\prod_{i=1}^n \frac{|\mathbf{k}_i|}{k_i^0}\right) \left(\sum_{j=1}^n \frac{|\mathbf{k}_j|^2}{k_j^0}\right)^{-1} \left(\sum_{j=1}^n |\mathbf{k}_j|\right)^{2n-3}, \tag{34}$$

which is bounded from above by the massless weight, eq. (31), so that the distribution can be unweighted by accepting the generated massive phase-space point with the probability

$$P_{\text{accept}} = \prod_{i=1}^n \frac{|\mathbf{k}_i|}{k_i^0} \overbrace{\left(\frac{\sum_j |\mathbf{k}_j|}{\sum_j \frac{|\mathbf{k}_j|^2}{k_j^0}}\right)}^{<1} \overbrace{\left(\frac{\sum_j |\mathbf{k}_j|}{M}\right)^{2n-4}}^{<1}, \tag{35}$$

which (by construction) tends to unity in the massless limit.

2.3 Process-generation basics

Particle-physics cross sections can crudely be divided into two categories: perturbative and non-perturbative. Both kinds of processes play crucial roles in PYTHIA. The former can be computed order by order in perturbation theory, *e.g.* based on Feynman-diagram rules. For the electroweak sector, the couplings are sufficiently small that higher-order corrections should offer a rapidly converging series. The exception is the enhanced emission of soft or collinear photons, but this is a well-understood issue. For the strong sector, on the other hand, the large α_s coupling leads to a slower convergence. It can still work well for QCD processes involving large momentum transfers. In the opposite limit, at low momentum transfers, the perturbative coupling diverges and perturbation theory breaks down. Therefore the total cross section in hadron-hadron collisions, which is dominated by such low scales, can only be described in terms of effective, phenomenological models. The same applies for its main components — elastic, diffractive and nondiffractive cross sections — which therefore are classified as non-perturbative processes.

In the current section, the focus will be on perturbative processes, introducing how these are defined and generated inside PYTHIA. Non-perturbative processes are discussed separately in section 6. There are also components that partly bridge the gap between the two, such as multiparton interactions (MPIs), hard diffraction, and photoproduction processes. These are also discussed in section 6, along with further aspects specific to simulating cross sections in heavy-ion collisions.

To begin the discussion of perturbative process generation, consider a process $a + b \to f_n$, where a and b are two incoming particles that together create a final state f consisting of n particles. The differential cross section can then be written as

$$\frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}\Phi_n} = \frac{|\mathcal{M}|^2}{2\sqrt{\lambda(\hat{s}, m_a^2, m_b^2)}} \approx \frac{|\mathcal{M}|^2}{2\hat{s}}, \tag{36}$$

where $\hat{s} = (p_a + p_b)^2$ is the squared invariant mass of the collision system. Usually m_a and m_b are negligible in comparison with $\sqrt{\hat{s}}$, and then the last expression is obtained. The process-specific physics is encapsulated in the matrix element $\mathcal{M}$, which we shall assume can be calculated perturbatively. The $|\mathcal{M}|^2$ expression also has to be averaged over incoming spin and colour configurations, and summed over outgoing spin and colour configurations, where relevant.

In some rare cases a and b are the actual incoming beam particles. Normally, however, a and b are constituents of the true beam particles, A and B. Then one needs to introduce parton distribution functions (PDFs), $f_a^A(x, Q^2)$ (and $f_b^B(x, Q^2)$), that to leading order describe the probability to find a parton a inside the particle A, with a fraction x of the particle four-momentum, if the hard-collision process probes the particle at a (factorization) scale Q^2. The cross section then reads

$$\sigma = \int \mathrm{d}x_1 f_a^A(x_1, Q^2) \int \mathrm{d}x_2 f_b^B(x_2, Q^2) \int \frac{\mathrm{d}\hat{\sigma}(\hat{s}, Q^2)}{\mathrm{d}\Phi_n} \mathrm{d}\Phi_n, \tag{37}$$

where

$$\hat{s} = x_1 x_2 s, \quad \text{with} \ s = (p_A + p_B)^2. \tag{38}$$

The nature of the PDFs varies depending on what kind of particle is concerned: hadrons, nuclei, leptons, photons, or pomerons. They will therefore be discussed further in the respective beam context. The most commonly used and best studied are the proton PDFs, *cf.* section 3.12, and for these we will omit the A and B superscripts.

2.3.1 $2 \to 2$ processes

Massless Kinematics: In a massless $2 \to 2$ subprocess $a(p_1) + b(p_2) \to c(p_3) + d(p_4)$ it is conventional to write the cross section in terms of the Mandelstam variables

$$\hat{s} = (p_1 + p_2)^2 = (p_3 + p_4)^2, \tag{39}$$

$$\hat{t} = (p_1 - p_3)^2 = (p_2 - p_4)^2 = -\frac{\hat{s}}{2}(1 - \cos\hat{\theta}), \tag{40}$$

$$\hat{u} = (p_1 - p_4)^2 = (p_2 - p_3)^2 = -\frac{\hat{s}}{2}(1 + \cos\hat{\theta}), \tag{41}$$

where $\hat{\theta}$ is the scattering angle, defined as the polar angle of particle 3, in the rest frame of the collision. Since $d\Phi_2 \propto d\cos\hat{\theta} \propto d\hat{t}$ (assuming a trivial flat φ dependence, as is the case unless the incoming beams are transversely polarized), it is common to recast eq. (37) accordingly. Furthermore, $dx_1\,dx_2 = d\tau\,dy$, where $\tau = x_1 x_2 = \hat{s}/s$ and $y = (1/2)\ln(x_1/x_2)$. It is also standard to use $xf(x)$ rather than $f(x)$. In total this gives

$$\sigma = \iiint \frac{d\tau}{\tau} dy\, d\hat{t}\, x_1 f_a^A(x_1, Q^2)\, x_2 f_b^B(x_2, Q^2) \frac{d\hat{\sigma}(\hat{s}, \hat{t}, Q^2)}{d\hat{t}}. \tag{42}$$

The $\hat{u}$ variable is redundant since $\hat{s} + \hat{t} + \hat{u} = 0$, but often symmetry properties of matrix elements are apparent if it is used judiciously. In a frame where a and b come in back-to-back, moving in the $\pm z$ directions, $\hat{p}_\perp^2 = \hat{t}\hat{u}/\hat{s}$ is the squared transverse momentum of the outgoing c and d. A frequent choice is to put $Q^2 = \hat{p}_\perp$ as the factorization scale.

The sampling ranges for each of the $(\tau, y, \hat{t})$ variables depends on whether phase-space cuts are imposed at the process-generation level, *cf.* section 3.13. Generically, they are:

$$\frac{\hat{s}_{\min}}{s} < \tau \qquad < \frac{\hat{s}_{\max}}{s}, \tag{43}$$

$$-\frac{1}{2}|\ln\tau| < y(\tau) < \frac{1}{2}|\ln\tau|, \tag{44}$$

$$\sqrt{1 - \frac{4\hat{p}_{\perp\max}^2}{\tau s}} < |z(\tau)| < \sqrt{1 - \frac{4\hat{p}_{\perp\min}^2}{\tau s}}, \tag{45}$$

where $\hat{t}$ has been replaced by $z = \cos\hat{\theta}$ via eq. (40), and we emphasize that there are solutions for both positive and negative z. The phase-space boundaries are set via the (user-specifiable) parameters $\hat{m}_{\min,\max}$, $\hat{p}_{\perp\min,\max}$, and/or $\hat{Q}_{\min}^2$, *cf.* section 3.13. To give some examples:

- For processes containing an s-channel resonance, it may be desirable to only generate phase-space points within a specific range of $\hat{m}$ values. Processes involving resonance production and decay are discussed in more detail in section 2.3.3.

- A restriction like $\hat{p}_\perp > \hat{p}_{\perp\min}$, implying $\hat{s}_{\min} = \max(\hat{m}_{\min}^2, 4\hat{p}_{\perp\min}^2)$, is mandatory for matrix elements that diverge in the $\hat{p}_\perp \to 0$ limit; this includes in particular massless t-channel QCD processes. It may also be convenient for studies focusing on the high-$p_\perp$ tail of "hard" $2 \to 2$ processes.

- The option to specify a $Q_{\min}^2$ value is intended for t-channel DIS-type processes with distinguishable final-state particles, *cf.* section 3.13, in which case $\hat{s}_{\min} \geq Q_{\min}^2$ and $z(\tau) \leq 1 - 2Q_{\min}^2/(\tau s)$.

The selection of phase-space points (τ, y, z) is described in detail in [14], and remains unchanged. The basic strategy is to use multichannel sampling in each of the three variables

separately. Thus, for instance, the τ dependence is modelled as a mix of sampling according to either $1/\tau$ or $1/\tau^2$. The normalization factors of the two possibilities are determined at initialization, and would depend on the process, the choice of PDFs and α_s, and the $p_{\perp\min}$ cut. That way an upper envelope is found for the real cross-section expression. The probability that a trial phase-space point is retained is given by the ratio of the full differential cross section to the multichannel overestimate, and the accepted events are assigned a standard weight of unity. There is always the risk that the intended upper estimate of the cross section is exceeded by the full expression in some corner of phase space, even if it is not common. Such points are associated with a weight correspondingly above unity.

The cross section for a process is obtained in parallel with the generation of events, using the multidimensional generalization of eq. (4). Thus the error decreases with the number of events generated.

When several processes are to be generated simultaneously, an upper envelope is found for each differential cross section separately. The size of integrated envelopes, *i.e.* the upper estimate of the respective cross sections, is used as a relative weight when the next process is selected. If the trial phase-space point is rejected then a new process choice is made. That is, a larger overestimate will make a given process more likely to be picked, but then afterwards also more likely to be rejected. In the end, all processes are generated in proportion to their correct integrated cross sections.

Generation in (τ, y, z) is only one possible choice. It has the advantage that additional τ terms can be used for the sampling of resonances in the cross section, *cf.* section 2.3.3. For the generation of MPIs, however, it is essential to use $\hat{p}_\perp^2$ rather than $\hat{t}$, *cf.* section 6.2. Then one may instead note that

$$\frac{dx_1}{x_1}\frac{dx_2}{x_2}\,d\hat{t} = \frac{d\tau}{\tau}\,dy\,d\hat{t} = dy_3\,dy_4\,d\hat{p}_\perp^2\,, \tag{46}$$

where y_3 and y_4 are the rapidities of the two outgoing particles.

Massive Kinematics: So far we have considered massless kinematics. It is quite common to have cases where one or both of the outgoing particles are massive, while the incoming ones still are assumed massless. In some cases, such as elastic scattering, both incoming and outgoing masses need to be taken into account. The fully general $\hat{t}$ expression is

$$\hat{t} = -\frac{\hat{s}^2 - \hat{s}(m_1^2 + m_2^2 + m_3^3 + m_4^2) + (m_1^2 - m_2^2)(m_3^2 - m_4^2) - \sqrt{\lambda(\hat{s}, m_1^2, m_2^2)\,\lambda(\hat{s}, m_3^2, m_4^2)}\,\cos\hat{\theta}}{2\hat{s}}\,, \tag{47}$$

with $\hat{u}$ obtained by $m_3^2 \leftrightarrow m_4^2$ and $\cos\hat{\theta} \to -\cos\hat{\theta}$, resulting in

$$\hat{s} + \hat{t} + \hat{u} = m_1^2 + m_2^2 + m_3^3 + m_4^2\,. \tag{48}$$

The limits $\hat{t}_{\min} < \hat{t} < \hat{t}_{\max}$ (all negative or, for $\hat{t}_{\max}$, zero) are obtained for $\cos\hat{\theta} = \mp 1$. Often $\hat{t}_{\max}$ is close to zero and a numerically safer recipe for it is obtained by noting that

$$\hat{t}_{\min}\,\hat{t}_{\max} = (m_3^2 - m_1^2)(m_4^2 - m_2^2) + \frac{(m_1^2 + m_4^2 - m_2^2 - m_3^2)(m_1^2 m_4^2 - m_2^2 m_3^2)}{\hat{s}}\,. \tag{49}$$

If $m_1^2 = m_2^2 = 0$ then $\hat{t}_{\min}\,\hat{t}_{\max} = m_3^2 m_4^2$ and $\hat{p}_\perp^2 = (\hat{t}\hat{u} - m_3^2 m_4^2)/\hat{s}$.

2.3.2 2 → 3 processes

In pure s-channel $2 \to 3$ processes, say (unpolarized) $e^+e^- \to \gamma^*/Z \to q\bar{q}g$, cross sections factorize into production and decay steps, and the decay phase space is easy to generate in terms

of two energy variables and three angles. Such decays are not coded as explicit hard processes, however, but instead are handled during the parton-level shower evolution. Three-body final states such as $e^+e^- \to \gamma^*/Z \to q\bar{q}g$ are then reached via showering from $e^+e^- \to \gamma^*/Z \to q\bar{q}$ (*cf.* section 2.3.3 on resonances and section 4 on parton showers), with matrix-element corrections applied to the extent available and switched on, *cf.* section 5.

For $2 \to 3$ hard processes that do not factorize into resonance production and decay plus shower, it becomes much more messy to set up phase space, since there are more possibilities for peaks in different places. PYTHIA does not have a general-purpose machinery to handle generic cross sections. Instead, the main assumption is that such processes are provided via the Les Houches accord, *cf.* section 10.1.1, from external programs that have their own phase-space generators.

There are a few internal $2 \to 3$ processes, however, for very specific tasks. These are generated according to one of three different prescriptions, tailored to the squared amplitudes for massless QCD $2 \to 3$ processes, Vector Boson Fusion (VBF), and central diffractive processes, respectively. These were developed separately and employ somewhat different notation in the code, here relabelled for clarity. Note that all three assume a cylindrical symmetry with respect to the collision axis.

Massless QCD $a(p_1)+b(p_2) \to c(p_3)+d(p_4)+e(p_5)$ cross sections contain divergences when any of the final-state particles become collinear to the beam, collinear to each other, and/or soft. It is therefore important to choose a set of phase-space variables that allows for the isolation of these singularities. The parameterization used in PYTHIA is $(y_3, y_4, y_5, p_{\perp 3}^2, p_{\perp 4}^2, \varphi_3, \varphi_4)$. The rapidity sampling here is simple and consistent, while the $p_\perp$ selection is not, unfortunately. The $p_{\perp 5}$ is fixed opposite to the vector sum of the other two, and in the first instance gets a different $p_\perp$ spectrum than them. Notably, a requirement for all $p_\perp > p_{\perp \min}$ can be imposed with full efficiency for two, but is inefficient for the third. It is also important to avoid the collinear singularity by an additional cut on $R = \sqrt{(\Delta y)^2 + (\Delta \varphi)^2}$ for all outgoing pairs. A process of special interest is vector-boson fusion to a Higgs boson, $W^+W^- \to H$ and $ZZ \to H$ (and/or $W^+W^+ \to H^{++}$ in some BSM scenarios). Since the bosons are emitted from fermion lines this results in $2 \to 3$ processes of the character $f_1(p_1) + f_2(p_2) \to f_3(p_3) + f_4(p_4) + H(p_5)$. The variables chosen in this case are $(\tau, y, y_5, p_{\perp 3}^2, p_{\perp 4}^2, \varphi_3, \varphi_4)$. Here, special care is taken in the modelling of $p_{\perp 3}^2$ and $p_{\perp 4}^2$ which, unlike the QCD cross sections, have no $p_\perp \to 0$ divergence but instead are fairly flat out to the gauge-boson mass. Note that the physics of the process here naturally singles out the Higgs $p_\perp$ as having a different shape than the other two, again different from the QCD case. The same machinery is also used for heavy-quark fusion to Higgs, $q\bar{q} \to Q\overline{Q}H$ and $gg \to Q\overline{Q}H$, where top masses are selected with a Breit–Wigner shape.

Another special case is central diffraction, *e.g.* $p(p_1)+p(p_2) \to p(p_3)+p(p_4)+X(p_5)$, where X is the central diffractive system. Here sampling in $t_1 = (p_1 - p_3)^2$ and $t_2 = (p_2 - p_4)^2$ is crucial to impose an exponential fall-off in these variables. The energy fractions x_1 and x_2 taken from the incoming proton defines $m_X^2 = p_5^2 = x_1 x_2 s$ in the collinear limit $t_1, t_2 \to 0$. Away from it also the φ_3 and φ_4 angles play a role, and one requires a more elaborate definition of x_1 and x_2.

2.3.3 Processes involving resonances

The term "resonance" has a specific meaning in PYTHIA and refers to particles whose decays are considered to be part of the hard process. This enables PYTHIA to modify the total calculated cross section depending on which decay channels are open or closed (including effects of sequential decays, such as $t \to bW^+$ followed by $W^+ \to e^+ \nu_e$), and also provides a natural framework for incorporating process-specific aspects such as spin correlations and/or finite-width effects. Here, we focus aspects of phase-space generation common to all processes involving resonances. Details on cross-section considerations, process-specific features,

and some further sophistications are explained in section 3.11, while user implementations of "semi-internal" resonances is described in section 9.7.3, and the handling of SUSY Les Houches Accord (SLHA) decay tables is covered in section 10.1.2.

We focus first on the *simplest* treatment available in PYTHIA, with partial widths and branching fractions fixed to their on-shell values. Technically, in the code this corresponds to decay channels that are assigned meMode = 100. Values of 101, 102, and 103 additionally include some simple kinematic threshold effects, also discussed below. Lastly, we emphasize that the *default* treatment often goes further than this, with most decay modes of SM resonances (and some BSM ones) assigned meMode = 0 implying the use of dedicated matrix-element expressions for branching fractions that can vary over a reasonably broad resonance peak; this is covered separately in section 3.11; see further section 9.7.3 for user implementations of such expressions.

Starting from a cross section computed in the zero-width approximation, *i.e.* for stable final-state resonances, the simplest shape modelling available in PYTHIA is a relativistic Breit–Wigner substitution of the type

$$1 = \int \delta(m^2 - m_0^2)dm^2 \rightarrow \int_{m_{\min}^2}^{m_{\max}^2} \frac{1}{\pi} \frac{m_0 \Gamma_0}{(m^2 - m_0^2)^2 + m_0^2 \Gamma_0^2} dm^2 , \qquad (50)$$

for each final-state resonance, where m_0 and Γ_0 are the nominal (on-shell) mass and widths of the resonance, respectively, and m is allowed to vary in a range $m \in [m_{\min}, m_{\max}]$ that can be specified individually for each resonance in PYTHIA's particle data table. Note that choosing a small range will reduce the total cross sections accordingly.

The phase-space integral in eq. (42) is then extended to include integrations over m^2 for each resonance, and the sampled values[1] for these masses are used instead of the on-shell ones in the evaluation of $d\hat{\sigma}/d\hat{t}$ and also in the relations between kinematic variables such as between $\hat{t}$ and $\cos\hat{\theta}$. This offers a crude level of approximation to the expected mass dependence of the full cross section, at least in the vicinity of the resonance(s) where the resonant amplitudes can still be assumed to dominate over any (non-resonant) background processes.

A complication arises for processes that involve pair production of the same kind of particle, such as $t\bar{t}$, W^+W^-, or Z^0Z^0 production. For such processes, on-shell matrix elements are phrased in terms of a single pole-mass value, while the procedure above produces two different values, m_3 and m_4. For the specific case of double-vector-boson production, PYTHIA uses 4-fermion matrix elements that include the full mass dependence (as well as the full γ^*/Z interference). However, for more general processes involving two of the same kind of resonance (such as $t\bar{t}$ production), the choice made in PYTHIA is to use an average squared mass,

$$\bar{m}^2 = \frac{m_3^2 + m_4^2}{2} - \frac{(m_3^2 - m_4^2)^2}{4\hat{s}} , \qquad (51)$$

which is defined so that $(\hat{s}, m_3^2, m_4^2)$ and $(\hat{s}, \bar{m}^2, \bar{m}^2)$ correspond to the same CM-frame three-momenta,

$$|\mathbf{p}^*(\hat{s}, \bar{m}^2, \bar{m}^2)|^2 = |\mathbf{p}^*(\hat{s}, m_3^2, m_4^2)|^2 = \frac{1}{4\hat{s}} (\hat{s} - (m_3 + m_4)^2)(\hat{s} - (m_3 - m_4)^2) . \qquad (52)$$

Analogous modified values for the $\hat{t}$ and $\hat{u}$ variables are defined to correspond to the same

---

[1]See [14, sec. 7.4.2] for details on the sampling procedure.

CM-frame scattering angle,

$$\bar{\hat{t}} = \hat{t} - \frac{(m_3^2 - m_4^2)^2}{4\hat{s}} = -\frac{1}{2}\left(\hat{s} - 2\bar{m}^2 - 2|\mathbf{p}^*|\sqrt{\hat{s}}\cos\hat{\theta}\right), \tag{53}$$

$$\bar{\hat{u}} = \hat{u} - \frac{(m_3^2 - m_4^2)^2}{4\hat{s}} = -\frac{1}{2}\left(\hat{s} - 2\bar{m}^2 + 2|\mathbf{p}^*|\sqrt{\hat{s}}\cos\hat{\theta}\right), \tag{54}$$

and these variables ($\bar{m}$, $\bar{\hat{t}}$, and $\bar{\hat{u}}$) are then used in the evaluation of the on-shell cross-section formula. If in doubt whether full matrix elements or the mass-symmetrized approximation represented by eqs. (51) to (54) is used for a given process, the corresponding `sigmaKin()` method can be inspected in the code (with $\bar{m}^2$ then typically denoted `s34Avg`). We note that, when gauge bosons are involved, the procedure is not guaranteed to be gauge invariant, nor positive definite, and breakdowns should be expected if any resonance masses are far from their on-shell values. The alternative would be to change to use full 4- or 6-body matrix elements instead (as already done for double-vector-boson production), *e.g.* by interfacing external hard-process generators, *cf.* section 10.1.

Be aware that, if a decay mode has been assigned `meMode = 100` and $m_{\min}$ is such that the decaying resonance can fluctuate down in mass to below the nominal threshold for the given decay mode (*i.e.*, $m_{\min} < \sum_j m_j$ with m_j the on-shell daughter masses for the decay mode in question), it is assumed that at least one of the daughters could also fluctuate down to keep the channel open. Otherwise the program will hit an impasse.

Alternatively, simple step functions $\Theta(m - \sum_j m_j)$ can be applied to impose kinematic thresholds; this is done for decay channels that are assigned `meMode = 101`. A slightly more sophisticated alternative is to use a smooth threshold factor,

$$\beta = \sqrt{\left(1 - \frac{m_1^2 + m_2^2}{m^2}\right)^2 - 4\frac{m_1^2 m_2^2}{m^4}}, \tag{55}$$

for two-body decay modes, and

$$\sqrt{1 - \frac{\sum_j m_j}{m}}, \tag{56}$$

for multi-body ones, again with m_j equal to the on-shell masses of the decay products for the given mode. The former correctly encodes the shrinking size of the phase space near threshold (but would still miss any non-trivial matrix-element factors) while the latter is only a crude simplification. Two separate options exist for this, depending on whether the stored on-shell branching fraction should be considered to already include this factor (`meMode = 103`) or whether it should be modified by it (`meMode = 102`). In the former case (`meMode = 103`), the actual factor applied is the ratio of the above to the corresponding value for $m = m_0$, with a safety limit imposed in case that denominator turns out to be very small, to avoid unintentionally large rescalings at large m.

Among the options discussed thus far, only the no-threshold one (`meMode = 100`) allows for purely off-shell decay modes, *i.e.* ones for which the on-shell daughter mass values exceed m_0; as noted above one or more of the daughters must then be able to fluctuate down in mass, or there will be trouble. The remaining options (`meMode = 101 – 103`) are all restricted to phase-space points satisfying $\sum m_j < m$, with m_j the on-shell daughter-mass values.

Currently, the only higher level of sophistication available in PYTHIA is to go all-in and implement dedicated decay-rate calculations specific to each given resonance and decay mode; this is obtained for `meMode = 0`. As mentioned above, this is the default for most SM resonance decays in PYTHIA as well as for some BSM ones, meaning that such code exists in the program (in the form of process-specific `SigmaProcess::weightDecay()` methods and

resonance-specific `ResonanceWidths::calcWidth()` methods) and is used by default. See further section 3.11.

Finally, note that both PYTHIA's simple shower as well as the VINCIA antenna shower allow for the insertion of resonance decays as $1 \rightarrow n$ branchings in the overall perturbative evolution, at decay-specific perturbative scales. This is called interleaved resonance decays [23] and is also further discussed in section 3.11.

Part II

Physics content

The PYTHIA event generator is the product of a physics development program in close touch with experimental reality. The two have often gone hand in hand, by making it possible to check which ideas work and which do not. Many of the concepts that today form the accepted picture of high-energy collisions can be traced directly to this undertaking, including string fragmentation, dipole showers, multiparton interactions, colour reconnection, and more. This part of the manual provides details on these and the other physics models encoded in PYTHIA. Many of these models still evolve to handle new experimental input, or to accommodate the progress in our theoretical understanding of the standard model of particle physics.

The first section describes the physics processes — sometimes denoted "hard processes" — internal to PYTHIA. These processes are those that can be calculated in leading-order perturbation theory in the standard model or simple extensions. While some of the calculations are currently outclassed, and are more of interest as a cross check or for quick studies, others, particularly the treatment of jet production and BSM physics models, are still actively used for comparisons with data. Further sections describe the core of the PYTHIA engine: parton showers, multiparton interactions, hadronization, and particle decays. An important complement to these sections is the one on matching and merging, which documents the methods for interfacing external calculations (that are more precise in perturbation theory) with the PYTHIA engine.

Notable additions to the PYTHIA 8.3 presentation here are the descriptions of two parton-shower plugins — DIRE and VINCIA— and the heavy-ion collision machinery.

3 Internal process types

PYTHIA 8.3 includes a good selection of native hard processes at Leading Order (LO). The hard processes are generated by sampling the allowed phase space using matrix elements, and convoluted with the PDFs, as a weight. Usually the multichannel sampling strategy results in accepted events with a common unit weight, but there are exceptions to this rule, so it is wise to be prepared for non-unit weights in event-analysis programs. In addition to the internal processes, there are several ways to feed in externally generated hard processes for showering and hadronization, including several options for matching and merging of higher-order processes. These options are discussed in detail in section 5 and section 10. This section classifies and lists internally defined hard processes and discusses about special features and appropriate settings for given processes. All internally defined processes are listed in appendix A, where also references to the cross-section formulae are given. If a process is included for both quark and lepton initial or final particles, the process is written with an "f" (denoting fermion) whereas a "q" is used when only quarks are expected. Charge-conjugate processes are always implicitly included.

3.1 Hard QCD

The internal QCD processes can be classified in three categories: (1) $2 \to 2$ scattering of light quarks and gluons, (2) production of heavy flavours (charm and bottom) in $2 \to 2$ processes, and (3) $2 \to 3$ processes involving light quarks and gluons. For hard processes suitable, process-dependent, phase-space cuts need to be applied to avoid soft and collinear singularities of perturbative QCD. In addition to these, soft QCD processes are included. These are discussed in section 6.1 and include also a unitarized version of the hard $2 \to 2$ cross sections,

where the divergences in the $p_\perp \to 0$ limit have been regulated with the screening parameter $p_{\perp 0}$, see section 6.2 for more details. Normally the hard and soft QCD processes would not be used simultaneously, since typically they target different kinds of physics studies. If they are combined nevertheless, relevant phase-space cuts should be introduced to separate the regions handled by each machinery, to prevent double counting.

3.1.1 Light quarks and gluons

This subclass of processes contain all possible $2 \to 2$ scatterings of massless quarks and gluons. In total there are six possibilities:

- $gg \to gg$

- $gg \to q\bar{q}$

- $qg \to qg$

- $qq' \to qq'$ where incoming and outgoing flavours are the same

- $q\bar{q} \to gg$

- $q\bar{q} \to q'\bar{q}'$

By default the light flavours include u, d and s quarks but it is also possible to produce c and b quarks with the massless matrix elements.

3.1.2 Heavy flavours

These processes provide heavy-quark pair production where heavy flavours stand for c and b. In LO, there are two possible processes each:

- $gg \to c\bar{c}$

- $q\bar{q} \to c\bar{c}$

- $gg \to b\bar{b}$

- $q\bar{q} \to b\bar{b}$

Unlike the case of massless partons, the finite mass also makes the matrix element expressions finite in the $p_\perp \to 0$ limit, so there is no need to introduce phase-space cuts to avoid divergences. However, it is also possible to generate these processes within the regularized soft QCD framework, though the mass effects are then not accounted for in the matrix elements. When considering heavy-quark production, one should keep in mind that especially c quarks are abundantly produced in the parton shower at LHC energies [24]. Therefore, to obtain *e.g.* inclusive D-meson spectra, these processes should be combined with the light-parton processes above. This combination will also provide the total QCD jet cross section in LO. Notice also that the $qg \to qg$ process is available only in the massless approximation above. (The massive matrix element for this process incorrectly sets the incoming quark on mass shell, so it is not a better alternative.)

3.1.3 Three-parton processes

In addition to $2 \to 2$ QCD processes, LO expressions for processes with three final state partons are also included in PYTHIA. These contain only light partons, but if needed the massive quarks can be dealt with using the massless matrix elements. One should also keep in mind that, since three-jet events can be formed from two-parton final states in the parton showers, mixing these with the $2 \to 2$ QCD processes would lead to double counting. So far this section is partly incomplete, *e.g.* colour flows are rather simple, so the purpose is rather to provide a way to check cross sections in specific kinematics where *e.g.* all three jets need to be above a certain $p_\perp$. Included processes are listed below:

- $gg \to ggg$

- $q\bar{q} \to ggg$

- $qg \to qgg$

- $qq' \to qq'g$ where q and q' are different flavours

- $qq \to qqg$ where incoming and outgoing flavours are the same

- $q\bar{q} \to q'\bar{q}'g$ where q and q' are different flavours

- $q\bar{q} \to q\bar{q}g$ where incoming and outgoing flavours are the same

- $gg \to q\bar{q}g$

- $qg \to qq'\bar{q}'$ where q and q' are different flavours

- $qg \to qq\bar{q}$ where incoming and outgoing flavours are the same

3.2 Electroweak

The internally defined electroweak (EW) processes contain prompt photon production, processes with photons in the initial state, and processes including electroweak bosons as an intermediate state or in the final state.

3.2.1 Prompt photon production

These processes include parton-initiated production that have one or two photons in the final state. The partonic cross sections are at LO in QCD for massless partons and contain only $2 \to 2$ processes. Thus, similarly as for light-flavour QCD, the expressions diverge in the $p_\perp \to 0$ limit and require a minimum $p_\perp$ cut to avoid QCD singularities. These processes are, however, also included in the eikonalized description of the soft QCD process class, where the divergences are regulated with the $p_{\perp 0}$. Therefore this event class should be preferred when low $p_\perp$ photons are considered. The available processes are

- $qg \to q\gamma$

- $q\bar{q} \to g\gamma$

- $gg \to g\gamma$

- $q\bar{q} \to \gamma\gamma$

- $gg \to \gamma\gamma$

Notice that the processes with two gluons in the initial state are box graphs. By default, it is assumed that the five massless quarks may form the loop, such that the expressions should be valid in a region where $p_\perp$ is between the b and t quark masses. It is, however, possible to change the number of active flavours inside the loop if a different region is considered. In addition to the photons produced in the hard scattering (prompt photons), photons may also be formed in parton showers and hadron decays. Therefore QCD processes might be needed to obtain a realistic rates for photon production.

3.2.2 Weak bosons

This section includes processes with standard model EW gauge bosons, γ^*/Z and $W^\pm$. The processes are classified into single and pair production, where the single production is associated with a parton and as an intermediate particle in t-channel exchange between two fermions.

As a highly-virtual photon γ^* cannot be distinguished from a Z boson with equal quantum numbers, typically both contributions should be accounted for to include interference effects. It is, however, possible to consider these two components separately, without the interference. This applies for all of the following processes including neutral EW bosons.

Boson exchange, DIS   The EW boson t-channel exchange is mainly relevant in Deep Inelastic Scattering (DIS) processes in lepton-hadron collisions but may also be applied for other types of collisions. There are two different contributions, one with neutral and one with charged EW bosons, namely

- $f f' \to f f'$ where a neutral γ^*/Z boson is exchanged so that the initial- and final-state fermion pair remains the same.

- $f_1 f_2 \to f_3 f_4$ where a charged $W^\pm$ boson is exchanged so that the initial- and final-state fermions are different. This includes charged current DIS with a charged-lepton beam and DIS with neutrino beams.

In pp collisions the factorization and renormalization scales are usually related to the transverse momentum, $p_\perp$, of the final-state particles. However, in DIS a more appropriate hard scale is usually the virtuality of the intermediate particle, Q^2. Therefore, when studying DIS with these processes, it is advised to set the renormalization and factorization scales appropriately, see section 3.10 for details. Similarly, rather than having a phase-space cut on $p_\perp$ to avoid divergences, here it is more appropriate to set a minimum Q^2 value to make sure that the relevant phase space is covered. Furthermore, since the default parton shower distributes the emission recoils globally, it is not well suited for DIS studies where the scattered lepton is expected to stay intact. Instead it is recommended to use either the dipole-recoil option or the DIRE shower, see section 4.1 and section 4.3 for further details.

Single boson production   Two different options are included for the single EW-boson production (s-channel) processes. In the first case the process is described as $2 \to 1$ scattering where the final state is either γ^*/Z or $W^\pm$:

- $\bar{f}f \to \gamma^*/Z$

- $ff' \to W^\pm$

The decay products of the short-lived (or virtual) particles and their kinematics are then derived as described in section 2.3.

The other possibility is to consider single EW-boson production as a $2 \to 2$ process where the decay products can be predetermined. This is useful if only certain final states are studied

and enables one to set phase-space cuts for the final state, *e.g.* for the $p_\perp$ of the produced lepton. These overlap with the first class of single-boson processes so one should not mix these processes to avoid double counting. The possible processes are:

- $f\bar{f} \to \gamma^* \to f'\bar{f}'$

- $f\bar{f} \to \gamma^*/Z \to f'\bar{f}'$

- $f_1\bar{f}_2 \to W^\pm \to f_3\bar{f}_4$

The difference between the first two is that in the first, the final state can be any of three possible lepton generations or five possible quark flavours, whereas in the second, the decay channels are set by the Z-decay modes. In the former, only γ^* exchange is included and the process is part of the MPI framework. In the latter, it is also possible to select between pure γ^*, Z, and the full interference modes. For the last, $W^\pm$ production, the decay channels are always the same for W^+ and W^-. These are set for W^+ and charge-conjugated channels are applied for W^-. No quark-mass effects are included for the angular distribution of the decay products of the $W^\pm$.

Boson pair production   These processes describe possible combinations of two EW-boson production, also including LO correlations for 4-lepton final states [25].

- $f\bar{f}' \to \gamma^*/Z\,\gamma^*/Z$

- $f\bar{f}' \to Z\,W^\pm$

- $f\bar{f} \to W^+W^-$

Notice that for the second process, no contribution from a virtual photon is included. In addition, it is possible to produce EW bosons in the parton shower as described in section 4.1.4 and section 4.2.4. Therefore, a full EW-boson pair production might require a combination of different processes with some additional care to avoid possible double counting.

Boson and parton production   These processes produce events where an EW boson is produced in association with a parton, where the latter in this case refers either to a quark, gluon, photon, or lepton. The possible channels are:

- $q\bar{q} \to \gamma^*/Z\,g$

- $qg \to \gamma^*/Z\,q$

- $f\bar{f} \to \gamma^*/Z\,\gamma$

- $f\gamma \to \gamma^*/Z\,f$

- $q\bar{q} \to W^\pm\,g$

- $qg \to W^\pm\,q$

- $f\bar{f} \to W^\pm\,\gamma$

- $f\gamma \to W^\pm\,f$

Again, there will be overlap with the single-boson production channels and the appropriate process depends on the final state and considered kinematics. For fully inclusive EW-boson production, the single-boson class is the relevant one, but for the high-$p_\perp$ tail these processes would provide more accurate kinematics. These processes should also be favoured when EW-boson production is studied with an associated high-$p_\perp$ jet or lepton.

3.2.3 Photon collisions

Many modern PDF sets include perturbatively-generated photons as a constituent of protons. In addition, all electrically-charged particles accelerated to high energies may emit photons that act as initiators for hard processes. The difference between these two cases is that in the former case, the beam hadron will be resolved, whereas in the latter case, the beam particle will stay intact. The following two-photon initiated processes are included in PYTHIA 8.3 and can be applied for resolved and unresolved beams:

- $\gamma\gamma \to q\bar{q}$

- $\gamma\gamma \to c\bar{c}$

- $\gamma\gamma \to b\bar{b}$

- $\gamma\gamma \to e^+e^-$

- $\gamma\gamma \to \mu^+\mu^-$

- $\gamma\gamma \to \tau^+\tau^-$

3.2.4 Photon-parton scattering

A few processes with a photon and a parton as initiators have been included. These are mainly relevant for photoproduction in ep collisions but can also be applied to other collision types with beams that may provide photons and partons. Similarly as pure-QCD processes with light partons, these processes also contain collinear and soft divergences so a suitable phase-space cut, *e.g.* on partonic $p_\perp$, must be applied to obtain finite cross sections. Unlike for the pure-QCD processes, no regularized description applicable at any $p_\perp$ is present for any of the photon-initiated processes. The included processes for photon-parton collisions are:

- $g\gamma \to q\bar{q}$

- $g\gamma \to c\bar{c}$

- $g\gamma \to b\bar{b}$

- $q\gamma \to qg$

- $q\gamma \to q\gamma$

Here, the heavy-quark pair production processes contain the full mass dependence in the matrix elements. Similar to the case of pure-QCD processes, at high-enough collision energies heavy quarks, at least charm, may be produced via parton-shower emissions so several processes might need to be considered to obtain realistic heavy-quark production rates.

3.3 Onia

Hard processes involving charmonium and bottomonium are provided using Non–Relativistic Quantum Chromodynamics (NRQCD) [26], which includes both colour-singlet and colour-octet contributions. The spectroscopic notation $^{2s+1}L_J$ specifies the necessary quantum numbers to define a state: spin s, orbital angular momentum L, and total angular momentum J. Processes are available for the $^3S_1$, ^3P_J, and ^3D_J states containing $c\bar{c}$ or $b\bar{b}$, given an arbitrary radial excitation n, *e.g.* any $\Upsilon(nS)$ for the $^3S_1$ $b\bar{b}$ onia states. Double onium production is also available for double-$^3S_1$ $c\bar{c}$ and $b\bar{b}$ processes, but only with colour-singlet contributions provided. Because of the long-standing discrepancy between polarization in data and NRQCD

predictions, only unpolarized processes are provided, with isotropic decays, which can then be reweighted accordingly by the user for a given polarization model.

Within NRQCD, the inclusive cross-section for a heavy onium state, H, can be written as,

$$d\sigma(pp \to H + X) = \sum_{s,L,J} d\hat{\sigma}(pp \to Q\overline{Q}[^{2s+1}L_J] + X)\langle\mathcal{O}^H[^{2s+1}L_J]\rangle, \tag{57}$$

where the cross section is factorized into a sum of products between short-distance matrix elements, $d\hat{\sigma}$, and long-distance matrix elements $\langle\mathcal{O}^H[^{2s+1}L_J]\rangle$. The short-distance matrix elements are calculated with perturbative QCD [27–32], while the long-distance matrix elements are determined from phenomenological fits to parameters [30, 33, 34]. Default settings for these parameters are provided for the J/ψ, $\psi(2S)$, χ_{c0}, χ_{c1}, χ_{c2}, $\psi(3770)$, $\Upsilon(1S)$, $\Upsilon(2S)$, $\Upsilon(3S)$, χ_{b0}, χ_{b1}, and χ_{b2} states.

The sum in eq. (57) for a given physical onium state $\left|H[^{2s+1}L_J]\right\rangle$ is over the expansion of its Fock states,

$$\left|H[^{2s+1}L_J]\right\rangle = \mathcal{O}(1)\left|Q\overline{Q}[^{2s+1}L_J^{(1)}]\right\rangle + \mathcal{O}(v)\left|Q\overline{Q}[^{2s+1}(L\pm 1)_{J'}^{(8)}]g\right\rangle$$
$$+ \mathcal{O}(v^2)\left|Q\overline{Q}[^{2s+1}(L\pm 1)_{J'}^{(8)}]gg\right\rangle + \dots, \tag{58}$$

where the superscript (1) indicates a colour-singlet state, (8) a colour-octet state, and the expansion is in the velocity v of the heavy-quark system. Consequently, a long-distance and short-distance matrix element must be provided for each state in the expansion.

For the physical $^3S_1$ states the following processes are available.

- $gg \to \left|Q\overline{Q}(^3S_1)[^3S_1^{(1)}]\right\rangle g$

- $gg \to \left|Q\overline{Q}(^3S_1)[^3S_1^{(1)}]\right\rangle \gamma$

- $gg \to \left|Q\overline{Q}(^3S_1)[^3S_1^{(8)}]\right\rangle g$

- $qg \to \left|Q\overline{Q}(^3S_1)[^3S_1^{(8)}]\right\rangle q$

- $q\overline{q} \to \left|Q\overline{Q}(^3S_1)[^3S_1^{(8)}]\right\rangle q$

- $gg \to \left|Q\overline{Q}(^3S_1)[^1S_0^{(8)}]\right\rangle g$

- $qg \to \left|Q\overline{Q}(^3S_1)[^1S_0^{(8)}]\right\rangle q$

- $q\overline{q} \to \left|Q\overline{Q}(^3S_1)[^1S_0^{(8)}]\right\rangle q$

- $gg \to \left|Q\overline{Q}(^3S_1)[^3P_J^{(8)}]\right\rangle g$

- $qg \to \left|Q\overline{Q}(^3S_1)[^3P_J^{(8)}]\right\rangle q$

- $q\overline{q} \to \left|Q\overline{Q}(^3S_1)[^3P_J^{(8)}]\right\rangle q$

The $^3P_J^{(8)}$ Fock states are a summation of contributions for $J = 0, 1, 2$. The $^3P_1^{(8)}$ and $^3P_2^{(8)}$ long-distance matrix elements are calculated from the $^3P_0^{(1)}$ long-distance matrix element.

The following processes are available for the physical ^3P_J states, again with $J = 0, 1, 2$.

- $gg \rightarrow \left| Q\overline{Q}(^3P_J)[^3P_J^{(1)}] \right\rangle g$

- $qg \rightarrow \left| Q\overline{Q}(^3P_J)[^3P_J^{(1)}] \right\rangle q$

- $q\overline{q} \rightarrow \left| Q\overline{Q}(^3P_J)[^3P_J^{(1)}] \right\rangle q$

- $gg \rightarrow \left| Q\overline{Q}(^3P_J)[^3S_1^{(8)}] \right\rangle g$

- $qg \rightarrow \left| Q\overline{Q}(^3P_J)[^3S_1^{(8)}] \right\rangle q$

- $q\overline{q} \rightarrow \left| Q\overline{Q}(^3P_J)[^3S_1^{(8)}] \right\rangle q$

Similar to the $^3P_J^{(8)}$ states, the colour-singlet $^3P_1^{(1)}$ and $^3P_2^{(1)}$ long-distance matrix elements are calculated from the $^3P_0^{(1)}$ long-distance matrix element.

For physical ^3D_J production, the following processes are provided:

- $gg \rightarrow \left| Q\overline{Q}(^3D_J)[^3D_J^{(1)}] \right\rangle g$

- $gg \rightarrow \left| Q\overline{Q}(^3D_J)[^3P_J^{(8)}] \right\rangle g$

- $qg \rightarrow \left| Q\overline{Q}(^3D_J)[^3P_J^{(8)}] \right\rangle q$

- $q\overline{q} \rightarrow \left| Q\overline{Q}(^3D_J)[^3P_J^{(8)}] \right\rangle q$

The colour-octet $^3P_J^{(8)}$ contributions are treated in the same fashion as for the physical $^3S_1$ state. Finally, double onium production is available for any arbitrary same-flavour $^3S_1$ configuration.

- $gg \rightarrow \left| Q\overline{Q}(^3S_1)[^3S_1^{(1)}] \right\rangle \left| Q\overline{Q}(^3S_1)[^3S_1^{(1)}] \right\rangle$

- $q\overline{q} \rightarrow \left| Q\overline{Q}(^3S_1)[^3S_1^{(1)}] \right\rangle \left| Q\overline{Q}(^3S_1)[^3S_1^{(1)}] \right\rangle$

The default configuration for double-onium production is to provide all possible combinations of the same-flavour physical $^3S_1$ states.

Many of the short-distance matrix elements diverge as $p_\perp \rightarrow 0$, and consequently must be regulated either with a hard cutoff or a smooth damping factor. Onium can be produced in a hard process, but also in multiparton interactions, except for double onium. In a hard process, a hard cutoff in $p_{\perp 0}$ is used, although it is also possible to implement smooth damping through a user defined scheme, see section 9.7.2. In multiparton interactions, instead, a smooth damping is performed with a cutoff scale $p_{\perp 0}$ for a given energy E_0 with an evolution parameter. See section 6.2 for more details.

The colour-octet states are defined in the event record using a non-standard numbering scheme, $99n_q n_s n_r n_L n_J$, where n_q is the quark flavour of the state and n_s is the colour-octet state type. Here, 0 is $^3S_1$, 1 is $^1S_0$, and 2 is ^3P_J. All remaining numbers follow the standard Particle Data Group (PDG) numbering scheme [35, sec. 45]. As an example, 9941003 is the $^1S_0^{(8)}$ $c\overline{c}$ colour-octet state for the colour-singlet J/ψ. After the parton shower and hadronization, all colour-octet states are forced to isotropically decay into their corresponding physical colour-singlet state and a soft gluon. A user-configurable mass splitting is used to set the mass of the colour-octet states for a given colour-singlet. This determines the softness of the gluon emitted in the octet to singlet transition.

Colour-octet states are allowed to evolve under the timelike QCD parton shower, see section 4 for more details on parton showers. This is meant to account for the competing effects of unbound $Q\overline{Q}$ states that emit additional gluons to become a semi-bound state, and semi-bound $Q\overline{Q}$ states that are broken apart by additional gluon radiation. The combination is approximated by allowing the colour-octet states to radiate in the parton shower with twice the $q \to qg$ splitting probability. Both the probability of a colour-octet state being considered in the parton shower and the pre-factor for the splitting kernel can be configured.

This treatment of colour-octet production in the parton shower is a simplification. The colour-octet state can be treated as a gluon, and so a factor of 9/4 rather than 2 may be used. Using a $q \to qg$ splitting kernel, rather than $g \to gg$, is roughly equivalent to always following the path of the harder gluon, resulting in harder final-state onia. Additionally, soft gluons producing heavy-quark will have little phase space to produce hard semi-bound states. However, after the $g \to Q\overline{Q}$ splitting, each heavy quark carries only approximately half the onium energy, reducing the energy of the gluon emissions. In principle, these two effects between softer and harder gluon emissions should approximately balance. However, comparisons to measurements of prompt J/ψ production in jets from pp collisions indicate that this treatment underestimates the local radiation surrounding onia [36, 37].

3.4 Top production

Standard model top production has now been part of standard measurements for over two decades and state-of-the-art experimental observations now make use of higher-order calculations. However, we still maintain a minimum set of top-production processes that can be used either with a K-factor for quick testing or for designing searches for non-standard decay modes by modifying the top-decay table by hand.

Production processes available include:

- $gg \to t\bar{t}$

- $qq \to t\bar{t}$

- $f\bar{f} \to t\bar{t}$ (via t-channel W or s-channel Z/γ separately)

- $\gamma\gamma \to t\bar{t}$

- $g\gamma \to t\bar{t}$

- $qq' \to q''t$ (single top via s-channel W)

It may be possible, for example, to test for the production of charged Higgses in top decays by adding the decay mode $t \to bH^+$ to the decay table and using the BSM Higgs sector (see the next section for details of setting BSM Higgs parameters).

3.5 Higgs

Pythia includes the capability of simulating production of standard model or BSM Higgses via the Two–Higgs Doublet Model (2HDM). The production processes for the SM Higgs include:

- $f\bar{f} \to H$

- $gg \to H$ (via 1-loop)

- $qg \to Hq$ (via 1-loop)

- $\gamma\gamma \to H$ (via 1-loop)

- $\mathrm{f\bar{f} \to Z\,H}$ (via s-channel Z)

- $\mathrm{f\bar{f} \to W^{\pm}\,H}$ (via s-channel W)

- $\mathrm{f\bar{f} \to H f\bar{f}}$ (vector-boson fusion, ZZ and $\mathrm{W^{\pm}W^{\pm}}$ can be selected separately)

- $\mathrm{f\bar{f} \to HQ\bar{Q}}$ where $\mathrm{Q = b, t}$

BSM Higgses can be produced using `Higgs:useBSM = on`. To allow for CP-violating cases, the neutral Higgses are named $\mathrm{H_1, H_2, A_3}$, which in the CP-conserving case refer to two scalar and one pseudoscalar Higgs, respectively. The neutral Higgses are ordered by mass. All processes mentioned above for the SM Higgs are also available for BSM ones by replacing H with the required BSM Higgs name. Further processes available for BSM Higgses are the pair-production processes.

- $\mathrm{f\bar{f} \to H_{1,2}A_3}$

- $\mathrm{f\bar{f} \to H^+H_{1,2}}$

- $\mathrm{f\bar{f} \to H^+H^-}$

The couplings of each Higgs boson to SM fermions can be set independently to account for all possible 2HDM structures. Further selection of the parity of each Higgs is also possible. We refer the user to the [online manual](#) for a description of each parameter.

The decay of the Higgses is also calculated automatically based on input parameters. The decay table can be overwritten by the user, either using the PYTHIA settings structure or using the SLHA interface (see section 10.1.2). Since the LHC cross-section working group recommends the usage of Next–to–Leading Order (NLO) decay widths, we use a multiplicative factor for all internally-calculated widths. The factor is calculated for $m_\mathrm{H} = 125$ GeV, but should be sensible for a range of masses. Furthermore, the Breit–Wigner shape of the Higgs resonance is complicated due to a dependence on mass. For resonance searches, it may be useful to "clip the wings" of the Breit–Wigner shape using `Higgs:clipWings` and `Higgs:wingsFac` (what factor of width beyond which to clip) parameters.

3.6 Supersymmetry

The implementation of the Minimal Supersymmetric Simplified Model (MSSM) allows fully general, complex 6×6 mixing in the squark sector, and up to five neutral gauginos (corresponding to next-to-minimal MSSM). We also allow all four kinds of R-parity violating couplings (one bi-linear and three tri-linear). Users are expected to input parameters via an SLHA file (see section 10.1.2). Typically, the Higgs sector of Supersymmetry (SUSY) is identical to a type-2 2HDM model and can be generated via the Higgs processes described above. PYTHIA is also capable of calculating decay widths in the standard channels for all SUSY particles. However, if a decay table is provided in the SLHA file, the internal calculation is turned off. For very low-width particles, the lifetime is set as the inverse of the total decay width. All particles with a decay width set to zero are set as stable.

Pair production of squarks ($\tilde{q}_i$), gluinos ($\tilde{g}$), and gauginos ($\tilde{\chi}_j^0$, $\tilde{\chi}^{\pm}$), including pairs like squark-gluino, squark-gaugino, and gluino-gaugino, are implemented with EW contributions. We also implement resonant production of squarks via R–Parity Violating (RPV) λ'' couplings, with corresponding modification to showering and hadronization to include the new colour structure. Here follows a full list of the processes available.

- squark-pair production (including anti-squarks and EW interference)

  - $\mathrm{f\bar{f} \to \tilde{q}_i \tilde{q}_j^{(*)}}$

- $gg \to \tilde{q}_i \tilde{q}_j^{(*)}$

- gluino pair and gluino-squark production

  - $q\bar{q} \to \tilde{g}\tilde{g}$
  - $gg \to \tilde{g}\tilde{g}$
  - $q_i g \to \tilde{q}_i \tilde{g}$ (and charge conjugate)

- electroweak-gaugino pair production

  - $f\bar{f} \to \tilde{\chi}^{\pm}\tilde{\chi}^{\pm}$
  - $f\bar{f} \to \tilde{\chi}_i^0 \tilde{\chi}_j^0$
  - $f\bar{f} \to \tilde{\chi}^{\pm}\tilde{\chi}_j^0$

- gaugino-gluino and gaugino-squark production

  - $f\bar{f} \to \tilde{g}\tilde{\chi}^{\mp}$
  - $f\bar{f} \to \tilde{g}\tilde{\chi}_i^0$
  - $f\bar{f} \to \tilde{q}_j^{(*)}\tilde{\chi}^{\mp}$
  - $f\bar{f} \to \tilde{q}_j^{(*)}\tilde{\chi}_i^0$

- slepton or sneutrino-pair production

  - $f\bar{f} \to \tilde{\ell}_i \tilde{\ell}_j^{(*)}$

- resonant production of a squark via an R-parity violating process

  - $q_i \bar{q}_j \to \tilde{q}_k$ (RPV)

Further selection of what processes to turn on is also possible by specifying individual PDG IDs of particles. All supersymmetric particles are given PDG codes greater than 1000000, with the superpartners generally carrying the corresponding code to their SM partner, *e.g.* an up quark is 2 and the two up squarks are named 1000002 and 2000002. The full list of PDG codes is available in the published review [35, sec. 45].

3.7 Hidden valley

Hidden Valley (HV) refers to a range of scenarios characterized by a gauge-symmetric dark sector with various possibilities of portals into the "valley". PYTHIA currently is the only general-purpose Monte Carlo code that implements a HV scenario, including running of gauge couplings, showering, and hadronization in the dark sector [38, 39]. There are multiple particle spectra and production modes available which together can cover a wide range of phenomenology.

First, based on the rank N of the dark **SU(N)**, radiation to either dark photons (*i.e.* **U(1)**) or dark gluons (*i.e.* **SU(N)**) is implemented. The matter content is modelled of in two separate ways — first via partners of the SM fermions (named dark-u, dark-d, dark-e and so on) that carry both the SM charges of their partner as well as fundamental of the dark **SU(N)**, and second via a "dark quark" that carries only the dark charge but does not carry any SM charges. In the first case, dark sector particles can be produced via normal SM gauge bosons and radiate to both SM and dark bosons based on their mass and relative strengths of the dark and SM couplings. In the second case, we implement an extra Z' portal to produce said quarks via a

kinetic mixing with the SM photon. The spin of the dark-sector particles (aside from the gauge bosons) can be set by the user to be either scalars, fermions, or vectors.

Two kinds of models are available in PYTHIA, depending on the charge of the new fermions. The first case is where the new fermions also carry some standard-model charge and can therefore be produced via one of the standard-model gauge bosons. The radiation of the final state-fermions then includes both dark-sector radiation as well as SM radiation. The processes that fall in this category include:

- $gg \to F_v \bar{F}_v$ via intermediate gluon where F_v is the hidden sector quark — either one of the quarks U_v, D_v, S_v, C_v, B_v, T_v, or generic quark Q_v

- $q\bar{q} \to F_v \bar{F}_v$ via intermediate gluon where F_v is the hidden sector fermion — either one of the quarks U_v, D_v, S_v, C_v, B_v, T_v, or generic quark Q_v

- $\bar{f}f \to F_v \bar{F}_v$ via intermediate Z or γ^*. F_v includes all the quarks above, plus the "leptons" E_v, v_{Ev}, and similarly for μ and τ flavours

It is possible to simulate a hidden sector where the new fermions do not carry any SM charges, but in this case, we need a new portal, which PYTHIA assumes is a new vector Z'. This Z' is then expected to be able to decay to both SM fermions as well as dark-sector fermions. Pair production of dark-sector fermions via this portal can be done using:

- $q\bar{q} \to Z_v$ followed by $Z_v \to F_v \bar{F}_v$

An important phenomenological effect is the running of the hidden-sector strong coupling which can make significant changes to the radiation pattern in the dark sector. This is by default taken into account by using the one-loop beta function of **SU(N)** once the number of colours and flavours of new fermions is set. The running can also be turned off by the user to use a fixed-coupling value instead. There is also an inherent ambiguity in the composition of the hadrons in the dark sector. PYTHIA allows the user to manually set the ratio of scalar to vector mesons as well as the parameters of the Lund symmetric fragmentation function or the dark sector (see section 7.1 for details of the fragmentation functions). The decay table of the hidden mesons back into the standard model (should this be desirable), can be done by the user at run time using the standard particle data scheme that PYTHIA uses for all particles.

3.8 Dark matter

Multiple models for Dark Matter (DM) are currently implemented in PYTHIA. They may be separated into two different categories — production via *s*-channel mediator and production via pair production of mediators (typically seen in co-annihilation or co-scattering scenarios of DM). In all cases, the DM is assumed to be fermionic. We provide the possibility to produce dark matter with one associated jet for the *s*-channel models (vector or axial-vector Z' and scalar or pseudoscalar *A*). For the mediator pair-production processes, all mediators are produced via Drell–Yan production.

The PDG provides some standard codes for common DM particles and mediators, *cf.* [35, sec. 45]. Of these, the fermionic DM code 52, the *s*-channel scalar mediator (54) and vector mediator (55) are used in this implementation. The new mediators are either charged scalar (with PDG code 56), charged vector-like fermion (PDG code 57), and doubly charged fermion (PDG code 59). The neutral partner that accompanies the charged mediators is given the PDG code 58.

The singlet model contains a scalar singlet with quantum numbers identical to a right-handed lepton. Therefore, it couples via a Yukawa-like coupling to a SM right-handed lepton

and the DM is a Dirac fermion. Both the scalar and the DM are odd under a Z_2 symmetry to ensure the stability of the DM.

$$\mathcal{L} = \partial_\mu \phi^* \partial^\mu \phi + \bar{\chi}(i\gamma^\nu \partial_\nu \chi) - m_\phi^2 |\phi|^2 - m\bar{\chi}\chi - (y_\ell \bar{\ell}\phi\chi + \text{h.c.}). \tag{59}$$

The fermionic mediators are based on models that have mixing between a singlet and an n-plet vector fermion, both charged under a Z_2 symmetry for which all of SM particles are even. The mixing between the singlet and n-plet is then calculated based on the value of n. The lightest neutral state is denoted as dark matter.

$$\mathcal{L} = \bar{\chi}(i\gamma^\nu \partial_\nu \chi) + \bar{\psi}(i\gamma^\nu D_\nu \psi) - m_1 \bar{\chi}\chi - m_2 \bar{\psi}\psi. \tag{60}$$

The mixing term depends on the representation of ψ. For example, for a triplet case, we have

$$\mathcal{L}_{\text{mix}} = \frac{c}{\Lambda^2} \left(\bar{\chi}(\Phi_h^\dagger \tau^a \Phi_h)\psi^a + \text{h.c.} \right), \tag{61}$$

where Φ_h is the SM-Higgs doublet, ψ is the triplet fermion, τ are the Pauli matrices and χ is the singlet fermion.

Production of DM can be studied in two ways — either by directly producing the s-channel mediator, which then decays to DM, or by producing the charged partner of DM via Drell–Yan followed by the decay of the partner. The production processes therefore are

- $q\bar{q} \to Z' \to \bar{\chi}\chi$

- $g\bar{g} \to S \to \bar{\chi}\chi$, note that 1-loop $g\bar{g} \to S$ via top-loop is included in this production.

- $q\bar{q} \to Z'g$ (mono-jet)

- $g\bar{g} \to Sg$ (also mono-jet, via 1-loop in production)

- $q\bar{q} \to Z'H$, *i.e.* mono-Higgs production (coupling of the SM to the new Z' has to be set by the user)

- $f\bar{f} \to \psi\bar{\psi}$ where $\psi = \tilde{\ell}^\pm$ (scalar with leptonic quantum numbers), $\chi^\pm$ (singly charged fermion), or $\chi^{\pm\pm}$ (doubly charged fermion), followed by decay of ψ into DM (Drell–Yan for charged partners)

Couplings of quarks and leptons to the mediators are assumed to be generation universal, however vector and axial-vector components (or equivalently scalar and pseudoscalar components for the scalar mediator) can be set individually for up type, down type, charged lepton, neutrino, and dark-matter fermions. In case of Z', it is also possible to choose a kinetic-mixing parameter ϵ which then automatically sets the rest.

3.9 Other exotica

Finally, we mention other models of new physics that are implemented in PYTHIA, though they are perhaps not as popular as they once were. We refer the reader to the `online manual` for the detailed descriptions of the model parameters and only provide a list here.

- **Fourth generation** includes production of fourth-generation quarks or leptons via the usual SM-mediated processes.

- **New gauge boson Z', W' and horizontal gauge boson** production can be performed through $f\bar{f} \to V$ production followed by decay. For Z', full interference with SM γ, Z in the s-channel production is taken into account. It is possible to have both universal and non-universal models where couplings to each generation should be set by hand by the user.

- The **left-right symmetry** model includes a right handed **SU(2)** sector. Along with the gauge bosons W' and Z', it also includes heavy right-handed neutrinos that can be used to study signatures of heavy neutral leptons.

- **Leptoquark production** includes resonant single production or pair production of scalar leptoquarks via gluon-mediated diagrams. The flavour of the leptoquark should be set by the user by explicitly setting the decay table of the leptoquark.

- **Compositeness** models include simple models of excited fermions and contact interactions that modify standard QCD dijet or Drell–Yan production of leptons.

- **Extra dimensions** includes production of the graviton or the extra Kaluza–Klein gauge boson (*e.g.* KK-gluon) of the Randall–Sundrum model. Further processes include modification of SM dijet/dilepton production due to extra KK-bosons in the TeV-scale or large extra dimension models. Finally, **Unparticle** emission is modelled associated with a jet or photon.

3.10 Couplings and scales for internal processes

The perturbatively calculated cross sections for QCD and Quantum Electrodynamics (QED) processes depend directly on the value of the relevant coupling evaluated at the scale at which the hard scattering occurs. The scale dependence of the couplings arises due to the renormalization procedure required to obtain finite cross sections and can be calculated by solving the renormalization group equations of the applied theory.

In PYTHIA 8.3 the running of the QCD coupling, $\alpha_s(Q^2)$, is implemented up to second order and applied at first order by default to match the precision of the internally-calculated cross sections. A fixed value can also be used, but the potential usage is limited to special cases and generally a running coupling should be applied for realistic cross-section estimates. The coefficients related to the value of the coupling are fixed by setting the $\alpha_s(Q^2)$ value at the mass of the Z boson.

Similarly, running of the QED coupling $\alpha_{em}(Q^2)$ has been implemented in PYTHIA 8.3. This, however, runs much slower than the QCD one and only first-order running is implemented. An option to use a fixed value for $\alpha_{em}(Q^2)$ is included, either by setting the value directly at the mass of the Z boson or by matching to its value at vanishing momentum transfer. In addition, it is possible to globally scale the cross sections with a K-factor if such behaviour is desired.

There are two relevant scales that needs to be set. The renormalization scale, Q^2_{ren}, arises from the renormalization procedure and defines at which scale the couplings are evaluated. The factorization scale Q^2_{fact} arises from factorizing the short-distance phenomena (hard scattering) from the large-distance (soft) structure of hadrons. This scale determines at which Q^2 the PDFs of resolved beams are probed.

As the scale dependencies arise from an approximated description of QCD, there is some amount of freedom in the scale choices. The only solid guideline is that the scales should be related to the hardness of the scattering process and therefore the optimal choice depends on the type of the studied process. Multiple options for the scale choices have been implemented into PYTHIA 8.3, and all options are available for both Q^2_{ren} and Q^2_{fact}.

For $2 \to 1$ processes two options exist:

- the squared invariant mass, $\hat{s}$, *i.e.* the mass of the produced particle;

- and a fixed scale.

For $2 \to 2$ a few more options are included:

- the smaller of the squared transverse masses of the outgoing particles, $\min(m^2_{\perp,3}, m^2_{\perp,4})$;

- the geometric mean of the squared transverse masses of the outgoing particles, $m_{\perp,3} \cdot m_{\perp,4}$;

- the arithmetic mean of the squared transverse masses of the outgoing particles, $(m^2_{\perp,3} + m^2_{\perp,4})/2$;

- the squared invariant mass of the system, $\hat{s}$, relevant for s-channel processes;

- the squared invariant momentum transfer $-\hat{t}$, relevant especially for DIS events as this coincides with the virtuality of the intermediate photon Q^2;

- and a fixed scale.

For $2 \to 3$ processes the possible choices are:

- the smallest of the squared transverse mass of the outgoing particles, $\min(m^2_{\perp,3}, m^2_{\perp,4}, m^2_{\perp,5})$;

- the geometric mean of the two smallest squared transverse masses of the outgoing particles, $\sqrt{m^2_{\perp,3} \cdot m^2_{\perp,4} \cdot m^2_{\perp,5}/\max(m^2_{\perp,3}, m^2_{\perp,4}, m^2_{\perp,5})}$;

- the geometric mean of the squared transverse masses of the outgoing particles, $(m^2_{\perp,3} \cdot m^2_{\perp,4} \cdot m^2_{\perp,5})^{1/3}$;

- the arithmetic mean of the squared transverse masses of the outgoing particles, $(m^2_{\perp,3} + m^2_{\perp,4} + m^2_{\perp,5})/3$;

- the squared invariant mass of the system, $\hat{s}$, relevant for s-channel processes;

- and a fixed scale.

For vector-boson-fusion (VBF) processes, such as $f_1 f_2 \to f_3 H f_4$, the virtualities of the intermediate bosons would not be accounted for with the above options and would likely underestimate the relevant scales. Therefore modified scale choices where, instead of the transverse mass of the final-state particle, a virtuality estimate $m^2_{\perp,Vi} = m^2_V + p^2_{\perp,i}$ can be used in the options above when relevant.

Traditionally, the theoretical uncertainties related to the truncated pQCD expansion are estimated by varying the QCD scales by a factor of two or so. To enable such variations, options to multiply the scales determined by the options above by constant factors have been implemented. In a basic form, these variations will, however, require to generate a completely new set of events, so mapping out all possible uncertainties might become computationally demanding. Therefore, options to calculate weights for each event based on different scale variations have been implemented in PYTHIA 8.3 for more efficient uncertainty estimation, see section 9.8 for details. Notice also that the couplings and scales can be set separately for MPIs and initial- and final-state showers.

3.11 Handling of resonances and their decays

By default, the SM electroweak gauge bosons, top quarks, the Higgs boson, and generally all BSM particles are classified as resonances. Note that all of these have on-shell masses above 20 GeV (with the exception of some hypothetical weakly interacting and stable particles such as the gravitino, which are also considered resonances).

Importantly, neither hadrons nor any particles that can be produced in hadron decays, such as τ leptons, are included in this category. The decays of such particles are performed after hadronization, and changing their decay channels will *not* automatically affect the reported cross section. For example, allowing only the decay $Z \rightarrow \mu^+\mu^-$ will reduce the total cross section reported by PYTHIA for hard processes like pp $\rightarrow$ Z by the appropriate branching fraction, while allowing only the decay $J/\psi \rightarrow \mu^+\mu^-$ will *not* change the cross section for gg $\rightarrow J/\psi$g. The reason for this is that hadron and τ decays involve multistep chains that cannot be predicted beforehand: a hard process like gg $\rightarrow$ gg can develop a shower with a g $\rightarrow$ b$\bar{\text{b}}$ branching, where the b hadronizes to a $\bar{\text{B}}^0$ that oscillates to a B^0 that decays to a J/ψ. Any bias at the hard-process level would not affect these other production mechanisms and could thus be misleading. Instead, the user must consider all relevant production sources and perform their own careful bookkeeping.

Both types, "resonances" and "unstable particles", can have Breit–Wigner distributed mass spectra (at least when generated by internal PYTHIA processes); more on this below. For the remainder of this subsection we focus on the production and decay of those particles that are classified as resonances, referring to section 8 for the treatment of hadron and τ decays.

Note that the cross-section reduction factors to account for decay modes that have been switched off are always evaluated at initialization, for nominal masses. For instance, in the example above, the $Z \rightarrow \mu^+\mu^-$ reduction factor is evaluated at the nominal Z mass, even when that factor is used, later on, say in the description of the decay of a 125 GeV Higgs boson, where at least one Z would be produced below this mass. We know of no case where this approximation has any serious consequences, however.

Note also that, for the specific case of electroweak showers (*cf.* section 4.1.4 and section 4.2.4), the decays of any resonances that are produced by the shower (*i.e.* not by the hard process) are treated inclusively, ignoring any user restrictions on which channels should be open or closed. It is then up to the user to select the final states of interest and reject the rest.

Finally, a word of caution: the above logic implies that switching off *all* of the decay channels of a resonance will result in cross sections evaluating to zero, precluding PYTHIA from being able to generate any events. Instead, to force a resonance to be treated as stable for a given run, set `NN:mayDecay = false`, with `NN` being its particle ID code.

Total and partial widths: For resonances, the partial widths to different decay channels are typically perturbatively calculable, given the parameters of the respective model. By default, during initialization PYTHIA therefore computes the hadronic widths of W, Z, t, and SM Higgs bosons at NLO in QCD, with

$$\Gamma^{\text{NLO}}_{\text{V}\rightarrow\text{q}\bar{\text{q}}} = \left(1 + \frac{\alpha_s(m_V^2)}{\pi}\right)\Gamma^{\text{LO}}_{\text{V}\rightarrow\text{q}\bar{\text{q}}},$$

$$\Gamma^{\text{NLO}}_{\text{t}\rightarrow\text{bW}} = \left(1 - \frac{5\alpha_s(m_Z^2)}{2\pi}\right)\Gamma^{\text{LO}}_{\text{t}\rightarrow\text{bW}}, \tag{62}$$

where V is a generic vector boson. For H^0, the default is a set of channel-specific numerical NLO rescaling factors recommended by the LHCXSWG [40], with current values given in table 1 valid for a reasonable range around the nominal Higgs mass of $m_{\text{H}} = 125$ GeV. Note also that PYTHIA 8 computes the LO partial widths for H$^0 \rightarrow \gamma\gamma$ and H$^0 \rightarrow$ gg using running quark-mass values in the loop integrals (evaluated at m_{H}); this gives a non-negligible shift relative to PYTHIA 6 which used pole-mass values in the same expressions. For comparisons, the LHCXSWG rescaling factors can optionally be replaced by simple $(1 + \alpha_s/\pi)$ correction for the decays to quarks, and for the loop-induced decays the running mass values can be replaced by pole ones.

Table 1: Numerical correction factors applied to the LO SM-Higgs decay partial width, based on LHCXSWG recommendations [40]. Note that the strong coupling is fixed to $\alpha_s = 0.12833$ in this context.

SM H^0 Decay Mode:	gg	$\gamma\gamma$	γZ	ZZ	WW	b$\bar{\text{b}}$	c$\bar{\text{c}}$	$\mu^+\mu^-$	$\tau^+\tau^-$
NLO rescaling factor:	1.47	0.88	0.95	1.10	1.09	1.11	0.98	0.974	0.992

For BSM resonances, PYTHIA applies the $(1 + \alpha_s/\pi)$ factor to all integer-spin BSM particle decays to quark-antiquark pairs and to semi-leptonic decays of right-handed neutrinos, while the $(1 - 5\alpha_s/(2\pi))$ one is applied to t$' \to$ qW decays.

At the technical level, these decay-rate calculations are performed by dedicated `calcWidth()` methods in the derived `ResonanceWidths` class for the given resonance. Note that this means that the tabulated widths for these particles stored in the program's particle data table are purely dummy values, overridden at initialization. To force a resonance with ID code NN to have a certain user-defined width, Γ, set `NN:doForceWidth = on` and `NN:mWidth` = Γ. Input of resonance widths via the SLHA interface is discussed separately below.

Breit–Wigner modelling: We now turn to PYTHIA's modelling of resonance *shapes*. Note that this applies to resonances that are produced by PYTHIA (*i.e.* in PYTHIA's internal hard processes and/or in decays performed by PYTHIA). For externally generated ones, *cf.* section 10.1, it is the responsibility of the external generator to model the shape of the produced resonances, though PYTHIA's modelling may still apply to any resonances produced by subsequent decays of particles that are kept stable in the external process.

An important note in the specific context that an external generator is responsible not only for resonance production, but also for one or more of their decays is that the total invariant mass of the resonance-decay products (and hence the resonance shape) is only guaranteed to be preserved during parton showering if an explicit resonance mother (with Les Houches status code 2) is present in the externally provided event record. This is particularly relevant for any coloured resonances (such as top quarks), for which the reconstructible resonance-mass distribution will otherwise be impacted by unphysically large QCD recoil effects to parton(s) outside the resonance-decay system. In principle, the same issue exists for QED recoil effects in decays of electrically charged resonances.

The basics of phase-space generation and Breit–Wigner sampling in the context of processes involving resonances were covered in section 2.3.3. As already mentioned there, decay-rate calculations specific to each given resonance and decay mode are the default for most SM-resonance decays in PYTHIA as well as for some BSM ones, via process-specific `SigmaProcess::weightDecay()` methods and resonance-specific `ResonanceWidths::calcWidth()` methods, enabled for decay channels assigned `meMode` = 0. For resonances that include such channels, eq. (50) of section 2.3.3 is generalized to

$$\frac{1}{\pi} \frac{m \sum_j \Gamma_j(m)}{(m^2 - m_0^2)^2 + m^2\Gamma_{\text{tot}}^2(m)} \,, \tag{63}$$

where both the partial widths Γ_j and the total width Γ_{tot} are in principle allowed to depend on m. There are two main sources of m dependence:

- Running couplings in the relevant matrix elements. This also applies *e.g.* to the NLO normalizations given by eq. (62), in which $\alpha_s(m_0^2)$ is replaced by $\alpha_s(m^2)$. The SM-Higgs resonance is sufficiently narrow that no appreciable running effects are expected, hence the partial widths given in table 1 are left unchanged.

- Threshold effects. For bosonic resonances (Z, W, H, and particles that are trivially related to them such as Z', W', H^+, and A bosons), decays to same-flavour fermion pairs are associated with the following threshold factors:

$$
\Gamma(m) = \frac{m\Gamma_0}{m_0}\Theta(\hat{s} - 4m_f^2) \begin{cases} \beta^3 & : & \text{scalar}, \\ \beta & : & \text{pseudoscalar}, \\ \beta(3 - \beta^2)/2 & : & \text{vector}, \\ \beta^3 & : & \text{axial-vector}, \end{cases} \tag{64}
$$

where Γ_0 is the on-shell partial width and $\beta = \sqrt{1 - 4m_f^2/m^2}$ is the fermion velocity in the rest frame of the decay. Resonances that have both vector and axial-vector (or both scalar and pseudoscalar) couplings use appropriate mixtures of these factors, and analogous but more complicated expressions are used for decays into unequal masses *e.g.* of the W^+. For other decays, the m dependence is typically more complicated.

We refer to the corresponding implementations in the `weightDecay()` and `calcWidth()` methods in the code, which can be inspected for more details about the treatment of a given process and/or decay mode, respectively.

Decay angular distributions: In many cases, non-trivial angular distributions are encoded in PYTHIA via process-specific LO matrix elements that include the relevant decays. For example, for the hard process $\bar{f}f \to W^+W^-$ (with f denoting a generic fermion), PYTHIA generates the angular distributions for the two W decays at the same time, using the full $\bar{f}f \to W^+W^- \to 4$-fermion matrix elements.

This allows for an accounting of the effects of spin correlations between the production and decay stages. Note, however, that only diagrams with the same resonant structure as the production process are included; interference with background processes is not accounted for by this method.

Using V to denote a generic weak boson ($W^\pm$ or Z^0, with the latter typically including γ^*/Z interference where relevant) and H to denote a generic neutral Higgs boson, processes for which such matrix-element-corrected resonance-decay distributions are generated by PYTHIA 8.3 include:

- Decays of (unpolarized) top quarks: $t \to bW^+ \to b\,2f$.

- Electroweak decays of neutral Higgs bosons: $H \to VV \to 4f$ and $H \to \gamma Z \to \gamma\,2f$, in both cases allowing for generic (BSM) mixed-CP states.

- Electroweak resonant s-channel processes $2f \to V \to 2f$. Note: this extends to BSM vector bosons such as V' and V_R, and also includes the full $\gamma^*/Z/Z'$ interference for Z' ones.

- Electroweak resonant $2 \to 4$ processes $2f \to VV \to 4f$ and $2f \to HV \to 4f$. Also $2f \to V' \to VV \to 4f$.

- W decays in $\bar{f}f \to g/\gamma\,W \to g/\gamma\,2f$.

- BSM excited-graviton decays in $2f \to G^*$ and $gg \to G^*$ processes, *cf.* [41].

- BSM compositeness excited-fermion decays in $2 \to f^* \to g/\gamma\,f$ and $2 \to f^* \to Vf$, with V decaying isotropically for the latter.

A prominent example of a process that is absent from this list is top-quark pair production, implying that internally generated $t\bar{t}$ events in PYTHIA do not exhibit non-trivial correlations *between* the two top decays. Note also that, for externally provided events (*cf.* section 10.1), only the top- and Higgs-decay correlations in the two first points above are applied. When interfacing external hard processes it is therefore important to consider whether, and how, resonance decays are treated by the external generator.

At the technical level, these process-specific angular distributions are implemented via dedicated `weightDecay()` methods in the derived `SigmaProcess` class for the given hard process.

Effects of PDFs on resonance shapes: Often, the observable resonance shape results from a convolution with non-trivial parton distribution functions. For hadrons, these tend to be strongly peaked towards small x, with a typical asymptotic behaviour roughly like $f(x) \propto 1/x$. When convoluted with the Breit–Wigner shape, this tilts the overall resonance shape; the parton-parton luminosity is higher in the low-mass tail than it is in the high-mass tail.

If the low-mass enhancement is strong enough, the wide tails of the Breit–Wigner can even lead to a secondary peaking of the cross section towards very low masses. This is obviously unphysical, as the resonant approximation is invalid that far from the resonance, and non-resonant background processes would anyway normally dominate in that region. The desire to cut away such behaviour is one reason for the default choices made in PYTHIA for the m_{min} limits in eq. (50). For non-standard PDFs, or when making user-defined modifications to the nominal mass and/or width values (*e.g.* for BSM particles), it is up to the user to check that sensible m_{min} limits are imposed.

Interleaved resonance decays: Rounding off the discussion of resonance production and decays, PYTHIA also allows for interleaving resonance decays with the final-state shower evolution, as described in [23]. Currently, this is only done by default for the VINCIA shower model, while it exists as a non-default option for PYTHIA's simple showers.

When interleaved resonance decays are enabled, resonance decays are inserted into the final-state shower evolution as $1 \rightarrow n$ branchings, at a scale which by default is given by the following measure of the off-shellness of the resonance propagator,

$$Q_{\mathrm{RES}}^2 = \frac{(m^2 - m_0^2)^2}{m_0^2}, \tag{65}$$

with median value $\langle Q_{\mathrm{RES}} \rangle = \Gamma$. (A few alternative choices are also offered, including an option to use a fixed scale $Q_{\mathrm{RES}} \equiv \Gamma$.) As part of the resonance-decay branching process, a "resonance shower" is also performed, in the region $m_0 > Q > Q_{\mathrm{RES}}$. This shower stage only involves the decaying resonance and its decay products, with no recoils to any other partons. Note that any nested resonance decays associated with intermediate scales (*e.g.* the W boson produced in a $t \rightarrow bW$ decay) are also performed during this stage, along with their corresponding resonance showers, while any decays associated with scales below Q_{RES} occur afterwards, sequentially.

The main consequence is that resonances are prevented from participating as emitters or recoilers for radiation at scales below Q_{RES}; only their decay products can do that. We refer to [23] for further details.

3.12   Parton distribution functions

Parton distribution functions provide number distributions of a parton flavour i at a given momentum fraction x when a hadron is probed at scale Q^2, and are a necessary input for any hard

process generation with hadron beams [42]. Here, we focus on PDFs for hadrons and nuclei — PDFs for other types of beams (including leptons, photons, and pomerons) are discussed separately in section 6. The scale evolution of the PDFs is provided by the Dokshitzer–Gribov–Lipatov–Altarelli–Parisi (DGLAP) equations [43–45] and usually these are derived in a global QCD analysis where the non-perturbative input at an initial scale is fitted to a wide range of experimental data. Further constraints are provided by the momentum- and baryon-number sum rules. Nowadays, it is common that in addition to the best fit, the PDF sets also provide error sets that can be used to quantify how the uncertainties in the applied data propagate into other observables.

In the case of protons, the high-precision DIS data from HERA collider form the backbone of the PDF analyses. On top of this, the modern PDF sets incorporate a wealth of different LHC data to increase the kinematic reach of the analysis and to obtain further constraints for the flavour dependence. With this, in kinematic regions relevant for LHC studies, the proton structure is known with a percent-level accuracy, except for in a few regions like the very small-x region. PYTHIA comes with some 20 different proton PDF sets. There are a few (pre-HERA) sets that are out of date, *e.g.* GRV94L and CTEQ5L, but are kept in for historical reasons as some earlier tunes were based on these. In addition, there are a few sets that include HERA data but did not have any from LHC data (*e.g.* CTEQ6L) which mainly differ from the older ones due different small-x gluon behaviour. The more modern ones include several data sets from LHC experiments which provide further constraints for gluon PDFs and flavour separation between different quarks. Another recent development in the PDFs is the inclusion of QED evolution that enables inclusion of photons as a part of the hadron structure. The current default set is NNPDF2.3 QCD+QED LO, which does contain some datasets from LHC, but not the most recent ones. It is important to note, however, that the default Monash tune is based on this PDF set, so updating to a more recent PDF set would not lead to an improved description unless a complete retuning of pp parameters is performed. Many further sets are accessible via the LHAPDF interface, *cf.* section 10.1.4. This runs slightly slower than the built-in sets, but also offers further facilities such as error bands around the central PDF member. Notice also that there might be small differences between the internally defined PDFs sets and the corresponding LHAPDF grids due to different interpolation routines and different extrapolations beyond the provided interpolation grid.

The neutron PDF is obtained from the proton one by isospin conjugation. This is not quite correct for some recent sets where the QCD evolution is combined with a QED one, *i.e.* where the quarks can radiate off photons, but in practice it is good enough except for photon physics.

For pions, the main set is based on GRS 99 [46]. This work makes the ansatz that valence, gluon, and sea PDFs are of the form $N x^a (1-x)^b (1 + A\sqrt{x} + Bx)$ at an initial scale $Q_0^2 = 0.26\,\text{GeV}^2$, with the parameters fitted to data. By choosing a small Q_0, the distributions can be assigned a valence-quark-like shape at that scale, and strange and heavier quarks can be taken to vanish. An older set based on GRV 92 [47] is available, but is deprecated in favour of GRS 99. A similar PDF is also available for the kaon [48].

For other hadrons, rough estimates for PDFs have been made based on the form above, with $A = B = 0$. No data is available, so the parameters a and b have been chosen heuristically, based on the guiding principle that all valence quarks should have roughly the same velocity for the hadron to stay together over time, and thus heavier quarks must take a larger average momentum fraction. The N are fixed by the flavour and momentum sum relations. For details on this procedure, see [49]. These PDFs are referred to as the SU21 sets, and are stored in the LHAPDF format and distributed with PYTHIA 8.3. Specifically, the PDFs included this way are available for the following hadrons: p, π^+, K^+, ϕ^0, η, D^0, D_s^+, J/ψ, B^+, B_s^0, B_c^+, Υ, Σ^+, Ξ^+, Ω^-, Σ_c^{++}, Ξ_c^+, Ω_c^0, Σ_b^+, Ξ_b^-, and Ω_b^-. The SU21 p and π PDFs are less accurate than other available sets, so they should not to be used in real studies, but are included for completeness.

Hadrons with the same quark contents as the ones above are assumed to have the same PDFs. Furthermore, other cases can be defined using isospin conjugation, since no QED effects are included in the SU21 sets. Mixed cases such as π^0 and Σ^0 are assumed to have equal u and d contents, which are given by the averages for the corresponding implemented PDF (*i.e.* π^+ and Σ^+, respectively). Using such rules, all normal hadrons can be simulated, except for baryons with more than one charm or bottom quark. One final technical point is that in the SU21 LHAPDF files for flavour-diagonal mesons, the antiquark content represents the sea, in order to make it possible to separate valence and sea (*e.g.* for J/ψ, the c column represents the charm content, while the $\bar{c}$ column represents charm sea).

Also, a few nuclear PDF sets have been included internally. These can be used to estimate the leading nuclear effect for inclusive high-$p_\perp$ observables, such as jet production, but for more involved studies it is recommended to use the full heavy-ion machinery, see section 6.8. More nPDFs are available as LHAPDF grids, but the advantage of the internally defined sets is that any proton baseline PDF can be applied and, if needed, the number of protons and neutrons can be redefined event-by-event.

A fair fraction of the internal PDFs are LO ones. This ensures a sensible behaviour also for processes at low x and/or Q^2 (discussed further below), but also some NLO and NNLO proton sets are available, for the modelling of hard processes. In this context, it can be mentioned that, at large x and Q^2, NLO corrections to the PDF shape are often more important than those for the matrix elements, such that NLO PDFs and LO MEs can be a viable combination. Further, in the large-(x, Q^2) region where the behaviour is nowadays rather well understood, PDFs do not risk turning negative.

For showers and MPIs, the case is less clear; they both connect to low-$p_\perp$ scales around or below 1 GeV, and especially MPIs can probe extremely small x values, down to around 10^{-8} at LHC energies, *cf.* section 6.2. In this region, all PDF components are poorly known, especially the dominant gluonic one. In an LO description, the PDFs are required to be non-negative, and HERA data in combination with Regge theory provide some reasonable constraints on the low-x behaviour. PDFs need not be positive definite at higher orders, NLO or NNLO, since it is only the convolution of NLO (NNLO) hard-process matrix elements with NLO (NNLO) PDFs that should be non-negative, up to NNLO (N^{3}LO) terms. Actually, at scales $p_\perp \sim 1$ GeV the whole perturbative expansion is poorly convergent, since α_s is large. Some recent PDFs attempt a resummed description of the small-x behaviour to restore a guaranteed PDF positivity [50]. Nevertheless, in general, the criteria for what constitutes an optimal or at least sensible PDF choice for the hard process are not necessarily the same as for showers and MPIs; for this reason, PYTHIA 8.3 allows for the use of one PDF set for the hard process and a different set for showers and MPIs. This can also be useful to preserve shower- and underlying-event tuning properties while changing PDFs for the hard process.

It is also possible to pick different PDF sets for the two incoming beam particles, which may be convenient as a technical trick but has no physics motivation when colliding beams are the same.

3.13 Phase-space cuts for hard processes

Several different phase-space cuts have been implemented for the internal hard processes in PYTHIA 8.3. These serve two purposes: to properly set values that ensure the approximations in the theory description are valid, and to allow for more efficient event generation when only a certain part of the available phase space is considered. The principal example is the lower limit of the partonic $p_\perp$ of $2 \to 2$ processes, that needs to be set to a high enough value such that the divergent behaviour of the massless matrix elements in the $p_\perp \to 0$ limit is avoided. Similarly, a suitable lower limit for $p_\perp$ should be applied when considering *e.g.* jet production at higher values of $p_\perp$, to avoid the inefficiency otherwise associated with a rapidly dropping

$p_\perp$ spectrum. (But also see comment at the end of this subsection.)

The number of implemented phase-space cuts for the hard scattering depends on the number of final-state particles of the process. For $2 \to 1$ only two options are included:

- the minimum invariant mass m_{min}

- the maximum invariant mass m_{max}

If the value of the latter is lower than the value of the former, the invariant mass will be limited from above by the collision energy. The same cuts also apply to $2 \to 2$ and $2 \to 3$ processes.

For $2 \to 2$ processes some more options appear. The first three are related to invariant transverse momentum of the process:

- the minimum transverse momentum $p_{\perp\text{min}}$

- the maximum transverse momentum $p_{\perp\text{max}}$

- an additional lower transverse-momentum cut $p_{\perp\text{diverge}}$

The latter is to prevent divergences in the $p_\perp \to 0$ limit for processes where a particle has a mass smaller than the set $p_{\perp\text{diverge}}$. In these cases, however, the larger of the $p_{\perp\text{min}}$ and $p_{\perp\text{diverge}}$ is always applied for the $p_\perp$ selection. The next set of cuts is related to limits of Breit–Wigner (BW) mass distributions. By default, the mass selection based on BW shapes is always applied for particles with a width above a certain threshold. There are two different thresholds that can be set:

- the minimum width of a resonance for which the Breit–Wigner shape can be deformed by the variation of the cross section across the peak;

- and the minimum width of a resonance that is below the former threshold, for which a simplified treatment is applied instead, where a symmetric Breit–Wigner selection is decoupled from the hard-process cross section.

Notice that the allowed mass range of a given particle can be set by modifying the particle properties. In case of DIS, instead of $p_\perp$, the most relevant phase-space cut is the lower limit for the allowed virtuality of the intermediate photon:

- minimum Q^2 for t-channel processes with non-identical particles

Notice that the cuts for $p_\perp$ will also be applied when a non-zero cut for Q^2 is applied.

For $2 \to 3$ processes that do not contain soft or collinear singularities, such as Higgs production in EW-boson fusion, the same cuts as in the $2 \to 2$ case can be applied. For QCD processes, where such singularities need to be accounted for, alternative cuts are defined. Also, since the outgoing partons are no longer back-to-back, cuts for individual partons can be used for a more detailed phase-space mapping:

- the minimum transverse momentum for the highest-$p_\perp$ parton

- the maximum transverse momentum for the highest-$p_\perp$ parton

- the minimum transverse momentum for the lowest-$p_\perp$ parton

- the maximum transverse momentum for the lowest-$p_\perp$ parton

- the minimum separation $R \, (= \sqrt{(\Delta\eta)^2 + (\Delta\phi)^2})$ between any two outgoing partons

The last one needs to have a high-enough value to avoid collinear divergences associated with the outgoing partons.

As described above, the phase-space cuts can be used to improve the sampling efficiency by focusing on a particular phase-space volume, _e.g._ defined by cuts on partonic $p_\perp$. In some cases this might, however, require several runs that need to be combined later on. Similar improvement in efficiency can also be achieved by reweighting the cross section of the hard process with a suitable kinematic variable. In PYTHIA 8.3 the events can easily be reweighted by $p_\perp^{-\alpha}$, where α is a power that could _e.g._ approximate the $p_\perp$ dependence of the hard cross section. This allows for a more uniform filling of the phase space, even when the cross section itself drops rapidly. The downside is that when reweighting is applied, each event comes with a weight that needs to be accounted for _e.g._ when filling histograms. In addition to this built-in reweighting of internally defined $2 \to 2$ hard processes, there are also more involved options for reweighting with different variables that can be enabled with the user hooks described in section 9.7.2.

An important aspect is that the described phase-space cuts are applied only for the hard scattering, _i.e._ before any showering or hadronization. As the shower emissions will modify the four-momentum of outgoing partons; a jet formed from the final particles will have a somewhat different transverse momentum than the parton that originated the jet. Final-state radiation and hadronization can reduce the energy of the jet, whereas initial-state radiation and multiparton interactions may enhance it. Therefore a "fiducial phase-space volume" is needed, _i.e._ hard processes must be generated in a larger volume than the volume of interest for final-state observables, at the unfortunate cost of generating many events that will be thrown away. The necessary amount of oversampling depends highly on the kinematics and beam configuration considered, so it needs to be checked case-by-case. For jet studies, this is usually done by plotting the hard-process $p_\perp$ associated with accepted jets or events. If a non-negligible fraction of events near the $p_{\perp\mathrm{min}}$ scale are accepted then $p_{\perp\mathrm{min}}$ is too high.

3.14 Second hard process

The MPI framework in PYTHIA will generate a variable number of partonic $2 \to 2$ interactions in addition to the selected hard process itself. These, mainly QCD processes, will form the underlying event, typically consisting of rather soft particles. Occasionally, they may also contain a hard scattering but, due to power-law falloff of the relevant cross sections, such events are rare. There are, however, cases when the studied observable is such that more control over the kinematics of the second scattering can significantly improve the sampling efficiency (_e.g._ of four-jet final states), or the second process is not included as a part of current MPI generation (_e.g._ the production of an EW boson together with a jet). The machinery for a second hard process can be used in these situations. It can be viewed as an approach to generate so-called Double Parton Scattering (DPS) events, but with two key distinctions. First, the DPS framework as used for theoretical studies typically assumes that there are exactly two hard interactions in an event, while the second-hard setup allows there to be further MPIs just like when starting out from one hard interaction. Second, the MPI machinery uniquely fixes how two hard cross sections should be combined into a total, while this usually involves a free parameter in the DPS expressions.

The basic approach in the PYTHIA implementation for the generation of two hard processes in a single event is that, first, the two processes are selected completely independently and, afterwards, momentum conservation and the possible correlations in the PDFs are accounted for by the rejection of a fraction of the topologies. This makes the process sampling symmetric and thus the distinction between "first" and "second" is used only for bookkeeping. Furthermore, as long as there is some overlap in phase space of the two processes, any of the two can be the hardest one. In principle, this construction would allow the generation of any two inter-

nally (or externally) defined processes, but in practice there is no need for a very fine-grained control of both processes, and furthermore the combination of two rare processes would give a negligible cross section. Therefore a somewhat more limited set of second processes have been implemented. Still, the first process can be selected from the complete list of processes (appendix A), or even provided externally. The processes that can be enabled as a second hard one include:

- standard QCD $2 \rightarrow 2$ processes, *i.e.* two-jet production

- a prompt photon and a jet

- two prompt photons

- charmonium production, colour singlet and octet

- bottomonium production, colour singlet and octet

- γ^*/Z production with full interference

- single $W^\pm$ production

- production of a γ^*/Z and a parton

- production of a $W^\pm$ and a parton

- top-pair production

- single-top production

- bottom-pair production

Technically these can be combined freely, but some combinations would double count and therefore must be avoided. This includes a $\gamma^*/Z/W^\pm$ together with a jet or on its own, and $b\bar{b}$ production as part of the QCD $2 \rightarrow 2$ processes or on its own. Also, since the last one will include only $b\bar{b}$ production explicitly in the hard scattering, the pairs produced by gluon splittings in the parton showers will not be present in that sample. Thus, depending on the kinematics, this might or might not be enough to give realistic cross-section estimates.

By default, the phase-space cuts, couplings, and scales for the second hard process are the same as for the primary scattering. It is, however, possible to set different cuts for the second one, and, due to fully symmetric treatment of the two processes, the cuts for the second process can be set higher or lower than for the primary one. The cuts that can be separately specified are the minimum and maximum values for the invariant mass and transverse momentum of the process.

It is instructive to consider some Poissonian statistics before showing how the cross sections of two processes should be combined. If the average number of subcollisions, $\langle n \rangle$, is known, the probability for n of them to occur is given by

$$P_n = \langle n \rangle^n \frac{e^{-\langle n \rangle}}{n!} \,. \tag{66}$$

In case where $\langle n \rangle$ is small, as it is for hard processes, we can approximate $e^{-\langle n \rangle} = 1$. The probability for one event to happen is then $P_1 = \langle n \rangle$, and correspondingly for two such events we find $P_2 = \langle n \rangle^2/2 = P_1^2/2$. Now consider two independent event types a and b, such that $\langle n \rangle = \langle n_a \rangle + \langle n_b \rangle = P_{1a} + P_{1b}$. The probability for any combination of two events a and b is then given by

$$P_2 = \frac{(P_{1a} + P_{1b})^2}{2} = \frac{P_{1a}^2 + 2P_{1a}P_{1b} + P_{1b}^2}{2} \,. \tag{67}$$

From this it can be read off that a probability for having two different-type events comes with a factor 2 relative to the same-type cases. If modelled in terms of increasing time, or decreasing hard scale (say $p_\perp$), the mixed combination can occur in two ways, either where the event a happens before b, or the other way around, which explains the factor of 2.

The proper way to evaluate the resulting cross section thus depends on whether the two processes are the same, and on whether the phase-space regions overlap. The simplest case is when the two processes do not overlap, *i.e.* either the phase-space regions are completely separated or the two process are different. An example of the latter would be a combination of processes where the first produces two jets and the second two photons. When the a and b cross sections are small fractions of the total non-diffractive cross sections σ_{ND}, naively the probabilities $P_{a,b} = \sigma_{a,b}/\sigma_{\mathrm{ND}}$ enter multiplicatively. Thus their combined cross section is

$$\sigma_{2ab}^{\mathrm{naive}} = P_a \, P_b \, \sigma_{\mathrm{ND}} = \frac{\sigma_{1a}\sigma_{1b}}{\sigma_{\mathrm{ND}}} \, . \tag{68}$$

This simplification neglects the dependence on collision geometry, however. The probability for a hard process is enhanced in central collisions, *i.e.* for small impact parameter, while it is depleted in peripheral ones. This leads to a so-called "trigger bias" effect, where events containing a first hard process predominantly occur in central collisions, which thereby enhances the likelihood of a second hard process. In the context of traditional MPIs this is known as the "pedestal effect", where a selected high-$p_\perp$ process has more underlying-event activity than an average event, see more details in section 6.2.2. When the colliding matter profiles have been specified, along with the parameters that set the $\langle n_{\mathrm{MPI}} \rangle$, a correction factor f_{impact} can be derived event-by-event within the MPI framework. Its average value gives a corrected combined cross section

$$\sigma_{2ab} = \langle f_{\mathrm{impact}} \rangle \frac{\sigma_{1a}\sigma_{1b}}{\sigma_{\mathrm{ND}}} = \frac{\sigma_{1a}\sigma_{1b}}{\sigma_{\mathrm{eff}}} \, . \tag{69}$$

In the last step we introduce σ_{eff}, which is the conventional parameter that many experimental results are expressed in terms of, but here it is a prediction of the model.

The cross section σ_{2aa} of two identical processes follows the same pattern, except for the extra factor of $1/2$ that has already been explained. Often a would itself be the sum of several subprocesses, *e.g.* the six main classes of $2 \rightarrow 2$ QCD processes that contribute to two-jet production. If so, then a compensating factor of 2 will automatically occur for the mixed-subprocess configurations, in the same spirit as eq. (67).

The cross section calculation becomes somewhat more complicated in cases when there is partial overlap between the two processes. An example would be identical processes with different, but partly overlapping, cuts on $p_\perp$. In such cases it is useful to split the problem into two completely independent processes a and b and a common process c. The first (second) process can be selected according to $\sigma_a + \sigma_c$ ($\sigma_b + \sigma_c$). Half of the events should be discarded if both processes are chosen as c, and the combined cross section should be reduced accordingly.

So far it has been assumed that the generation of the two processes can be done independently, apart from the geometrical correction factor for the final cross sections. This obviously misses all possible correlations between the PDFs and, perhaps more importantly, may violate energy-momentum conservation. Part of the selected events will be discarded to account for these effects, even though each process would be acceptable on its own. The correlations in multiparton PDFs implemented in PYTHIA are described further in section 6.2.4. The PDF reduction factor is obtained as the average of the two possible orderings, where either the second or first PDF is corrected for the parton taken out either by the first or second process.

In the end, the cross sections provided by PYTHIA after the event generation do account for all these effects, including the correction factor $\langle f_{\mathrm{impact}} \rangle$ and the PDF rescaling. The error estimates provided by PYTHIA are statistical ones and do not cover the potentially large model uncertainties, as usual. When the first process is provided externally, PYTHIA does not have

the information whether there is an overlap between the first and the second process, and so will assume that this is not the case. The proper correction for an overlap then rests with the user.

4 Parton showers

The most violent pp collisions at the LHC may have five to ten easily separated jets. Zooming in on these, they display a substructure of jets-inside-jets-inside-jets, associated with the perturbative production of increasingly nearby partons. Such a fractal nature is expected to continue down to the hadronization scale, a bit below 1 GeV. At that scale, the event may contain up to a hundred partons, even if the full partonic structure is masked by the subsequent non-perturbative hadronization process. There is no way to perform matrix-element calculations to describe such complicated event topologies. Instead, the standard approach is to start out from a matrix-element calculation with only a few well-separated partons, and then apply a parton shower to that.

Parton showers attempt to describe how a basic hard process is dressed up by emissions at successively "softer" (longer-wavelength) and/or more "collinear" (smaller-angle) resolution scales, to give an approximate but realistic picture of the (sub)structure of the partonic state across the full range of (perturbative) resolution scales. Such a shower is constructed in a recursive manner, from the large scale of the hard process down to a lower cutoff at around the hadronization scale. In each step, the number of partons is increased by one, or in very special cases two, and the random nature of the steps leads to a large variability of final states. It is worth emphasizing that, although often thought of in the context of QCD, parton (or more generally particle) showers are in fact common to any quantum field theory with several (quasi-)massless particles. Thus, showers are present in QCD, QED, and the EW theory above the symmetry breaking scale and as such, dedicated modules describing all of these are part of PYTHIA 8.3.

One starting point is to study the ratio of two differential matrix elements, $d\sigma_{n+1}/d\sigma_n$, where the numerator corresponds to the emission of one more gluon in the final state. It then turns out that this ratio is given by universal expressions, *i.e.* independent of which specific process is considered, if this gluon is either soft, or collinear with one of the already existing partons. This means that one can formulate a generic scheme that can be applied to any process of interest. Such schemes started to be developed in the late 1970s. A key ingredient has been the DGLAP evolution equations [43–45], which describe near-collinear emissions. Modern showers, like the three available with PYTHIA 8.3, are based on many subsequent developments, intended to make them cover the full phase space as well as possible. These aspects are described later, but initially we introduce the simpler, classical (collinear "leading-log") framework that helps in understanding the overall picture.

Historically, showers are split into two kinds, ISR and FSR, which occur respectively before or after the hard process. Alternatively, they may be referred to as spacelike and timelike showers, respectively, since their representation in terms of Feynman diagrams contain off-shell intermediate particles that are either spacelike or timelike. The more virtual such a particle is, the shorter it may exist. Therefore, the highest virtualities occur in and closest to the hard interaction, and then showers with decreasing virtualities stretch backwards (for ISR) or forwards (for FSR) in time. LHC processes usually contain both ISR and FSR, and outside the strictly collinear limits the distinction can be blurred, just like interfering Feynman graphs of a different nature may contribute to a given final state. A decay $\gamma^*/Z \to q\bar{q}$ is pure FSR, however, while its production $q\bar{q} \to \gamma^*/Z$ can be discussed in terms of ISR only, so these are often used as textbook examples. (Conversely, ISR-FSR interference can be exemplified by

t-channel colour-singlet exchange, such as in deep inelastic scattering or vector boson fusion.)

FSR   Starting from $\gamma^*/Z \to q\bar{q}$, either the q or $\bar{q}$ may emit a g, *e.g.* q $\to$ qg. This produces a q$\bar{q}$g state, after which either of the three partons may branch, and so on. The differential probability for a parton to branch can be written as

$$d\mathcal{P}_a(z, Q^2) = \frac{dQ^2}{Q^2} \frac{\alpha_s(Q^2)}{2\pi} \sum_{b,c} P_{a \to bc}(z) \, dz \,. \tag{70}$$

Here a is the mother that splits into partons b and c, where the momentum-energy of the mother is split such that b takes fraction z and c takes $1-z$. The Q scale, used to order emissions in a falling sequence, is a key distinguishing feature of different shower implementations, and may be chosen *e.g.* as mass, transverse momentum or energy-weighted emission angle. That is, z parameterizes the longitudinal and Q the transverse evolution of the shower. There is also an azimuthal angle φ that determines the orientation of the decay plane; typically, and for the purpose of this brief introduction, this is assumed to be distributed isotropically, though we note that PYTHIA does allow for non-uniform distributions as well, *e.g.* to reflect gluon polarization effects.

A key issue that distinguishes parton showers from so-called analytic resummation approaches, is that the latter only maintain exact energy and momentum conservation in the strict soft and collinear limits while showers do so over all of phase space. This difference leads to the crucial aspect of *recoil effects* in parton showers, which will play an important role when we introduce dipole showers later on.

There are three different DGLAP splitting kernels,

$$P_{q \to qg}(z) = \frac{4}{3} \frac{1 + z^2}{1 - z} \,, \tag{71}$$

$$P_{g \to gg}(z) = 3 \frac{\left(1 - z(1-z)\right)^2}{z(1-z)} \,, \tag{72}$$

$$P_{g \to q\bar{q}}(z) = \frac{1}{2} \left(z^2 + (1-z)^2\right) \,. \tag{73}$$

These obey the trivial symmetry relations $P_{a \to cb}(z) = P_{a \to bc}(1-z)$. The $P_{g \to q\bar{q}}$ kernel is normalized for one quark flavour only, and has to be summed over all kinematically allowed channels.

The same approach can also be used for other branchings, notably QED ones, where α_s in eq. (70) is replaced by α_{em} and the splitting kernels are

$$P_{f \to f\gamma}(z) = e_f^2 \frac{1 + z^2}{1 - z} \,, \tag{74}$$

$$P_{\gamma \to f\bar{f}}(z) = N_c \, e_f^2 \left(z^2 + (1-z)^2\right) \,, \tag{75}$$

where $N_c = 3$ if f is a quark and $N_c = 1$ if a charged lepton.

The DGLAP kernels are often written with additional terms that modify the behaviour at $z = 1$ and 0, in order to conserve momentum-energy and flavour in analytic calculations. This is not necessary in event generators, partly because the 0 and 1 limits are never reached, and partly because conservation issues are handled explicitly: parton a is removed at the same time as b and c are inserted in the list of currently existing partons.

The branching probability in eq. (70) can be integrated over the kinematically allowed z range

$$d\mathcal{P}_a(Q^2) = \frac{dQ^2}{Q^2} \frac{\alpha_s(Q^2)}{2\pi} \sum_{b,c} \int_{z_{min}(Q^2)}^{z_{max}(Q^2)} P_{a \to bc}(z) \, dz \,, \tag{76}$$

to express the infinitesimal probability that a branches in a dQ^2 infinitesimal step. (Strictly speaking $|dQ^2|$ since Q^2 is decreasing in the evolution.) The probability for a *not* to branch in the same step thus is $1 - d\mathcal{P}_a(Q^2)$. By multiplication of the no-emission probabilities (exponentiation), the probability for a not to branch between an initial scale Q_1^2 and a final lower Q_2^2 becomes the Sudakov factor [51]

$$\Pi_a(Q_1^2, Q_2^2) = \exp\left(-\int_{Q_2^2}^{Q_1^2} d\mathcal{P}_a(Q^2)\right). \tag{77}$$

The differential probability for a to evolve from a Q_{max}^2 to a Q^2 and then branch at the latter scale, thus is $\Pi_a(Q_{\mathrm{max}}^2, Q^2) d\mathcal{P}_a(Q^2)$. Note that the introduction of a Sudakov factor ensures that the total probability for a to branch cannot exceed unity, something that is not guaranteed for $d\mathcal{P}_a$ alone.

We observe that the Sudakov factor plays a crucial role in the selection of a branching scale. The veto-algorithm technology in section 2.2.3 is eminently suited to handle cases where the Q^2 and z integrations cannot be done analytically. The Sudakov factor also is closely related to virtual corrections of matrix elements, *i.e.* loop corrections. This will play a key role for the matching and merging methods presented in the next section.

ISR   The ISR description starts out from the evolution equation for Parton Distribution Function (PDF)s,

$$\begin{aligned}
df_b(x, Q^2) &= \frac{dQ^2}{Q^2} \frac{\alpha_{\mathrm{s}}(Q^2)}{2\pi} \sum_a \int f_a(x', Q^2) dx' \int P_{b/a}(z) dz\, \delta(x - x'z) \\
&= \frac{dQ^2}{Q^2} \frac{\alpha_{\mathrm{s}}(Q^2)}{2\pi} \sum_a \int \frac{dz}{z} f_a\left(x' = \frac{x}{z}, Q^2\right) P_{b/a}(z),
\end{aligned} \tag{78}$$

where $f_i(x, Q^2)$ is the probability to find a parton i inside a hadron, with i carrying a fraction x of the full hadron momentum if the hadron is probed at a scale Q^2.

As for FSR, the evolution is driven by branchings $a \to bc$ but, where FSR is formulated in terms of the decay rate of a, ISR is given in terms of the production rate of b. The simple splitting kernels are easily related, $P_{b/a}(z) = P_{a \to bc}(z)$, except that $P_{g/g}(z) = 2P_{g \to gg}(z)$, since two gluons are produced for each gluon that decays.

The evolution of PDFs starts at some low scale Q_0^2 and then proceeds towards the Q^2 scale of the hard process, where they enter into the cross-section expression. While eq. (78) describes the evolution of an inclusive distribution, an exclusive shower formulation similar to the FSR one is possible, although more complicated. A key problem is that the two incoming cascades, one from each side of the event, may not end up as the colliding partons one is interested in. For example, in $gg \to H$ the two incoming gluons must have an invariant mass that matches the Higgs mass.

The solution to this problem is backwards evolution [52]. In this method, the evolved PDFs are first used to select the hard process of interest, say $q\bar{q} \to \gamma^*/Z$. Only afterwards are the incoming showers then constructed backwards in time, from the high Q^2 scale down to the low Q_0^2. To this end, we introduce

$$d\mathcal{P}_b(x, Q^2) = \frac{df_b(x, Q^2)}{f_b(x, Q^2)} = \frac{dQ^2}{Q^2} \frac{\alpha_{\mathrm{s}}(Q^2)}{2\pi} \sum_a \int_{z_{\mathrm{min}}(Q^2)}^{z_{\mathrm{max}}(Q^2)} dz\, \frac{x' f_a(x', Q^2)}{x f_b(x, Q^2)} P_{b/a}(z), \tag{79}$$

where we have used that $z = x/x'$. Here $d\mathcal{P}_b$ is the probability that parton b becomes associated with a branching $a \to bc$ during the interval dQ^2. A no-branching probability

$\Pi_b(x, Q_1^2, Q_2^2)$ can be defined in analogy with the Sudakov factor eq. (77). The corrected probability for a parton b that branches or interacts at Q_{max}^2 to be assigned a mother a at Q^2 then is $\Pi_b(x, Q_{max}^2, Q^2) \, d\mathcal{P}_b(x, Q^2)$. This a in its turn must be evolved to yet lower Q^2 to find its mother at an even higher x value.

Recoils and dipoles   An isolated parton cannot branch, if energy and momentum is to be preserved. Take as an example $\gamma^*/Z \to q\bar{q} \to q^*\bar{q} \to q\bar{q}g$, where q^* is the off-shell quark that branches as $q^* \to qg$. Initially, the q and $\bar{q}$ can split the energy equally, but the off-shell q^* acquires a larger mass than $\bar{q}$, and so it must have a larger energy while the $\bar{q}$ receives a smaller one. In this case we would call q the radiator (or emitter) and $\bar{q}$ the recoiler, but note that at the end, both may yield energy to create the g. Also, considering the existence of $\bar{q} \to \bar{q}g$ branchings, it may be simpler to say that it is the q$\bar{q}$ pair that jointly radiates the g. Note that q and $\bar{q}$ have opposite and compensating colours and thus form a colour dipole, hence the concept of dipole radiation.

This picture generalizes to the subsequent emission of further gluons [53, 54]. In the limit of infinitely many colours, $N_c \to \infty$ [55], the q$\bar{q}$g system exactly splits into one qg dipole and one g$\bar{q}$ dipole. These can radiate independently, and the recoil is distributed within each dipole. It is still possible, but not necessary, to split the radiation inside each dipole as being associated with either dipole end.

To allow dipole showers to operate, unique colour indices (in the $N_c \to \infty$ limit) are assigned to all coloured partons, both ones produced in the hard process and ones in the subsequent shower evolution. For the extension to ISR, and to decays like $t \to bW^+$, one should note that the hole left behind by a scattered or decayed colour parton can act like its anticolour.

Formal basis of parton showers   In the previous discussion, we have developed the basic idea of parton showers, similarly to the historical development. We now want to turn to a more in-depth treatment about the formal basis of modern shower algorithms as the three implemented in PYTHIA 8.3.

We have seen above that parton showers build upon the factorization of (squared) amplitudes in soft and collinear limits. Technically, this means that whenever either two (or more) particles become collinear or one (or more) particle becomes soft, the full (squared) amplitude can be well approximated by the (squared) matrix element without the unresolved particle times a universal radiation function. It is the latter, which takes the effect of the soft or collinear radiation into account. This factorization is reminiscent of the perturbative physics of the hard process and occurs, because an intermediate, almost on-shell, propagator can be replaced by a polarization sum, such that the amplitude may be split into two independent pieces. Vital for the construction of showers is that this factorization is universal in the sense that it is process and multiplicity independent. This means that the same radiation functions can be used for different squared matrix elements and at any multiplicity, as long as only single-unresolved radiation is concerned. The latter comment serves to emphasize that at higher multiplicities also multiple-unresolved limits occur, in which, for instance, two particles become simultaneously soft or three particles become simultaneously collinear. For such configurations, it should be obvious that higher-order radiation functions are needed and the ones describing single-soft or (double-)collinear radiation are not sufficient. At the same time, it is always possible to factorize phase-space integration measures into on-shell steps by introducing delta functions to factorize the decay system, and introducing recoiling systems to guarantee four-momentum conservation. Taking matrix-element and phase-space factorization together, it follows that cross sections can be factorized. This allows for iteration of the approximation, as long as the measure of "softness" or "collinearity" remains appropriate. In

this context, the requirement of an appropriate measure leads to the notion of strong ordering, which means that radiation of soft particles is yet softer and radiation of collinear particles is yet more collinear. Although different possibilities to factorize matrix elements exist, all inherit that the approximation should recover the singularities of fixed-order results. On the one hand, DGLAP evolution is driven by collinear radiation; on the other hand, factorization-breaking (so-called non-global) logarithms are driven by soft radiation. These are the limits any parton shower resumming the leading, *i.e.* largest, logarithms should recover.

Based on the above, we can start thinking about the construction of a shower model. It should be emphasized that the construction of showers is by no means unique. As the bare minimum, a shower algorithm must define the following.

1. Radiation functions, *i.e.* the matrix-element factorization.

2. A phase-space factorization and recoil procedure.

3. An ordering variable, *i.e.* a measure of "softness" and/or "collinearity".

For each of these points, different choices are possible and used, motivated by different desires to obtain certain objectives: simplicity, extendability, or simply to describe specific processes better at the cost of describing others worse.

Using somewhat general language for now, we can denote the radiation functions by $K_{j/\tilde{i}\tilde{k}}$, describing the radiation of particle j from the two parent particles $\tilde{i}$ and $\tilde{k}$, *i.e.* the branching $\tilde{i}\tilde{k} \mapsto ijk$. Depending on the specifics of the shower algorithm, one of the two parents $\tilde{i}$ and $\tilde{k}$ may be distinguished as the "emitter" while the other, the "recoiler", only ensures four-momentum conservation, or both parents act as emitters and recoilers in an agnostic way. The former is how both PYTHIA's simple shower and DIRE are structured, whereas the latter describes the antenna picture employed in the VINCIA shower. No matter which specific choice of radiation functions is made, the sum of terms must reproduce all single-unresolved limits of the full real-emission cross section,

$$\mathrm{d}\sigma_{n+1} \xrightarrow{\text{single-unresolved}} \sum_j K_{j/\tilde{i}\tilde{k}}\,\mathrm{d}\Phi_{+1}\,\mathrm{d}\sigma_n =: K_{n\mapsto n+1}\,\mathrm{d}\Phi_{+1}\,\mathrm{d}\sigma_n \,, \tag{80}$$

with the cross sections σ defined as in eq. (37). This factorization consists of two parts: the factorization of the squared matrix element and the factorization of the phase space.

Specifically, in the case of two particles i and j becoming collinear, the $n+1$-particle matrix element factorizes into a product of the n-particle matrix element and the DGLAP splitting kernels eqs. (71) to (75),

$$|\mathcal{M}_{n+1}|^2 \xrightarrow{i\|j} \frac{8\pi\alpha}{2p_i \cdot p_j} P_{\tilde{i}\to ij}(z)\,|\mathcal{M}_n|^2 + \text{angular terms}\,. \tag{81}$$

Generally, the collinear limit involves spin correlations between the factorized matrix element and the (spin-dependent) DGLAP kernels, here indicated by the additional "angular terms". These terms vanish upon azimuthal integration and are therefore not necessarily implemented in a parton-shower algorithm. It is, however, vital to account for these terms in so-called NLO subtraction schemes to ensure point-wise cancellation of singularities. In the limit of a single gauge boson becoming soft, however, the emission of the soft boson can be described by a universal factor known as the soft eikonal. Different to the collinear limit, soft radiation is an intrinsically coherent phenomenon, meaning that the boson is emitted by the whole particle ensemble, introducing a sum over radiators:

$$|\mathcal{M}_{n+1}|^2 \xrightarrow{E_j \to 0} 8\pi\alpha \sum_{i<k} \mathcal{C}_{ik} \frac{2p_i \cdot p_k}{(2p_i \cdot p_j)(2p_j \cdot p_k)}\,|\mathcal{M}_n|^2 \,, \tag{82}$$

with charge factors $\mathcal{C}_{ik}$ depending on the charges of the radiators i and k. Especially in the case of QCD, these charge factors introduce intricate colour correlations for soft gluon emissions. It is because of these complications that most parton showers only consider the leading-colour limit, *i.e.* neglect any contributions in the above sum that correspond to emissions from non-neighbouring partons.

Besides the factorization of matrix elements, in eq. (80) we used that the $n + 1$-particle phase space exactly factorizes into a product of an n-particle phase space and the branching phase space $\mathrm{d}\Phi_{+1}$, obtained through a formal insertion of an intermediate off-shell particle with mass $m_{ij}^2 = (p_i + p_j)^2$,

$$
\begin{aligned}
\mathrm{d}\Phi_{n+1}(q; p_1, \ldots, p_i, p_j, p_k, \ldots, p_{n+1}) &= \mathrm{d}\Phi_n(q; p_1, \ldots, p_{\tilde{\imath}}, p_{\tilde{k}}, \ldots, p_n) \\
&\quad \times \left| J(p_{\tilde{\imath}}, p_{\tilde{k}}; p_{ij}, p_k) \right| \frac{\mathrm{d}m_{ij}^2}{2\pi} \mathrm{d}\Phi_2(p_{ij}; p_i, p_j) \\
&\equiv \mathrm{d}\Phi_n(q; p_1, \ldots, p_{\tilde{\imath}}, p_{\tilde{k}}, \ldots, p_n) \, \mathrm{d}\Phi_{+1}(p_i, p_j, p_k). \quad (83)
\end{aligned}
$$

It must be emphasized that the n-particle phase-space measure is here written with on-shell momenta $p_{\tilde{\imath}}$ and $p_{\tilde{k}}$ instead of an off-shell intermediate momentum p_{ij}. This means we here assume an on-shell phase-space factorization, *i.e.* that after each emission, all momenta are separately physical and momentum is conserved at each step in the shower,

$$
p_{\tilde{\imath}} + p_{\tilde{k}} = p_i + p_j + p_k. \quad (84)
$$

The change from the off-shell momenta $\{p_{ij}, p_k\}$ to the on-shell momenta $\{p_{\tilde{\imath}}, p_{\tilde{k}}\}$ is represented by the Jacobian $\left| J(p_{\tilde{\imath}}, p_{\tilde{k}}; p_{ij}, p_k) \right|$. Specific forms of kinematic mappings $\{p_{\tilde{\imath}}, p_{\tilde{k}}\} \mapsto \{p_i, p_j, p_k\}$ (or "recoil schemes") are again shower specific. Presently, however, all showers in PYTHIA 8.3 employ an on-shell factorization as described here. While this might not generally be required, this is a key requirement for the matching and merging techniques utilized in PYTHIA 8.3, *cf.* section 5.

The branching phase space $\mathrm{d}\Phi_{+1}$ accounts for the degrees of freedom entering through the emission of one particle from the n-particle configuration and can generally be expressed in terms of three "shower variables" t, z, and ϕ,

$$
\mathrm{d}\Phi_{+1}(p_i, p_j, p_k) = |J(t, z, \phi)| \, \mathrm{d}\Phi_{+1}(t, z, \phi) = \frac{1}{16\pi^2} |J(t, z, \phi)| \, \mathrm{d}t \, \mathrm{d}z \, \mathrm{d}\phi. \quad (85)
$$

Usually, t is interpreted as the ordering variable of the shower, z as some kind of energy-sharing variable, and ϕ as the angle about the branching plane in the i-j-k rest frame. However, different showers make different choices which may be more or less connected with this analogy.

Addressing point 3 of the list above, it is instructive to start by noting that various choices of ordering variables are formally equivalent at the Leading Logarithmic (LL) level, as can be seen by comparing the differentials as they enter through the matrix-element and phase-space factorizations described above,

$$
\frac{\mathrm{d}t}{t} = \frac{\mathrm{d}p_{\perp,j}^2}{p_{\perp,j}^2} = \frac{\mathrm{d}m_{ij}^2}{m_{ij}^2} = \frac{\mathrm{d}\theta_{ij}^2}{\theta_{ij}^2}, \quad (86)
$$

and by noting that in the collinear limit $p_{\perp,j}^2 \sim z(1-z)m_{ij}^2 \sim z^2(1-z)^2 E_j^2 \theta_{ij}^2$. It is straightforward to see that all these choices represent a certain measure of softness or collinearity, as required above. The requirement that this measure remains appropriate during the shower evolution then translates into strong ordering of emissions, *i.e.* subsequent emissions evolve down in the ordering scale: $t_0 > t_1 > t_2 > \ldots t_n$.

Putting the above together, a no-branching probability, often also called Sudakov factor, can be defined:

$$
\Pi_n(t_n, t_{n+1}; \Phi_n) = \exp\left\{ -\int_{t_{n+1}}^{t_n} K_{n\mapsto n+1}(\Phi_n, \Phi_{+1}(t', z', \phi')) \, d\Phi_{+1}(t', z', \phi') \right\}.
\tag{87}
$$

It describes the evolution from an n-particle state at scale t_n to an $n+1$-particle state at scale $t_{n+1} < t_n$. By rewriting $K_{n\mapsto n+1}$ as the sum of radiation functions $K_{j/\tilde{i}\tilde{k}}$, Π_n can written as the product of $\tilde{i}\tilde{k} \mapsto ijk$ no-branching probabilities:

$$
\begin{aligned}
\Pi_n(t_n, t_{n+1}; \Phi_n) &= \exp\left\{ -\sum_j \int_{t_{n+1}}^{t_n} \int_{z_{\min}}^{z_{\max}} \int_0^{2\pi} \frac{1}{16\pi^2} K_{j/\tilde{i}\tilde{k}}(t', z', \phi') \left| J(t', z', \phi') \right| \frac{d\phi'}{2\pi} dz' \, dt' \right\} \\
&= \prod_j \exp\left\{ -\int_{t_{n+1}}^{t_n} \int_{z_{\min}}^{z_{\max}} \int_0^{2\pi} \frac{1}{16\pi^2} K_{j/\tilde{i}\tilde{k}}(t', z', \phi') \left| J(t', z', \phi') \right| \frac{d\phi'}{2\pi} dz' \, dt' \right\} \\
&= \prod_j \Pi_{j/\tilde{i}\tilde{k}}(t_n, t_{n+1}; \Phi_n).
\end{aligned}
\tag{88}
$$

Written this way, it is also emphasized that each branching $\tilde{i}\tilde{k} \mapsto ijk$ comes with its own branching phase space and kinematic mapping. This is how the full no-branching probability $\Pi_{n\mapsto n+1}$ is implemented in shower algorithms in practice.

For the calculation of the expected value of an observable O, the no-branching probabilities enter to describe the shower evolution as a Markov chain,

$$
\langle O \rangle_n^{\mathrm{PS}} = \int \frac{d\sigma_n}{d\Phi_n} \mathcal{S}_n(t, O) \, d\Phi_n,
\tag{89}
$$

which is generated by a "shower operator" $\mathcal{S}_n(t, O)$, defined recursively as

$$
\mathcal{S}_n(t, O) := \Pi_n(t, t_c; \Phi_n) O(\Phi_n) + \int_{t_c}^t K_{n\mapsto n+1} \Pi_n(t, t'; \Phi_n) \mathcal{S}_{n+1}(t', O) \, d\Phi_{+1}(t', z', \phi').
\tag{90}
$$

This shower operator makes the unitarity of the shower explicit. The first term implicitly accounts for all unresolved radiation and virtual corrections between the shower starting scale t and the shower cutoff t_c. The second term, on the other hand, describes the emission of a single particle, approximated by the sum of radiation functions $K_{n\mapsto n+1}$, and includes all unresolved and virtual corrections between the shower starting and cutoff scale.

It is instructive to make the form of the no-branching probability eq. (88) more explicit for QCD showers. Implicitly, the radiation functions $K_{j/\tilde{i}\tilde{k}}$ above contain the strong-coupling constant, a colour factor, and, for ISR, a ratio of PDFs,

$$
K_{j/\tilde{i}\tilde{k}}(t, z, \phi) = g_s^2(t) R_{\mathrm{PDF}}(t, z) \mathcal{C}_{j/\tilde{i}\tilde{k}} \bar{K}_{j/\tilde{i}\tilde{k}}(t, z, \phi) = 4\pi\alpha_s(t) R_{\mathrm{PDF}}(t, z) \mathcal{C}_{j/\tilde{i}\tilde{k}} \bar{K}_{j/\tilde{i}\tilde{k}}(t, z, \phi),
\tag{91}
$$

where we have introduced the coupling-, PDF-, and colour-factor-stripped radiation function $\bar{K}_{j/\tilde{i}\tilde{k}}$, which depends solely on the branching kinematics. For FSR, the PDF ratio is equal to

unity, $R_{\mathrm{PDF}} = 1$, as the initial-state momenta do not change due to the branching. Differentially in the evolution variable t, the integral in the exponent of $\Pi_{j/\tilde{\imath}\tilde{k}}$ can thus be written as

$$\frac{\mathrm{d}\mathcal{P}_{j/\tilde{\imath}\tilde{k}}^{\mathrm{FSR}}(t)}{\mathrm{d}t} = \frac{\alpha_{\mathrm{s}}(t)}{2\pi} \frac{\mathcal{C}_{j/\tilde{\imath}\tilde{k}}}{2} \int\limits_{z_{\min}}^{z_{\max}} \int\limits_{0}^{2\pi} \bar{K}_{j/\tilde{\imath}\tilde{k}}(t,z',\phi') \left| J(t,z',\phi') \right| \frac{\mathrm{d}\phi'}{2\pi} \,\mathrm{d}z' , \qquad (92)$$

$$\frac{\mathrm{d}\mathcal{P}_{j/\tilde{\imath}\tilde{k}}^{\mathrm{ISR}}(t)}{\mathrm{d}t} = \frac{\alpha_{\mathrm{s}}(t)}{2\pi} \frac{\mathcal{C}_{j/\tilde{\imath}\tilde{k}}}{2} \int\limits_{z_{\min}}^{z_{\max}} \int\limits_{0}^{2\pi} R_{\mathrm{PDF}}(t,z) \bar{K}_{j/\tilde{\imath}\tilde{k}}(t,z',\phi') \left| J(t,z',\phi') \right| \frac{\mathrm{d}\phi'}{2\pi} \,\mathrm{d}z' , \qquad (93)$$

for FSR and ISR, respectively. Written this way, the connection to eqs. (76) and (79), respectively, is immediately evident. It is worthwhile to point out here that typically different shower algorithms are inconsistent as to whether colour factors are included or excluded in radiation functions. Moreover, depending on whether a shower aims at describing the evolution of a single initial-state leg at a time or both at the same time, the PDF ratios R_{PDF} have to include one PDF ratio,

$$R_{\mathrm{PDF}}(t,z) = \frac{x_i f_i(x_i,t)}{x_{\tilde{\imath}j} f_{\tilde{\imath}}(x_{\tilde{\imath}},t)} , \qquad (94)$$

or two PDF ratios if both initial-state particles are evolved at the same time,

$$R_{\mathrm{PDF}}(t,z) = \frac{x_i f_i(x_i,t)}{x_{\tilde{\imath}} f_{\tilde{\imath}}(x_{\tilde{\imath}},t)} \frac{x_k f_k(x_k,t)}{x_{\tilde{k}} f_{\tilde{k}}(x_{\tilde{k}},t)} . \qquad (95)$$

The x-fractions $p_i = x_i P$, with P the incoming hadron momentum, depend on the shower variables t and z.

A similar analysis can be done in the cases of QED or EW showers, where the QCD coupling has to be replaced by the electromagnetic/electroweak coupling and QCD colour factors by the appropriate QED/EW charges.

Formal accuracy   Despite their success in describing wide classes of observables with often impressive agreement with experimental data, parton showers commonly work with a number of approximations. It is not an easy task to formally assess the accuracy of a given shower model, *i.e.* to determine which exact terms of a perturbative series a shower includes. For a start, there are three expansions to be considered:

1. the perturbative expansion in the coupling constant $\alpha^n(t)$, determining the accuracy of the hard process, *e.g.* leading-order (LO), next-to-leading order (NLO), *etc.*;

2. the perturbative expansion in large logarithms $\alpha^n(t)\log^m(t_{\mathrm{hard}}/t)$, determining the accuracy of the resummation, *e.g.* Leading Logarithmic (LL), Next–to–Leading Logarithmic (NLL), *etc.*;

3. and for QCD showers, the expansion in the number of colours (N_{c}), determining the accuracy of the colour factors in the resummation, *e.g.* Leading Colour (LC), Next–to–Leading Colour (NLC), *etc.*

A baseline shower would for example start from a LO matrix element and (typically) generate the LL corrections arising from additional radiation under the LC assumption of planar colour flows. Such a shower could be assigned a LO+LL+LC accuracy. This can be expected from virtually all common shower models, although observables may exist for which a given shower does not correctly include the LL terms. It is more interesting, however, to determine if and

for which observables showers reach sub-leading, *i.e.* higher, accuracy than the LO+LL+LC minimum.

Increasing the accuracy on the fixed-order side can be addressed by matching and merging methods, which are described in detail in section 5. Matching and merging at LO and NLO have de-facto become state of the art for all showers and processes.

Assessing and increasing the logarithmic accuracy of showers has become a highly-active field, where no general solution has yet been developed. Different approaches to assess the logarithmic accuracy of showers have been developed in the recent past, such as ones based on comparison of analytic and numerical resummation [56, 57], analytic examination of the logarithmic structure of showers [58, 59], or numerical checks of logarithmic terms [60, 61]. Moreover, for simple processes such as e^+e^- annihilation to jets, first shower models have been developed that can be shown to give NLL accuracy for a wider range of observables [62, 63]. Most common shower models currently only obtain a formal LL accuracy, with varying, observable-dependent subleading accuracy.

Lastly, the inclusion of sub-leading colour corrections in parton showers is an active field as well, with approaches based on matrix-element corrections [59, 64–68], sampling of colours [69, 70], quantum-probability density-matrix arguments [71, 72], or amplitude-level evolution [73, 74]. Sub-leading colour corrections are not in general universally applied in parton showers.

Showers in PYTHIA 8.3   There are three different shower modules available in PYTHIA 8.3: the original/default simple shower, the VINCIA antenna shower, and DIRE. These will be discussed in detail below in section 4.1, section 4.2, and section 4.3, respectively.

4.1   The simple shower

The "simple shower" is the oldest parton-shower algorithm in PYTHIA 8 and is also the default shower model in PYTHIA 8.3. It has its origin in the mass-ordered showers in JET-SET/PYTHIA [52, 75–77], with the transition to $p_\perp$ ordering [78] partly influenced by the Lund dipole picture [54] and partly by the desire to combine the ISR and FSR shower evolution with MPI in a single interleaved sequence [78].

Over the years, significant revisions and extensions have been introduced, many of them only available in recent PYTHIA versions. This includes:

- Full interleaving of ISR, FSR, and MPI [79].

- Options for a dipole-style treatment of initial-final colour flows [80].

- $f \to f\gamma$ and $\gamma \to f\bar{f}$ splittings (where f represents charged fermions).

- Matrix element corrections for resonance decays and a few other processes [76, 77, 81].

- Extensive facilities for matching and merging (*cf.* section 5).

- Reweighted shower branchings and uncertainty bands [20].

- A flexible treatment of showers in baryon-number-violating processes [82].

- Weak showers [83].

- Hidden-sector showers [38, 39].

The name "simple" shower here refers to the limited aim of a consistent leading-logarithmic (and beyond) shower evolution, with several known shortcomings [58, 59], as opposed to the more sophisticated goals of the alternative VINCIA (*cf.* section 4.2) and DIRE (*cf.* section 4.3) shower options, also available in PYTHIA 8.3. It should be emphasized that, by virtue of its longer history, many features are only available in the simple shower, and that as such the naming might be slightly misleading. As an example, the simple shower offers a much larger selection of matching and merging methods than does VINCIA or DIRE.

The shower machinery consists of one algorithm for FSR and one for ISR. These two are evolved together into one combined sequence of decreasing $p_\perp$ scales. As an example, consider a partonic process $a + b \to c + d$, where a and b are extracted from the beams A and B. It is then possible for c and d to undergo FSR branchings, and for a and b backwards-evolution ISR ones. Starting from some maximal scale $p_{\perp\max}$, downwards evolution gives a possible branching $p_\perp$ scale for each of the four partons. The one with largest $p_\perp$ is the winner that undergoes a branching, leading to a new state of five partons. The selected $p_\perp$ value is taken as the new starting point for all five partons to evolve further down in $p_\perp$, giving a new branching. This is applied iteratively until some lower cutoff is reached and the evolution is stopped. Also, MPIs will form part of this evolution, see section 6.2.5.

4.1.1 Basic shower branchings

The description of showers in the introduction of this section is valid for the simple shower framework. Notably the branching probabilities $d\mathcal{P}_a$ of eq. (76) and $d\mathcal{P}_b$ of eq. (79) play a central part, but with two key additions.

One is that evolution is performed in terms of transverse momenta, *i.e.* the generic Q^2 scale in eq. (76) and eq. (79) is replaced by a $p_{\perp\text{evol}}^2$. The use of transverse momentum as an evolution variable has been shown to catch key coherence features and therefore is a preferred choice [53, 54].

The other is that a dipole picture is being used, although with some exceptions. In it each coloured parton has a unique anticolour partner, and together the two form a dipole. Radiation is split into one contribution from each dipole end. When one end radiates, the other end has to take a recoil such that total energy and momentum is preserved.

Shower evolution   To understand basic kinematics in a branching $a \to bc$, expressions become especially simple using light-cone (LC) $p^\pm = E \pm p_z$, for which $p^+ p^- = m_\perp^2 = m^2 + p_\perp^2$. When a moves along the $+z$ axis, with $p_b^+ = z_{\text{LC}} p_a^+$ and $p_c^+ = (1 - z_{\text{LC}}) p_a^+$, p^- conservation then gives

$$m_a^2 = \frac{m_b^2 + p_\perp^2}{z_{\text{LC}}} + \frac{m_c^2 + p_\perp^2}{1 - z_{\text{LC}}}, \tag{96}$$

or equivalently

$$p_\perp^2 = z_{\text{LC}}(1 - z_{\text{LC}})m_a^2 - (1 - z_{\text{LC}})m_b^2 - z m_c^2 = p_{\perp\text{LC}}^2. \tag{97}$$

For a timelike branching $Q^2 = m_a^2$ and $m_b = m_c = 0$, assuming massless partons, so then $p_{\perp\text{LC}}^2 = z_{\text{LC}}(1 - z_{\text{LC}})Q^2$. For a spacelike branching $Q^2 = -m_b^2$ and $m_a = m_c = 0$, where b is the parton that will enter the hard interaction, so instead $p_{\perp\text{LC}}^2 = (1 - z_{\text{LC}})Q^2$. We are inspired by these relations to define abstract evolution variables

$$p_{\perp\text{evol}}^2 = z(1 - z)Q^2 \quad \text{for FSR}, \tag{98}$$

$$p_{\perp\text{evol}}^2 = (1 - z)Q^2 \quad \text{for ISR}, \tag{99}$$

in which to order the sequence of shower emissions. The z_{LC} definitions will be replaced by invariant-mass-based z for the final kinematics definitions, for better Lorentz invariance properties, and as a consequence $p_{\perp\text{evol}} \neq p_{\perp\text{LC}}$. Further details on this are given later.

The evolution is now carried out, downwards in $p^2_{\perp\text{evol}}$ from some starting scale $p^2_{\perp\text{evol,max}}$, for FSR by parton a branching to $b + c$, for ISR by parton b being reconstructed as coming from the branching of an earlier a. The branching probabilities of eq. (76) and eq. (79), with the addition of no-branching probabilities Π, eq. (77), gives

$$d\mathcal{P}_{\text{FSR}} = \Pi_a(p^2_{\perp\text{evol,max}}, p^2_{\perp\text{evol}})\, d\mathcal{P}_a(p^2_{\perp\text{evol}}), \tag{100}$$

$$d\mathcal{P}_{\text{ISR}} = \Pi_b(x, p^2_{\perp\text{evol,max}}, p^2_{\perp\text{evol}})\, d\mathcal{P}_b(x, p^2_{\perp\text{evol}}). \tag{101}$$

A $p^2_{\perp\text{evol}}$ scale is selected for each existing dipole end, and the end with the largest value is chosen to branch.

The selection of a branching means that $p^2_{\perp\text{evol}}$ and z are fixed. From these, one can derive the virtuality of the evolving parton

$$m^2_a = Q^2 = \frac{p^2_{\perp\text{evol}}}{z(1-z)} \quad \text{for FSR}, \tag{102}$$

$$-m^2_b = Q^2 = \frac{p^2_{\perp\text{evol}}}{(1-z)Q^2} \quad \text{for ISR}. \tag{103}$$

What now remains is to construct the kinematics of the branching. This works rather differently for FSR and for ISR, so the two cases are presented separately.

FSR branching kinematics Study the radiation inside a dipole, consisting of a radiator a and a recoiler r, in the dipole rest frame, with a moving in the $+z$ direction, and with $m^2_{ar} = (p_a + p_r)^2$.

For massless partons, the introduction of an off-shell $Q^2 = m^2_a$ increases E_a from $m_{ar}/2$ to $(m^2_{ar} + Q^2)/2m_{ar}$, with E_r reduced by the same amount, or in terms of four-momenta

$$p_{a'} = p_a + \frac{Q^2}{m^2_{ar}} p_r, \qquad p_{r'} = \left(1 - \frac{Q^2}{m^2_{ar}}\right) p_r. \tag{104}$$

The two daughters share the energy according to $E_b = z E_a$ and $E_c = (1-z)E_a$. With the modified a still along the $+z$ axis, the transverse momentum of the two daughters then becomes

$$p^2_{\perp b,c} = \frac{z(1-z)(m^2_{ar} + Q^2)^2 - m^2_{ar}Q^2}{(m^2_{ar} - Q^2)^2} Q^2 \ \leq \ z(1-z)Q^2 = p^2_{\perp\text{evol}}. \tag{105}$$

The kinematics can now be completed, including a random φ orientation of the $p_\perp$. Also, if the original dipole had to be boosted and rotated to its rest frame, the new system should be transformed back to the original frame.

Colours are also assigned in the branching, such that the new colour-dipole picture is set up. This is well defined in the $N_c \to \infty$ limit, except for g $\to$ gg branchings. Here a rewriting [54],

$$P_{\text{g}\to\text{gg}}(z) = 3\frac{\left(1 - z(1-z)\right)^2}{z(1-z)} = \frac{3}{2}\frac{1+z^3}{1-z} + \frac{3}{2}\frac{1+(1-z)^3}{z} \simeq 3\frac{1+z^3}{1-z}, \tag{106}$$

allows the gluon that takes the (usually smaller) $1-z$ fraction to be the "radiated" gluon that connects the "radiator" gluon to the recoiler.

Of note is that the light-cone sharing of momenta between daughters, suggested initially, here is replaced by an energy sharing. It has the advantage that $p^2_{\perp\text{evol}}$ and this z together exactly match on to the singularity structure of matrix elements, such as the textbook

$\gamma^*/Z \to q(1) + \overline{q}(2) + g(3)$ one, when $q \to qg$ and $\overline{q} \to \overline{q}g$ radiation from the two dipole ends is combined

$$\frac{dp^2_{\perp\text{evol,q}}}{p^2_{\perp\text{evol,q}}} \frac{dz_q}{1-z_q} + \frac{dp^2_{\perp\text{evol},\overline{q}}}{p^2_{\perp\text{evol},\overline{q}}} \frac{dz_{\overline{q}}}{1-z_{\overline{q}}} = \frac{dx_1 dx_2}{(1-x_2)x_3} + \frac{dx_1 dx_2}{(1-x_1)x_3} = \frac{dx_1 dx_2}{(1-x_1)(1-x_2)}, \quad (107)$$

with $x_i = 2E_i/E_{\text{tot}}$. Corrections to fully reproduce several important matrix elements therefore are easily implemented.

Incidentally, note that $1 - x_2 \propto \cos\theta_{qg}$ and $1 - x_1 \propto \cos\theta_{\overline{q}g}$, so eq. (107) provides a prescription for how radiation from the full dipole smoothly can be split into radiation from the two ends as a function of the gluon emission angle. This split also decides which of the two original partons is the recoiler, the one that keeps its direction of motion.

The kinematics need to be modified when quark masses are included, with full expressions in [77]. There are two key points, however. First, if the branching parton a has an on-shell mass m_a and off-shell mass $m_{a'}$, then eq. (98) needs to be modified to

$$p^2_{\perp\text{evol}} = z(1-z)Q^2 = z(1-z)(m^2_{a'} - m^2_a), \quad (108)$$

to reproduce the singularities in matrix elements. Second, if the daughters are initially assigned four-momenta $p^{(0)}_b$ and $p^{(0)}_c$ as if they were massless, then massive four-vectors can be constructed as

$$p_b = (1-k_b)p^{(0)}_b + k_c p^{(0)}_c, \quad (109)$$

$$p_c = (1-k_c)p^{(0)}_c + k_b p^{(0)}_b, \quad (110)$$

$$k_{b,c} = \frac{m^2_a - \sqrt{(m^2_a - m^2_b - m^2_c)^2 - 4m^2_b m^2_c} \pm (m^2_c - m^2_b)}{2m^2_a}. \quad (111)$$

The $p_{\perp b,c}$ is also reduced in the process, by a factor $1 - k_b - k_c$.

ISR branching kinematics The handling of ISR branching kinematics is somewhat more complicated. At any scale $p^2_{\perp\text{evol}}$, two initial partons, one from each incoming beam particle, are identified by the ISR algorithm. These partons are taken to be massless and collinear with the beams, and are the mothers of the respective incoming ISR cascade.

By the backwards evolution, the resolution scale is gradually reduced, and then either of these two partons may be reconstructed as the daughter b of a previous branching $a \to bc$. The other of the two partons takes on the role of recoiler r. The parton b, previously taken as massless, is now assigned a spacelike virtuality $m^2_b = -Q^2$, and the recoiler is needed for consistent reconstruction of the kinematics. The kinematics redefinition is performed such that the invariant mass of the $b + r$ system is unaffected, since this mass is given by the set of produced particles, which in a case like $gg \to H$ must not be modified. The final-state system must be rotated and boosted, however, since b not only acquires a virtuality but also a transverse momentum. If previously b was assumed to travel along the event axis, now it is a that does so.

The current massless mothers should, at any step of the cascade, have four-momenta in the rest frame of the two incoming beam particles given by $p_i = x_i(\sqrt{s}/2)(1; 0, 0, \pm 1)$, ensuring that $\hat{s} = x_1 x_2 s$. This relation should be preserved in the $a \to bc$ branching, so the $z = x_b/x_a$ should fulfil $z = m^2_{br}/m^2_{ar} = (p_b + p_r)^2/(p_a + p_r)^2$. This constraint is used to construct the

kinematics in the $a + r$ rest frame, with a moving along the $+z$ axis, and if c is massless:

$$p_{a,r} = \frac{m_{ar}}{2}(1; 0, 0, \pm 1), \tag{112}$$

$$p_b = \left(\frac{m_{ar}}{2} z; p_{\perp b,c} \cos\varphi, p_{\perp b,c} \sin\varphi, \frac{m_{ar}}{2}\left(z + \frac{2Q^2}{m_{ar}^2} \right) \right), \tag{113}$$

$$p_c = \left(\frac{m_{ar}}{2}(1-z); -p_{\perp b,c} \cos\varphi, -p_{\perp b,c} \sin\varphi, \frac{m_{ar}}{2}\left(1 - z - \frac{2Q^2}{m_{ar}^2} \right) \right), \tag{114}$$

$$p_{\perp b,c}^2 = (1-z)Q^2 - \frac{Q^4}{m_{ar}^2} < (1-z)Q^2 = p_{\perp\text{evol}}^2. \tag{115}$$

The $p_{\perp b,c}^2$ and $p_{\perp\text{evol}}^2$ measures agree well for small Q^2 values. With increasing Q^2, the $p_{\perp b,c}^2$ will gradually turn over and decrease again (for fixed z and m_{ar}). The maximum $p_{\perp b,c}^2$ occurs for $p_{zc} = 0$, and a decreasing $p_{\perp b,c}^2$ is obtained for an increasingly negative p_{zc}. Thus the drop of $p_{\perp b,c}^2$ is deceptive. The $p_{\perp\text{evol}}^2$ therefore makes more sense than $p_{\perp b,c}^2$ as an evolution variable, in spite of its nontrivial kinematics interpretation. Furthermore, emissions with negative p_{zc} should preferably be associated with radiation off the other incoming parton, where it is collinearly enhanced, so this region is less important.

Quark-mass effects are less crucial for ISR than for FSR: nothing heavier than charm and bottom need be considered as beam constituents, and even those are suppressed. Kinematics have to be slightly modified if the outgoing parton c is not massless, *e.g.* in a $g \to q\overline{q}$ branching. The main effect is a modified evolution $p_\perp$, with eq. (99) replaced by

$$p_{\perp\text{evol}}^2 = (1-z)(Q^2 + m_c^2), \tag{116}$$

and a reduced $p_\perp$ in the branching, replacing eq. (115) by

$$p_{\perp b,c}^2 = (1-z)Q^2 - \frac{Q^4}{m_{ar}^2} - m_c^2\left(z + \frac{Q^2}{m_{ar}^2} \right) = Q^2 - z\frac{(Q^2 + m_c^2)(m_{br}^2 + Q^2)}{m_{br}^2}. \tag{117}$$

Charm and bottom quarks raise another issue, namely what to do in the threshold region, *i.e.* around the Q_{thr}^2 scale where $g \to c\overline{c}$ or $g \to b\overline{b}$ branchings are turned on in the PDF evolution. Normally, it is assumed that these quark PDFs vanish below Q_{thr}^2 and then evolve above it as a massless quark would. Initially, thus $f_q(x, Q^2) \propto \ln(Q^2/Q_{\text{thr}}^2)$. In backwards evolution of a c/b quark, this leads to a diverging $d\mathcal{P}_b$ in eq. (79) for $Q^2 \to Q_{\text{thr}}^2$, and a vanishing no-branching probability. While such a behaviour is possible to handle by evolving with gradually smaller Q^2 steps as the threshold is approached, the chosen solution is instead to rely on the known forwards-evolution PDF shape. Therefore, once $p_{\perp\text{evol}}^2 < f\, m_q^2$, with f a parameter of the order of 2, a $p_{\perp\text{evol}}^2$ is chosen logarithmically evenly between m_q^2 and $f\, m_q^2$, and a z flat in the allowed range. Acceptance is based on the product of three factors, representing the running of α_s, the splitting kernel (including the mass term) and the gluon density weight. At failure, a new $p_{\perp\text{evol}}^2$ is chosen in the same range, *i.e.* is not required to be lower since no no-branching probability is involved.

As for FSR, the choices of $p_{\perp\text{evol}}^2$ and z offers a possibility to match onto the singularity structure of common matrix elements, and thereby easily correct to matrix-element expressions. Consider *e.g.* $q\overline{q}' \to gW^\pm$ [81]. The $q \to qg$ branching gives a denominator $\hat{t}(\hat{t} + \hat{u})$ and $\overline{q}' \to \overline{q}'g$ a denominator $\hat{u}(\hat{t} + \hat{u})$, which combine to $\hat{t}\hat{u}$, in agreement with the matrix element. This also illustrates how the full ISR radiation pattern can be subdivided into contributions from the two sides.

One special option in the ISR implementation, on by default, is the possibility to order the emissions in rapidity, or equivalently in angle, *i.e.* to veto any trial emission that leads to un-ordered emitted partons [79]. The backwards evolution is one towards smaller $p_\perp$ and larger

x values, so angular ordering is already implicit to first approximation, but the unordered emissions have a non-negligible impact that appears to be detrimental for some distributions. There are good arguments for a rapidity ordering to be a legitimate choice [84], to provide a consistent separation between ISR and FSR. But that was for a somewhat different algorithm, so this option should more be seen as one possible variation beyond the basic LL accuracy of the shower.

Strong coupling   By default a first-order running $\alpha_s(p_{\perp\text{evol}}^2)$ is used, but alternatives are a fixed value or second-order running. Tuned $\alpha_s(m_Z^2)$ values typically tend to come out somewhat above the PDG $\overline{\text{MS}}$ one [35]. This can be understood as absent higher-order effects, in splitting kernels and shower kinematics, being absorbed into effective values. Since these higher-order corrections differ between ISR and FSR, the $\alpha_s(m_Z^2)$ are also set separately for the two.

Furthermore, in the soft-gluon limit, it can be shown that the dominant $\mathcal{O}(\alpha_s^2)$ splitting-function term, which generates contributions starting from $\mathcal{O}(\alpha_s^2 \ln^2)$ at the integrated level, can be absorbed into the LO splitting functions by translating to the so-called Catani–Marchesini–Webber (CMW) (also known as MC) scheme [85]. This means that an $\overline{\text{MS}}$ $\alpha_s(m_Z^2) = 0.1185$ would translate into an MC $\alpha_s(m_Z^2) = 0.126$. This goes some of the way towards explaining the PYTHIA default $\alpha_s(m_Z^2) = 0.1365$. It is possible to switch on the usage of the CMW rescaling procedure to allow a lower input $\alpha_s(m_Z^2)$, but physics is only mildly modified by this.

Another consequence of staying at leading order is that usage of LO parton distributions is vastly to be preferred. If not, the description of ISR branchings at low scales becomes quite unreliable, for physical and technical reasons. The former are covered elsewhere, the latter are reflected in the need to have positive PDFs in eq. (79), which is not guaranteed at NLO.

Shower cutoff   A lower cutoff scale $p_{\perp\min}$ is needed both for ISR and FSR, but the two need not be same. The FSR one is related to the transition from partons to hadrons, and LEP experience gives us some understanding that too high a value does affect event shapes detrimentally. The ISR case is less clear cut. Experimental signals, such as the $p_\perp$ spectrum of Z bosons in pp/p$\overline{\text{p}}$ collisions, are affected by the non-trivial interplay with primordial $k_\perp$, *cf.* section 6.3.3. A lower $p_{\perp\min}$ means more $p_\perp$ kicks to the Z, but a shower initiator with a larger x, which means more dilution of its $k_\perp$ in the cascade. One reasonable strategy therefore is to assume the ISR is damped in the same way as MPIs are, *i.e.* the $dp_\perp^2/p_\perp^2$ divergence is replaced by a $dp_\perp^2/(p_{\perp 0}^2 + p_\perp^2)$ one. Alternatively, it is also possible to use a sharp cutoff.

Interleaving   Multiparton interactions and ISR are in direct competition for the beam-remnant momentum. Therefore, a combined downwards evolution in $p_\perp$ of the two gives precedence to the harder parts of the event activities. There is no corresponding competition requirement for FSR to be interleaved, and FSR can also be viewed as occurring after the other two components in time. Interleaving is allowed, however, since it can be argued that a high-$p_\perp$ FSR occurs on shorter time scales than a low-$p_\perp$ MPI, say. Backwards evolution of ISR is also an example that physical time is not the only possible ordering principle. Rather, one can work with conditional probabilities: given the partonic picture at a specific $p_\perp$ resolution scale, what possibilities are open for a modified picture at a slightly lower $p_\perp$ scale, either by MPI, ISR, or FSR? This is the default approach taken.

It is possible to switch off the interleaving, and consider FSR after MPI and ISR. In that case it is also possible to allow FSR dipoles to be formed between matching colour-anticolour pairs in two different MPIs, whereas normally dipoles are local to each MPI separately.

Another ordering issue is when resonance decays and their showers are considered. By default, this is done after the ISR/FSR/MPI evolution of the hard process, and also after the handling of beam remnants and colour reconnections (CR). An option for "early resonance decays" allows for the resonance-decay to be handled before remnants and CR; this does not alter the perturbative evolution, but partons from resonance decays can then participate in CR on an equal footing with partons from the production process. The option for "interleaved resonance decays" [23] moves the resonance-decay handling even earlier, interleaving it with the ISR/FSR/MPI evolution of the hard process, with a few different options for which value of the perturbative evolution scale to associate to resonance decays, the default being of order the width of the resonance. This effectively represents an alternative treatment of finite-width effects; it is not a big effect for the standard-model particles, none of which has widths much larger than the shower cutoff, but could be relevant for precision studies and/or in BSM scenarios.

4.1.2   The dipole evolution

The previous subsection described the kinematics of a single branching. The full evolution in an event requires some further consideration, in particular related to the overall colour flow and the resulting set of radiating dipoles. In hadronic collisions the dipole pattern can be quite complicated. Consider the example of gg → gg scattering, as shown in fig. 7a, which is one of the six possible colour topologies for this process in the $N_c \to \infty$ limit. Each radiation now is characterized by whether the radiator is in the initial (I) or final (F) state, combined with the same classification for the recoiler, so in general four different emission types need to be considered.

Final-final radiation   To begin with, consider the simple $e^+ e^- \to \gamma^*/Z \to q\bar{q}$ event. The first emission of a gluon, to give $q\bar{q}g$, follows the pattern already outlined. Now the $N_c \to \infty$ limit is applied to split the event into two dipoles qg and $g\bar{q}$. Each can be considered in its respective rest frame, with the $p_{\perp\text{evol}}$ scale of the branching setting the upper limit for the continued evolution. In this evolution, the full emission rate of g → gg has to be split between the two dipoles. Using eq. (106), the effective splitting kernel becomes $P_{g \to gg}(z) = (3/2)(1 + z^3)/(1 - z)$. Here, the emitter gluon takes the fraction z and the emitted $1 - z$, where the latter is the one straddling the two new dipoles. The radiation function from the q (or $\bar{q}$) and g ends of the dipole have almost the same shape, the main difference being between the colour factors 4/3 *vs.* 3/2, which are smoothly mixed around the middle of the dipole, as already discussed for the angular dependence of q → qg *vs.* $\bar{q} \to \bar{q}g$. There are known shortcomings with this colour factor treatment [59, 86], but these are of order $1/N_c^2$ and are neglected here. On the kinematics side, note that an emission in one dipole also affects the kinematics of adjacent ones, by virtue of sharing one gluon with changed momentum.

Initial–Initial (II) radiation   The ISR and FSR descriptions can be separated so long as colour does not flow between the initial and the final state, as for the first emission in $q\bar{q} \to Z$, which is pure II. But once a gluon has been emitted, *cf.* fig. 7b, the two dipoles now bypass the Z, and the Z does not receive any further $p_\perp$ recoil during the subsequent evolution. This runs counter to standard perturbation- and resummation-theory results, which is the reason why traditionally ISR has only been handled as II dipoles. That is, as shown in fig. 7b, the emission of a second gluon is handled as occurring from the (new) $q\bar{q}$ dipole, with Z + g together taking the recoil. Similarly, as shown in fig. 7a, the two IF dipole ends are replaced by doubling the strength of the II dipole.
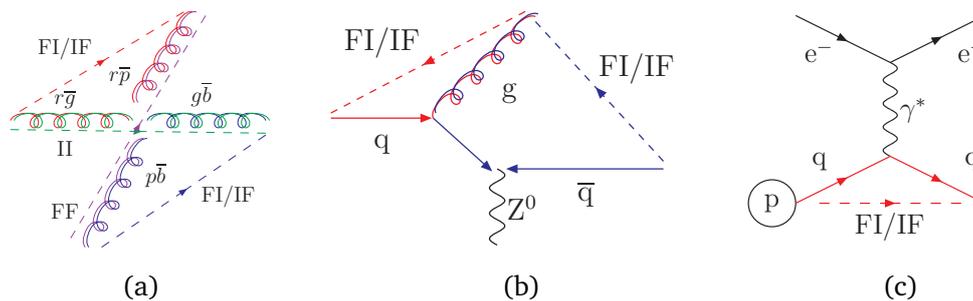
Figure 7: (a) Colour flow for the process $g(r\overline{g}) + g(g\overline{b}) \to g(r\overline{p}) + g(p\overline{b})$. Here, the $N_c \to \infty$ limit is used so that p stands for the new colour purple. The dashed lines represent the colour lines stretching between the dipole ends. The type of dipole is indicated. (b) $q\overline{q} \to Zg$, again with colour lines and dipole types. (c) Deeply inelastic scattering, again with colour lines and dipole types.

Final–Initial (FI) radiation Normally it should be possible to replace the FI ends by FF ones, with an arbitrary matching of the dipole ends, and such an option exists for exploratory purposes, but this is not the default. Instead, the recoiler r is now the incoming colour-connected parton. In a branching, studied in the dipole rest frame, a fraction Q^2/m_{ar}^2 of the recoiler energy should be given from the recoiler to the emitter, as in eq. (104). But now the increase of a momentum is not compensated anywhere in the final state. Instead, it is the momentum of the incoming recoiler that is increased, by the same amount as the emitter. Thus its momentum fraction x is scaled up as

$$x_{r'} = \left(1 + \frac{Q^2}{m_{ar}^2}\right) x_r \,. \tag{118}$$

The direction along the incoming beam axis is not affected by this rescaling. The kinematics construction therefore is similar to that of Catani–Seymour dipoles [87]. It leads to an increased dipole mass m_{ar} and increased squared subcollision mass $\hat{s}$, the latter by the same factor as x_r. The increased x value again leads to an extra PDF weight

$$\frac{x_{r'} f_r(x_{r'}, p_\perp^2)}{x_r f_r(x_r, p_\perp^2)} \,, \tag{119}$$

in the emission and no-emission probabilities, to maintain a proper damping of radiation in the $x_{r'} \to 1$ limit. The splitting of the full dipole radiation pattern is not as well understood in this case as for an FF dipole, however. Some rough estimates of how to share the full dipole-emission rapidity range can be made [79]. Based on these, an extra damping factor is applied by default, of the form $Q_{\text{hard}}^2/(Q^2 + Q_{\text{hard}}^2)$, where Q_{hard}^2 is the relevant hard scale of the process, like $4p_\perp^2$ for QCD $2 \to 2$ processes.

Initial–Final (IF) radiation Finally, a non-default option exists, where IF dipole ends are treated in their own right [80]. It then suffers from the above-mentioned problems with $p_{\perp Z}$ resummation, but it enables handling *e.g.* of deeply inelastic scattering (DIS), *cf.* fig. 7c, where II radiation is not an option (using the e^- as recoiler would upset DIS kinematics), and presumably offers a more realistic description *e.g.* of weak-gauge-boson fusion to a Higgs. The kinematics step from $b + r$, where r is the colour-connected recoiler in the final state, to $a + c + r'$, as a consequence of the $a \to bc$ step, is easiest constructed in the $b + r$ rest frame.

There

$$p_a = \frac{1}{z} p_b \,, \tag{120}$$

$$p_c = \left(\frac{1}{z} - 1\right) p_b + p_{\text{shift}} \,, \tag{121}$$

$$p_{r'} = p_r - p_{\text{shift}} \,, \tag{122}$$

$$p_{\text{shift}} = \left(\frac{(2z-1)Q^2}{2m_{br}} + z\frac{m_c^2}{m_{br}} ; p_\perp \cos\varphi, p_\perp \sin\varphi, -\frac{Q^2}{2m_{br}} - z\frac{m_r^2}{m_{br}}\frac{Q^2 + m_c^2}{m_{br}^2 - m_r^2}\right), \tag{123}$$

$$p_\perp^2 = \left((1-z)(Q^2 + m_c^2) - m_c^2\right)\left(1 - z\frac{Q^2 + m_c^2}{m_{br}^2 - m_r^2}\right) - m_r^2\left(z\frac{Q^2 + m_c^2}{m_{br}^2 - m_r^2}\right)^2 \,. \tag{124}$$

The same set of rotations and boosts as used to recover the $b+r$ rest frame can then be inverted to bring c and r' back to the event rest frame.

Special cases What remains is to combine IF and FI emissions consistently. In the specific case of the first gluon emission from a DIS process, it turns out that the IF-type branching q $\to$ qg exactly reproduces the soft- and collinear-singularity structure of the γ^*q $\to$ qg matrix element on its own, with only a mild mismatch in the numerator (which vanishes in the soft-gluon limit). Therefore, it would be possible to leave aside FI emissions in this case, and the same holds for g $\to$ gg splittings, but not for g $\to$ q$\bar{\text{q}}$ ones. So in general. both IF and FI contributions have to be used. One simplifying factor is that the incoming parton must always be along the beam axis, so there will only be one common phase-space mapping, unlike the case of FF or II dipoles. Nevertheless, the details become technical and we refer to [80] for further discussion. One small comment, however: when an emission from a qg dipole is considered, the two ends radiate with different colour charges, 4/3 and 3/2, respectively. The colour factors of the two ends are then mixed in proportion to the $1/m^2$ values of the emitted parton to the two dipole ends.

Another set of problems occurs in the decays of coloured resonances, say t $\to$ bW. In this case the colour dipole is stretched between the b and the hole left behind by the decayed t. In order to conserve momentum-energy, the b uses the W as a recoiler, and this choice is unique. Once a gluon has been radiated, however, it is possible to either still have the unmatched colour (inherited by the gluon) recoiling against the W, or to let it recoil against the b for this dipole as well. The former could give unphysical radiation patterns, so the latter is chosen by default, although it is not perfect either. A more detailed discussion of this issue can be found in [88]. The same issue exists for a second emission of QED radiation, *e.g.* in W$^+$ $\to$ e$^+\nu_\text{e}$, but is obviously less significant there.

4.1.3 Matrix-element and other corrections

In this subsection we give a survey of some methods used to make the shower reproduce, or at least better approximate, known matrix-element behaviours. The methods to match and merge external matrix-element input to the showers are covered separately in section 5, so here we mainly describe program elements internal to the simple shower. Included are also some other "correction" aspects, that should offer improvements to the shower, or at least provide increased understanding by controlled variations.

Matrix-element corrections One key capability is the first-order correction to resonance decays $a \to bc$, where a gluon is emitted to give an $a \to bcg$ final state. The foremost example of this is e$^+$e$^-$ $\to$ γ^*/Z $\to$ q$\bar{\text{q}}$ $\to$ q$\bar{\text{q}}$g [75]. This works because eq. (107) provides a way that

the parton shower exactly can reproduce the singularity structure of the matrix element, *i.e.* of the generic ratio

$$\frac{1}{\sigma_{a\to bc}}\frac{\mathrm{d}\sigma_{a\to bcg}}{\mathrm{d}x_1\,\mathrm{d}x_2}\,. \tag{125}$$

The $1+z^2$ numerator of the splitting kernels also combines to an expression that overestimates the numerator of the matrix elements, *e.g.* $x_1^2+x_2^2$ for $\mathrm{e}^+\mathrm{e}^-$ annihilation. In the veto-algorithmic downwards evolution of the shower, it is therefore trivial to use the ratio of the correct numerator to the shower-kernel numerator, as a probability that a trial emission will be retained. In fact, for the evolution down to the first branching, it is as simple as putting the numerator equal to 2 and correct down from that.

This approach has then been extended to all combinations of colours and spins for a, b and c that can occur within the SM and MSSM [77], and can be reused for other models where the same colour and spin combinations occur. The inclusion of b and c masses as in eq. (108) also reproduces the proper propagator poles $1/(m_{b'}^2-m_b^2)$ and $1/(m_{c'}^2-m_c^2)$ that are found in the matrix elements, such that all correction factors are well behaved over the whole phase space. Although the matrix elements are calculated for a first emission only, they are reused in a suitably modified form to include mass effects also in subsequent steps.

Similarly, there are a few processes where the first branching of an ISR shower are corrected to the respective matrix element [81], based on a common singularity structure. These include $\mathrm{q}\bar{\mathrm{q}}\to\mathrm{V}\mathrm{g}$, $\mathrm{q}\mathrm{g}\to\mathrm{V}\mathrm{q}$, $\mathrm{f}\bar{\mathrm{f}}\to\mathrm{V}\gamma$, and $\mathrm{f}\gamma\to\mathrm{V}\mathrm{f}$, where $\mathrm{V}=\gamma^*/\mathrm{Z}/\mathrm{W}^\pm/\mathrm{Z}'/\dots$ is a colour-singlet vector boson. In the point-like-coupling approximation, also Higgs production $\mathrm{g}\mathrm{g}\to\mathrm{H}$ and $\gamma\gamma\to\mathrm{H}$ is handled.

It should be feasible to include a matrix element correction to DIS in the same fashion as already outlined, but this has not been done yet. A generic and more detailed discussion of matrix-element corrections is given in section 5.

Power and wimpy showers   In the cases above, the ISR/FSR showers are allowed to cover the full phase space, so-called *power showers* [89]. We have seen that they can reach the furthest corners no worse than being a factor two off, which then could be fixed by modest reweighting. One guess is that this would hold true also in other processes, where no matrix-element correction factors have been implemented. But there are counterexamples. Consider QCD jet production, say, starting out from $2\to 2$ partonic processes. Then a low-$p_\perp$ $2\to 2$ process could not be allowed to shower further partons at high $p_\perp$, or else such high-$p_\perp$ production would be double counted and the whole perturbative framework would be undermined. So the logical $p_{\perp\mathrm{evol,max}}$ shower starting scale is the $p_\perp$ scale of the $2\to 2$ process, *i.e.* the factorization scale, giving *wimpy showers*. Comparisons with $2\to 3$ matrix elements confirm that such a scale choice is close to optimal [79].

In general, it is possible for the user to choose between power and wimpy showers, even separately for ISR and FSR. The default option involves a choice between the two based on the likelihood of double counting:

- If the final state of the hard process (not counting subsequent resonance decays) contains at least one quark $(\mathrm{u,d,s,c,b})$, gluon, or photon then $p_{\perp\mathrm{evol,max}}$ is chosen to be the factorization scale for internal processes and the `scale` value for Les Houches input, *i.e.* wimpy showers.

- Else, emissions are allowed to go all the way up to the kinematic limit, *i.e.* power showers.

The reasoning is that in the former set of processes, the ISR emission of yet another quark, gluon, or photon could lead to double counting, while no such danger exists in the latter case.

In cases where more is known about the context of a particular event sample, *e.g.* when doing matching and merging, it is important to make use of this knowledge to override the default behaviour. One example is to start out with power showers but then implement a user hook to reject those emissions that would double count the particular cuts of the event sample.

Damped showers   While there are processes where power or wimpy showers are appropriate, there are also ones where the actual behaviour lies in between. It is relevant to recall that the characteristic cross-section shape of a shower emission is $dp_\perp^2 / p_\perp^2$, while that of QCD $2 \to 2$ process is $dp_\perp^2 / p_\perp^4$. That is, the $p_\perp$ spectrum of a parton ought to begin to drop faster around the scale where it goes from being a soft add-on to being a part of the core hard process. For top-pair production $gg \to t\bar{t}g$, *e.g.* the gluon emission can be approximated by a shape

$$\frac{d\mathcal{P}}{dp_{\perp g}^2} \propto \frac{1}{p_{\perp g}^2} \frac{k^2 M^2}{k^2 M^2 + p_{\perp g}^2} \,, \tag{126}$$

where M^2 is a reasonable scale to associate with the hard process and k^2 is a fudge factor of order unity. This generalizes into the possibility to use a power shower with an additional damping factor $k^2 M^2 / (k^2 M^2 + p_{\perp \mathrm{evol}}^2)$. Studies [90] show that this is a reasonable approach for coloured final states, *e.g.* for pairs of supersymmetric coloured particles, whereas a simple power shower is more appropriate for pair production of uncoloured particles. This can be understood as reduced emission by a destructive interference between ISR and FSR when colours flow from the initial to the final state [91], but only if there is such flow.

Gluon splittings   The pure s-channel nature of $g \to q\bar{q}$ splittings motivates the introduction of an option with $\alpha_s(m_{q\bar{q}}^2)$ rather than $\alpha_s(p_{\perp \mathrm{evol}}^2)$, where $m_{q\bar{q}}$ is the invariant mass of the $q\bar{q}$ pair. More importantly, the cuts on the allowed z range during the FSR evolution imply that the branching rate is reduced relative to expectations from matrix elements. Therefore, for this branching only, the default option is to weigh up the splitting kernel inside the allowed z range to give the correct integrated matrix-element weight. Furthermore, this range is afterwards remapped to cover the full range of decay angles, disregarding the normal $p_\perp$ ordering. This treatment is especially important for charm and bottom quarks, where the mass is not negligible and mass corrections should be reproduced both in rate and in angular distributions. As a final twist, the matrix element for $H \to gg \to gq\bar{q}$ does reproduce the expected behaviour *e.g.* from $e^+e^- \to \gamma^* \to q\bar{q}$, but times a factor $(1 - m_{q\bar{q}}^2 / m_H^2)^3$. The default option uses this factor, with the radiating dipole mass replacing the Higgs one, to suppress high-mass branchings.

Dead cones   For topologies where a gluon recoils against a massive quark (or another massive coloured particle) there are no suitable ME corrections implemented into PYTHIA. When the dipole radiation pattern is split into two ends, with a smooth transition between the two, this means that the gluon end can radiate into the quark hemisphere as if the quark were massless. The "dead cone" effect, that radiation collinear with a massive quark is strongly suppressed, thereby is not fully respected. (Unlike radiation from the quark end itself, where mass effects are included.) By default, a further suppression is therefore introduced for $g \to gg$ branchings, derived as the massive/massless ratio of the eikonal expression for dipole radiation, which eliminates radiation collinear with the quark. The $g \to q\bar{q}$ branchings currently are not affected; the absence of a soft singularity implies that there is hardly any radiation into the recoiler hemisphere anyway.

Global recoil   The default ISR and FSR showers differ, in that the former uses a global recoil while the latter uses a dipole one. That is, the recoil from an emission is carried by all final-state

particles in ISR, but only by a single one in FSR. Then we introduced an option where dipole recoil can be used for ISR. As it turns out, there is also an option to obtain a global recoil in FSR. In such a scenario, the radiation pattern is unrelated to colour correlations, which could be seen as a disadvantage. It is convenient for some matching algorithms, however, where a full analytic knowledge of the shower radiation pattern is needed to avoid double counting, so it is by such user requests that the option is made available.

Technically, the radiation pattern is most conveniently represented in the rest frame of the final state of the hard subprocess. Then, for each parton at a time, the rest of the final state can be viewed as a single effective parton. This "parton" has a fixed invariant mass during the emission process, and takes the recoil without any changed direction of motion. The momenta of the individual new recoilers are then obtained by a simple common boost of the original ones. With the whole subcollision mass as "dipole" mass, the phase space for subsequent emissions is larger than for the normal dipole algorithm, which leads to a too steep multiplication of soft gluons. Therefore, the main application is for the first one or few emissions of the shower, where a potential overestimate of the emission rate is to be corrected by a matching to the relevant matrix elements. Thereafter, subsequent emissions should be handled as before, *i.e.* with dipoles spanned between nearby partons. Several process-dependent settings are needed to use this option.

Azimuthal asymmetries Parton-shower branchings are assumed to occur isotropically in azimuthal angle φ, in the rest frame of the respective dipole. The boost to the overall CM frame then gives rise to the familiar "string effect" [92, 93] coherence phenomenon, where particle production is enhanced in the region between two colour-connected partons. But there are also azimuthal correlations arising from parton polarization [94]. Notably, gluons tend to be plane polarized, with the decay plane of $g \rightarrow gg$ branchings favourably aligned with the production plane, while $g \rightarrow q\bar{q}$ ones tend to be aligned orthogonal to it. The former branching type is common but with small asymmetries, while the opposite holds for the latter branching type, so that net effects are small. They are included nevertheless, since they may have some effect in charm and bottom production.

User hooks   There are also other user hooks that can be used to modify the shower evolution. The ones that allow an ISR or FSR emission to be vetoed play a key role in matching and merging schemes and therefore are described in section 5.

4.1.4   QED, electroweak and other showers

The simple shower includes several extensions beyond the QCD core discussed so far. Characteristic is that these form part of the same evolution in a common $p_{\perp\text{evol}}$ scale, although with some distinguishing features.

QED shower   The most obvious extension is to QED. The required branching kernels have been presented in eqs. (74) and (75). In the evolution equations $\alpha_s(p_{\perp\text{evol}}^2)$ is replaced by $\alpha_{\text{em}}(p_{\perp\text{evol}}^2)$, but otherwise most that has been written about $q \rightarrow qg$ and $g \rightarrow q\bar{q}$ carries over. A dipole language is used also for QED emissions, but the dipoles may be different from the QCD ones. An example is $e^+e^- \rightarrow \gamma^*/Z \rightarrow q\bar{q} \rightarrow q\bar{q}g$, where the last stage contains two colour dipoles qg and $g\bar{q}$, but only one charge dipole $q\bar{q}$, since the gluon carries no electrical charge. The complete multipole radiation pattern may be poorly represented by a set of simple dipoles in cases with multiple charges, since there is no confinement mechanism in QED to further a unique dipole setup. In reality, few events contain multiple QED charges to consider, and if

so, often the event history suggests a reasonable division, *e.g.* when a new dipole arises from a $\gamma \to f\bar{f}$ branching.

The lower cutoff on QED radiation in a hadron beam is not bound to be the same as the QCD one, *i.e.* since there is no issue of α_{em} diverging at low scales. Nevertheless it is plausible to assume that the QCD cutoff is related to the transition from quarks to hadrons, and thus should be applied to all radiation. For radiation off a lepton, there is no such restriction, and PYTHIA then by default sets $p_{\perp evol,min} = 10^{-6}$ GeV for FSR and $5 \cdot 10^{-4}$ for ISR. These values are fully sufficient to cover the emission of any photons observable at a collider. They are also adjusted to be in a region where kinematic reconstruction still works well in double precision. It has been pointed out that they are not sufficiently low to generate the full observable-photon spectrum when PYTHIA is applied to whatever processes could give the highest-energy cosmic rays.

The branching of a photon, $\gamma \to f\bar{f}$, does not fit well into the dipole picture. The choice of a recoiler is based on the history to the largest extent possible, *i.e.* based on what the photon was produced in association with. The photon branchings in part compete with the hard processes involving γ^*/Z production. In order to avoid overlap it makes sense to correlate the maximum γ mass allowed in showers with the minimum γ^*/Z mass allowed in hard processes, by default at 10 GeV. In addition, the shower contribution only contains the pure γ^* contribution, *i.e.* not the Z part, so the mass spectrum above around 50 GeV would not be well described.

Electroweak shower The emission of $W^\pm$ and Z gauge bosons off fermions is an integrated part of the ISR and FSR frameworks, and is fully interleaved with QCD and QED emissions [83]. It is off by default, however, since it takes some time to generate trial emissions, whereof very few result in real emissions unless the fermion transverse momenta are much larger than the W/Z masses. These masses also have a considerable impact on the phase space of emissions, which the shower is not set up to handle with a particularly good accuracy. Therefore, the weak-shower emissions are always matched to the matrix element for emissions off an $f\bar{f}$ weak dipole, or some other $2 \to 3$ matrix element that resembles the topology at hand. Even if the match may not be perfect, at least the main features should be caught that way. Notably, the correction procedure is used throughout the shower evolution, not only for the emission closest to the hard $2 \to 2$ process. Also, the angular distribution in the subsequent $V = W^\pm/Z$ decay is matched to the matrix-element expression for $f\bar{f} \to f\bar{f}V \to f\bar{f}f'\bar{f}'$ (FSR) and $f\bar{f} \to g^*V \to g^*f'\bar{f}'$ (ISR). Afterwards, the $f'\bar{f}'$ system undergoes showers and hadronization just like any $W^\pm/Z$ decay products would.

Special for the weak showers is that couplings are different for left- and right-handed fermions. With incoming unpolarized beams this should average out, at least so long as only one weak emission occurs. In the case of several weak emissions off the same fermion, the correlation between them will carry a memory of the fermion helicity. Such a memory is retained for the affected dipole end. The flavour-changing character of $W^\pm$ emissions also affects the tight relation between the real-emission evolution and Sudakov factors, so-called Bloch–Nordsieck violations. These effects are not expected to be large, but they are not properly included. Another restriction is that there is no simulation of the full γ^*/Z interference: at low masses, the QED shower involves a pure γ^* component, whereas the weak shower generates a pure Z. Finally, it should be remembered that this is not a full (electro)weak shower, which would also have required interactions among gauge bosons, and even involved the Higgs boson. These interactions are included, *e.g.* in the VINCIA EW shower, *cf.* section 4.2.4.

Onia Hard production of charmonium and bottomonium can proceed either through colour-singlet or colour-octet mechanisms. In the former case, the state does not radiate and the onium is therefore produced in isolation, while it is sensible to assume that a shower can

evolve in the latter case, giving an onium state embedded in some amount of jet activity. Currently, both cases are initiated by $2 \to 2$ interactions directly producing an onium state; the alternative mechanism of producing onia during the shower evolution itself [95] is not (yet) implemented. Emissions off an octet-onium state could easily break up a semi-bound quark pair, but might also create a new semi-bound state, and to some approximation these two effects should balance in the onium production rate. The showering implemented here therefore should not be viewed as an accurate description of the emission history step by step, but rather as an effective approach to build up the onium environment. The simulation of branchings is based on the assumption that the full radiation is provided by an incoherent sum of radiation off the quark and off the antiquark of the onium state. Thus, the splitting kernel is taken to be the normal q → qg one, multiplied by a factor of two. A number of corrections to this picture could be imagined; since they would come with opposite signs the assumption is that they cancel out. Further discussion is also included in section 3.3.

Baryon-number-violating decays A complicated case for showering is baryon-number-violating decays, *e.g.* a neutralino decaying to three quarks. It is then not possible to assign an ordinary dipole configuration. Instead half-strength dipoles are constructed between each pair of quarks. That way the total emission rate from each quark is at normal strength, and the recoil can be taken by either of the other two quarks. Similar reduced-showering-rate dipoles can be selected also in a few other cases.

Hidden Valley processes The Hidden Valley (HV) scenario, introduced in section 3.7, has been developed specifically to allow the study of visible consequences of radiation in a hidden sector, either by recoil effects or by leakage back into standard-model particles. A key aspect therefore is that the normal timelike showering machinery has been expanded with a third kind of radiation, in addition to the QCD and QED(+EW) ones [38, 39]. These three kinds are fully interleaved, *i.e.* evolution occurs in a common $p_\perp$-ordered sequence. This radiation may be described either within a (possibly broken) $\mathbf{U(1)}$ or an unbroken $\mathbf{SU(N)}$ gauge group, but not both simultaneously. Thus, one has either HV-photons or HV-gluons as interaction carriers, where the latter are non-Abelian and may branch into more HV-gluons. A set of 12 new particles mirrors the standard-model flavour structure, and is charged under both the SM and the HV symmetry groups, so that they can radiate both into the visible and invisible sector. There is also a new massive particle with only HV charge, sitting in the fundamental representation of the HV gauge group, denoted an HV-quark.

HV particles are only produced in or after the hard process, so only FSR needs to be considered. The HV radiation defines its own set of dipoles, usually between opposite charges. Decays of massive particles can give rise to the same kind of issues as for top decays, *i.e.* that a dipole properly involves the hole of the decaying particle. Matrix-element corrections are implemented for a number of decay processes, with colour, spin, and mass effects included, as for SM processes. These were calculated within the context of the particle content of the MSSM, however, which does not include spin-1 particles with unit colour charge. In such cases spin 0 is assumed instead. By experience, the main effects come from mass and colour flow anyway, so this is not a bad approximation. In the case of a broken $\mathbf{U(1)}$ symmetry, the HV-photon is massive, which requires some kinematics corrections relative to ordinary QED radiation. If decays back to the SM occur, *e.g.* the HV-photon by mixing with the ordinary γ, then also ordinary showers are allowed. By default the coupling strength is fixed, but running is allowed, given the gauge group and the contributing matter content.

4.1.5 Algorithms for automated shower variations and enhanced splittings

Several variations of the simple shower are available in an automated fashion, to help construct uncertainty bands for predictions [20]. That is, weights are constructed and associated with the shower evolution under different alternative conditions, at the same time as the normal showers (with unit weight) are evolved. The properties of an event only need to be analysed once, but can then be filled in one histogram for each distinct variation, with its associated event weight, and at the end these histograms can be combined to provide the uncertainty band. Variations can be set for the renormalization scale for ISR and FSR QCD emissions (separately), for the inclusion of non-singular terms in the ISR and FSR splitting kernels (separately), and for different PDF members.

The veto algorithm is used to generate parton-shower histories for the physics parameters chosen at initialization as normal. Using eq. (16), we can compute sets of weights (which we call variations) for each event reflecting the changed probability for that event under different possible choices of physics parameters. The number of variations calculated is limited only by finite computing and memory resources.

While the proof of unitarity is more easily realized using eq. (16), the algorithm is employed discretely. Thus the factors

$$\frac{f(t')}{g(t')r(t')} \text{ (acc) and } \frac{1-f(t')/g(t')}{1-r(t')} \text{ (disc)}, \tag{127}$$

can be calculated at each discrete step and book-kept during the shower to calculate an event weight. The factors (acc) account for the effect in accepted splittings, while the factors (disc) preserve unitarity from the discarded trial splittings.

Parton-Shower Variations Parton-shower variations are calculated by considering an alternative shower algorithm with an alternative radiation kernel $P'(t,z)$. The physical trial-accept probability, P'_{acc}, for this alternative shower is the ratio of the alternative kernel to the unchanged oversampling kernel. An alternative radiation kernel could have a different treatment of α_{s}, different non-singular terms in the splitting kernels, and/or different effective higher-order terms. In principle, one could also imagine varying the definition of the shower evolution scale, t, as was done in [96], or other shower aspects such as the splitting variable z or the recoil scheme, but calculations of weights for such variations are currently not available in PYTHIA. Suppressing the z dependence for clarity, the algorithm to compute the probability of an event generated by P' based on an event generated using P is [20, 64]:

1. Initialize all weights to the input value of the event $w' = w$.

2. For an accepted branching, multiply the value of the parton-shower-variation weight w' by the ratio of accept probabilities,

$$R'_{\text{acc}}(t) = \frac{P'_{\text{acc}}(t)}{P_{\text{acc}}(t)} = \frac{P'(t)}{P(t)}. \tag{128}$$

3. For a rejected branching, multiply w' by the ratio of rejection probabilities,

$$R'_{\text{disc}}(t) = \frac{P'_{\text{disc}}(t)}{P_{\text{disc}}(t)} = \frac{1-P'_{\text{acc}}(t)}{1-P_{\text{acc}}(t)} = \frac{\hat{P}(t)-P'(t)}{\hat{P}(t)-P(t)}, \tag{129}$$

where $\hat{P}$ is the trial overestimate used for generating a branching.

Note that steps 2 and 3 adjust the branching and no-branching probabilities, respectively, so that the set of weights w' represents a separately unitary event sample. In other words, the samples integrate to the same total cross section, *i.e.* $\langle w' \rangle = \langle w \rangle$. The denominator of the relative discard-ratio, eq. (129), is given by the difference $\hat{P} - P$. If the trial overestimate, $\hat{P}$, is very close to P, the denominator can become close to zero, producing large weights that may be numerically unstable. This occurs because the variation algorithm effectively uses failed trials to estimate the modifications to the no-branching probabilities, and if the algorightm is "too efficient", with few failed trials, then each failed trial must represent a rather large correction, resulting in large fluctuations. A technical fix is to apply a "headroom factor" to the trial functions when automated uncertainty variations are requested so that there is always a non-negligible probability for trials to be discarded. These enhanced rejections come at the cost of computational speed.

Every accepted trial emission contributes a factor R'_{acc} to the final event weight w' in the full shower evolution, while every discarded trial emission contributes a factor R'_{disc}, hence the total weight is just the product:

$$
w' = \prod_{i \in \text{accepted}} \frac{P'_{i,\text{acc}}}{P_{i,\text{acc}}} \prod_{j \in \text{discarded}} \frac{P'_{j,\text{disc}}}{P_{j,\text{disc}}}.
\tag{130}
$$

Technically, a few limiting factors are imposed on the variation weights, as will be described further below, mainly in order to prevent large fluctuations from low-scale branchings that do not have much impact on (infrared and collinear safe) observable distributions.

Renormalization-Scale Variations   A standard method to estimate uncertainties from unknown higher-order contributions is to vary the QCD (and/or QED) renormalization scale(s). For a coherent shower algorithm based on $p_\perp$, a judicious choice of a renormalization-scale prefactor — the so-called CMW scale factor — takes leading second-order corrections from soft-gluon emissions into account, and the framework for shower renormalization-scale variations in PYTHIA is constructed to preserve this property. This is done by including an (optional) explicit $\mathcal{O}(\alpha_s^2)$ compensating term that restores the soft limit of the NLO expansion of the effective scale choice.

Specifically, for each shower kernel $P(z)$, we define renormalization-scale variations by:

$$
P(t,z) = \frac{\alpha_s(p_\perp)}{2\pi} \frac{P(z)}{t} \quad \rightarrow \quad P'(t,z) = \frac{\alpha_s(k p_\perp)}{2\pi} \big(1 + \delta_{\text{soft}}(k)\big) \frac{P(z)}{t},
\tag{131}
$$

where k is the renormalization-scale-variation factor (relative to the default argument $p_\perp$) and $\delta_{\text{soft}}(k)$ is the soft compensation term, which we define as

$$
\delta_{\text{soft}}(k) = (1 - \zeta) \frac{\alpha_s(\mu_{\text{max}})}{2\pi} \beta_0 \ln k,
\tag{132}
$$

with $\beta_0 = (11 N_c - 2 n_F)/3$, $N_c = 3$, and n_F the number of active flavours at the scale $\mu = p_\perp$. The purpose of the factor $(1 - \zeta)$ is to suppress the compensation term outside the soft limit. We define it via:

$$
\zeta = \begin{cases}
z, & \text{for splittings with a } 1/z \text{ singularity}, \\
1 - z, & \text{for splittings with a } 1/(1-z) \text{ singularity}, \\
\min(z, 1-z), & \text{for splittings with a } 1/(z(1-z)) \text{ singularity}.
\end{cases}
\tag{133}
$$

The choice of renormalization scale for the α_s factor in eq. (132) is beyond NLO accuracy. To be conservative and to avoid the risk of overcompensating, our choice is to use the largest local

scale in the problem. Specifically, $\mu_{\text{max}} = \max(m_{\text{dip}}, kp_\perp)$, where m_{dip} is the invariant mass of the emitting colour dipole. Note, this compensation term is only applied to gluon emissions. For $g \rightarrow q\bar{q}$ splittings, the full (uncompensated) variation, $\alpha_{\text{s}}(kp_\perp)/\alpha_{\text{s}}(p_\perp)$ is applied.

A subtlety is that, if the choice of k causes the scale to cross any quark-mass thresholds, $\alpha_{\text{s}}(kp_\perp)$ will not be evaluated with the same number of active flavours n_F as the nominal shower would. For reasonable values of k, this is a small effect since matching conditions are applied in PYTHIA to make the running continuous across thresholds. In principle, an additional term $\alpha_{\text{s}}/(2\pi) \ln(m_q/(kp_\perp))/3$ could be added to compensate for the different β_0 coefficients used in the region. But since this would reduce the variation, we consider it more conservative to leave it out, which may then also partially reflect the additional ambiguity in the running in the vicinity of thresholds.

Thus, we arrive at the following form for the reweighting factor for accepted branchings,

$$R'_{\text{acc}}(t,z) \;=\; \frac{P'_{\text{acc}}(t,z)}{P_{\text{acc}}(t,z)} \;=\; \frac{\alpha_{\text{s}}(kp_\perp)}{\alpha_{\text{s}}(p_\perp)} \left(1 + (1-\zeta)\frac{\alpha_{\text{s}}(\mu_{\text{max}})}{2\pi}\beta_0 \ln k \right). \tag{134}$$

The user has a wide range of options available for calculating parton-shower scale variations for ISR or FSR. Technically, to limit large weight fluctuations between events with and without a very soft branching near the end of the evolution, we impose an absolute limit on the allowed amount of α_{s} variation. By default, $|\Delta\alpha_{\text{s}}| \leq 0.2$. This choice is not a significant restriction on the range of variation for perturbative branchings (even when $\alpha_{\text{s}} \sim 0.5$, a full 40% amount of variation is still allowed), but it does prevent variations corresponding to branchings very near the cutoff, which should *not* have a significant impact.

Finite-Term Variations   Another class of parton-shower variations is related to non-singular terms in the splitting kernels, also called "finite terms". The exact forms of the parton-shower branching kernels are only uniquely determined in the singular (infrared soft and/or collinear) limits. Away from these limits, one is in principle free to choose shower kernels that differ from, say, the DGLAP kernels, by non-singular contributions. (Notably, this freedom is exploited in the context of matrix-element corrections and in the POWHEG method, *cf.* section 5.) Setting the non-singular terms to zero is an arbitrary choice and is anyway not stable against reparameterizations [64, 97]. In the absence of matrix-element corrections, it therefore makes sense to vary the finite terms, *e.g.* provide an estimate for how significant the effects of the missing matrix-element corrections could be.

The effects of finite-term variations are qualitatively different from those of renormalization-scale variations. By construction, the latter are proportional to the shower kernels. But away from the singular limits, the pole terms in the radiation function are not enhanced, and may not bear much resemblance to a corresponding matrix-element based calculation for the process at hand. In exactly these regions, modest finite terms can produce larger variations than scale choices can. PYTHIA provides the possibility to explore such variations.

To account for non-singular terms, the radiation function based on DGLAP is modified as follows:

$$\frac{P(z)}{Q^2}\, dQ^2 \;\rightarrow\; \left(\frac{P(z)}{Q^2} + \frac{c}{m_{\text{dip}}^2} \right) dQ^2 = \left(P(z) + \frac{c\, Q^2}{m_{\text{dip}}^2} \right) \frac{dt}{t}, \tag{135}$$

where m_{dip} is the invariant mass of the dipole in which the splitting occurs and the finite-term variation parameter c is a dimensionless constant of order unity. This expression uses the identity $dQ^2/Q^2 = dt/t$, which holds for any $t = f(z)Q^2$ and covers all the PYTHIA evolution

variables. The alternative splitting kernel is then

$$P'(t,z) \ = \ \frac{\alpha_s}{2\pi} \, \mathcal{C} \left(\frac{P(z) \ + \ c \, Q^2/m_{\text{dip}}^2}{t} \right), \tag{136}$$

where $\mathcal{C}$ is the colour factor, leading to the acceptance factor:

$$R'_{\text{acc}} \ = \ \frac{P'_{\text{acc}}}{P_{\text{acc}}} \ = \ 1 + \frac{c \, Q^2/m_{\text{dip}}^2}{P(z)}. \tag{137}$$

From this expression, it is immediately apparent that the relative variation explicitly vanishes when $Q^2 \to 0$ or $P(z) \to \infty$. For gluon emission off timelike massive quarks, Q^2 is changed to the off-shellness of the massive quark, defined as $Q^2 = (p_b + p_g)^2 - m_b^2 = 2p_b \cdot p_g$ [77], with p_b the four-momentum of the massive quark and p_g that of the emitted gluon.

We suggest variations on the order of unity ($c \in [-2, 2]$ to be conservative) as a baseline. This is motivated by PYTHIA's Matrix Element Corrections (MEC) for Z-boson decays [75] and by explicit numerical studies [97] that found order-unity differences between different physical processes and different antenna-shower formalisms.

Veto Algorithm with Biased Kernels   The veto algorithm can also be modified to enhance the rate of some or all splittings in the shower. In this case, the event weight is modidified to account for the relative overpopulation of events with such splittings (and underpopulation of events without). This has applications, *e.g.* for enhancing the rate of relatively rare splittings, like $q \to q\gamma$ and $g \to b\bar{b}$. The biasing method implemented in PYTHIA 8.3 is described in [15, 20] and is formally identical to that of [98].

For simplicity, assume that one of many splitting types is to be biased by an enhanced trial factor $E > 1$. The (new) oversampling kernel will be

$$\hat{P}_{\text{biased}} = E\hat{P}, \tag{138}$$

so that more of the desired splittings are encountered in the veto algorithm. Keeping the (old) accept probability unchanged, the bias can be incorporated into a weight inversely proportional to the biasing factor:

$$P_{\text{acc}} \ = \ \frac{P}{\hat{P}} \ \ ; \ \ R_{\text{acc}} \ = \ \frac{\hat{P}}{\hat{P}_{\text{biased}}} \ = \ \frac{1}{E}. \tag{139}$$

With this modification, the original distribution for the desired splitting is recovered, since $R_{\text{acc}} P_{\text{acc}} \hat{P}_{\text{biased}} = P$. Similarly, when a biased trial branching is not accepted, a different weight is applied to the event:

$$R_{\text{disc}} \ = \ \frac{1 - P_{\text{acc}} R_{\text{acc}}}{1 - P_{\text{acc}}} \ = \ \frac{\hat{P}}{\hat{P} - P} \left(1 - \frac{P}{\hat{P}_{\text{biased}}} \right) \ \overset{P \ll \hat{P}_{\text{biased}}}{\to} \ \frac{\hat{P}}{\hat{P} - P}, \tag{140}$$

For very large values of E, the reweighting factor becomes independent of the bias, but is still important for recovering the physical no-branching probability. Strictly speaking, E does not have to be a constant and could be allowed to depend upon the shower kinematics, but currently only a simple constant factor is implemented in PYTHIA, which can be specified independently for each type of shower splitting.

4.2 The VINCIA antenna shower

The VINCIA shower implements an interleaved $p_\perp$-ordered evolution based on the so-called antenna formalism. In event-generator contexts, this type of shower was first pioneered by the ARIADNE model [4, 54], which was widely used *e.g.* at LEP. For completeness, we note that the objects we call "antennae" here were actually called dipoles in that context, but today the term dipole has taken on a different meaning, see, *e.g.* section 4.3 on DIRE.

Especially for FSR QCD radiation, VINCIA shares many features with ARIADNE, including its evolution-variable definition and its antenna-style $2 \mapsto 3$ approach to parton branchings in which both parents can acquire transverse recoil and the soft eikonal remains unpartitioned. These latter two properties are specific to antenna showers.

For ISR, VINCIA's treatment is quite different from that of ARIADNE, with VINCIA extending the concept of (interleaved) backwards evolution [52, 78] to the antenna picture [99] via coherent II and IF antennae [100], as well as so-called Resonance–Final (RF) ones [88]. The latter are relevant to the decays of coloured resonances, such as top quarks. They come with their own, dedicated kinematic mapping which is constructed to preserve the invariant mass of the decaying resonance. Since all of its building blocks are explicitly coherent (at least at leading colour) and interleaved in a single common sequence of decreasing $p_\perp$ values, VINCIA should exhibit a quite reliable description of soft coherence effects across essentially all physical contexts.

This extends to QED, for which VINCIA's default antenna functions [101, 102] include fully coherent (multipole) soft interference effects in addition to the collinear DGLAP structures. We are not aware of any other multipole QED treatment that can be interleaved with the QCD evolution. (*e.g.* the YFS formalism [103] is constructed purely as an "afterburner", *i.e.* not interleaved with the QCD shower, and collinear logarithms can only be included order by order.)

A further difference with respect to ARIADNE is that VINCIA's QCD and multipole QED showers are constructed as so-called "sector" antenna showers, in which the phase space is divided into distinct (non-overlapping) colour and kinematics sectors, each of which only receives contributions from one specific antenna-branching kernel. This has a number of mainly technical consequences which will be elaborated on further below, to do with making the incorporation of higher-order corrections as straightforward and efficient as possible. For the time being, ARIADNE-style "global" showers also remain available as a non-default option.

Effects of particle masses are systematically included, both via mass corrections to the antenna functions such that all relevant (quasi-)collinear limits are reproduced [100, 104], and by the use of exact massive phase-space factorizations. The current default behaviour is to treat bottom and heavier quarks, and muons and tau leptons, as massive in VINCIA, though this can be changed if desired. (Weak bosons are always treated as massive.) A subtlety arises in the treatment of incoming heavy-flavour quarks (and, potentially, muons). Kinematically, such partons are book kept as massless, similarly to the choice made in PYTHIA's simple shower. The consequence is that the treatment of mass effects for initial-state partons in VINCIA is less rigorous than for final-state ones. One should also be aware that there can be a non-trivial interplay with the flavour scheme employed by the chosen PDF set.

As a complementary option to the multipole QED shower, VINCIA also includes a module for full-fledged electroweak showers [23, 105]. This option includes the full set of EW-branching kernels, including both Higgs couplings and gauge-boson self-interactions, tallying to more than 1000 EW-antenna functions in total. The main limitation is that only the relevant (quasi-)collinear limits are implemented, not the full soft interference structure; thus, also the QED treatment is effectively reduced to a DGLAP-style treatment when using this option. Note also that the EW module is based explicitly on VINCIA's underlying formalism for helicity-dependent showers [106, 107] (*e.g.* to tell left- and right-handed weak bosons apart).

This module therefore requires Born partons with assigned helicities, which is not the default in PYTHIA, and must be provided either via external LHEF events with helicity information, or via VINCIA's dedicated option for hard-process helicity selection. (The latter is based on PYTHIA's run-time interface to external matrix-element libraries; see the program's `online manual` and example programs for configuration and linking instructions.)

A further feature that was originally introduced with VINCIA's electroweak-shower module, but is now applied independently of it, is a novel treatment of finite-width effects, called interleaved resonance decays [23], which are the default in VINCIA. This means that decays of short-lived resonances, such as top quarks, W/Z bosons, or BSM particles, are inserted in the shower evolution at an evolution scale of order the off-shellness of the resonance, instead of being treated sequentially, after the shower of the hard process. This reflects the physical picture that short-lived particles should not be able to radiate at frequencies lower than the inverse of their lifetime; only their decay products can do that. This can produce subtle changes in reconstructed invariant-mass distributions, relative to conventional (non-interleaved) resonance decays.

All of VINCIA's shower modules are fully interleaved with PYTHIA's treatment of multiparton interactions (MPI), in the same manner as for the simple-shower model.

4.2.1   Common features

Some aspects are common to all of VINCIA's shower modules. This includes the definition of the evolution variable as well as recoil schemes and phase-space factorizations. These common features are discussed in this subsection before going into further detail on each of the specific components of VINCIA's shower implementations.

Evolution variables   All showers in VINCIA, including the QED and EW ones, are evolved in a Lorentz-invariant scaled notion of off-shellness, based on a generalized version of the ARIADNE definition of transverse momentum. For a generic branching $IK \to ijk$,

$$p_{\perp j}^2 = \frac{\bar{q}_{ij}^2 \bar{q}_{jk}^2}{s_{\max}}, \tag{141}$$

where the off-shellness for final-state partons is defined as

$$\bar{q}_{ij}^2 = (p_i + p_j)^2 - m_I^2 = m_{ij}^2 - m_I^2, \qquad i \text{ is final}, \tag{142}$$

and that for initial-state partons is obtained via crossing (and an overall sign change to make it positive),

$$\bar{q}_{ij}^2 = -(p_i - p_j)^2 + m_I^2, \qquad i \text{ is initial}. \tag{143}$$

These both involve the positive invariant $2p_i \cdot p_j$ but differ in the signs of pre- *vs.* post-branching parton masses. This reflects the underlying crossing and sign change, combined with the propagator structure of backwards evolution. For convenience, we define the dimensionful invariant

$$s_{ij} \equiv 2p_i \cdot p_j, \tag{144}$$

regardless of whether particles i and j are massless or not. The maximal antenna invariant, $s_{\max}$, is then defined by

$$s_{\max} = \begin{cases} s_{IK} & \text{FF}, \\ s_{aj} + s_{jk} & \text{IF \& RF}, \\ s_{ab} & \text{II}, \end{cases} \tag{145}$$

where initial-state partons are labelled with letters from the beginning of the alphabet (a and b) and final-state ones are labelled by i, j, k, …. Below, that labelling convention will be used systematically to distinguish initial- and final-state partons.

We also define dimensionless (scaled) invariants and masses,

$$y_{ij} = \frac{s_{ij}}{s_{\max}} \quad ; \quad \mu_i^2 = \frac{m_i^2}{s_{\max}} . \tag{146}$$

For massless kinematics, the scaled invariants have very simple relations to the z variables of DGLAP-style approaches. Thus, for final-final (FF) antennae, the CM energy fractions are

$$x_k = \frac{2E_k}{\sqrt{s_{IK}}} = 1 - y_{ij}, \tag{147}$$

and similarly for the two other permutations of (i, j, k). For initial-initial (II) antennae, the incoming legs are always massless and the y_{AB} invariant can be identified with the z variable, since

$$y_{AB} = \frac{s_{AB}}{s_{ab}} = \frac{x_A x_B}{x_a x_b} = z_a z_b, \tag{148}$$

where we have emphasized that, for antenna-II branchings, in general both x values change, with

$$z_a = \frac{x_A}{x_a} = \sqrt{y_{AB} \frac{1 - y_{jb}}{1 - y_{aj}}}, \quad \text{and} \quad z_b = \frac{x_B}{x_b} = \sqrt{y_{AB} \frac{1 - y_{aj}}{1 - y_{jb}}}. \tag{149}$$

There is still the constraint that, in the a-collinear limit $z_b \to 1$ and *vice versa* (for massless j).

For initial-final (IF) ones, the exact relations are more involved but again the collinear limits can be examined via

$$z_a = 1 - y_{jk}, \qquad z_k = 1 - y_{aj}. \tag{150}$$

Recoil schemes In the antenna formalism, the branching recoil is shared between both antenna parents for FSR emissions in an on-shell kinematics map along the lines of [104,108,109] including full mass dependence. This is illustrated in the top row in fig. 8. In the collinear limits, any transverse momentum is fully absorbed within the collinear pair, and the anti-collinear parent recoils purely longitudinally; therefore, in these limits the map agrees with the conventional dipole ones, *cf.* [87,110,111]. This means that the post-branching momenta are constructed as

$$p_i^\mu = (E_i, 0, 0, |\vec{p}_i|), \tag{151}$$

$$p_j^\mu = (E_j, -|\vec{p}_j| \sin \theta_{ij}, 0, |\vec{p}_j| \cos \theta_{ij}), \tag{152}$$

$$p_k^\mu = (E_k, |\vec{p}_k| \sin \theta_{ik}, 0, |\vec{p}_k| \cos \theta_{ik}), \tag{153}$$

in the rest frame of the parent I-K antenna. Here, the energies are given by

$$E_i = \frac{s_{ij} + s_{ik} + 2m_i^2}{2m_{IK}}, \qquad E_j = \frac{s_{ij} + s_{jk} + 2m_j^2}{2m_{IK}}, \qquad E_k = \frac{s_{ik} + s_{jk} + 2m_k^2}{2m_{IK}}, \tag{154}$$

and the angles by

$$\cos \theta_{ij} = \frac{2E_i E_j - s_{ij}}{2|\vec{p}_i||\vec{p}_j|}, \qquad \cos \theta_{ik} = \frac{2E_i E_k - s_{ik}}{2|\vec{p}_i||\vec{p}_k|} . \tag{155}$$
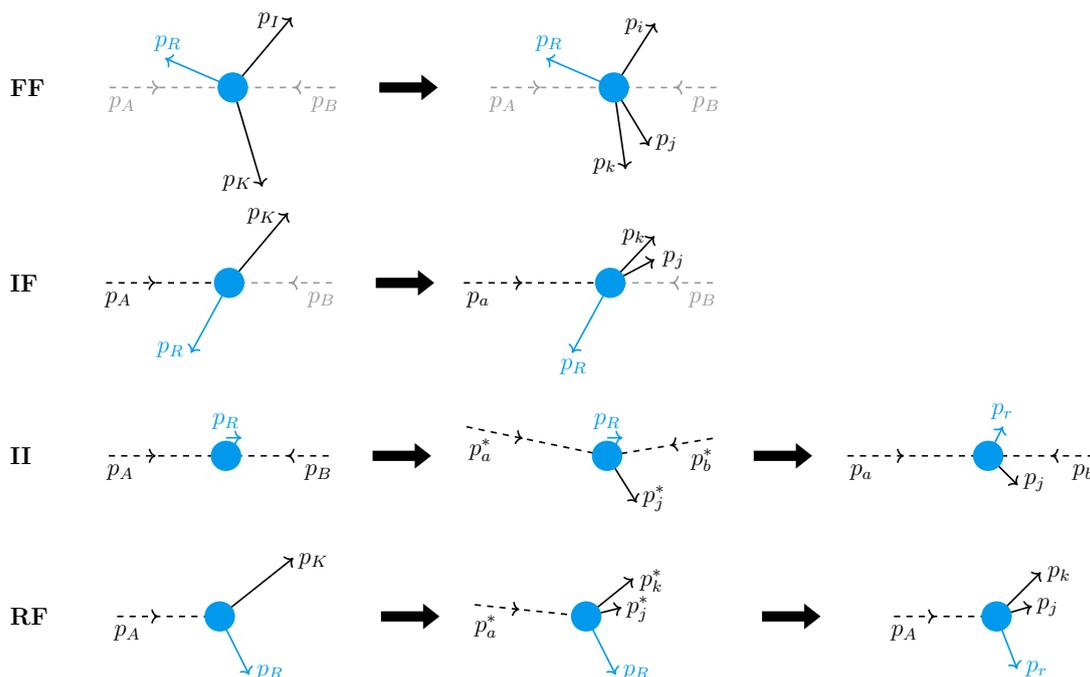
Figure 8: Illustration of VINCIA's kinematic maps, for final-final (FF), initial-final (IF), initial-initial (II), and resonance-final (RF) branchings. Dashed lines represent initial-state momenta, non-participating legs are shaded grey, and the set of final-state spectators (R) is shown in cyan. For II and RF branchings, the frame reinterpretation done in the last step imparts a collective recoil to the final-state spectators, $p_R \rightarrow p_r$.

The branching plane is then rotated in the x-y plane, by an angle ϕ, which is uniformly sampled in the range $[0, 2\pi]$. The azimuthal rotation by ϕ is combined with a rotation by a polar angle ψ whose value is only uniquely constrained in the two collinear limits. The case $\psi \rightarrow 0$ corresponds to parton i being oriented in the same direction as parton I. This limiting value is mandated in the K-collinear limit in which parton i must recoil purely longitudinally. In the I-collinear limit, $\psi \rightarrow \pi - \theta_{ik}$ ensures that parton k recoils longitudinally. A few different options are provided for how to interpolate between those limits, *cf.* [104, 109].

For initial-state radiation, one (or both) parents must remain collinear to the beam axis, and instead the hard system can now acquire transverse recoil. This makes it more complicated to define a truly antenna-like recoil scheme, and VINCIA's choices [100, 112, 113] are more similar to dipole treatments such as the ones in [114–117]. In the case of IF antennae, this amounts to constructing the post-branching momenta as

$$x_a = x_A/y_{AK} \qquad \left(\implies p_a^\mu = \frac{1}{y_{AK}} p_A^\mu \right), \tag{156}$$

$$p_j^\mu = \frac{(y_{ak} + \mu_j^2 - \mu_k^2) + (y_{ak} - y_{aj})\mu_K^2 - y_{AK}y_{ak}}{y_{AK}} p_A^\mu + y_{aj} p_K^\mu + \sqrt{\Gamma_{ajk}} \, q_{\perp\max}^\mu, \tag{157}$$

$$p_k^\mu = \frac{(y_{aj} - \mu_j^2 + \mu_k^2) + (y_{aj} - y_{ak})\mu_K^2 - y_{AK}y_{aj}}{y_{AK}} p_A^\mu + y_{ak} p_K^\mu - \sqrt{\Gamma_{ajk}} \, q_{\perp\max}^\mu, \tag{158}$$

in the A-K rest frame, as illustrated in the second row in fig. 8. In this context, $\Gamma_{ajk} = y_{aj}y_{jk}y_{ak}$ and $q_{\perp\max}^\mu$ denotes the transverse component in terms of a spacelike four-vector that is perpendicular to p_A and p_K and obeys $q_{\perp\max}^2 = -(s_{aj} + s_{ak})$.

For II antennae, both initial-state particles are evolved at the same time and therefore both momentum fractions change simultaneously [99, 118], *cf.* the third row in fig. 8 for an illustration. (Note that this is different to "dipole" kinematics, in which only one of the incoming x fractions can change in each branching.) Consequently, the post-branching momenta are constructed as

$$p_B^\mu, \tag{159}$$

$$x_a = x_A/z_a \qquad (\implies p_a^\mu = p_A^\mu/z_a), \tag{160}$$

$$x_b = x_B/z_b \qquad (\implies p_b^\mu = p_B^\mu/z_b), \tag{161}$$

$$p_j^\mu = y_{jb}p_a^\mu + y_{aj}p_b^\mu + \sqrt{y_{aj}y_{jb} - \mu_j^2}\, q_{\perp\mathrm{max}}^\mu, \tag{162}$$

$$p_r^\mu = p_a^\mu + p_b^\mu - p_j^\mu, \tag{163}$$

where the $z_{a,b}$ fractions are defined in eq. (149), $q_{\perp\mathrm{max}}^\mu$ is again a spacelike four-vector perpendicular to p_A and p_B with $q_{\perp\mathrm{max}}^2 = -s_{ab}$, and r denotes the recoiling spectator system whose combined invariant mass and rapidity are both unchanged by the branching: $p_r^2 = p_R^2$ and $y_r = y_R$.

In both IF and II antennae, all momenta are rotated about the branching plane by a uniformly distributed angle $\phi \in [0, 2\pi]$.

For RF antennae [88], the invariant mass of the resonance must be kept fixed, $p_a^2 = p_A^2 = m_A^2$. The post-branching kinematics are therefore constructed in the resonance rest frame with the z-axis defined along p_K, so that

$$p_A^\mu = p_a^\mu = (m_A, 0, 0, 0), \tag{164}$$

$$p_k^\mu = \left(E_k, 0, 0, \sqrt{E_j^2 - m_k^2}\right), \tag{165}$$

$$p_j^\mu = \left(E_j, \sqrt{E_j^2 - m_j^2}\sin\theta_{jk}, 0, \sqrt{E_j^2 - m_j^2}\cos\theta_{jk}\right), \tag{166}$$

$$p_r^\mu = \left(m_A - E_k - E_j, -\sqrt{E_j^2 - m_j^2}\sin\theta_{jk}, 0, -\sqrt{E_k^2 - m_k^2} - \sqrt{E_j^2 - m_j^2}\cos\theta_{jk}\right), \tag{167}$$

where r denotes the remainder of the resonance decay system and

$$E_j = \frac{s_{aj}}{2m_a}, \quad E_k = \frac{s_{ak}}{2m_a}, \quad \cos\theta_{jk} = \frac{2E_b E_g - s_{jk}}{2\sqrt{(E_k^2 - m_k^2)(E_j^2 - m_j^2)}}. \tag{168}$$

These momenta are rotated about the y axis such that the set of recoilers are along -z, so that only j and k receive transverse recoil. Again, the momenta are subsequently rotated by a uniformly sampled angle $\phi \in [0, 2\pi]$ about the z axis. The original orientation of p_K with respect to z is then recovered in a final step. This map is illustrated in the bottom row of fig. 8.

Helicity Dependence   All of VINCIA's QCD and EW (but not QED) antenna functions are implemented with full helicity dependence, *i.e.* decomposed into distinct terms for each set of contributing helicities. This facilitates helicity-dependent showering and matching, given a polarized Born state [107]. For brevity, the QCD antenna functions shown below are averaged over pre-branching helicities and summed over post-branching ones; see [100] for details on their individual helicity components.

Biased branchings and uncertainty weights   Just as for the simple shower, VINCIA contains several options for artificially increasing (or suppressing) the probabilities for different branching types to occur, accompanied by non-unity event weights to compensate for how over- or

under-represented each generated event becomes in the resulting sample. This can be especially useful to enhance the rate of rare splittings, such as $g \to b\bar{b}$. The general procedure is described in section 4.1.5 and follows the formalism presented in [20].

As a relatively minor extension, VINCIA also allows for "enhancement" factors smaller than unity, which then act to suppress the corresponding branchings. The intended use case is to focus on Sudakov-suppressed regions of phase space. In the algorithms described in [15, 20], enhancement factors smaller than unity are not guaranteed to produce positive weights. In VINCIA's implementation, this issue is sidestepped by letting trial branchings be enhanced by a factor $\max(1, E)$,

$$\hat{P}_{\text{biased}} = \max(1, E)\hat{P} \,, \tag{169}$$

where $\hat{P}$ is the unenhanced trial-generation probability density and E is the enhancement (or suppression) factor. Thus for $E < 1$ the trial probability is not modified. Conversely, each trial branching is accepted with a probability

$$P_{\text{acc}} = \frac{\min(1, E)P}{\hat{P}} \,, \tag{170}$$

where $P/\hat{P}$ is the unbiased accept probability. The reweighting factor for an accepted trial branching remains $R_{\text{acc}} = 1/E$ (*cf.* section 4.1.5), while the reweighting factor for a discarded one generalizes to

$$R_{\text{disc}} = \frac{\hat{P}_{\text{biased}} - P}{\hat{P}_{\text{biased}} - EP} \,. \tag{171}$$

Thus, in VINCIA's version of the enhancement algorithm, both R_{acc} and R_{disc} are positive definite for any $E > 0$ and $P < \hat{P}$.

Although automated shower-variation weights were a signature feature of early versions of VINCIA [64], such variations have not yet been incorporated into the current VINCIA implementation in PYTHIA 8.3 but remain planned for a future revision. See the program's online manual for updates.

4.2.2 QCD showers

In their present incarnation, VINCIA's QCD showers are fully developed within the so-called sector framework [97, 100, 106, 108, 119–121], in which only a single branching contributes per phase-space point. This is enforced by dividing the phase space into sectors according to a decomposition of unity as given by the following sum of Heaviside step functions,

$$1 = \sum_j \Theta_j^{\text{sct}}(p_{\perp j}^2, \zeta_j, \phi_j) = \sum_j \theta\left(\min_k \{Q_{\text{res},k}^2\} - Q_{\text{res},j}^2\right). \tag{172}$$

To discriminate between the different sectors, a "sector resolution" variable is used, which we define to be [97]

$$Q_{\text{res},j}^2 = \begin{cases} p_{\perp j}^2, & \text{if } j \text{ a gluon}, \\ \bar{q}_{ij}^2 \sqrt{\dfrac{\bar{q}_{jk}^2}{s_{\max}}}, & \text{if } (i,j) \text{ a quark} - \text{antiquark pair}, \end{cases} \tag{173}$$

with $p_{\perp j}^2$ and $\bar{q}_{ij}$ as defined in eq. (141). The asymmetric choice for quark-antiquark pairs accounts for the fact that in gluon splittings with an arbitrary colour-connected recoiler $X_I g_K \mapsto X_i q_j \bar{q}_k$, there is no singularity associated to the i-j-collinear limit [97].

The shower evolution is given by the exponentiation of leading-order antenna functions [118–126], specifically sector-antenna ones defined by the ratio of colour-ordered squared amplitudes,

$$
A_{j/IK} = g_s^2 \mathcal{C}_{j/IK} \bar{A}_{j/IK} = \frac{\left| \mathcal{M}_{ijk}(q; p_i, p_j, p_k) \right|^2}{\left| \mathcal{M}_{IK}(q; p_I, p_K) \right|^2}, \tag{174}
$$

and the coupling- and colour-factor-stripped antenna function $\bar{A}_{j/IK}$. Antenna functions for quark-antiquark, quark-gluon, and gluon-gluon parents can be derived from off-shell photon decays $\gamma \to q\bar{q}$ [123], neutralino decays $\tilde{\chi}^0 \to \tilde{g}g$ [124], and Higgs decays $H \to gg$ [125], respectively. An antenna function derived in this way will include the full single-unresolved singularity structure of all colour dipoles in the given colour-ordered amplitude.

When multiple colour dipoles are present in the three-particle state used to derive the function, these can be divided into sub-antenna functions,

$$
\begin{aligned}
A_{g/qg}(p_i, p_j, p_k) &= A_{g/qg}^{gl}(p_i, p_j, p_k) + A_{g/qg}^{gl}(p_i, p_k, p_j), \\
A_{g/gg}(p_i, p_j, p_k) &= A_{g/gg}^{gl}(p_i, p_j, p_k) + A_{g/gg}^{gl}(p_i, p_k, p_j) + A_{g/gg}^{gl}(p_j, p_i, p_k).
\end{aligned} \tag{175}
$$

Such functions build the basis for so-called *global* antenna showers, in which every antenna radiates over all of its branching phase space, and only the sum of all antennae recovers the full single-unresolved singularity structure. Denoting these by $A_{j/IK}^{gl}$, the specific choices for global final-state antenna functions in VINCIA are

$$
A_{g/q\bar{q}}^{gl}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2, \mu_k^2) = \frac{g_s^2 \mathcal{C}_{g/q\bar{q}}}{s_{IK}} \left[\frac{(1-y_{ij})^2 + (1-y_{jk})^2}{y_{ij}y_{jk}} + 1 - \frac{2\mu_i^2}{y_{ij}^2} - \frac{2\mu_k^2}{y_{jk}^2} \right], \tag{176}
$$

$$
A_{g/qg}^{gl}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2) = \frac{g_s^2 \mathcal{C}_{g/qg}}{s_{IK}} \left[\frac{(1-y_{ij})^3 + (1-y_{jk})^2}{y_{ij}y_{jk}} + 2 - y_{ij} - \frac{y_{jk}}{2} - \frac{2\mu_i^2}{y_{ij}^2} \right], \tag{177}
$$

$$
A_{g/gg}^{gl}(s_{IK}; y_{ij}, y_{jk}) = \frac{g_s^2 \mathcal{C}_{g/gg}}{s_{IK}} \left[\frac{(1-y_{ij})^3 + (1-y_{jk})^3}{y_{ij}y_{jk}} + 3 - \frac{3}{2}y_{ij} - \frac{3}{2}y_{jk} \right], \tag{178}
$$

$$
A_{q/gX}^{gl}(s_{IK}; y_{ij}, y_{jk}, y_{ik}, \mu_Q^2) = \frac{g_s^2 \mathcal{C}_{q/Xg}}{2s_{IK}} \frac{1}{y_{ij} + 2\mu_Q^2} \left[y_{ik}^2 + y_{jk}^2 + \frac{2\mu_Q^2}{y_{ij} + 2\mu_Q^2} \right]. \tag{179}
$$

They only differ from the ones given in [118–126] by non-singular terms. Below, we show how VINCIA's sector-antenna functions, $A_{j/IK}^{sct}$, are constructed from these building blocks.

Single-unresolved limits In the sector shower formalism, there is only a single branching kernel that contributes per phase-space point. In order to capture the correct leading-logarithmic structure of QCD matrix elements, it is therefore vital that sector-antenna functions fully incorporate all single-unresolved limits of a given antenna/dipole. This means, that a single sector-antenna function has to reproduce the full eikonal in the soft-gluon limit,

$$
A_{j/IK}^{sct}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2, \mu_k^2) \xrightarrow{g_j \text{ soft}} g_s^2 \mathcal{C}_{j/IK} \left[\frac{2s_{ik}}{s_{ij}s_{jk}} - \frac{2m_i^2}{s_{ij}^2} - \frac{2m_k^2}{s_{jk}^2} \right], \tag{180}
$$

while reproducing the full massive DGLAP splitting kernel $P_{I \to ij}(z, \mu_i^2)$ (or $P_{I \to ij}(z, \mu_i^2)/z$ for initial-state partons) in any (quasi-)collinear limit,

$$
A_{j/IK}^{sct}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2, \mu_k^2) \xrightarrow{i \| j} g_s^2 \mathcal{C}_{j/IK} \frac{P_{I \to ij}(z, \mu_i^2)}{s_{ij}}. \tag{181}
$$

This differs from conventional (non-sector) parton-shower algorithms, in which the soft and/or collinear singularity structures are partial fractioned onto different branching kernels. *E.g.* in DGLAP-based and dipole approaches, the soft eikonal is partial fractioned onto two separate kernels (which are associated with two different recoil maps), and the same is true of gluon-collinear singularities in both dipole and global-antenna showers. In the sector-antenna formalism, each antenna function reproduces both the full eikonal and the full DGLAP kernel in the respective limits, and double counting is avoided by allowing only one such antenna function to contribute per phase-space point.

FSR antenna functions   In VINCIA, final-final (FF) sector-antenna functions for gluon emissions are constructed from their global counterparts, eqs. (176) to (179), by symmetrizing over colour-connected gluons in the following way,

$$
\begin{aligned}
A^{\mathrm{sct}}_{g/IK}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2, \mu_k^2) = {} & A^{\mathrm{gl}}_{g/IK}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2, \mu_k^2) \\
& + \delta_{Ig} A^{\mathrm{gl}}_{g/IK}(s_{IK}; y_{ij}, 1 - y_{jk}, \mu_i^2, \mu_k^2) \\
& + \delta_{Kg} A^{\mathrm{gl}}_{g/IK}(s_{IK}; 1 - y_{ij}, y_{jk}, \mu_i^2, \mu_k^2),
\end{aligned}
\tag{182}
$$

where $\delta_{Ig} = 1$ if I is a gluon and zero otherwise, and similarly for K. Note that the symmetrization is done on the CM energy fraction of the relevant gluon(s), as in $(y_{jk} = 1 - x_i) \to (1 - y_{jk} = x_i)$ in the symmetrization for $I \to ij$, instead of via explicit permutations of the i and j momenta as in eq. (175), which would correspond to $y_{jk} \to (y_{ik} = 1 - y_{jk} - y_{ij})$. This slight difference (which vanishes in the relevant collinear limit $y_{ij} \to 0$) is to ensure finiteness in phase-space regions close to the "hard" boundary y_{ik}. Although the $y_{ik} = 0$ region will never belong to the j-emission sector, this damping of the singularity is important as it allows for the sampling of the sector-antenna function over all of phase space with a post-hoc imposed sector veto. Additionally, it ensures numerical stability whenever sector boundaries become close to the y_{ik}-singular region.

For gluon-splitting sector-antenna functions, an equivalent procedure yields

$$
A^{\mathrm{sct}}_{q/gX}(s_{IK}; y_{ij}, y_{jk}, y_{ik}, \mu_Q^2) = 2 A^{\mathrm{gl}}_{q/gX}(s_{IK}; y_{ij}, y_{jk}, y_{ik}, \mu_Q^2).
\tag{183}
$$

Antenna functions for final-state partons that are colour-connected to incoming ones, as in initial-final (IF) or resonance-final (RF) colour flows, are discussed below.

ISR antenna functions   As for final-state radiation, sector-antenna functions involving initial-state partons can be obtained by symmetrizing corresponding global ones over final-state gluons. The reason initial-state legs do not need to be symmetrized is that there is no sector for "emission into the initial state". (Analogously, while jet algorithms may decide to cluster final-state partons either with each other or with the beam, the beam itself is hard by definition and cannot be clustered away.)

This means that, even in the global-antenna approach, beam-collinear singularities do not need to be partial-fractioned. Hence, for II antennae, there is no difference between global and sector-antenna functions; while for initial-final gluon emissions, antenna functions with two final-state gluons are symmetrized as follows,

$$
\begin{aligned}
A^{\mathrm{sct,IF}}_{g/AK}(s_{AK}; y_{aj}, y_{jk}, \mu_a^2, \mu_k^2) = {} & A^{\mathrm{sct,IF}}_{g/AK}(s_{AK}; y_{aj}, y_{jk}, \mu_a^2, \mu_k^2) \\
& + \delta_{gK} A^{\mathrm{sct,IF}}_{g/AK}(s_{AK}; 1 - y_{aj} + y_{jk}, y_{jk}, \mu_a^2, \mu_k^2).
\end{aligned}
\tag{184}
$$

Finiteness close to the spurious $y_{ak} \to 0$ singularity is here again ensured by adding y_{jk} to the symmetrized argument. Initial-final antenna functions describing final-state gluon splittings are obtained in exactly the same way as in eq. (183).

Global initial-final and initial-initial antenna functions are obtained from eqs. (176) to (179) by crossing partons from the final state into the initial state. For initial-initial antennae, the crossing $(I, K, i, k) \rightarrow (-A, -B, -a, -b)$ implies:

$$
\begin{aligned}
y_{ij} = \frac{s_{ij}}{s_{IK}} &\quad \rightarrow \quad \frac{-s_{aj}}{s_{AB}} = -\frac{y_{aj}}{y_{AB}}, \\
y_{jk} = \frac{s_{jk}}{s_{IK}} &\quad \rightarrow \quad \frac{-s_{jb}}{s_{AB}} = -\frac{y_{jb}}{y_{AB}}, \\
y_{ik} = \frac{s_{ik}}{s_{IK}} &\quad \rightarrow \quad \frac{s_{ab}}{s_{AB}} = \frac{1}{y_{AB}},
\end{aligned}
\tag{185}
$$

while for initial-final antennae, the crossing $(I, i) \rightarrow (-A, -a)$ yields:

$$
\begin{aligned}
y_{ij} = \frac{s_{ij}}{s_{IK}} &\quad \rightarrow \quad \frac{-s_{aj}}{-s_{AK}} = \frac{y_{aj}}{y_{AK}}, \\
y_{jk} = \frac{s_{jk}}{s_{IK}} &\quad \rightarrow \quad \frac{s_{jk}}{-s_{AK}} = -\frac{y_{jk}}{y_{AK}}, \\
y_{ik} = \frac{s_{ik}}{s_{IK}} &\quad \rightarrow \quad \frac{-s_{ak}}{-s_{AK}} = \frac{y_{ak}}{y_{AK}}.
\end{aligned}
\tag{186}
$$

The RF antenna functions are identical to the IF ones. The full set of VINCIA antenna functions, including their helicity contributions, can be found in [100].

The strong coupling VINCIA offers the same basic options for the strong coupling as the simple shower does, with up to 2-loop running matched across flavour thresholds and an option to use the CMW scheme. However, whereas the main tuneable parameters in the simple shower are the effective values of $\alpha_s^{\text{ISR}}(M_Z^2)$ and $\alpha_s^{\text{FSR}}(M_Z^2)$ (which may then be interpreted as being given in a renormalization scheme not necessarily identical to $\overline{\text{MS}}$), in VINCIA one instead specifies a single common value for $\alpha_s^{\overline{\text{MS}}}(M_Z^2)$ — normally just set to agree with a reasonable global average value such as that given by the PDG [35] — with different effective values for different branching types obtained via user-specifiable renormalization-scale prefactors,

$$
\alpha_s^{\overline{\text{MS}}}(M_Z^2) \rightarrow
\begin{cases}
\alpha_s(k_{\text{E}}^{\text{F}} p_{\perp j}^2 + \mu_0^2), & \text{for} \quad \text{FF and RF gluon emissions}, \\
\alpha_s(k_{\text{S}}^{\text{F}} p_{\perp j}^2 + \mu_0^2), & \text{for} \quad \text{final-state gluon splittings}, \\
\alpha_s(k_{\text{E}}^{\text{I}} p_{\perp j}^2 + \mu_0^2), & \text{for} \quad \text{II and IF gluon emissions}, \\
\alpha_s(k_{\text{S}}^{\text{I}} p_{\perp j}^2 + \mu_0^2), & \text{for} \quad \text{initial-state gluon splittings}, \\
\alpha_s(k_{\text{C}}^{\text{I}} p_{\perp j}^2 + \mu_0^2), & \text{for} \quad \text{initial-state gluon conversions},
\end{cases}
\tag{187}
$$

where the scheme can be either $\overline{\text{MS}}$ or CMW and $\mu_0 \sim \mathcal{O}(\Lambda_{\text{QCD}})$ is a fixed scale that forces the effective coupling to asymptote to $\alpha_s(\mu_0^2)$ for $p_{\perp j} \rightarrow 0$. A maximum value can also be specified beyond which α_s is not allowed to grow, effectively freezing the coupling at that value in the infrared.

Evolution equations The differential branching probability as implemented by the sector shower is given as the sum of individual $IK \mapsto ijk$ antenna branching probabilities,

$$
\frac{\mathrm{d}\mathcal{P}}{\mathrm{d}p_{\perp j}^2} = \sum_j \frac{\mathrm{d}\mathcal{P}_{j/IK}}{\mathrm{d}p_{\perp j}^2},
\tag{188}
$$

which can be written in terms of the shower evolution variable $p_{\perp j}^2$ and an arbitrary complementary phase-space variable ζ as

$$\frac{\mathrm{d}\mathcal{P}_{j/IK}}{\mathrm{d}p_{\perp j}^2} = \frac{\alpha_s(p_{\perp j}^2)}{4\pi} \mathcal{C}_{j/IK} \int\limits_{\zeta_{\min}(p_{\perp j}^2)}^{\zeta_{\max}(p_{\perp j}^2)} \int\limits_0^{2\pi} \bar{A}_{j/IK}^{\mathrm{sct}}(p_{\perp j}^2, \zeta) R_{\mathrm{PDF}} F_\Phi \left| J(p_{\perp j}^2, \zeta) \right| \Theta^{\mathrm{sct}}(p_{\perp j}^2, \zeta, \phi) \frac{\mathrm{d}\phi}{2\pi} \mathrm{d}\zeta. \tag{189}$$

Here, the Jacobian $J(p_{\perp j}^2, \zeta)$ accounts for the change to the shower variables $(y_{ij}, y_{jk}) \mapsto (p_{\perp j}^2, \zeta)$, for which different choices are implemented in VINCIA, depending on the branching type, *cf.* [100, section 2.5]. Note that, since the starting point is an exact phase-space factorization and the Jacobian factor $J(p_{\perp j}^2, \zeta)$ accounts for the mapping to shower variables, there is no physical dependence on the choice of ζ in VINCIA; it only affects how simple or complicated the trial integrals become, and the efficiency with which trial branchings can be generated. The phase-space factor

$$F_\Phi = \begin{cases} s_{IK} \dfrac{s_{IK}}{\sqrt{\lambda(s_{IK}, m_i^2, m_k^2)}} & \text{FF}, \\[3ex] \dfrac{s_{AK} + m_j^2 + m_k^2 - m_K^2}{(1 - y_{jk})^3} \dfrac{s_{AK} + m_j^2 + m_k^2 - m_K^2}{\sqrt{\lambda(m_A^2, m_{AK}^2, m_K^2)}} & \text{RF}, \\[3ex] \dfrac{s_{AK}}{1 - y_{jk}} & \text{IF}, \\[3ex] \dfrac{s_{AB}}{1 - y_{aj} - y_{jb}} & \text{II}, \end{cases} \tag{190}$$

accounts for the relative size of the post-branching phase space to the pre-branching phase space. For ISR, a PDF ratio is included for every initial-state parton,

$$R_{\mathrm{PDF}} = \begin{cases} 1 & \text{FF \& RF}, \\[3ex] \dfrac{f_a(x_a, p_{\perp j}^2)}{f_A(x_A, p_{\perp j}^2)} & \text{IF}, \\[3ex] \dfrac{f_a(x_a, p_{\perp j}^2)}{f_A(x_A, p_{\perp j}^2)} \dfrac{f_b(x_b, p_{\perp j}^2)}{f_B(x_B, p_{\perp j}^2)} & \text{II}. \end{cases} \tag{191}$$

Two things should be noted in eq. (189). First, the colour factor $\mathcal{C}_{j/IK}$ is normalized such that the integral prefactor is always $1/4\pi$ (as opposed to $1/2\pi$), with the specific VINCIA choices being

$$\mathcal{C}_{g/q\bar{q}} = 2C_F = \frac{8}{3}, \tag{192}$$

$$\mathcal{C}_{g/qg} = \frac{1}{2}(2C_F + C_A) = \frac{17}{6}, \tag{193}$$

$$\mathcal{C}_{g/gg} = C_A = 3, \tag{194}$$

$$\mathcal{C}_{q/Xg} = 2T_R = 1, \tag{195}$$

where an interpolation between $2C_F$ and C_A,

$$\mathcal{C}_{g/qg} = \frac{(1 - y_{ij})2C_F + (1 - y_{jk})C_A}{2 - y_{ij} - y_{jk}}, \tag{196}$$

is available for qg antennae. Second, the azimuthal integration is made explicit although the antenna functions have no azimuthal dependence. This is to emphasize a potentially non-trivial azimuthal dependence of the sector veto Θ^{sct}.

Matching, merging, and matrix-element corrections   A unique property of VINCIA's sector-based approach to parton showers is that there is only a very small number of "shower histories" leading to each distinct parton configuration. For gluon emissions, VINCIA's sector shower is entirely bijective, *i.e.* there is only a single unique shower history leading from the Born to any given Born+n-gluon parton configuration. For $g \to q\bar{q}$ splittings, one has to sum over all possible same-flavour quark-antiquark pairings, but the number of contributing histories for a given phase-space point is still drastically reduced relative to conventional, non-sectorized, showers. We say that the sector shower is "maximally bijective", and this provides an optimal framework for high-multiplicity matching and merging, discussed further in section 5.4.

Infrared cutoffs   For $p_\perp$ scales below 1 GeV or so, perturbative approximations become increasingly inaccurate as $\alpha_s(p_\perp)$ shoots towards divergence at $\Lambda_{\text{QCD}} \sim 200 - 300$ MeV. Like for the simple shower model, VINCIA's perturbative shower evolution is therefore also halted some distance above Λ_{QCD}, at which point the parton system is handed to PYTHIA's string-fragmentation model for hadronization. In VINCIA, the precise scale at which the shower is stopped can be set independently for FF, IF, and II antennae.

The shower cutoff for FF antennae in VINCIA is analogous to the FSR cutoff in the simple-shower model. It can be regarded as the effective factorization scale between the perturbative and non-perturbative parts of the overall fragmentation description. It therefore has the interpretation as the scale at which the parameters of the non-perturbative hadronization modelling are defined. Ideally, the hadronization parameters should "run" with the shower cutoff, but since the relevant running equations are not known, in practice the hadronization parameters simply have to be retuned for each new value of the cutoff. In other words, the FF-cutoff value can be considered part of the fragmentation tuning. In general, one would seek not to leave too much of a gap between the lowest $p_\perp$ scales generated by shower branchings (down to the cutoff) and the highest $p_\perp$ scales generated by string breaks (with typical size set by the fragmentation $p_\perp$ width, *cf.* section 7.1.1).

For II antennae, the cutoff can be regarded as an effective colour-screening resolution scale, or a lowest scale for which partons inside hadrons can be said to be well represented by plane waves. This could possibly be tied to the physics of parton saturation, though no explicit such connection is made here. The practical considerations are similar as for the ISR shower cutoff in the simple-shower model, striking a balance between $p_\perp$ kicks generated by the shower and contributions from so-called "primordial $k_\perp$", *cf.* section 6.3.3.

For IF antennae, the fact that VINCIA's default recoil strategy is fully local, and does not impart $p_\perp$ recoil to any partons outside of the $2 \to 3$ branching itself, leads to some pathologies. In particular, each IF branching *dilutes* the primordial $k_\perp$, and does not add any perturbative $p_\perp$ of its own, to the hard system. The non-smooth interplay between the II and IF recoil strategies can make it challenging to describe the soft peak of experimental signals such as the Drell–Yan $p_\perp$ spectrum, and can produce seemingly counter-intuitive scaling with the value of the IF cutoff.

4.2.3   QED showers

The VINCIA shower offers a number of options for the inclusion of electromagnetic and weak corrections. They all share common features, like the phase-space factorizations and ordering scale, with the QCD shower, *cf.* section 4.2.1. In this section, we describe the first (and default) option, which is a pure QED shower that incorporates a fully coherent multipole treatment of the simulation of photon radiation off systems of charged fermions, vectors, and scalar particles, as well as photon splittings to pairs of charged fermions [101, 102]. We also include a simpler and somewhat faster alternative, in which the full multipole sum is replaced by

individual dipole terms according to a principle of maximal screening, analogous to how QED is handled in the simple-shower model.

The basic building block for VINCIA's treatment of photon radiation is the photon-emission antenna function for a single pair of final-state charged particles i and k,

$$
\begin{aligned}
A^{\text{sct}}_{\gamma/IK}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2, \mu_k^2) = 2g^2 \frac{Q_I Q_K}{s_{IK}} \Bigg[& 2\frac{y_{ik}}{y_{ij} y_{jk}} - 2\frac{\mu_i^2}{y_{ij}^2} - 2\frac{\mu_k^2}{y_{jk}^2} + \delta_{If}\frac{y_{ij}}{y_{jk}} + \delta_{Kf}\frac{y_{ij}}{y_{jk}} \\
& + \delta_{IW}\frac{4}{3} y_{ij}\left(\frac{y_{jk}}{y_{IK} - y_{jk}} + \frac{y_{jk}(y_{IK} - y_{jk})}{y_{IK}^2} \right) \\
& + \delta_{KW}\frac{4}{3} y_{jk}\left(\frac{y_{ij}}{y_{IK} - y_{ij}} + \frac{y_{ij}(y_{IK} - y_{ij})}{y_{IK}^2} \right) \Bigg], \quad (197)
\end{aligned}
$$

where the Kronecker deltas ensure the correct collinear terms are incorporated in the cases where I and K are fermions or W bosons. The factors Q_I and Q_K represent the relative electromagnetic charges of I and K, respectively. The II and IF antennae may be found by crossing symmetry following eqs. (185) and (186). For notational convenience we define

$$
A^{\text{sct}}_{\gamma/IK}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2, \mu_k^2) = g^2 Q_I Q_K \bar{A}^{\text{sct}}_{\gamma/IK}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2, \mu_k^2), \quad (198)
$$

similar to the QCD equivalent eq. (174).

While possibly counter intuitive, the definition of a coherent QED shower using eq. (198) is not as straightforward as for its (leading-colour) QCD shower counterpart. The reason is the absence of an equivalent of the leading-colour approximation, which in QCD allows one to discard the majority of the soft eikonal contributions that are subleading in colour. Conversely, in QED no eikonal is subleading to any other, and full coherence can only be accomplished by the inclusion of all of them simultaneously. VINCIA's most sophisticated photon emission algorithm accomplishes this by the definition of a single branching kernel

$$
\bar{A}^{\text{sct}}_{\gamma/\text{coh}} = \sum_{\{IK\}} \sigma_I Q_I \sigma_K Q_K \bar{A}^{\text{sct}}_{\gamma/IK}(s_{IK}; y_{ij}, y_{jk}, \mu_i^2, \mu_k^2), \quad (199)
$$

where $\{IK\}$ runs over all pairs of charged particles, and σ_I and σ_K are sign factors that have $\sigma_I = 1$ for final-state particles and $\sigma_I = -1$ for initial-state particles. This branching kernel includes all soft multipole terms, as well as the correct collinear limits [101], but its singular structure is highly complex. The coherent algorithm is able to sample it by sectorizing the phase space according to

$$
\frac{\mathrm{d}\mathcal{P}_{j,\text{coh}}}{\mathrm{d}p_{\perp j}^2} = \sum_{\{ik\}} \int_{\zeta_{\min}(p_{\perp j}^2)}^{\zeta_{\max}(p_{\perp j}^2)} \int_0^{2\pi} \frac{\alpha_{\text{em}}(p_{\perp j}^2)}{4\pi} \bar{A}^{\text{sct}}_{\gamma/\text{coh}}(p_\perp^2, \zeta)
$$

$$
\times R_{\text{PDF}} F_\Phi \left| J(p_{\perp j}^2, \zeta) \right| \Theta^{\text{sct}}_{ik}(p_{\perp j}^2, \zeta, \phi) \frac{\mathrm{d}\phi}{2\pi} \mathrm{d}\zeta, \quad (200)
$$

where $\Theta^{\text{sct}}_{ik}(p_{\perp j}^2, \zeta, \phi)$ is given by eq. (172), but with a sum over charged-particle pairs, and the sector resolution is the same as that of a gluon emission as given by eq. (173). This procedure ensures the soft and collinear singularities are correctly regularized by the transverse momenta of the photon with respect to all pairs of charged particles. The coherent emission algorithm is the default choice, but in some specific high-multiplicity cases it may be slow due to the large number of sectors that need to be sampled.

As a backup, a faster, unsectorized alternative is implemented that rephrases the photon emission probability as

$$\frac{d\mathcal{P}_{\text{pair}}}{dp_{\perp j}^2} = \sum_{[IK]} \int_{\zeta_{\min}(p_{\perp j}^2)}^{\zeta_{\max}(p_{\perp j}^2)} \int_0^{2\pi} \frac{\alpha_{\text{em}}(p_{\perp j}^2)}{4\pi} Q_{[IK]}^2 \bar{A}_{\gamma/IK}^{\text{sct}}(p_{\perp j}^2, \zeta) R_{\text{PDF}} F_\Phi \left| J(p_{\perp j}^2, \zeta) \right| \frac{d\phi}{2\pi} d\zeta, \qquad (201)$$

where $[IK]$ now runs over all pairings of charged particles with identical but opposite charge $Q_{[IK]}$. The factors σ_I and σ_K have been absorbed into the definition of $Q_{[IK]}$, meaning that a final-state charged particle may be paired with a same-sign initial-state particle. That is, every charged particle now only appears once, and pairings are constructed to minimize the sum of dipole-antenna invariant masses as per the principle of maximum screening [102]. The task of pairing particles under such a constraint in $\mathcal{O}(n^3)$ time complexity is accomplished using the Hungarian algorithm [127–129]. While this algorithm is generally faster, it only approximates the complete multipole structure. Furthermore, it may not always be possible to pair up all charges. For instance, in a $W^+ \to u\bar{d}$ decay, no pairings are possible at all. In such cases, as many charges as possible are paired up, and the fully coherent algorithm is used for the remainder.

The QED shower also includes photon splittings to charged fermions, which use antennae that are kinematically identical to their gluon-splitting counterparts. Furthermore, while photon radiation off quarks is cut off at a scale of order the hadronization scale, leptonic photon radiation continues to much lower scales and has its own cutoff scale. Since the system of leptons is not necessarily charge conserving by itself, which is a requirement for the above algorithms, the pool of charges is supplemented with the colour-neutral strings that enter the hadronization stage. When acting as the recoiler of a lepton, the antenna function is replaced by a dipole function that only contains the singular limits relevant to the lepton.

QED radiation off charged hadrons and/or in hadron decays, is not present in the current implementation but may be included in future work; see the program's `online manual` for updates.

4.2.4 EW showers

As an alternative to the coherent QED shower described above, VINCIA also offers the option to interleave the QCD shower with a full-fledged EW shower, in which all possible branchings from the EW sector are incorporated, albeit only in a collinear approximation without any attempt at incorporating soft-interference effects [23,105]. For each given application, one must therefore choose whether weak-shower corrections are more important than QED coherence effects for the study at hand, the default choice being the coherent-QED one.

When enabled, VINCIA's EW option includes not only the branchings that are also available in the simple shower (heavy vector-boson emissions off initial- and final-state fermions) but also final-state triple vector-boson branchings, Higgs emissions, and decay-like splittings. Like the QED module, the EW one also shares the common features of the QCD shower, allowing for a sensible interleaving of the two. However, it is important to be aware that the EW shower relies on the helicity-dependent evolution described in section 4.2.1, which must therefore also be enabled. The resulting intermediate states of definite helicity are vital in the EW sector due to its chiral nature. Helicity-dependent antenna functions are present for all EW branchings, capturing their associated quasi-collinear limits. Due to the rich physics landscape of the EW sector of the SM and the many different helicity combinations, there are hundreds of distinct polarized collinear-splitting kernels. The antenna functions are therefore not included here, but they may be found in [23]. Note that, since the EW shower does not incorporate soft-

interference effects, the antenna functions are more like dipole functions, only including the single quasi-collinear limit of the branching particle, while the other just functions as a recoiler.

A number of features unique to the EW sector are incorporated. For example, in a shower sequence like $e^- \rightarrow e^- Z/\gamma \rightarrow e^- W^+ W^-$, the interference between the Z and the γ can be of $\mathcal{O}(1)$ [105, 130]. A full treatment of this effect may for instance be accomplished by the evolution of density matrices, which can quickly become prohibitively expensive. VINCIA instead implements a simplified approach, in which the emission probability is corrected at first order by an event weight. This weight is computed using quasi-collinear amplitude-level branching amplitudes using the spinor-helicity formalism.

These same amplitudes are also used to determine recoilers for the quasi-collinear branchings of the EW shower. Unlike in the QCD sector, where recoilers are typically chosen to be a colour-connected parton, no such mechanism is available in the EW sector. Furthermore, because the EW shower only models quasi-collinear branchings without soft-interference effects, the choice of recoiler is formally arbitrary. One can however select the recoiler probabilistically such that the kinematic effects of recoil on previous branchings is minimized [105].

Another peculiar feature of the EW sector is the fact that branchings like $t \rightarrow bW^+$ and $Z \rightarrow q\bar{q}$ appear both as shower branchings and as resonance decays. For off-shellness scales $Q^2 = m^2 - m_0^2 \sim \Gamma^2$, the physics is best described by a Breit–Wigner distribution, while for scales above the electroweak scale Q_{EW}^2, the EW shower is most accurate. In the intermediate region a matching procedure is required. When the EW shower produces a heavy resonance like one of the EW gauge bosons, a top quark, or a Higgs boson, its mass is sampled from a helicity-dependent Breit–Wigner distribution (see [23] for details)

$$\mathrm{BW}(Q^2) \propto \frac{m_0 \Gamma(m)}{Q^4 + m_0^2 \Gamma(m)^2}. \tag{202}$$

This procedure mirrors the treatment of resonances that are part of the hard process as described in section 2.3.3, which can also be branched by the EW shower. The shower is matched to the Breit–Wigner distribution by applying a suppression factor $Q^4/(Q^2 + Q_{EW}^2)^2$, and the resonance is decayed when the evolution scale reaches the sampled resonance off-shellness without generating an EW branching. In that case, if the EW shower produced the resonance, the decay is distributed according to the appropriate helicity-dependent $1 \rightarrow 2$ matrix element. If instead the resonance was part of the hard process, the decay has already been generated and is inserted.

Finally, double counting issues appear with the inclusion of EW branchings in the shower. For instance, the state $pp \rightarrow VVj$ may be reached by starting from $pp \rightarrow VV$ and performing an initial-state QCD emission, or from $pp \rightarrow Vj$ and performing an EW emission. To avoid double counting such phase-space points, VINCIA implements an overlap veto procedure that can be used when overlapping matrix elements are enabled. It is based on the k_T jet algorithm [131] distance measures, generalized to account for the massive states that appear in the EW sector, given by

$$d_{iB} = k_{T,i}^2,$$
$$d_{ij} = \min(k_{T,i}^2, k_{T,j}^2) \frac{\Delta_{ij}}{R} + |m_i^2 + m_j^2 - m_I^2|. \tag{203}$$

The distance between the beam and final-state particle i is measured by d_{iB}, while d_{ij} measures the distance between two final-state particles i and j. If, for example, a gluon is emitted by the QCD shower, the distances with respect to its colour-connected partons are computed. Furthermore, the distances of all possible $2 \rightarrow 1$ EW clusterings of the state after the gluon emission are also evaluated. If one of these distances is smaller than the QCD ones, then the current phase-space point should be populated by the EW shower rather than the QCD shower,

and the gluon emission is vetoed. This procedure ensures no double counting occurs and the QCD and EW showers populate the regions of phase space they are most accurate in.

4.3 The DIRE shower

The DIRE parton shower, introduced in [117], offers another alternative showering model. It aims to combine aspects traditionally associated with $2 \to 3$ dipole (antenna) showers with features of "conventional" $1 \to 2$ parton showers. The goal of this hybrid is to inherit the modelling of soft-emission effects from dipole showers, while keeping an explicit association of splittings with specific collinear directions. This should, in principle, allow for an uncomplicated comparison to the ingredients in QCD factorization theorems. The physics aspects of DIRE have been developed in a series of articles [117, 132–137], and we refer the reader to these publications for details. Below, we will summarize the most important choices, virtues and current limitations.

4.3.1 Phase-space coverage and ordering

DIRE employs an exact factorization of the single- and double-emission phase spaces. The single-emission phase space is adapted from [111, 138], and allows for any combination of masses in $2 \to 3$ branchings. The construction of post-emission momenta through the mapping $M^{(1)}$ can be sketched by

$$M^{(1)}\left(\begin{array}{c} \text{Radiator } \tilde{i1} \\ \\ \text{Recoiler } \tilde{k} \end{array} \oplus\ t^{(1)}, z^{(1)}, \phi^{(1)} \right) = \begin{array}{c} \text{Radiator } i \\ \\ \text{Emission 1} \quad \text{Recoiler } k \end{array}, \qquad (204)$$

where $t^{(1)}$ is the evolution variable, $z^{(1)}$ a momentum-sharing variable, and $\phi^{(1)}$ an azimuthal angle. Note that under the mapping $M^{(1)}$, the direction of the recoiler is not affected by the branching. Only its longitudinal momentum components change. This deliberate *choice* ensures that the collinear direction defined by the recoiler, and consequently its mapping onto factorization theorems, remains intact. A caveat to this approach — related to initial-state emissions — is discussed below.

The double-emission phase space — relevant for NLO parton evolution [132] — may similarly be illustrated by a map $M^{(2)}$

$$M^{(2)}\left(\begin{array}{c} \text{Radiator } \tilde{i12} \\ \\ \text{Recoiler } \tilde{k} \end{array} \oplus\ t^{(12)}, z^{(12)}, \phi^{(12)}, s_{12}, x, \phi' \right) = \begin{array}{c} \text{Radiator } i \\ \text{Emission 1} \\ \text{Emission 2} \quad \text{Recoiler } k \end{array}. \qquad (205)$$

Here, $t^{(12)}$ is the evolution variable assigned to the emission of the system (12), $z^{(12)}$ ($\phi^{(12)}$) a momentum-sharing (azimuthal angle) variable, and s_{12} the virtuality of the system (12), while x and ϕ' are related to the momentum sharing (azimuthal angle) between emissions 1 and 2. Again, the direction of the "recoiler" is preserved.

The momentum mappings $M^{(1)}$ and $M^{(2)}$ are (re)arranged to ensure that the phase-space coverage is fully symmetric between radiation from the "radiator" or the "recoiler", *i.e.* given a fixed post-branching phase-space point and fixed branching variables, an identical pre-branching phase-space point is produced, independent of assigning the emissions to the radiator or recoiler. However, it should be noted that the momentum-sharing variables z are *not symmetric* under exchange of the "emission" for one of the other involved particles, since the

limit $z \to 1$ is associated with a soft emission. DIRE separates the generation of post-branching momenta into four distinct cases:

FF, *i.e.* emission from a final-state particle, using a final-state recoiler:
This case has the fewest kinematic constraints, but the richest set of combinations of possible masses. Thus, the mapping is constructed to ensure that regions of phase space in which mass-corrected transition rates would lead to negative contributions are outside the physical phase-space boundaries.

FI, *i.e.* emission from a final-state particle, using an initial-state recoiler:
This case again has few kinematic constraints, after the choice of keeping the recoiler direction intact. DIRE will, if not instructed otherwise, treat incoming particles as massless for the purpose of phase-space generation. This means that in this configuration, negative transition rates due to mass corrections may occur. This is handled by a weighted shower algorithm.

IF, *i.e.* emission from an initial-state particle, using a final-state recoiler:
This case has several kinematic constraints that need to be considered. In fact, the system is over-constrained if both the initial-state particle and the final-state recoiler should retain their directions. In this case, the transverse momentum generated in the branching can be balanced by extending the set of particles that may change their momentum. DIRE offers the possibility to employ a "global"-recoil strategy, in which the transverse momentum of the splitting is balanced by all final-state particles within the decaying system[2]. It is also possible to instruct DIRE to relax the condition that the final-state recoiler retains its direction. In this "local" strategy, the system of particles that change their momentum does not need to be extended.

II, *i.e.* emission from an initial-state particle, using an initial-state recoiler:
This case also has several kinematic constraints, and is over constrained, since both initial-state particles should retain their directions. Here, no attempt is made to construct a "local"-recoil strategy. Instead, the transverse momentum of the branching is collectively balanced by all final-state particles within the decaying system.

In general, the construction of post-branching momenta is subject to many choices. The choices above have foremost been guided by providing a simple procedure and Jacobian factors, such that analytic integrations of the emission patterns are as straight-forward as possible. This helps when improving the evolution with next-to-leading order corrections.

The evolution variables in DIRE are chosen to lead to a symmetric phase-space sampling and simple phase-space boundaries. Soft transverse momenta fulfil these criteria, if defined by

$$t^{(a)} \propto \frac{(p_i p_a)(p_k p_a)}{Q^2} \propto p_a^+ p_a^- \, , \tag{206}$$

where Q^2 is a maximal scale, p_a may be a sum of one or two emission momenta, and p_i and p_k are the radiator and recoiler post-branching momenta. Thus, in all of the four cases above, and for both single-emission and double-emission contributions, DIRE employs soft transverse momentum as ordering variable, see [117, 132] for details.

4.3.2 Transition rates

The DIRE parton shower aims to model configurations containing soft particles or collinear configurations with high fidelity. As in a traditional parton shower, separate transition rates

---

[2]Here, "decaying system" refers to the particle content of a single $2 \to n$ scattering, in case several such scatterings exist due to the inclusion of multiparton interactions.

are used in each collinear direction. It might be helpful to explain this choice with an example. Imagine a dipole stretched between a quark and a gluon. The primary contributions to radiation collinear to the quark should be proportional to the colour factor C_{F}, while the radiation pattern collinear to the gluon should, up to small corrections, be proportional to C_{A}. Similarly, higher-order corrections to the radiation pattern in either region differ.

However, simultaneously radiating from both dipole "ends" with the full rate expected in the collinear limit (given by the DGLAP splitting functions) will naively lead to an incorrect pattern in the soft limit. This problem is circumvented by replacing collinear-soft parts of DGLAP kernels by an improved description. The latter may be obtained by distributing the correct soft radiation pattern among all coherently radiating particles,

$$\frac{(p_i p_k)}{(p_i p_a)(p_k p_a)} = \underbrace{\frac{1}{(p_i p_a)}}_{\text{combine with Jacobian}} \quad \underbrace{\frac{(p_i p_k)}{(p_i p_a) + (p_k p_a)}}_{\text{rewrite in } t \text{ and } z} \quad + (i \leftrightarrow k)$$

$$\frac{1}{t} \qquad \underbrace{\frac{2(1-z)}{(1-z)^2 + t/Q^2}}_{\text{collinear limit } t \to 0} \longrightarrow \frac{2}{1-z} \,. \tag{207}$$

The resulting soft-collinear pattern is supplemented with hard-collinear terms [87]. As an extension of [87], the $1/z$-terms present in DGLAP kernels are also shifted $\frac{1}{z} \to \frac{z}{z^2 + t/Q^2}$ to ensure that sum rules for the splitting kernels are maintained. Finally, mass-dependent corrections based on [111] are added. The exact splitting kernels used in DIRE are listed in [117]. The above chain of reasoning is used for all branching types in DIRE.

QCD The above reasoning directly applies to QCD branchings at leading order, *i.e.* when increasing the multiplicity by one particle, and while not including explicit virtual corrections to single-parton emission. At leading order:

- Dipoles are formed from radiator-recoiler pairs connected by a colour flow in the $N_{\mathrm{c}} \to \infty$ limit. At the point of compiling this manual, the fixed-colour corrections discussed in [137] have not been included in the core PYTHIA code.

- Colour factors due to colour-charge correlators in the $N_{\mathrm{c}} \to \infty$ limit are given by:

    1. gluon-radiation off (anti)quarks $\propto C_{\mathrm{F}}$,
    2. gluon-radiation off gluons $\propto C_{\mathrm{A}}/\#(\text{possible recoilers}) = C_{\mathrm{A}}/2$,
    3. and gluon branching to quark pairs $\propto T_{\mathrm{R}}$.

- Coupling-factors α_s for all QCD splittings are evaluated with dynamic arguments, with the preferred scheme being $\alpha_s(t)$. However, it should be noted that the emergence of the running coupling is driven by soft-gluon emissions, and thus, it is *a priori* not obvious if the evaluation $\alpha_s(t)$ extends also to hard-collinear configurations. Thus, the user may instruct PYTHIA to use different arguments to evaluate α_s: the running coupling may be evaluated using the "collinear transverse momenta" $\mathbf{k}_\perp^2$ defined as evolution variables in [114], *i.e.* $\alpha_s(\mathbf{k}_\perp^2)$, or it may be evaluated using the strict definition of the (inverse) eikonal term, *i.e.* $\alpha_s\left(\frac{(p_i p_a)(p_k p_a)}{(p_i p_k)}\right)$.

The usage of a running coupling effectively includes "universal" virtual corrections to the emission rates. For inclusive soft-gluon emission, it is possible to include further next-to-leading

order corrections rescaling the soft-gluon emission rate:

$$\frac{\alpha_s}{2\pi}\left[\frac{2(1-z)}{(1-z)^2+t/Q^2}+(\text{hard-collinear terms})\right]$$

$$\to \frac{\alpha_s}{2\pi}\left(1+K\frac{\alpha_s}{2\pi}\right)\frac{2(1-z)}{(1-z)^2+t/Q^2}+\frac{\alpha_s}{2\pi}\cdot(\text{hard-collinear terms})\,. \tag{208}$$

This may be considered as a conservative implementation of the conventional CMW (or MC) scheme [85]. Note that different strategies for the evaluation of running couplings will induce *different* higher-order corrections. Without better modelling, none of these *ad hoc* choices are completely satisfactory.

Higher-order corrections to QCD evolution have been known for a long time. DIRE implements several aspects of QCD evolution at next-to-leading order:

- Inclusive branching rates can be augmented with hard-collinear corrections by employing NLO DGLAP splitting functions. These improvements are available for both initial-state and final-state branchings. The benefit of such corrections is mainly in a more consistent treatment of PDF evolution in backwards initial-state evolution, since the latter relies on the parton shower distributing emissions according to the rates used to evolve externally pre-tabulated PDFs from low to high scales.

- Correlated triple-collinear emissions, *i.e.* branchings of the form $1\oplus 1\to 3\oplus 1$, have been included to yield NLO DGLAP (initial- or final-state) evolution in the collinear limit from fully differential double-emission matrix elements.

- Correlated double-soft emissions and explicit real-virtual corrections, *i.e.* branchings of the form $2\to 4$ and $2\to 3$ at 1-loop, can be employed for final-state branchings. The inclusion of such NLO corrections is mainly a reduction of the renormalization-scale uncertainty of the parton shower. At the point of writing this manual, the consistent combination of triple-collinear and double-soft NLO corrections outlined in [139] has not been included in a public PYTHIA release.

QED The description of QED in DIRE [135, 137] follows a very similar structure to that of QCD branchings, and is inspired by [140].

- Dipoles are formed from all electrically charged radiator-recoiler pairs, much like the fixed-colour QCD dipole assignments discussed in [69, 137].

- Charge-factors due to electric-charge correlators are determined from

$$Q^2=-\frac{\eta_{\tilde{i}\tilde{1}}\eta_{\tilde{k}}Q_{\tilde{i}\tilde{1}}Q_{\tilde{k}}}{Q_{\tilde{i}\tilde{1}}^2}\ (\text{photon emission})\,,$$

$$Q^2=\frac{1}{\#\text{recoilers}}\quad (\text{photon splitting})\,, \tag{209}$$

where $Q_{\tilde{i}\tilde{1}}$ and $Q_{\tilde{k}}$ are the charges of the radiator and recoiler, respectively, and $\eta_i=+1(-1)$ if i is a final- or initial-state particle. These correlators multiply the splitting functions in place of the QCD colour factors, and may readily lead to negative contributions to the transition rates. Thus, a weighted shower algorithm is crucial for the QED modelling in DIRE.

- Coupling-factors α_{em} for all QED splittings are evaluated in the Thompson limit, *i.e.* no running QED coupling is employed in the shower.

Kinetically mixed dark photons   DIRE further implements kinetically mixed dark photon interactions, featuring dark photon emission from and decay into standard-model particles. These transitions are handled analogously to QED interactions, except that the dark photon may be massive. The decay width of the dark photon is currently ignored for both dark photon emission and decay.

Electroweak effects   Finally, DIRE allows for electroweak-boson radiation and fermionic weak-boson decay, using a simplistic model similar to the ideas of [83, 141]. Electroweak effects are mainly included because of the necessity for consistent matrix-element merging at LHC energies: to avoid an overly QCD-evolution biased scale setting for vector-boson plus jets configurations that exhibit giant K-factors [142], the inclusion of parton-shower histories containing electroweak clusterings are mandatory for some showers [143, 144]. Thus, DIRE implements electroweak evolution using:

- Transition rates are determined from partial-fractioned massive dipole kernels [111].

- Dipoles are formed from all pairs of particles that may emit the same electroweak vector bosons. Electroweak-boson decays employ the same recoiler selection as vector-boson radiation, much like the QED case [135, 137].

- Coupling factors are calculated under the assumption of chirality-summed evolution, *cf.* [141]. The coupling value is kept fixed, *i.e.* no running coupling is employed.

This overly simplified model may appropriately handle electroweak history effects in the context of multi-jet merging, especially for $W^{\pm}$-boson plus multi-jet configurations at the LHC [136]. Beyond this, studies of weak-boson radiation with DIRE are discouraged.

4.3.3   Weight handling aspects

The transition rates outlined in section 4.3.2 may be relatively complex. This results in some technical requirements that need to be met to produce a sound simulation since:

- it may not be possible to find efficient overestimates of complex transition kernels, such as for correlated double emission;

- and it may not be possible to guarantee positivity, *e.g.* due to mass effects, electric charge correlators, or higher-order corrections.

Both of these points (as well as the automated renormalization scale variations available in DIRE) may be addressed with the help of a weighted veto algorithm, which was discussed in section 2.2.3. DIRE employs this method more heavily than the rest of PYTHIA. Thus, the relevant features and extensions beyond the literature will be discussed here.

The core realization of weighted shower algorithms is that acceptance rates in the veto algorithm may be factored into a contribution that is "unweighted" via the veto algorithm, and an event-by-event "weight" factor that encapsulates the effect of sign changes or under-estimations of the transition rate. To preserve inclusive cross sections, this is naturally complemented by event-by-event weights that augment the rejection rate of the parton shower. Once acceptance and rejection weights have been introduced, it also becomes possible to only partially unweight the use of overestimates through the veto algorithm, and correct for the partial unweighting by amending the event weight. This allows for an enhancement of certain transitions beyond their natural rate, leading to an improved statistical error, at the expense of a larger weight variance. Finally, the veto algorithm may be implemented in a series of distinct

accept-reject steps. Each such step can be upgraded to incorporate complex rates[3]. At present, DIRE employs a weighted shower when choosing a branching according to splitting kernels, and another, stacked weighted shower to incorporate matrix-element corrections. A weighted approach to the latter is necessary as matrix-element corrections may induce sign changes, or because matrix elements are underestimated by the (sum of all possible) splitting kernels.

The weighted shower algorithm in DIRE rests on the realization that the rate of producing one transition in the shower after n rejections (through the veto algorithm) is given by

$$\mathcal{P}(t) = g(t) \frac{f(t)}{g(t)} \exp\left(-\int_t^{t_n} d\bar{t} g(\bar{t})\right) \prod_{i=1}^{n} \left[\frac{g(t_i) - f(t_i)}{g(t_i)}\right] g(t_i) \exp\left(-\int_{t_i}^{t_0} d\bar{t} g(\bar{t})\right). \quad (210)$$

This equation is the formal requirement for the validity of the veto algorithm, and does not strictly constrain the transition rates $f(t)$ by the "overestimate" $g(t)$ through $0 < f(t) < g(t)$. The numerical implementation of the equation does, however, require sensible acceptance *probabilities*. This may be achieved by introducing an auxiliary function $h(t)$ that guarantees acceptance probabilities $0 < f(t)/h(t) < 1$, see *e.g.* also the description at the end of section 2.2.3. Rewriting in terms of this auxiliary function leads to

$$\mathcal{P}(t) = \left[g(t) \frac{f(t)}{h(t)} \exp\left(-\int_t^{t_n} d\bar{t} g(\bar{t})\right) \prod_{i=1}^{n} \left[1 - \frac{f(t_i)}{h(t_i)}\right] g(t_i) \exp\left(-\int_{t_i}^{t_0} d\bar{t} g(\bar{t})\right)\right]$$
$$\otimes \frac{h(t)}{g(t)} \prod_{i=1}^{n} \frac{h(t_i)}{g(t_i)} \frac{g(t_i) - f(t_i)}{h(t_i) - f(t_i)}. \quad (211)$$

The second line may be interpreted as a corrective factor due to disconnecting the sampling and rejection distributions. It does not have to be bounded, and is implemented as an event weight. It is important to note that the acceptance rates $f(t)/h(t)$ only need to be bounded point-wise in t, *i.e.* they may be adjusted depending on the value of $f(t)$. In particular, DIRE uses

$$h(t) = \begin{cases} \text{sign}[f(t)]g(t), & \text{if } g(t) > |f(t)|, \\ kf(t), & \text{if } g(t) < |f(t)| \quad (\text{with } k \sim 1.1). \end{cases} \quad (212)$$

An artificially enhanced sampling may be achieved by shifting $g \to g' = Cg$, while keeping all rejection steps (*i.e.* the definition of $h(t)$) fixed to their original values. The compensating event weight will then be shifted to

$$\frac{1}{C} \frac{h(t)}{g(t)} \prod_{i=1}^{n} \frac{h(t_i)}{g(t_i)} \frac{g(t_i) - f(t_i)/C}{h(t_i) - f(t_i)}. \quad (213)$$

Once the weighted shower is in place, parton-shower variations may be included by keeping track of multiple weights of the form

$$w^{[k]} = \frac{1}{C} \frac{h(t)}{g(t)} \prod_{i=1}^{n} \frac{h(t_i)}{g(t_i)} \frac{g(t_i) - f^{[k]}(t_i)/C}{h(t_i) - f(t_i)}, \quad (214)$$

where $f^{[k]}(t_i)$ is the value of the varied transition kernel. DIRE allows for renormalization-scale variations in the argument of running-coupling evaluations, as well as variations of parton distribution functions.

Finally, DIRE stacks weighted-shower steps, especially to allow the incorporation of matrix-element corrections. The possibility for stacking relies on two realizations: weighted-shower

---

[3]For example, the algorithm of [137] relies on a three-step (un)weighting.

induced event weights are multiplicative, and after applying the event weight of previous steps, the shower rate will be correctly determined from the full splitting rate. Thus, for subsequent weighted shower steps, the full splitting kernel that would be obtained after applying the weight is the new sampling rate — or "overestimate" — for the next, stacked, weighted-shower step.

DIRE currently stacks two weighted-shower steps. Ignoring, for the sake of a simple presentation, splitting enhancements and variations, then the first weighted shower (used to exponentiate complicated splitting kernels) yields a weight

$$w_1 = \frac{h_1(t)}{g_1(t)} \prod_{i=1}^{n} \frac{h_1(t_i)}{g_1(t_i)} \frac{g_1(t_i) - f_1(t_i)}{h(t_i) - f(t_i)}, \tag{215}$$

while the stacked weighted shower (used within the context of matrix element corrections) further induces the weight

$$w_2 = \frac{h_2(t')}{f_1(t')} \prod_{i=1}^{m} \frac{h_2(t_i')}{f_1(t_i')} \frac{f_1(t_i') - f_2(t_i')}{h_2(t_i') - f_2(t_i')}, \tag{216}$$

with

$$f_2(t_i') = f_1(t_i') \otimes \text{ME-correction} \quad \text{and} \quad h_2(t) = \begin{cases} \text{sign}[f_2(t)] f_1(t), & \text{if } |f_1(t)| > |f_2(t)|, \\ k_2 f_2(t), & \text{if } |f_1(t)| < |f_2(t)|, \end{cases} \tag{217}$$

where $k_2 \sim 1.5$. Since matrix-element corrections are applied only once a viable splitting has been selected, and the corresponding phase-space point generated. Thus, the set of all t' is different, and smaller, than the set of all t. Currently, DIRE does not implement enhancements or variations in the stacked weighted-shower step, since there does not seem to be a strong need for such complications. Variations may, in the future, be used to embed uncertainties due to the underlying Lagrangian entering the matrix elements.

Note that only the product of all event weights is required. Thus, the stacked algorithm is identical to the original weighted-shower algorithm from an outside perspective.

5 Matching and merging

Matching and merging methods aim to augment the event generator with (multiple) calculations performed within fixed-order perturbation theory. This is rather straight forward for individual (and simple) hard-scattering calculations, which may be treated as the "hard process" from which further event generation steps start. When the fixed-order calculation includes virtual and/or real corrections, a consistent treatment quickly becomes more complex, such that dedicated schemes of combining external calculations with the event generator need to be developed.

Naive parton showers aim to reproduce the effect of many collinear or soft emissions, and thus require improvements when describing observables that depend on well-separated hard particles. Fixed-order perturbative calculations furnish, on the other hand, an appropriate description of events with a handful of well-separated particles, but may fail in the collinear and soft limits. At high-energy colliders, observables typically exhibit effects of both approximations. On top of that, both the bulk cross sections (of low jet multiplicity) and tails (depending on the correct rate of high jet multiplicities) are often equally important. Methods to perform a matching or a merging of the fixed-order calculations with parton showers aim to combine the strengths of both approaches.

Before going into the details of matching and merging methods, it is useful to discuss some aspects of fixed-order calculations. Higher-order calculations require the calculation of virtual and real corrections. The latter introduce additional final-state particles, so that the (next-to-)^{k}order prediction for an observable O is

$$\langle O \rangle = \int d\Phi_n \sum_{i=0}^{k} \frac{d\sigma_n^{(i)}}{d\Phi_n} O(\Phi_n) + \int d\Phi_{n+1} \sum_{i=0}^{k-1} \frac{d\sigma_{n+1}^{(i)}}{d\Phi_{n+1}} O(\Phi_{n+1}) + \int d\Phi_{n+1} \sum_{i=0}^{k-2} \frac{d\sigma_{n+2}^{(i)}}{d\Phi_{n+2}} O(\Phi_{n+2})$$

$$+ \cdots + \int d\Phi_{n+k} \frac{d\sigma_{n+k}^{(0)}}{d\Phi_{n+k}} O(\Phi_{n+k}), \tag{218}$$

where the superscript (i) determines the loop order. The symbols $d\Phi_n$, $d\Phi_{n+1}$, $\ldots$, $d\Phi_{n+k}$ refer to the n-, $(n+1)$-, $\ldots$, $(n+k)$-particle phase-space measures defined in eq. (19). We will refer to a "matching method" as a method to combine complete higher-order corrections (*i.e.* all terms for the order k) to a single inclusive process. A "merging method" combines several calculations for a lowest-multiplicity base process and related processes with additional well-separated jets (*i.e.* up to a certain multiplicity $n+m$, but possibly omitting some higher-(i) terms) with parton showering. The goals of these two approaches are often overlapping. Next-to-leading order matching methods aim to include the NLO prediction

$$\langle O \rangle_{\text{NLO,in}} = \int d\Phi_n \left(\frac{d\sigma_n^{(0)}}{d\Phi_n} + \frac{d\sigma_n^{(1)}}{d\Phi_n} \right) O(\Phi_n) + \int d\Phi_{n+1} \frac{d\sigma_{n+1}^{(0)}}{d\Phi_{n+1}} O(\Phi_{n+1}), \tag{219}$$

while a leading-order merging combines the calculation

$$\langle O \rangle_{\text{LO,in}} = \int d\Phi_n \frac{d\sigma_n^{(0)}}{d\Phi_n} O(\Phi_n) \Theta(Q(\Phi_n) - Q_{\text{MS}}) + \int d\Phi_{n+1} \frac{d\sigma_{n+1}^{(0)}}{d\Phi_{n+1}} O(\Phi_{n+1}) \Theta(Q(\Phi_{n+1}) - Q_{\text{MS}})$$

$$+ \cdots + \int d\Phi_{n+m} \frac{d\sigma_{n+m}^{(0)}}{d\Phi_{n+k}} O(\Phi_{n+k}) \Theta(Q(\Phi_{n+m}) - Q_{\text{MS}}), \tag{220}$$

with the parton shower. Here, Q_{MS} denotes the so-called merging scale, which separates the hard (fixed-order) region $Q > Q_{\text{MS}}$ from the soft/collinear (resummation) region $Q \leq Q_{\text{MS}}$. This scale is in principle arbitrary, and merging algorithms should not develop a strong dependence on the exact choice, as long as it amounts to a reasonably small scale.

Before any combination is attempted, it is important to remember that virtual and real corrections are separately infrared divergent, and only their sum is free of singularities. This means that the individual contributions need to be regularized carefully, making the (un-weighted) generation of events challenging. Matching and merging can help with this, as explained below. Furthermore, fixed-order calculations are *inclusive*, meaning that a calculation for the process $ab \rightarrow c + X$ includes real-emission corrections implicitly, as part of X. For example, a leading-order calculation for $pp \rightarrow e^+e^- + X$ implicitly includes the process $pp \rightarrow e^+e^- g$. Fixed-order calculations for different processes can thus not simply be added — they first have to be reorganized as *exclusive* cross sections. At fixed order, this is achieved by including all relevant virtual corrections. The parton shower employs Sudakov factors or no-emission probabilities to produce exclusive all-order cross sections — a reminder that Sudakov factors resum virtual corrections to all orders. High-multiplicity fixed-order calculations and showered low-multiplicity predictions may overlap as well.

Thus, various sources of overlap between calculations should be handled when combining fixed-order calculations with parton showers. Matching and merging methods typically employ a mix of subtraction, phase-space division and (emission or event) vetoes for this task. The subtractions that are required in matched or merged calculations can occur at fixed perturbative order or at all perturbative orders.

The aim of additional fixed-order subtractions is to remove the parton-shower approximations of real and/or virtual corrections from the fixed-order calculation, such that the resulting remnant can be showered without introducing overlaps. At next-to-leading order, this leads to (NLO) matching formulas that schematically have the form

$$
\langle O \rangle_{\text{NLO+PS}} = \int d\Phi_n \left(\frac{d\sigma_n^{(0)}}{d\Phi_n} + \frac{d\sigma_n^{(1)}}{d\Phi_n} - \frac{d\sigma_n^{\text{PS}(1)}}{d\Phi_n} \right) \mathcal{S}(O, \Phi_n)
$$
$$
+ \int d\Phi_{n+1} \left(\frac{d\sigma_{n+1}^{(0)}}{d\Phi_{n+1}} - \frac{d\sigma_{n+1}^{\text{PS}(0)}}{d\Phi_{n+1}} \right) \mathcal{S}(O, \Phi_{n+1}), \tag{221}
$$

where $\mathcal{S}$ is the shower operator defined in eq. (90). This also shows that these "parton-shower subtractions" are typically mandatory to allow for fixed-order event generation, since this would allow for the generation of the bracketed terms in eq. (221) as individual event samples. Additionally, this highlights that matrix-element-corrected parton showering can lead to simple NLO matching methods. Matrix-element corrections guarantee that the first emission in the parton shower is distributed according to the full tree-level rate by improving the splitting kernel eq. (80),

$$
K_{j/\tilde{i}\tilde{k}} \rightarrow P_{\text{MEC}} K_{j/\tilde{i}\tilde{k}} \quad \text{with} \quad P_{\text{MEC}} = \frac{\left| \mathcal{M}_{n+1}^{(0)} \right|^2}{\sum_j K_{j/\tilde{i}\tilde{k}} \left| \mathcal{M}_n^{(0)} \right|^2}. \tag{222}
$$

It is straightforward to see that in the sum over all branchings this reproduces the full $n + 1$-particle matrix element,

$$
\sum_j P_{\text{MEC}} K_{j/\tilde{i}\tilde{k}} \left| \mathcal{M}_n^{(0)} \right|^2 = \left| \mathcal{M}_{n+1}^{(0)} \right|^2. \tag{223}
$$

In practice, the correction factor P_{MEC} is implemented via an additional multiplicative factor in the accept probability of the shower veto algorithm, *cf.* section 2.2.3. It is worth noting that matrix-element-correction methods historically appeared well before generic NLO matching methods [75, 145–147]. MECs identified

$$
\frac{d\sigma_{n+1}^{\text{PS}(0)}}{d\Phi_{n+1}} = \frac{d\sigma_{n+1}^{(0)}}{d\Phi_{n+1}}, \tag{224}
$$

$$
\frac{d\sigma_n^{\text{PS}(1)}}{d\Phi_n} = - \int d\Phi_1 \frac{d\sigma_{n+1}^{(0)}}{d\Phi_{n+1}}, \tag{225}
$$

leading to the POWHEG matching prescription

$$
\langle O \rangle_{\text{NLO+PS}} = \int d\Phi_n \left(\frac{d\sigma_n^{(0)}}{d\Phi_n} + \frac{d\sigma_n^{(1)}}{d\Phi_n} + \int d\Phi_1 \frac{d\sigma_{n+1}^{(0)}}{d\Phi_{n+1}} \bigg|_{\Phi_n} \right) \mathcal{S}_{\text{MEC}}(O, \Phi_n). \tag{226}
$$

Since the matrix-element-corrected parton shower $\mathcal{S}_{\text{MEC}}$ would now produce real-emission events, it is not possible to combine this calculation naively with further pre-calculated multi-parton fixed-order predictions. It would, however, be possible to add new tree-level samples

if the contributions are also subtracted in the overall result:

$$
\begin{aligned}
\langle O \rangle_{\text{STACKED}} = \int \mathrm{d}\Phi_n \left(\frac{\mathrm{d}\sigma_n^{(0)}}{\mathrm{d}\Phi_n} + \frac{\mathrm{d}\sigma_n^{(1)}}{\mathrm{d}\Phi_n} + \int \mathrm{d}\Phi_1 \left. \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \right|_{\Phi_n} \right) O(\Phi_n) \\
- \int \mathrm{d}\Phi_n \mathrm{d}\Phi_1 \left. \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \right|_{\Phi_n} \Theta(Q(\Phi_{n+1}) - Q_{\text{MS}}) O(\Phi_n) \\
+ \int \mathrm{d}\Phi_{n+1} \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \Theta(Q(\Phi_{n+1}) - Q_{\text{MS}}) O(\Phi_{n+1}).
\end{aligned}
\tag{227}
$$

This somewhat academic exercise of "stacking" fixed-order calculations can be cast into a more familiar form related to the parton shower. For that, we introduce the "parton-shower weight" of an n-parton state w_n^{PS} as the exact parton-shower rate of the n-parton state, excluding the product of naive splitting probabilities. Thus,

$$
w_n^{\text{PS}} = f_0(x(\Phi_0), \mu_{f\text{PS}}^2) \prod_{i=0}^{n-1} \Pi_i(t_i, t_{i+1}; \Phi_i) \frac{\alpha_s(t_{i+1})}{2\pi} \frac{f_{i+1}(x(\Phi_{i+1}), t_{i+1})}{f_i(x(\Phi_i), t_{i+1})}.
\tag{228}
$$

Similarly, we may collect all coupling and PDF factors used at fixed order into the fixed-order weight

$$
w_n^{\text{FO}} = \left(\frac{\alpha_s(\mu_r^2)}{2\pi} \right)^n f_n(x(\Phi_n), \mu_f^2).
\tag{229}
$$

Applying the ratio of the former to the latter weight, to an n-parton fixed-order calculation introduces appropriate parton-shower higher orders. With this, we may instead add and subtract all-order contributions, leading to

$$
\begin{aligned}
\langle O \rangle_{\text{MERGED}} = \int \mathrm{d}\Phi_n \left(\frac{\mathrm{d}\sigma_n^{(0)}}{\mathrm{d}\Phi_n} + \frac{\mathrm{d}\sigma_n^{(1)}}{\mathrm{d}\Phi_n} + \int \mathrm{d}\Phi_1 \left. \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \right|_{\Phi_n} \right) \frac{w_n^{\text{PS}}}{w_n^{\text{FO}}} O(\Phi_n) \\
- \int \mathrm{d}\Phi_n \mathrm{d}\Phi_1 \frac{w_{n+1}^{\text{PS}}}{w_{n+1}^{\text{FO}}} \left. \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \right|_{\Phi_n} \Theta(Q(\Phi_{n+1}) - Q_{\text{MS}}) O(\Phi_n) \\
+ \int \mathrm{d}\Phi_{n+1} \frac{w_{n+1}^{\text{PS}}}{w_{n+1}^{\text{FO}}} \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \Theta(Q(\Phi_{n+1}) - Q_{\text{MS}}) O(\Phi_{n+1}) \\
\approx \int \mathrm{d}\Phi_n \left(\frac{\mathrm{d}\sigma_n^{(0)}}{\mathrm{d}\Phi_n} + \frac{\mathrm{d}\sigma_n^{(1)}}{\mathrm{d}\Phi_n} + \int \mathrm{d}\Phi_1 \left. \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \right|_{\Phi_n} \right) \frac{w_n^{\text{PS}}}{w_n^{\text{FO}}} \Pi_n(t_n, t_{cut}; \Phi_n; > Q_{\text{MS}}) O(\Phi_n) \\
+ \int \mathrm{d}\Phi_{n+1} \frac{w_{n+1}^{\text{PS}}}{w_{n+1}^{\text{FO}}} \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \Theta(Q(\Phi_{n+1}) - Q_{\text{MS}}) O(\Phi_{n+1}).
\end{aligned}
\tag{230}
$$

Here, the additional argument "$> Q_{\text{MS}}$" in the no-emission probability indicates that only emissions leading to states with $Q(\Phi_{n+1}) > Q_{\text{MS}}$ should be considered — leading to what is sometimes called the "vetoed shower" no-emission probability. The lines after the approximate equality would be fully equivalent to the lines before if the shower correctly reproduced the rate $\mathrm{d}\sigma_{n+1}^{(0)}/\mathrm{d}\Phi_{n+1}$.

This equation leads to an interesting interpretation: the inclusion of a no-emission probability Π_n on top of the fixed-order n-parton cross section *is producing a subtraction* that allows it to be combined the $(n + 1)$-parton event samples. Event samples that are made exclusive with the help of no-emission probabilities can be added without further complication. This

realization is the basis of merging methods, which extend the argument to the combination of several multiparton calculations. If the no-emission probabilities are approximated by jets after the complete evolution sequence, then the merging procedure can become independent of the shower details. This is the case for MLM jet matching. In the Catani–Krauss–Kuhn–Webber–Lönnblad (CKKW-L) method, the second equation and the exact (partonic) no-emission probabilities of the parton shower are used to calculate the rescalings $w^{\mathrm{PS}}_{n+1}/w^{\mathrm{FO}}_{n+1}$. Incidentally, such ratios are often called the "CKKW-L weight" or "merging weight". Unitarized merging methods retain the explicit add-subtract structure to guarantee the correct inclusive cross sections even if the parton shower does not accurately reproduce the (higher-order) emission pattern.

As of today, a broad spectrum of matching and merging techniques has been developed. Historically, the first method were matrix-element corrections (MECs) [75], where the shower kernel itself is corrected to the full matrix element after the first emission. This method has later been extended to include higher orders as well [64, 97, 109, 112, 148]. For NLO matching, two general schemes exist, namely MC@NLO [149] and POWHEG [150–152], with the former being automated in the MADGRAPH5_AMC@NLO [153] and SHERPA [154] event-generation frameworks and the latter available through the POWHEG BOX program [155]. Well-established tree-level merging methods are MLM [156,157] and CKKW [158,159], which utilize a simple jet-matching algorithm and analytic Sudakov factors, respectively. The CKKW-L method [160–162] and METS [163] extend the CKKW merging scheme to use numerical no-branching probabilities, generated by trial showers.

The Unitarised Matrix Element + Parton Shower (UMEPS) scheme [164, 165] improves the unitarity of tree-level merging and thereby addresses the issue that these change the inclusive cross section of the event samples. At NLO, multiple refinements of the aforementioned LO merging schemes exist. The NL3 technique [166] extends CKKW-L to NLO, just as UNLOPS [167, 168] does the same with UMEPS. At the same time, UNLOPS may be viewed as the unitarity-improved version of NL3. The MENLOPS scheme [169, 170] combines an NLO calculation for the lowest multiplicity with LO calculations for higher multiplicities in the METS scheme, while the full extension to NLO is treated in the MEPS@NLO scheme [171, 172]. The FxFx method [173] combines MC@NLO matching with MLM merging. The MiNLO scheme [174, 175] may be regarded as a scale-setting-improved NLO extension of the CKKW algorithm, with analytically calculated NLL Sudakov factors between the clustered states.

Before describing the matching and merging methods available in PYTHIA 8.3, it should be emphasized that (NLO) matching and merging methods introduce the stated fixed-order accuracy only up to the matched (merged) multiplicities. That is, an NLO-matched n-jet event sample has NLO accuracy only for n-jet observables, while $(n+1)$-jet observables will have LO accuracy, and $(n+2)$-jet as well as higher-multiplicity observables have parton-shower accuracy only. Similarly, a merged event sample with up to n jets at (N)LO accuracy, has (N)LO accuracy for m-jet observables with $m \leq n$. In the case of NLO merging, $(n+1)$-jet observables will have LO accuracy, while they will have parton-shower accuracy for LO merging. In either case, m-jet observables with $m > n + 1$ have only parton-shower accuracy. Another question is the accuracy of the inclusive cross section. In NLO matching schemes, the inclusive cross section is accurate to NLO by design. In merging schemes, the inclusive cross section of n-jet cross sections are individually retained only if the merging scheme is constructed to be unitary, such as UMEPS or UNLOPS. In non-unitary merging schemes, inclusive cross sections are changed by the inclusion of higher-multiplicity event samples.

5.1 PYTHIA methods for leading-order multi-jet merging

As discussed above, PYTHIA offers a variety of leading-order merging schemes. This allows for a test of the robustness of merged predictions, beyond assessing the uncertainty due to scheme-specific parameters. The main task for a leading-order merging scheme is to produce

an inclusive event sample that provides a simultaneous tree-level prediction for observables depending on any number of jets $\leq n$. This entails removing any overlap between the tree-level calculations for $\leq n$ partons. The second main task is to provide a smooth transition between the "well-separated region" ($Q(\Phi_n) > Q_{\mathrm{MS}}$) described by (reweighted) tree-level results, and the "soft/collinear region" ($Q(\Phi_n) < Q_{\mathrm{MS}}$) described by the parton-shower radiation pattern. Internal merging schemes in PYTHIA also ensure that the self-consistency of the PYTHIA event-generation chain is intact.

CKKW-L multi-jet merging is the oldest tree-level merging scheme implemented in PYTHIA. It allows both standard-model and BSM core processes[4], to which multiple several colour-charged partons or W bosons are added. Lepton and hadron-collider processes are accepted. The resulting tree-level samples are combined with each other and the default parton shower by employing the merging formula

$$
\langle O \rangle_{\mathrm{CKKW-L}} = \sum_{n=0}^{N-1} \int \mathrm{d}\Phi_n \frac{\mathrm{d}\sigma_n^{(0)}}{\mathrm{d}\Phi_n} \frac{w_n^{\mathrm{PS}}}{w_n^{\mathrm{FO}}} \Pi_n(t_n, t_{cut}; \Phi_n; > Q_{\mathrm{MS}}) \Theta(Q(\Phi_n) - Q_{\mathrm{MS}}) \, \mathcal{S}(O, \Phi_n; < Q_{\mathrm{MS}})
$$
$$
+ \int \mathrm{d}\Phi_N \frac{w_N^{\mathrm{PS}}}{w_N^{\mathrm{FO}}} \frac{\mathrm{d}\sigma_N^{(0)}}{\mathrm{d}\Phi_N} \Theta(Q(\Phi_N) - Q_{\mathrm{MS}}) \, \mathcal{S}(O, \Phi_N) , \tag{231}
$$

where the showers $\mathcal{S}(O, \Phi_n; < Q_{\mathrm{MS}})$ of all but the highest-multiplicity event sample fill in emissions below the merging scale Q_{MS}. The main challenge of CKKW-L merging lies in the correct calculation of the weights w_n^{PS}. Their calculation in PYTHIA ensures that the value of the weights is identical to the all-order weight the shower would had attached to the state Φ_n, had it produced the state internally. This requires the construction of all possible parton-shower histories, and an admixture of the (history-dependent) weight factors identical to that of the shower [161].

A theoretical drawback of CKKW-L is that inclusive jet cross sections change upon inclusion of higher-multiplicity tree-level samples. The size of the change is determined by the value of the merging scale Q_{MS}, leading to unacceptable changes for Q_{MS} of $\mathcal{O}(1\mathrm{GeV})$. For low merging scales, the exact "subtract what you add" strategy highlighted in eq. (230) has to be employed. For this purpose, the UMEPS method has been introduced in PYTHIA. The UMEPS implementation can handle the same process classes as the CKKW-L scheme. The UMEPS merging formula reads

$$
\langle O \rangle_{\mathrm{UMEPS}} = \sum_{n=0}^{N-1} \int \mathrm{d}\Phi_n \Bigg[\frac{\mathrm{d}\sigma_n^{(0)}}{\mathrm{d}\Phi_n} \frac{w_n^{\mathrm{PS}}}{w_n^{\mathrm{FO}}} \Theta(Q(\Phi_n) - Q_{\mathrm{MS}})
$$
$$
- \int \mathrm{d}\Phi_1 \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \frac{w_{n+1}^{\mathrm{PS}}}{w_{n+1}^{\mathrm{FO}}} \Theta(Q(\Phi_{n+1}) - Q_{\mathrm{MS}}) \Bigg] \mathcal{S}(O, \Phi_n; < Q_{\mathrm{MS}})
$$
$$
+ \int \mathrm{d}\Phi_N \frac{w_N^{\mathrm{PS}}}{w_N^{\mathrm{FO}}} \frac{\mathrm{d}\sigma_N^{(0)}}{\mathrm{d}\Phi_N} \Theta(Q(\Phi_N) - Q_{\mathrm{MS}}) \, \mathcal{S}(O, \Phi_N) , \tag{232}
$$

The subtractive samples in this formula are produced with the help of the parton-shower history employed to calculate the weights w_n^{PS} [164].

As a final remark on leading-order merging, it should be noted that PYTHIA offers detailed semi-internal `UserHooks` utilities for MLM jet matching [176]. This early approach to combining fixed-order matrix-element calculations with parton showers approximates the parton-shower no-emission probabilities necessary to remove overlap between samples with a

---

[4]Note, though, that no attempt is made at diagram removal or subtraction for colour-changed BSM resonances that decay into SM quarks.

pragmatic event-veto procedure: the parton ensemble at fixed order is stored, and compared to jets after showering the ensemble. If each jet directions overlaps with one parton direction, the event is retained. The rate of rejected events mimics the application of no-emission probabilities. This convenient approach has the benefit of simplicity and computational efficiency, at the expense of sacrificing a formal understanding of the result.

5.2 PYTHIA methods for matching

The POWHEG NLO matching formalism as given in eq. (226) provides an elegant and universal method for the combination of NLO calculations and parton showers. It is universal, as it does not depend on the exact implementation of the parton shower to be matched. This is, because in addition to the NLO-corrected Born-level event, the first emission is generated according to a matrix-element corrected no-branching probability

$$
\bar{\Pi}(Q_0^2, Q_1^2) = \exp\left\{ -\int_{Q_1^2}^{Q_0^2} \mathrm{d}\Phi_{+1} \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\sigma_n^{(0)}} \right\} ,
\tag{233}
$$

which is independent of the branching kernels used by the shower. In principle, an NLO-matched prediction could therefore be obtained with any given shower by starting the shower evolution at the scale of the first POWHEG emission. In practice, the ordering variable of the shower t will disagree with the ordering variable Q^2 used in the POWHEG formalism. To avoid over- or under-counting emissions, it is hence preferable to start the shower at the phase-space maximum (*i.e.* using a "power shower", *cf.* section 4.1.3) and vetoing emissions that are harder than the POWHEG emission according to the POWHEG ordering variable. This method was outlined in [90] and since then PYTHIA 8.3 provides the relevant POWHEG hook to supply consistent showering of POWHEG BOX events with the simple showers. It is worth noting that this procedure leads to the somewhat awkward situation that the first, *i.e.* hardest, jet is produced from the kinematics and colour configuration of the Born+1-jet state. To circumvent this, a more complete treatment would involve clustering the first emission and running a vetoed truncated shower off the actual Born state for the first emission. This is currently not available in PYTHIA 8.3.

It might potentially be regarded as an inelegance that the POWHEG no-branching probability eq. (233) exponentiates the full matrix element, including process-specific non-singular terms, and that the value of Q_0^2 is typically given by the (hadronic) phase-space limit, and not a scale that more adequately defines the transition between "hard" and "soft" radiation. These concerns are avoided in the (historically first developed) MC@NLO method, in which the real-radiation matrix element is separated into an infrared-singular ("soft") and infrared-regular ("hard") part,

$$
\mathrm{d}\sigma_{n+1}^{(0)} = \mathrm{d}\sigma_{n+1}^{S(0)} + \mathrm{d}\sigma_{n+1}^{H(0)} .
\tag{234}
$$

Therefore, only the singular part in the no-branching probability is retained,

$$
\bar{\Pi}^S(Q_0^2, Q_1^2) = \exp\left\{ -\int_{Q_1^2}^{Q_0^2} \mathrm{d}\Phi_{+1} \frac{\mathrm{d}\sigma_{n+1}^{S(0)}}{\mathrm{d}\sigma_n^{(0)}} \right\} ,
\tag{235}
$$

so that the MC@NLO matched expectation value of an observable O reads

$$
\langle O \rangle_{\text{NLO+PS}}^{\text{(MC@NLO)}} = \int d\Phi_n \left[\frac{d\sigma_n^{(0)}}{d\Phi_n} + \frac{d\sigma_n^{(1)}}{d\Phi_n} + \int d\Phi_{+1} \frac{d\sigma_{n+1}^{\text{PS(0)}}}{d\Phi_{n+1}} \right.
$$
$$
\left. + \int d\Phi_{+1} \left(\frac{d\sigma_{n+1}^{\text{S(0)}}}{d\Phi_{n+1}} - \frac{d\sigma_{n+1}^{\text{PS(0)}}}{d\Phi_{n+1}} \right) \right] \mathcal{S}(O, \Phi_n)
$$
$$
+ \int d\Phi_{n+1} \frac{d\sigma_{n+1}^{\text{H(0)}}}{d\Phi_{n+1}} . \tag{236}
$$

As evident from eq. (236), the MC@NLO method requires stringent consistency between the NLO calculation and the shower. Different to the POWHEG method, it therefore does not provide a universal scheme but needs to be implemented explicitly for each shower. To facilitate a simple implementation of the MC@NLO technique for PYTHIA's simple shower, PYTHIA 8.3 provides a global-recoil scheme, *cf.* section 4.1.3. A publicly available parton-level event generator supporting the generation of MC@NLO events for PYTHIA's simple shower is MAD-GRAPH5_AMC@NLO. Caution is, however, advised, as the global-recoil scheme might not be a suitable choice for each and every process.

Different to the POWHEG formalism, the MC@NLO scheme explicitly employs negative-weighted events (in fact POWHEG was designed to remove negative weights from MC@NLO). While not posing a problem technically, negative weights reduce the efficiency of any simulation, simply because they have to compensated for in histograms with positive-weighted events, of which more are needed to obtain the same statistics as for simulations with a strict probability interpretation. The fraction of negative-weighted events can be reduced by the MC@NLO-Δ formalism [177], which regulates the divergent structure of real-emission matrix elements via shower-generated no-branching probabilities. In addition, the MC@NLO-Δ prescription introduces an independent shower starting scale for each colour line in the hard process. Such multi-scale treatment is also required in the POWHEG formalism, if resonance-aware matching is pursued, *e.g.* when using the POWHEG BOX RES generator [178]. In both cases, PYTHIA 8.3 offers the necessary machinery to deal with multiple scale definitions [179] through `UserHooks` (see section 9.7.2). While PYTHIA 8.3 offers in-house implementations for MC@NLO and POWHEG matching as alluded to above, other matching schemes can conveniently be implemented via user hooks, *cf.* section 9.7.2. A prominent example is the NNLO+PS matching framework GENEVA [180–183].

5.3 PYTHIA methods for NLO multi-jet merging

A number of schemes to combine several matched NLO (QCD) calculations with each other and parton showering are available in PYTHIA. As was the case for tree-level merging (*cf.* section 5.1), this allows for an exploration — through comparison within the same code base — of the benefits and limitations of various approaches, as well as NLO merged predictions more generally.

Historically, the first two NLO merging schemes embedded in PYTHIA were NL^3 (an extension of the CKKW-L approach to NLO) and UNLOPS, the extension of UMEPS to NLO accuracy. Beyond the theoretical and computational challenges already present at leading order, NLO merging needs to ensure that the application of all-order weights $w_n^{\text{PS}}/w_n^{\text{FO}}$ does not invalidate the NLO precision of the input samples due to overlapping virtual corrections. This can be achieved by subtracting the first-order expansion of the shower weights attached to tree-level events. Thus, the main complication in calculation is in generating terms in the shower expansion [184]. PYTHIA uses trial showering to generate the expansion of no-emission probabilities, and analytic results to calculate the expansion of running-coupling and PDF factors.

Once these technicalities are under control, the NLO extension of CKKW-L implements the matching formula

$$
\begin{aligned}
\langle O \rangle_{\mathrm{NL}^3} = &\sum_{n=0}^{N} \int \mathrm{d}\Phi_n \frac{\mathrm{d}\sigma_n^{(0)}}{\mathrm{d}\Phi_n} \frac{w_n^{\mathrm{PS}}}{w_n^{\mathrm{FO}}} \Pi_n(t_n, t_{cut}; \Phi_n; > Q_{\mathrm{MS}}) \Theta(Q(\Phi_n) - Q_{\mathrm{MS}}) \, \mathcal{S}(O, \Phi_n; < Q_{\mathrm{MS}}) \\
&+ \sum_{n=0}^{N} \int \mathrm{d}\Phi_n \left(\frac{\mathrm{d}\sigma_n^{(1)}}{\mathrm{d}\Phi_n} + \int \mathrm{d}\Phi_1 \left. \frac{\mathrm{d}\sigma_{n+1}^{(0)}}{\mathrm{d}\Phi_{n+1}} \right|_{\Phi_n} \right. \\
&\qquad \left. - \frac{\mathrm{d}\sigma_n^{(0)}}{\mathrm{d}\Phi_n} \left[\frac{w_n^{\mathrm{PS}}}{w_n^{\mathrm{FO}}} \Pi_n(t_n, t_{cut}; \Phi_n; > Q_{\mathrm{MS}}) \right]_{\mathcal{O}(\alpha_s)} \right) \mathcal{S}(O, \Phi_n; < Q_{\mathrm{MS}}) \\
&+ \int \mathrm{d}\Phi_{N+1} \frac{\mathrm{d}\sigma_{N+1}^{(0)}}{\mathrm{d}\Phi_{N+1}} \frac{w_{N+1}^{\mathrm{PS}}}{w_{N+1}^{\mathrm{FO}}} \Theta(Q(\Phi_{N+1}) - Q_{\mathrm{MS}}) \, \mathcal{S}(O, \Phi_{N+1}).
\end{aligned}
\tag{237}
$$

where square bracket with subscript $\mathcal{O}(\alpha_s)$ indicate that the $\mathcal{O}(\alpha_s)$ term of the expansion of the bracketed term is required. The first and last line of eq. (237) are identical to the CKKW-L result. The second line incorporates the inclusive NLO correction, and the subtraction of double counting of virtual corrections. As was the case for CKKW-L, the calculation of all necessary terms in eq. (237) employs parton shower histories.

However, the NL^3 formula eq. (237) has the same theoretical drawback as the CKKW-L results: inclusive n-parton rates are changed when including corrections to $m > n$ parton rates. The size of the effect is determined by Q_{MS}, and can easily be of a similar numerical size as NLO corrections for moderately small Q_{MS}. Thus, PYTHIA also extends the UMEPS method, which corrects this behaviour, to NLO accuracy. The resulting UNLOPS merging formula can be found in [184]. UNLOPS is the preferred NLO merging scheme in PYTHIA.

Before moving on, it is interesting to observe that any reweighting of the second line in eq. (237) with higher-order terms will neither affect the NLO fixed-order nor the shower accuracy of the combined calculation. Thus, variants of NLO merged methods can be obtained from reweighting these contributions. This freedom, and the resulting uncertainty, is exposed in PYTHIA by offering three different variants of UNLOPS [185]. Sensible uncertainties from NLO merged calculations should include the envelope of these variants as "scheme uncertainty".

As part of its semi-internal implementation of MLM jet matching, PYTHIA also offers semi-internal utilities to combine input samples produced for FxFx merging [173]. This scheme extends the MLM method to NLO QCD accuracy, and handles the overlap between events of different multiplicity before showering in a hybrid scheme: fixed-order events are reweighted with analytic Sudakov form factors to produce results that are additive before showering. The overlap after showering is addressed in a pragmatic way, following along the lines of MLM jet matching. The resulting scheme is computationally efficient, especially since Sudakov form factors can directly be integrated into the fixed-order calculation code. However, this efficiency comes at the price of an unclear all-order structure of the prediction. Nevertheless, the scheme has found a large user base.

5.4 Matching and Merging in VINCIA

The unique "maximally bijective" property of VINCIA's sector antenna showers, *cf.* section 4.2, make them well suited for incorporating corrections from fixed-order matrix elements, especially at high multiplicities. The focus so far has been on QCD corrections, although the sector nature of VINCIA's coherent QED shower should make adaptations to include QED corrections as well fairly straightforward.

At leading order, a dedicated CKKW-L merging scheme has been implemented which exploits the properties of VINCIA's sector showers. This is discussed below in section 5.4.1. Details on the general PYTHIA CKKW-L implementation can be found in section 5.1.

Next-to-leading order matching in the antenna framework is so far not generally available, but VINCIA's QCD showers, including the resonance-final one, can be combined with NLO QCD calculations by shower-independent matching methods, such as POWHEG. This is described in section 5.4.2.

Although VINCIA currently has no built-in NLO matching schemes, a generalization of the scheme developed in [96, 186] may become available in the future. In a similar vein, VINCIA does not offer the merging of multiple NLO-matched samples in the current version. Schemes extending the ones described in section 5.3 to sector showers may, however, be implemented in the future.

VINCIA's NNLO matching approach presented in [187] is not yet part of the public PYTHIA 8.3 releases. We expect it to become available in a future release, once it has been applied and validated for a larger class of processes.

A signature feature in early developments of VINCIA, iterated matrix-element corrections [64, 106, 148, 188] have not yet been made available in PYTHIA 8.3. Plans are underway to do so, building on the new matrix-element interfaces described in section 10.1.6. Note also that VINCIA's weak shower, described in section 4.2.4, is currently not sectorized and hence full-fledged EW merging would presumably require some work. Finally, note that dedicated matching and merging strategies for VINCIA's interleaved treatment of resonance decays have not yet been worked out. See the program's `online manual` for updates.

5.4.1 Leading-order merging

Tree-level merging with VINCIA is done in the CKKW-L scheme [158, 160, 161] according to eq. (231), properly extended to sector showers [162]. The phase-space sectorization particularly facilitates the merging at very high multiplicities and offers increased control over highly-complex final states. Most of the merging method is identical to the PYTHIA implementation described in section 5.1, with the difference only in the construction of shower histories needed for the Sudakov reweighting. The settings relevant to VINCIA's CKKW-L implementation can be found in section 9.6.1.

In the default CKKW-L scheme, all possible shower histories are constructed and the one maximizing the branching probability is chosen, *cf.* section 5.1. In the sector-shower CKKW-L implementation, however, the construction of all possible histories is replaced by a deterministic inversion of the shower evolution. This is possible because VINCIA's sector showers generate branchings only if these correspond to the minimal sector-resolution variable, *cf.* section 4.2.2. The sector-resolution variable can then be used to exactly invert any branching. The only subtlety in this algorithm stems from the treatment of multiple quark pairs, for which all possible quark-antiquark clusterings must be taken into account. To this end, the same procedure as in the default CKKW-L method is utilized and a shower history is constructed for all viable permutations of colour strings between quark pairs, and the one maximizing the branching probability is picked. This algorithm replaces the shower history tree by maximally a few linear history branches, which not only positively affects the CPU time needed for the computation, but more importantly reduces the prohibiting scaling in memory allocation intrinsic to the default CKKW-L algorithm.

5.4.2 NLO matching

If an NLO-matched calculation with VINCIA is desirable, the POWHEG method [150, 151] with externally matched NLO event samples, as *e.g.* produced by the POWHEG BOX program [155,

178], can be used [88]. To this end, the difference in the POWHEG BOX and VINCIA evolution variables are properly accounted for by increasing the shower starting scale and vetoing emissions above the POWHEG scale [90]. The usage of POWHEG BOX with VINCIA is described in detail in [189, appendix A].

5.5 Matching and Merging in DIRE

At the time of compiling this manual, the matching and merging machinery with DIRE have not been validated within PYTHIA 8.3. Previous versions of DIRE + PYTHIA 8.2 included CKKW-L tree-level merging [135], UNLOPS NLO merging [136], iterated matrix-element corrections within the MOPS approach [135, 137], and TOMTE N3LO+PS matching [190]. We expect these previous developments to become accessible and validated in PYTHIA 8.3 in the future.

6 Soft and beam-specific processes

Hadrons and nuclei are composite objects, mainly made out of quarks and gluons. This requires the introduction of parton distribution functions (PDFs) $f_a^A(x, Q^2)$, expressing the probability density that parton a exists inside particle A with a momentum fraction x if the particle is probed at a resolution scale Q^2. Given such PDFs, hard collisions between the constituent partons can be described by perturbation theory, see section 2.3. But in the limit $p_\perp \to 0$ the cross section for perturbative QCD scattering diverges, and traditional perturbation theory breaks down.

The alternative offered already since before the advent of QCD is so-called Regge–Gribov theory [191–195], wherein an effective field theory is formulated in terms of the exchange of reggeon ($\mathbb{R}$) and pomeron ($\mathbb{P}$) objects between the colliding hadrons, with propagators and vertex-coupling strengths, the latter both to hadrons and among themselves. A reggeon (pomeron) contribution represents the resummed effect from the exchange of (an infinite set of) mesons (glueballs) with a common set of flavour quantum numbers, but ordered in a linear relationship (a "trajectory") between increasing orbital angular momentum L and increasing m^2. This language can be used to motivate expressions for total, elastic, and diffractive cross sections, even if today this is done in a pragmatic spirit, where not fully consistent adjustments of parameters can be made to better fit data.

Leptons are fundamental particles, unlike hadrons, and it would seem like traditional perturbation theory can always be applied. But a charged lepton is surrounded by a cloud of virtual photons, and these carry some of the total momentum. It therefore becomes necessary to introduce PDFs also to describe the distribution of a lepton and photons inside the whole charged lepton, as a function of Q^2. Either of these two components can then collide with constituents of the other beam. The photon, in its turn, can fluctuate further into a lepton or quark pair, and the latter again can have a non-perturbative behaviour. This requires a similar approach for photon interactions as for hadron ones, in fact with even more complexity. Since hadrons and nuclei also can contain or be surrounded by photons, by coupling to the charge of individual quarks or to the hadron as a whole, similar issues can arise in hadronic collisions.

Also fluxes of $W^\pm$ and Z bosons can be defined, and have been used in the past, both for leptons and for protons. The large weak-gauge-boson masses suppress the rate in the $p_\perp \to 0$ limit, however, and so their contributions are better handled as propagators in Feynman graphs, like the top quark and the Higgs boson. This also implies that neutrinos can be considered point-like for our purposes.

Heavy-ion collisions introduce further physics aspects, relative to hadronic collisions. Some of these are reasonably well understood, such as the role of the initial geometry, where models for the distribution of nucleons inside a nucleus can be used to find the "wounded" nucleons,

i.e. those that interact. But most of the subsequent physics is still open to interpretation, and different approaches exist. One such is the PYTHIA 8.3/ANGANTYR model, presented here.

6.1 Total and semi-inclusive cross sections

Here we introduce the components of the total hadron-hadron cross section, and how these vary as a function of the collision energy. The intention is not to go into the modelling of the collision processes as such, which is the main topic for subsequent subsections, but some such information is necessary when the differential cross sections are the basic building blocks, and the integrated ones only a consequence thereof. See also the `online manual`, under the heading "Total Cross Sections".

Throughout this section, we will discuss collisions between two high-energy hadrons A and B at a squared CM energy $s = E_{CM}^2$. By high energy, we mean roughly $E_{CM} > 10$ GeV, where the perturbative model is valid. Low-energy non-perturbative processes are discussed in section 6.1.5. The (strong-interaction) total cross section for the collision of two high-energy hadrons is conveniently subdivided into several components, typically

$$\begin{aligned}
\sigma_{tot}^{AB}(s) &= \sigma_{el}^{AB}(s) + \sigma_{inel}^{AB}(s) \\
&= \sigma_{el}^{AB}(s) + \sigma_{sd(XB)}^{AB}(s) + \sigma_{sd(AX)}^{AB}(s) + \sigma_{dd}^{AB}(s) + \sigma_{cd}^{AB}(s) + \sigma_{nd}^{AB}(s).
\end{aligned} \tag{238}$$

The components are:

- Elastic scattering (el) $AB \to AB$ where the hadrons are scattered through an angle but are otherwise unharmed. Everything else, where the final state is not AB, is collectively called inelastic.

- Single diffraction (sd) where either of the incoming hadrons becomes an excited system, while the other remains intact, $AB \to XB$ or $AB \to AX$. Here, X represents the excited system that eventually will produce two or more hadrons.

- Double diffraction (dd) where both hadrons are excited, $AB \to X_1 X_2$, but remain as separate objects.

- Central diffraction (cd), where both hadrons survive but lose energy to a new central system, $AB \to AXB$.

- Non-diffractive interactions (nd), or more formally inelastic non-diffractive ones, where both hadrons break up and form a common system, $AB \to X$, that is not (easily) subdivided into separate subsystems, unlike diffraction.

In principle, one could imagine higher diffractive topologies, say $AB \to X_1 X_2 X_3$, but these are expected to be small and are neglected here. For applications at low energies we will also introduce annihilation and resonance contributions.

The dividing line between these different components is unclear, notably between diffractive and non-diffractive events. Single- or double- diffractive systems X predominantly have low masses, and thus only produce a few particles at either end of the full rapidity range. In between, there is a large rapidity gap, *i.e.* a region in rapidity space without any particle production. That is unlike the non-diffractive events, where particle production is assumed to span the whole available rapidity range. But, since particles are discrete objects randomly produced, there will be a falling distribution of increasing gap sizes also in non-diffractive events. In contrast, the falling tail of large-mass diffractive systems can leave no obvious gap in a diffractive event. We therefore need to distinguish the theoretical modelling of cross-section classes and event properties presented here from the experimental-detector and analysis-dependent classification of observed events.
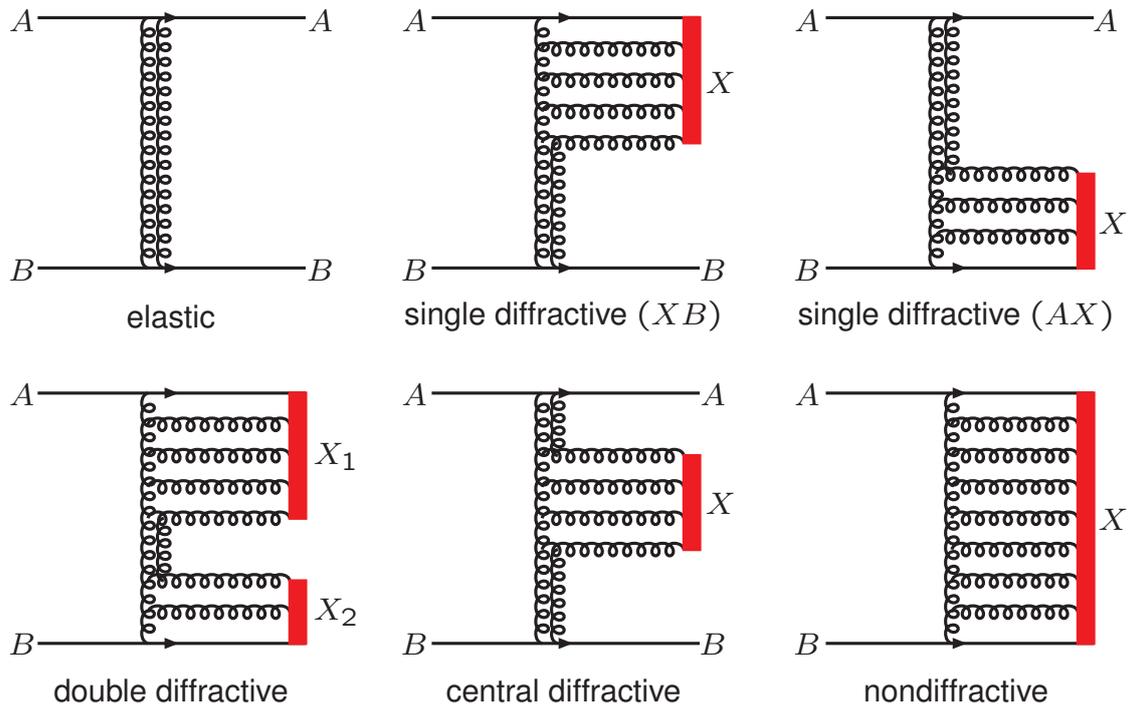
Figure 9: Schematic Feynman-diagram-style illustration of the six event classes in eq. (238). A pair of parallel vertical gluons represent a net colour-singlet exchange, a pomeron, while a single vertical gluon gives a colour-octet exchange, a cut pomeron. The vertical axis can be viewed as representing the rapidity range spanned between A and B, where horizontal gluons are regions with possible partonic final-state presence. The red bars represent the final regions where strings will be drawn and produce hadrons, whereas the regions without them are rapidity gaps.

In modern nomenclature, where a pomeron is viewed as shorthand for a two-gluon system in a colour-singlet state, the different event classes can be illustrated in terms of the colour and momentum-energy transfers between the two incoming hadrons, see fig. 9. An elastic scattering requires a pomeron (or reggeon) to be exchanged, so that the scattered hadrons remain colour singlets, but with (modestly) changed outgoing momenta. If only one gluon is exchanged, a so-called cut pomeron, then the colour transfer turns the A and B hadrons into colour-octet objects, which means they will be connected by colour strings that can fragment into hadrons over the whole rapidity range, *i.e.* this gives a non-diffractive event. Single diffraction, *e.g.* $AB \to AX$, can be viewed as a two-step process. First the emission of a $\mathbb{P}$ from A, carrying a fraction ξ of the A momentum. And second the collision between the $\mathbb{P}$ and B, giving rise to a system with $M_X^2 = \xi s$. For the first step a pomeron flux $f_{\mathbb{P}}^A(\xi, t)$ can be introduced in analogy with conventional PDFs, while the second step can be viewed as a non-diffractive-type $\mathbb{P}B$ subcollision, at least for large M_X. Double and central diffraction can be described in a similar manner.

The hadronic cross sections that will be discussed late are for reasonably high hadron-hadron CM energies, say $E_{CM} > 10$ GeV, corresponding to a fixed-target proton-proton beam energy of $E_{beam} \gtrsim 50$ GeV. Separate from this, low-energy cross sections will be discussed in the context of hadronic rescattering, section 6.1.5. To a large extent the same language can be used, but at low energies the contribution from exclusive resonances, $AB \to R \to AB$ or $AB \to R \to CD$, can give rise to rapid fluctuations in the cross section as a function of s.

6.1.1 Proton total cross sections

As already mentioned, pomeron and reggeon contributions play a large role in the modelling of cross sections. Both are expected to give an s^δ energy dependence, where $\delta \geq 0$ for pomerons and $\delta < 0$ for reggeons, such that high-energy cross sections are dominated by the pomeron term. The pomeron contribution is even, *i.e.* the same for AB and $A\overline{B}$, while reggeons can be even or odd, the latter giving opposite-sign contributions for the two processes. A hypothetical odderon contribution would be odd, as the name indicates, and have $\delta \geq 0$ like the pomeron, so that $\sigma^{A\overline{B}}(s) - \sigma^{AB}(s)$ would not vanish in the $s \to \infty$ limit. Its existence is supported by recent TOTEM data [196], but is still not included in most models.

The simple s^δ behaviour is obtained for the exchange of a single object, whereas multiple exchanges can come in with opposite signs and damp the rise of cross sections. The Froissart bound [197] shows that cross sections cannot rise faster than $\ln^2 s$ asymptotically, but that limit is far off. Other bounds are more relevant, for instance that diffractive cross sections cannot become larger than the total one [198].

The most important hadronic cross sections are the pp and p$\overline{\text{p}}$ ones. Here, four different $\sigma_{\text{tot}}(s)$ parameterizations are available for high-energy collisions in PYTHIA, plus one place-holder, see further the overview in [199]. They are roughly ordered in increasing number of free parameters tuned to data, with numbers corresponding to the options of the `SigmaTotal:mode` switch.

0. A zero option allows the user to set any value at the currently studied energy, *i.e.* it does not model any energy dependence.

1. The Donnachie–Landshoff (DL) form [200], with one pomeron and one reggeon term,

$$\sigma_{\text{tot}}^{AB}(s) = X^{AB} s^{0.0808} + Y^{AB} s^{-0.4525}, \tag{239}$$

   with s in units of GeV and σ in mb. The coefficients $X^{\text{p}\overline{\text{p}}} = X^{\text{pp}}$, as discussed above. There is no such symmetry for the Y^{AB}, which can be viewed as having one even and one odd reggeon, but with the same power.

2. The Minimum Bias Rockefeller (MBR) parameterization [201], which uses two different expressions. Below 1.8 TeV the form is given by one pomeron and two reggeon terms, whereof one odd and one even, with different δ. Above it a common Froissart-inspired form like $a \ln^2 s + b \ln s + c$ is used.

3. The ABMST model [202] includes a soft and a hard pomeron, *i.e.* lower or higher $\delta > 0$, an even and an odd reggeon, plus terms for two-pomeron and triple-gluon exchange.

4. The COMPAS/RPP parameterization [203] contains a total of six even and six odd terms, including pomeron, odderon, reggeon, and double-exchange ones.

The relevant cross section parameterizations are hard coded in options 1 – 4, and cannot easily be changed.

6.1.2 Proton elastic cross sections

Elastic cross sections are related to total ones via the optical theorem:

$$\frac{d\sigma_{\text{el}}}{dt}(s, t = 0) = \frac{1 + \rho^2}{16\pi} \sigma_{\text{tot}}^2(s), \tag{240}$$

where t represents the squared momentum transfer between the initial and final proton on the same side of the event. For detailed modelling, a suitable starting point is the elastic scattering

amplitude $T(s, t)$, from which one derives $(d\sigma_{el}/dt)(s, t) \propto |T(s, t)|^2$, $\sigma_{tot} \propto \text{Im}\, T(s, 0)$, and $\rho = \text{Re}\, T(s, 0)/\text{Im}\, T(s, 0)$. Typically ρ is close to 0 and can be neglected to first approximation. The total elastic cross section is obtained by integration over t.

For the simple Regge-theory-motivated ansatz, that $(d\sigma_{el}/dt)(s, t) \propto \exp(B_{el} t)$, one obtains

$$\sigma_{el}(s) = \frac{1 + \rho^2}{16\pi} \frac{\sigma_{tot}^2}{B_{el}}, \tag{241}$$

where, to a very good approximation at high energies, the t integration range has been extended to $[-\infty, 0]$. The ansatz also assumes that

$$B_{el}^{AB}(s) = 2b_A + 2b_B + 2\alpha' \ln(s/s_0), \tag{242}$$

where $b_{A,B}$ come from the respective hadronic form factors, with $b = 2.3$ GeV^{-2} for unflavoured baryons and 1.4 GeV^{-2} for mesons, $\alpha' = dL/dm^2 = 0.25$ GeV^{-2} is the slope of the pomeron trajectory, and $s_0 = 1/\alpha' = 4$ GeV^2 is a typical hadronic scale [198, 204].

In detail, the total-cross-section models above, as selected by `SigmaTotal:mode`, also handle elastic scattering as follows.

0. It is possible for the user to set their own σ_{el}, B_{el}, and ρ at the current energy.

1. The original DL model was extended to Schuler and Sjöstrand (SaS)/DL [198] by the simple Regge-theory ansatz above, but with the difference that the $\ln s$ dependence in eq. (242) is replaced by an $s^{0.0808}$ term to ensure that σ_{el} does not grow faster than σ_{tot} asymptotically. There is no modelling of ρ, but a value can be set by hand.

2. In MBR the ratio $\sigma_{el}(s)/\sigma_{tot}(s)$ is parameterized, separately below and above 1.8 TeV, and separately for pp and p$\bar{\text{p}}$ below it. Then eq. (241) is used to derive a $B_{el}(s)$ slope, with $\rho = 0$.

3. In ABMST the fundamental building block is a complex scattering amplitude $T(s, t)$, containing the six terms of the total cross section, each with a separate t dependence, usually, but not always, of an exponential character. From this, both total and elastic cross sections are derived, including the ρ parameter.

4. Also the COMPAS/RPP parameterization starts out from a complex $T(s, t)$, with the same comments as for ABMST, except that the number of terms now is larger.

There are no further free parameters in the code, beyond the ones mentioned above.

So far, only strong interactions have been considered. But, since protons are charged particles, there are also electromagnetic (EM) interactions. These are given by the traditional Coulomb scattering cross section, $d\sigma_{el}/dt \propto \alpha_{em}^2/t^2$, which blows up in the $t \to 0$ limit, *i.e.* for vanishing scattering angle. Therefore, it is always necessary to specify a minimal angle or equivalently a $t_{max} < 0$. There are two aspects that make it possible to disregard the EM contributions at the LHC, except for special runs. Firstly, the EM contribution exceeds the strong one only below a $|t|$ of order 10^{-3} GeV^2, which corresponds to extremely small scattering angles. Secondly, owing to this, inelastic EM collisions are completely negligible. By default, Coulomb corrections therefore are not taken into account, but can be switched on.

What complicates the issue is that the elastic scattering amplitude

$$T(s, t) = T_n(s, t) + e^{i\alpha_{em}\phi(t)} T_c(s, t), \tag{243}$$

contains a phase factor in front of the Coulomb T_c amplitude, relative to the definition of the real part of the nuclear/QCD T_n amplitude. Three different expressions are used, one for SaS/DL, and also for MBR and `SigmaTotal:mode0`, and one each for ABMST and COMPAS/RPP. Although written in slightly different ways, they give almost identical results.

6.1.3 Proton diffractive cross sections

Diffractive cross sections are differential in several variables: for single diffraction in t and M_X; for double diffraction in t, M_{X_1}, and M_{X_2}; and for central diffraction in t_1, t_2, and M_X. Here, M_X represents the mass of the respective diffractive system. Alternatively the scaled variable $\xi = M_X^2/s$ is often used, but it is less intuitive when modelling contributions from low-mass resonances. The fundamental objects are the differential expressions, and the integrated rates in general do not have simple closed forms. Within Regge theory it is possible to relate the differential expressions to the ones for total and elastic ones, with minor extensions. Specifically, single diffraction is modelled with triple-Regge exchange graphs that involve the same pomeron (or reggeon) propagators as before, but requires the introduction of triple-Regge couplings. If only pomerons are considered, as could be reasonable at high energies, then mass spectra will behave roughly like dM_X^2/M_X^2 and t spectra roughly like $\exp(Bt)$, where $B = B(s, M_X^2)$ depends on the process considered.

In reality it is more complicated, for a number of reasons. At low masses the experimental M_X spectrum is not smooth, but reflects the presence of well defined N and Δ resonance states. At high masses phase-space restrictions kick in, *e.g.* in the allowed t range, as well as a wish to minimize the overlap between diffractive and non-diffractive event topologies. In addition to the pomeron also reggeons should be considered, in various combinations, contributing to different mass distribution shapes and CM energy dependencies. Some terms increase faster with CM energy than the total cross section itself, implying that the description has to break down at some point. The solution to this is likely to involve the possibility of multiple exchanges of both a diffractive and non-diffractive nature, leading to a competition between the two [205].

Three different diffractive models are implemented [199], matching the first three descriptions of total and elastic cross sections, plus again an additional placeholder, enumerated in the `SigmaDiffractive:mode` switch in the same way as in `SigmaTotal:mode`. It is possible to combine the choice of total plus elastic and diffractive models freely.

0. One can set user defined single, double and central diffractive cross sections for the current energy. In this option there is a choice between seven possible M_X^2 spectra, with related t shapes.

1. The SaS model is based on pomeron contributions only, *i.e.* is of the form $(dM_X^2/M_X^2)\exp(Bt)$ to first approximation. At low masses a smooth enhancement is added, to provide a simple smeared-out further contribution from resonances. At large masses the rate is suppressed to reduce the rate of diffractive events with small rapidity gaps. The rise of the diffractive cross section with energy is given by integration. It turns out, however, that the initial ansatz gives a steeper rise than data, so energy-dependent damping factors have been introduced. Central diffraction is a rather recent addition, not included in many commonly used tunes, and therefore not on by default. The B slope is similar in spirit to eq. (242), but without any form factor contribution for protons that break up, and the logarithmic term is related to the rapidity gap size, *e.g.* $\ln(s/M_X^2)$ for single diffraction.

2. In the MBR model the single-, double- and central-diffractive cross sections are given as ratios of two integrals, one being the Regge cross section and the other a renormalized flux. These are matched so that the increase of diffractive cross sections is kept at an acceptably low rate. The differential distributions in M_X^2 and t are given by somewhat lengthier expressions than in SaS, but qualitatively similar.

3. The ABMST model is the most sophisticated one, in terms of number of components considered. The single-diffractive description is split into two parts, for high- and low-mass

diffraction. The former includes $\mathbb{PPP}$, $\mathbb{PPR}$, $\mathbb{RRP}$ and $\mathbb{RRR}$ graphs, plus pion exchange, each with a characteristic mass, t, and energy dependence. Four resonances are modelled in the low-mass regime, along with a background from the high-mass regime and a contact term matching the two regimes smoothly. The resonances are excited states of the proton, each a unit of angular momentum higher than the previous one. Taken together, the ABMST model gives a very good description of data at lower energies. Unfortunately the energy dependence of some terms is too steep, such that single diffraction at the LHC is overestimated by about a factor of two, and results at 100 TeV would be completely unphysical. A few different options have therefore been included in the PYTHIA implementation to damp this rise [199]. Another problem is that ABMST does not model double diffraction. One expects an approximate relationship [195]

$$\frac{\mathrm{d}^3 \sigma_{\mathrm{dd}}}{\mathrm{d}M_{X_1}^2 \, \mathrm{d}M_{X_2}^2 \, \mathrm{d}t} \approx \frac{\mathrm{d}^2 \sigma_{\mathrm{sd}}}{\mathrm{d}M_{X_1}^2 \, \mathrm{d}t} \frac{\mathrm{d}^2 \sigma_{\mathrm{sd}}}{\mathrm{d}M_{X_2}^2 \, \mathrm{d}t} \left(\frac{\mathrm{d}\sigma_{\mathrm{el}}}{\mathrm{d}t} \right)^{-1}, \tag{244}$$

however, and this has been used to extend the modelling. Also central diffraction can be introduced by a similar ansatz.

4. The COMPAS/RPP parameterization does not extend to diffraction, so there is no such option.

Each of the models above contain a wide selection of modifiable parameters, specific to that diffraction model. Both the integrated and the differential cross sections can be modified, notably affecting the dependence on CM energy and the shape of the M_X spectra.

In summary, the modelling of diffraction is highly nontrivial, and at a more primitive stage than that of total and elastic cross sections. There also exist alternative starting points to the Regge formalism we have worked with here, notably the Good–Walker one [206]. In it, it is assumed that the interaction eigenstates do not agree with the mass ones. That is, an incoming proton can be viewed as a coherent sum of interaction eigenstates. During the collision process, parts of these eigenstates are absorbed to give rise to non-diffractive events. The remaining parts of the incoming wave function can be projected back on to a spectrum of possible masses for the outgoing object, including one component corresponding to elastic scattering. Actually the "diffraction" name comes from the close analogue with optics, where an opaque disk put in a beam of light absorbs part of the light but also generates a quantum mechanical diffraction pattern in the remaining light. Such a picture implies that diffractive and elastic collisions are more peripheral than non-diffractive ones. The same also holds in the Regge-language-related MPI framework to be discussed in the next subsection, so even of the models seemingly are unrelated, there are many common traits.

6.1.4 Other cross sections

Except for the absence of Coulomb elastic scattering, collisions involving (anti)neutrons are assumed to have the same cross sections as (anti)protons in PYTHIA, and this similarity appears supported by data [200]. Therefore all of the models above can be used for pn, p$\bar{\mathrm{n}}$, nn and n$\bar{\mathrm{n}}$.

For other hadron combinations, the only alternative beyond the user-defined option is an extension of the SaS/DL setup. It encompasses the following collision types.

- Combinations where $\sigma_{\mathrm{tot}}(s)$ were fitted by DL [200]: π^+p, π^-p, K^+p, K^-p, and γp.

- SaS extensions [207]: ρ^0p, ϕ^0p, J/ψp, $\rho^0\rho^0$, $\rho^0\phi^0$, ρ^0J/ψ, $\phi^0\phi^0$, ϕ^0J/ψ, and J/ψJ/ψ. Particles with identical flavour content are assumed to give identical cross sections. The prime example is π^0, ρ^0, and ω. The emphasis on the interactions of vector mesons is related to the SaS modelling of γp and $\gamma\gamma$ physics, where an important

aspect is that a real photon can fluctuate into a vector meson like ρ^0, ω, ϕ^0, or J/ψ, and interact as such most of the time (Vector Meson Dominance (VMD), see section 6.6 and section 6.7). Total $\gamma\gamma$ cross sections are also provided.

- Later extensions in the SaS/DL spirit [49]: K^0p, ηp, η'p, $D^{0,+}$p, D_s^+p, $B^{0,+}$p, B_s^0p, B_c^+p, Υp, Λp, Ξp, Ωp, Λ_cp, Ξ_cp, Ω_cp, Λ_bp, Ξ_bp, and Ω_cp. Isospin symmetry is used to equate the cross sections of closely related particles, *e.g.* n with p and $\Sigma^{+,0,-,*+,*0,*-}$ with Λ. For the baryon-baryon processes, the corresponding baryon-antibaryon ones are also implemented. The purpose of these extensions is to allow the tracing of the evolution of cascades in matter, also in collisions with nuclei, meaning that essentially all hadronic collisions with protons or neutrons have to be included.

- As a final case, a $\sigma_{\text{tot}}^{\mathbb{P}\text{p}}(s)$ is defined for pomerons, but more for model studies of diffractive systems at a given mass than for comparisons with data.

In summary, by suitably mapping a particle onto one of equivalent flavour content, the possibilities above cover a fair fraction of all possible hadron collisions. The main exceptions are those involving baryons with more than one charm or bottom quark, and (most) collisions between two short-lived particles. Should the need arise, further extensions along the same lines would be possible for these cases.

It should be clear from the onset that the accuracy expected for these cross sections cannot compare with the pp and $\overline{\text{p}}$p ones. As a rule of thumb, the rarer the particle, the more uncertain the assumptions that went into deriving related cross sections. For many applications, notably the evolution of a cascade in matter, it is the average collision rates that count rather than the individual ones, however, one may assume that it should still work out reasonably well.

The starting point in all these total cross sections is the pomeron plus reggeon ansatz of eq. (239). The X^{AB} pomeron term strength appears to obey the Additive Quark Model (AQM) rule [208, 209], *i.e.* be proportional to the number n_i of valence quarks in each hadron, but with a reduced contribution for strange and heavier quarks. Thus we have made the ansatz that $X^{AB} \propto n_{\text{eff}}^A n_{\text{eff}}^B$, with

$$n_{\text{eff}} = n_{\text{d}} + n_{\text{u}} + 0.6 n_{\text{s}} + 0.2 n_{\text{c}} + 0.07 n_{\text{b}} . \tag{245}$$

The prefactors for heavier quarks have been assumed roughly inversely proportional to their respective constituent quark masses, which could be viewed as a reflection of a reduced size of their spatial wave functions.

The modelling of the Y^{AB} reggeon factors is considerably less systematic, since typically several reggeon trajectories may contribute. The mix of charge-even and charge-odd contributions gives $Y^{\overline{AB}} \neq Y^{AB}$, while $X^{\overline{AB}} = X^{AB}$. For baryon collisions $Y^{\overline{AB}} > Y^{AB}$, which can be viewed as a reflection of possible contributions from q$\overline{\text{q}}$ annihilation graphs. This is supported by the observation that $Y^{\phi\text{p}} \approx 0$, consistent with the OZI rule [210–212], and we assume that this suppression of couplings between light u/d quarks and s quarks extends to c and b. Thus, for baryons, the reggeon Y^{AB} and $Y^{\overline{AB}}$ values are assumed proportional to the number of u/d quarks only, scaled separately from the Y^{pp} and $Y^{\overline{\text{p}}\text{p}}$ reference values. Thereby baryons with the same flavour content, or only differing by the relative composition of u and d quarks, are taken to be equivalent, *i.e.* $\sigma^{\Lambda\text{p}}(s) = \sigma^{\Sigma^+\text{p}}(s) = \sigma^{\Sigma^0\text{p}}(s) = \sigma^{\Sigma^-\text{p}}(s)$. Another simplification is that $\overline{\text{D}}/\overline{\text{B}}$ mesons are assigned the same cross sections as the respective D/B, taken to be some average.

The B_{el}^{AB} slope for hadronic collisions is defined as in eq. (242), with a universal α' but $b_{A,B}$ taken to be 1.4 for mesons and 2.3 for baryons, except that mesons made only out of c and b quarks are assumed to be more tightly bound and thus have lower values, in the spirit of the AQM factors. Given this, and assuming $\rho \approx 0$, the integrated elastic cross sections

are given by eq. (241). For photon interactions, only the VMD part is assumed to undergo "elastic" scatterings, so the fractions of fluctuations to ρ^0, ω, ϕ^0, or J/ψ are combined with the expressions for these respective states to scatter elastically.

Diffractive cross sections are again calculated using the SaS ansatz. It provides differential cross-section expressions for relevant collision processes, which are integrated numerically, with results suitably parameterized, including a special threshold-region form [213]. The ansatz for the pomeron-exchange hadronic form factor is taken to be $\beta_{A\mathbb{P}}(t) = \beta_{A\mathbb{P}}(0)\exp(b_A t)$. With suitable normalization this translates to $X^{AB} = \beta_{A\mathbb{P}}(0)\beta_{B\mathbb{P}}(0)$. Thus $\beta_{p\mathbb{P}}(0) = \sqrt{X^{pp}}$ and other $\beta_{A\mathbb{P}}(0) = X^{Ap}/\beta_{p\mathbb{P}}(0)$. The β numbers enter in the prefactor of single diffractive cross sections, *e.g.* $\sigma_{AB \to AX} \propto \beta_{A\mathbb{P}}^2(0)\beta_{B\mathbb{P}}(0) = X^{AB}\beta_{A\mathbb{P}}(0)$. This expression can be viewed as a consequence of the optical theorem, wherein an (semi)elastic cross section is related to the square of a corresponding total one. Neither side is elastic in double diffraction $AB \to X_1 X_2$, and therefore the rate is directly proportional to X^{AB}. For photons again only the VMD parts undergo diffractive scatterings.

The descriptions mentioned so far are intended for cross sections at high energies. Specifically, the original DL ansatz was tuned to data down to 6 GeV. At low energies, different descriptions are used, as outlined in the next subsection, most of which are not intended to be used much above 10 GeV. In cases where the full energy range from threshold upwards needs to be used, a smooth interpolation is therefore applied between the low- and high-energy descriptions. More precisely, the transition is linear in the range between

$$E_{\text{CM}}^{\text{begin}} = E_{\text{min}} + \max(0., m_A - m_p) + \max(0., m_B - m_p), \quad \text{and} \tag{246}$$

$$E_{\text{CM}}^{\text{end}} = E_{\text{CM}}^{\text{begin}} + \Delta E, \tag{247}$$

where E_{min} is 6 GeV and ΔE is 8 GeV by default.

6.1.5 Low-energy processes

At low energies (below ~ 10 GeV), the perturbative framework described in this section breaks down. In modern high-energy physics, experimental beam energies lie far above this threshold, but processes at these energies still have applications for example in hadronic rescattering (see section 7.4). PYTHIA provides a framework for simulating such low-energy collisions. This framework can be used explicitly by enabling `LowEnergyQCD:*` processes, and is used implicitly inside PYTHIA when rescattering is turned on. The following gives a summary of the available low-energy processes:

Elastic scattering $AB \to AB$ is implemented similarly to elastic scattering at high energies, except the cross section is calculated differently, as described below.

Diffractive scattering   (both single and double) is also similar to how it is implemented at high energies. Central diffractive (AXB) has a very small cross section at low energies, and is thus not implemented. In addition, at low energies the diffractive system can sometimes be viewed as a resonance excitation, for example pp $\to$ pΔ^+. In PYTHIA 8.3, these excitation processes are implemented only for nucleon-nucleon interactions.

Non-diffractive scattering   Works similarly in principle to high-energy non-diffractive interactions, but with extra steps to ensure the process does not reduce to an elastic scattering at energies very close to the threshold.

Annihilation processes   Baryon-antibaryon interactions where one or two quarks annihilate.

Resonance formation   A meson interacting with a baryon or another meson can form a resonance particle, *e.g.* $p\pi^+ \to \Delta^{++}$ or $\pi^+\pi^- \to \rho^0$.

While several of these processes correspond to similar high-energy processes, their cross sections are in most cases calculated differently, as perturbative calculations cannot be used at these energies. Only a short overview of how the cross sections are calculated is given here, and the reader is referred to [213] for further details.

When PDG data is available[5], total cross sections are calculated using parameterizations or by fitting to data. The HPR_1R_2 parameterization is used when available, as is the case for *e.g.* nucleon-nucleon interactions [214]. For baryon-antibaryon interactions, a parameterization due to UrQMD is used [215]. $\pi\,\pi$ and $\pi\,K$ interactions use parameterizations by Pelaez *et al.* [216–218]. For other processes involving mesons, if the pair can form resonances, the total cross section is calculated by summing the contributions from each resonance, possibly also adding an elastic contribution. While these cases describe the most common processes, there is also a large set that is not covered. For these remaining processes, the total cross section is calculated using the additive quark model (AQM) [208,209] with small modifications introduced to also include charm and bottom quarks [213]. Specifically, the total AQM cross section is given by

$$\sigma_{\text{AQM}}^{AB} = (40 \text{ mb}) \frac{n_{\text{eff}}^A}{3} \frac{n_{\text{eff}}^B}{3}, \tag{248}$$

where n_{eff} is the "effective" number of quarks in each hadron, given by eq. (245). With this, low-energy processes are available for all possible hadron-hadron types.

In our description, we define elastic interactions as processes where the hadrons exchange momenta without ever changing their types, *e.g.* through a pomeron exchange. We do not include for example "pseudo-elastic" scattering through a resonance, $AB \to R \to AB$. Note that this distinction usually cannot be made experimentally, so one often considers a process elastic as long as the outgoing hadrons are of the same type as the incoming ones. For nucleon-nucleon and nucleon-pion interactions, the elastic cross section is found by fitting to data below 5 GeV [214], and by using the CERN/HERA parameterization above 5 GeV [219]. Elastic cross sections for baryon-antibaryon interactions are calculated using another parameterization by UrQMD [215], and for $\pi\,\pi$ and $\pi\,K$, we use parameterizations by Pelaez *et al.* [216–218]. Other cross sections are given by an elastic AQM-style parameterization. The angular distribution of the outgoing hadrons is the same as for the high-energy case (section 6.1.2).

Diffractive cross sections are calculated using the SaS model [198,207], with two modifications. First, the basic model is designed to deal with processes involving only p, $\bar{\text{p}}$, π, ρ, ω, and ϕ hadrons. In the low-energy framework, the generic case is calculated by replacing each incoming hadron by the most similar among these particles (*e.g.* treating each baryon as a proton), then rescaling the calculated cross section by the appropriate AQM factor. The second modification is due to the fact that the basic SaS model is intended for collision energies above 10 GeV. This is compensated for by multiplying by an *ad hoc* factor below 10 GeV. At low energies, diffractive processes can lead to the formation of explicit resonances, *e.g.* $pp \to p\Delta^+$. This is implemented in PYTHIA 8.3 only for nucleon-nucleon interactions, using the description by UrQMD [215].

Non-diffractive cross sections are calculated by subtracting all other partial cross sections from the total cross section. One important difference between non-diffractive interactions at low and high energies is that at low energies, the hadronization process might sometimes produce a hadron pair that is the same as the incoming pair, essentially reducing the interaction to an elastic process ($AB \to X_1X_2 \to AB$). This is a problem in cases where the calculated elastic cross section has already been adjusted to fit data. Several steps are taken to ensure

---

[5]https://pdg.lbl.gov/2018/hadronic-xsections/hadron.html [214]

that this does not give unexpected contributions to the elastic cross section, and are outlined in section 7.1.6.

Annihilation processes in our framework refer to baryon-antibaryon interactions where one or two quark-antiquark pairs are annihilated. Strings are drawn between the remaining quark-antiquark pairs, and hadronize to form outgoing hadrons. The cross section for annihilation in $p\bar{p}$ is given by a parameterization by Koch and Dover [220],

$$\sigma_{\text{ann}} = 120 \frac{s_0}{s} \left(\frac{A^2 s_0}{(s - s_0)^2 + A^2 s_0} + 0.6 \right), \tag{249}$$

where $s_0 = 4m_{\text{p}}^2$ and $A = 0.05$ GeV. The cross section for other $B\bar{B}$ interactions is found by rescaling this value by an AQM factor. The only exception is when the quark-contents make annihilation impossible, *e.g.* like in a $\Delta^{++} + \overline{\Sigma}$ interaction, in which case the annihilation cross section is set to zero.

Finally, resonance production refers to processes where the two hadrons combine to form one resonance particle. The cross section for the process $AB \to R$ is given by a non-relativistic Breit–Wigner [214],

$$\sigma_{AB \to R} = \frac{\pi}{p_{\text{CM}}^2} \frac{(2S_R + 1)}{(2S_A + 1)(2S_B + 1)} \frac{\Gamma_{R \to AB} \Gamma_R}{(m_R - \sqrt{s})^2 + \frac{1}{4}\Gamma_R^2}, \tag{250}$$

where S is the spin of each particle, p_{CM} is the CM momentum of the incoming particles, $\Gamma_{R \to AB}$ is the mass-dependent partial width of the decay $R \to AB$, and Γ_R is the mass-dependent total width of R. The full list of implemented resonances is given in [213]. For $\pi\pi$ and πK where the total cross section is calculated using the parameterization by Pelaez *et al.*, the partial cross sections are rescaled to ensure their sum equals the total cross section.

6.2 Multiparton interactions basics

Hadrons are composite objects. A proton consists of three valence quarks, plus countless gluons and sea quarks. When two hadrons collide there is a possibility for several parton pairs to collide — multiparton interactions (MPIs). Processes with exactly two parton pairs, double parton scattering (DPS), was proposed in the early days of QCD, but then viewed as a rare perturbative process [221, 222]. Regge–Gribov theory, on the other hand, allowed for events with multiple cut pomerons, *i.e.* several "strings" crossing from one rapidity end of the event to the other, each generating its sequence of low-$p_\perp$ hadrons [223]. The PYTHIA philosophy for the first time introduced a merger and extension of these two approaches [224]. In it, semiperturbative MPIs both generate multiple minijets, that contribute to the $p_\perp$ flow, and multiple colour connections between the beam remnants, that leads to events with higher multiplicity. This picture is now generally accepted in its essentials. An overview of MPI theory and phenomenology can be found in [225], with the PYTHIA perspective described in [226], with many further references. See also the `online manual` under the "Multiparton Interactions" heading.

6.2.1 The perturbative cross section

The $p_\perp$-differential perturbative QCD $2 \to 2$ cross section can, to leading order, be written as

$$\frac{d\sigma}{dp_\perp^2} = \sum_{i,j,k} \iiint f_i(x_1, Q^2) f_j(x_2, Q^2) \frac{d\hat{\sigma}_{ij}^k}{d\hat{t}} \delta\left(p_\perp^2 - \frac{\hat{t}\hat{u}}{\hat{s}} \right) dx_1 \, dx_2 \, d\hat{t}, \tag{251}$$

with $Q^2 = p_\perp^2$ as factorization and renormalization scale, partons assumed massless, and k running over processes with the same initial state but different final states (*cf.* eq. (37) and
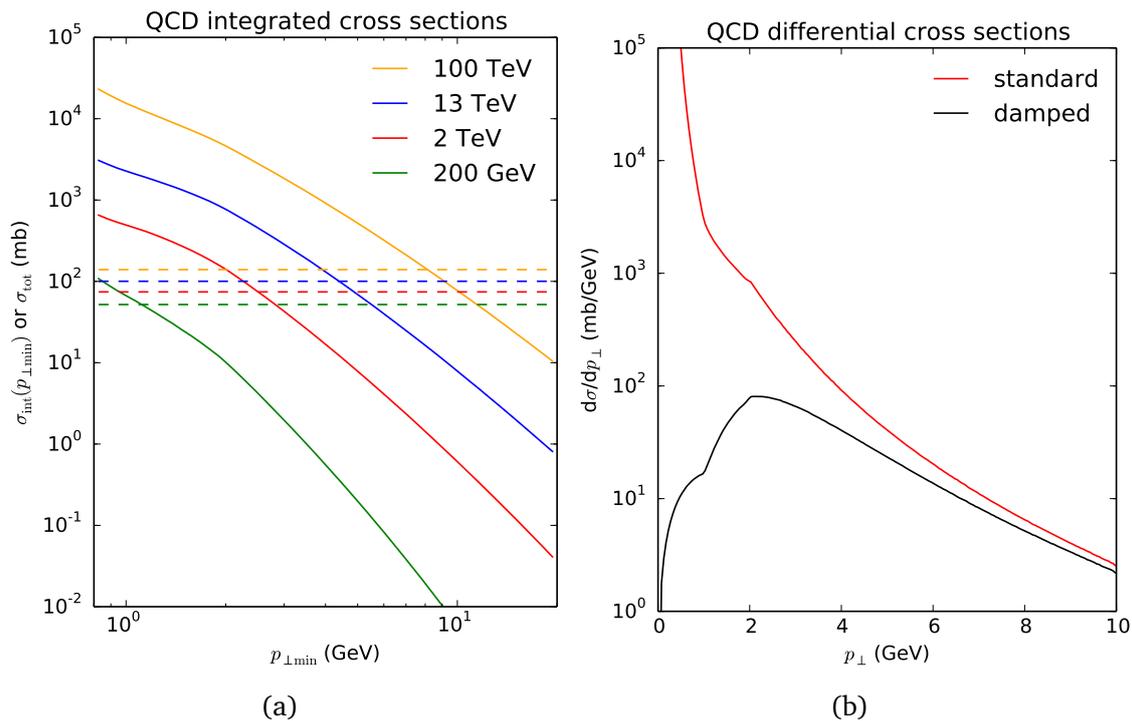
Figure 10: (a) Integrated standard $2 \to 2$ QCD cross section as a function of the lower cutoff $p_{\perp\min}$ for pp collisions at 200 GeV, 2 TeV, 13 TeV and 100 TeV, respectively. Horizontal dashed lines give the total cross section at their respective energy. (b) Differential $2 \to 2$ QCD cross section at 13 TeV, as obtained in standard perturbation theory, and after multiplication by the damping factor eq. (253). Minor breaks in slopes come from transitions, notably the freeze of PDFs below 1 GeV. Results have been obtained for the default PYTHIA 8.3 setup, and details depend *e.g.* on the choice of PDF set.

eq. (42)). The partonic cross section $\mathrm{d}\hat{\sigma}/\mathrm{d}\hat{t}$ is dominated by t-channel gluon exchange, *i.e.* $qq' \to qq'$, $qg \to qg$ and $gg \to gg$. (Including those u-channel graphs that easily can be relabelled into t-channel ones.) This cross section has an approximate behaviour

$$\frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}\hat{t}} \propto \frac{\alpha_s^2(Q^2)}{\hat{t}^2} \quad \Rightarrow \quad \frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}p_\perp^2} \propto \frac{\alpha_s^2(p_\perp^2)}{p_\perp^4}. \tag{252}$$

Evidently this cross section is divergent in the limit $p_\perp \to 0$, as shown in fig. 10. The integrated $2 \to 2$ cross section above some $p_{\perp\min}$ scale, $\sigma_{\mathrm{int}}(p_{\perp\min})$, is increasing with the pp collision energy. But, taking $p_{\perp\min} = 1$ GeV as a scale where perturbation theory would be expected to hold, already at a collision energy of 200 GeV, the σ_{int} value exceeds the total pp cross section σ_{tot} at this energy.

A further aspect is that σ_{tot} is subdivided into different components, as already discussed, and the $2 \to 2$ partonic interactions primarily occur within the non-diffractive one, which is what we will assume next. They are absent in elastic scattering and low-mass diffraction, while they can occur in high-mass diffraction. This is a small fraction of the total cross section, however, so to first approximation we may neglect it. Later on we will correct the picture.

Putting it together, one finds that $\sigma_{\mathrm{int}}(p_{\perp\min})$ is around 60 mb for $p_{\perp\min} \approx 5$ GeV at LHC energies, which is also the order of the non-diffractive pp cross section σ_{nd}. Going to lower $p_{\perp\min}$ scales the cross section rapidly explodes, $\sigma_{\mathrm{int}}(2\text{ GeV}) \approx 1000$ mb $\approx 15\,\sigma_{\mathrm{nd}}$. In the context of MPIs, this is not as bad as it may sound, since we may interpret the ratio $\sigma_{\mathrm{int}}(p_{\perp\min})/\sigma_{\mathrm{nd}}$ as

the average number of MPIs above the $p_{\perp\text{min}}$ scale that occur in a non-diffractive collision. Nevertheless, an infinity of MPIs in the $p_{\perp\text{min}} \to 0$ limit is not attractive.

A damping of the cross section at low $p_\perp$ can be viewed as a consequence of colour screening: in the $p_\perp \to 0$ limit a hypothetical exchanged gluon would not resolve individual partons but only (attempt to) couple to the vanishing net colour charge of the hadron. By contrast, traditional perturbation theory is based on the assumption of asymptotically free incoming and outgoing partons. To be specific, a multiplicative damping factor

$$\left(\frac{\alpha_s(p_{\perp 0}^2 + p_\perp^2)}{\alpha_s(p_\perp^2)} \frac{p_\perp^2}{p_{\perp 0}^2 + p_\perp^2} \right)^2 , \tag{253}$$

is introduced, with $p_{\perp 0}$ a free parameter. This means a modification to eq. (252)

$$\frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}p_\perp^2} \sim \frac{\alpha_s^2(p_\perp^2)}{p_\perp^4} \longrightarrow \frac{\alpha_s^2(p_{\perp 0}^2 + p_\perp^2)}{(p_{\perp 0}^2 + p_\perp^2)^2} , \tag{254}$$

which is finite in the limit $p_\perp \to 0$, *cf.* fig. 10b.

The $p_{\perp 0}$ value is not provided from first principles, although suggestions have been made to equate it with the saturation scale Q_s in colour glass condensate models [227, 228]. Fits to pp/p$\bar{\text{p}}$ data give a result that increases with energy, by default like

$$p_{\perp 0}(E_{\text{CM}}) = (2.28 \text{ GeV}) \left(\frac{E_{\text{CM}}}{7 \text{ TeV}} \right)^{0.215} , \tag{255}$$

but alternatively a logarithmic rise could be assumed. It should be noted that results are sensitive to the choice of PDF set, and especially to the low-x behaviour of the gluon distribution at small Q^2. The numbers are for the default NNPDF2.3 QCD+QED LO $\alpha_s(M_Z) = 0.130$ set [229]. The choice of an LO PDF is deliberate, since the description of partonic collisions is also an LO one, but in particular since NLO PDFs tend to become unphysical at small x and Q^2. This is why PYTHIA offers the possibility to use two different sets of PDFs, one for the hard processes, where these kinematic regions are not accessed, and one for MPIs and showers, where often they are.

The range of x values that can be accessed by MPI in PYTHIA is illustrated by the thick black lines in fig. 11, for hadronic CM energies ranging from 10 GeV (at the left-hand edge of the plot) to 100 TeV (at the right-hand edge). The shaded area emphasizes the region of low $x \leq 10^{-4}$ in which current PDFs are uncertain by a factor two or more. The red dashed line indicates the solution to $x^2 s = 4p_{\perp 0}^2$, for the default form of $p_{\perp 0}(E_{\text{CM}})$ given by eq. (255). Any partonic collision with $\hat{p}_\perp \sim p_{\perp 0}$ will involve at least one x value below this line. Thus, especially at LHC energies and beyond, it is important to keep in mind that the effective MPI cross section (and hence any observables derived from it) around $\hat{p}_\perp \sim p_{\perp 0}$ really depends on the *combination* of $p_{\perp 0}$ and the shape of the low-x PDF parameterization. Since the latter can change drastically between different PDF sets, any "tuned" values of $p_{\perp 0}$ should be considered valid only for the PDF set they were obtained with.

6.2.2 The impact-parameter model

A hadron is characterized not only by its longitudinal structure, as encoded in the PDFs, but also by its transverse one. That is, the "impact parameter" plane overlap of partons in the two hadrons influences the possible collisions. The hadrons are Lorentz contracted to pancake shapes in high-energy collisions, such as the LHC, so the partons can be considered as frozen during the short collision time.
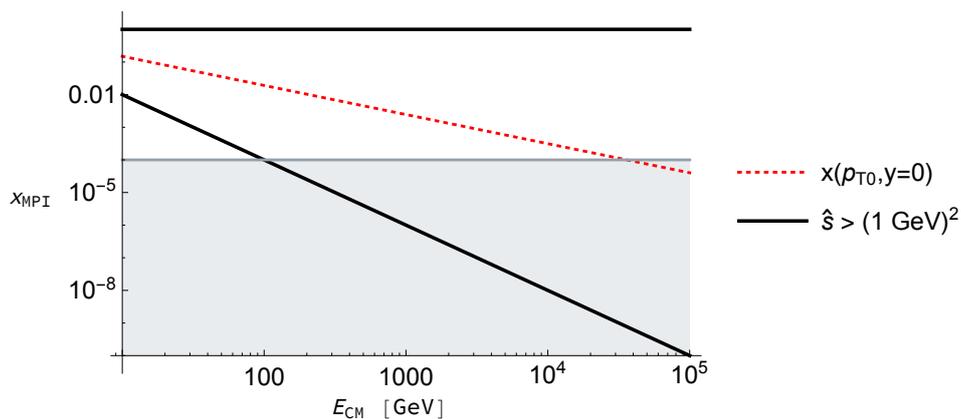
Figure 11: Range of x values accessible to MPI in PYTHIA, for $10\,\text{GeV} < E_{\text{CM}} < 100\,\text{TeV}$. Scatterings at $\hat{p}_\perp \sim p_{\perp 0}$ will involve at least one x fraction below the red dashed line. Grey shading highlights the low-x extrapolation region $x < 10^{-4}$ in which current PDFs are uncertain by a factor two or more.

As a first approximation we will assume a common spatial distribution $\rho(\mathbf{x})\,\text{d}^3 x = \rho(r)\,\text{d}^3 x$ for all parton types and momenta in a hadron. In the collision between two hadrons, passing by at an impact parameter b, the overlap between the two distributions is then given by

$$
\widetilde{\mathcal{O}}(b) = \iint \text{d}^3 x\, \text{d}t\, \rho_{\text{boosted}}\left(x - \frac{b}{2}, y, z - vt\right) \rho_{\text{boosted}}\left(x + \frac{b}{2}, y, z + vt\right)
$$
$$
\propto \iint \text{d}^3 x\, \text{d}t\, \rho(x, y, z)\, \rho(x, y, z - \sqrt{b^2 + t^2}), \tag{256}
$$

where the second line is obtained by suitable scale changes.

A few different ρ distributions have studied and made available as options. Using Gaussian distributions is especially convenient, since the convolution then becomes trivial. However, a single Gaussian does not give a good enough description of the data, and a better description is obtained with a sum of two Gaussians, with a small core region embedded in a larger hadron. This can be viewed as a manifestation of the "hot spot" concept [230, 231], wherein partons may tend to cluster in a few small regions, typically associated with the three valence quarks, as a consequence of partons cascading from them. Another alternative, that is currently the default, is a one-parameter shape

$$
\widetilde{\mathcal{O}}(b) \propto \exp\left(-b^d\right), \tag{257}
$$

where $d < 2$ gives more fluctuations than a Gaussian and $d > 2$ less. The default value is $d = 1.85$, i.e. slightly more peaked than a Gaussian. Note that the expression is for the overlap, not for the individual hadrons, for which no related simple analytic form is available.

It is now assumed that the interaction rate, to first approximation, is proportional to the overlap

$$
\langle \widetilde{n}_{\text{MPI}}(b) \rangle = k\, \widetilde{\mathcal{O}}(b). \tag{258}
$$

Interactions are assumed to occur independently of each other for a given b, to first approximation, which leads to a Poissonian number distribution. Zero interactions means that the hadrons pass each other without interacting. The $\widetilde{n}_{\text{MPI}}(b) \geq 1$ interaction probability therefore is

$$
\mathcal{P}_{\text{int}}(b) = 1 - \exp\left(-\langle \widetilde{n}_{\text{MPI}}(b) \rangle\right) = 1 - \exp\left(-k\, \widetilde{\mathcal{O}}(b)\right). \tag{259}
$$

We notice that $k\widetilde{\mathcal{O}}(b)$ is essentially the same as the eikonal $\Omega(s,b) = 2\,\mathrm{Im}\chi(s,b)$ of optical models [232–235], but split into one piece $\widetilde{\mathcal{O}}(b)$ that is purely geometrical and one $k = k(s)$ that carries the information on the parton-parton interaction cross section.

Simple algebra shows that the average number of interactions in events, *i.e.* hadronic passes with $n_{\mathrm{MPI}} \geq 1$, is given by

$$\langle n \rangle = \frac{\int k\,\widetilde{\mathcal{O}}(b)\,\mathrm{d}^2 b}{\int \mathcal{P}_{\mathrm{int}}(b)\,\mathrm{d}^2 b} = k\langle\widetilde{\mathcal{O}}\rangle = \frac{1}{\sigma_{\mathrm{nd}}} \int_0^{s/4} \frac{\mathrm{d}\sigma}{\mathrm{d}p_\perp^2}\,\mathrm{d}p_\perp^2 , \qquad (260)$$

which fixes the absolute value of k (numerically). We have also taken the occasion to introduce $\langle\widetilde{\mathcal{O}}\rangle$ as the average overlap. Hence $\widetilde{\mathcal{O}}(b)/\langle\widetilde{\mathcal{O}}\rangle$ represents the enhancement at small b and depletion at large b.

So far, we have assumed the transverse b-space profile to be decoupled from the longitudinal x one. This is not the expected behaviour, because low-x partons in a hadron should diffuse out towards larger r during the evolution down from higher-x ones [236]. Additionally, if $r = 0$ is defined as the centre of energy of a hadron, then by definition a parton with $x \to 1$ also implies $r \to 0$. In this spirit, there is a non-default PYTHIA option with correlated x and r [237]. It does not explicitly trace the evolution of cascades in x, but assumes that the r distribution of partons at any x can be described by a simple Gaussian, but with an x-dependent width:

$$\rho(r,x) \propto \frac{1}{a^3(x)} \exp\left(-\frac{r^2}{a^2(x)}\right), \quad \text{with} \quad a(x) = a_0\left(1 + a_1 \ln\frac{1}{x}\right), \qquad (261)$$

where a_0 and a_1 are free parameters to be determined. The overlap is then given by

$$\widetilde{\mathcal{O}}(b, x_1, x_2) = \frac{1}{\pi}\,\frac{1}{a^2(x_1) + a^2(x_2)} \exp\left(-\frac{b^2}{a^2(x_1) + a^2(x_2)}\right). \qquad (262)$$

In principle one could argue that also a third length scale should be included, related to the transverse distance the exchanged propagator particle, normally a gluon, could travel. This distance should be made dependent on the $p_\perp$ scale of the interaction. For simplicity, this further complication is not considered but, a finite effective radius is allowed also for $x \to 1$. The generation of events is more complicated with an x-dependent overlap, but largely involves the same basic principles. Until now, there is no evidence that this option provides a better description of data than the default, unfortunately.

6.2.3 The generation sequence

To introduce the MPI generation algorithm, leave aside the impact-parameter issue for a moment. The probability to have an MPI at a given $p_\perp$ in a non-diffractive event is then given by $(1/\sigma_{\mathrm{nd}})\mathrm{d}\sigma/\mathrm{d}p_\perp$. If interactions occur independently of each other, the number of MPIs would be distributed according to a Poissonian, with the zero suppressed. There are a few ways to generate such a Poissonian.

The PYTHIA approach is inspired by the parton-shower paradigm. The generation of consecutive MPIs is formulated as an evolution downwards in $p_\perp$, resulting in a sequence of n interactions with $\sqrt{s}/2 > p_{\perp 1} > p_{\perp 2} > \cdots > p_{\perp n} > 0$. The probability distribution for $p_{\perp 1}$ becomes

$$\frac{\mathrm{d}\mathcal{P}}{\mathrm{d}p_{\perp 1}} = \frac{1}{\sigma_{\mathrm{nd}}}\frac{\mathrm{d}\sigma}{\mathrm{d}p_{\perp 1}} \exp\left(-\int_{p_{\perp 1}}^{\sqrt{s}/2} \frac{1}{\sigma_{\mathrm{nd}}}\frac{\mathrm{d}\sigma}{\mathrm{d}p_\perp'}\,\mathrm{d}p_\perp'\right). \qquad (263)$$

Here the naive probability is corrected by an exponential factor expressing that there must not be any interaction in the range between $\sqrt{s}/2$ and $p_{\perp 1}$ for $p_{\perp 1}$ to be the hardest interaction. The procedure can be iterated, to give

$$\frac{\mathrm{d}\mathcal{P}}{\mathrm{d}p_{\perp i}} = \frac{1}{\sigma_{\mathrm{nd}}} \frac{\mathrm{d}\sigma}{\mathrm{d}p_{\perp i}} \exp\left(-\int_{p_{\perp i}}^{p_{\perp i-1}} \frac{1}{\sigma_{\mathrm{nd}}} \frac{\mathrm{d}\sigma}{\mathrm{d}p'_{\perp}} \mathrm{d}p'_{\perp}\right). \tag{264}$$

The exponential factors resemble Sudakov form factors of parton showers [51], or virtual corrections of "uncut pomerons" in the Regge–Gribov framework, and fills the same function of ensuring that probabilities are bounded by unity. We will use the Sudakov terminology to stress this similarity. Summing up the probability for a scattering at a given $p_{\perp}$ scale to happen at any step of the generation chain gives back $(1/\sigma_{\mathrm{nd}})\,\mathrm{d}\sigma/\mathrm{d}p_{\perp}$, and the number of interactions above any $p_{\perp}$ is a Poissonian with an average of $\sigma_{\mathrm{int}}(p_{\perp})/\sigma_{\mathrm{nd}}$, as it should. The downwards evolution in $p_{\perp}$ is handled by using the veto algorithm, like for showers. If no MPIs are generated in the evolution, a sequence is rejected and a new try made.

When the impact-parameter variability is to be included as well, eq. (263) generalizes to

$$\frac{\mathrm{d}\mathcal{P}}{\mathrm{d}^2 b\,\mathrm{d}p_{\perp 1}} = \frac{\widetilde{\mathcal{O}}(b)}{\langle\widetilde{\mathcal{O}}\rangle} \frac{1}{\sigma_{\mathrm{nd}}} \frac{\mathrm{d}\sigma}{\mathrm{d}p_{\perp 1}} \exp\left(-\frac{\widetilde{\mathcal{O}}(b)}{\langle\widetilde{\mathcal{O}}\rangle} \int_{p_{\perp 1}}^{\sqrt{s}/2} \frac{1}{\sigma_{\mathrm{nd}}} \frac{\mathrm{d}\sigma}{\mathrm{d}p'_{\perp}} \mathrm{d}p'_{\perp}\right). \tag{265}$$

This expression can be integrated over $p_{\perp 1}$ to give eq. (259). Once b has been chosen, the selection is similar to that in eq. (263), except that there is now a factor $\widetilde{\mathcal{O}}(b)/\langle\widetilde{\mathcal{O}}\rangle$ multiplying the rate. The same factor enters in the extension of eq. (264), for the continued evolution, to

$$\frac{\mathrm{d}\mathcal{P}}{\mathrm{d}p_{\perp i}} = \frac{\widetilde{\mathcal{O}}(b)}{\langle\widetilde{\mathcal{O}}\rangle} \frac{1}{\sigma_{\mathrm{nd}}} \frac{\mathrm{d}\sigma}{\mathrm{d}p_{\perp i}} \exp\left(-\frac{\widetilde{\mathcal{O}}(b)}{\langle\widetilde{\mathcal{O}}\rangle} \int_{p_{\perp i}}^{p_{\perp i-1}} \frac{1}{\sigma_{\mathrm{nd}}} \frac{\mathrm{d}\sigma}{\mathrm{d}p'_{\perp}} \mathrm{d}p'_{\perp}\right). \tag{266}$$

The usefulness of the doubly differential expression in eq. (265) is not so apparent in the generation of an inclusive non-diffractive event sample, where $p_{\perp 1}$ can be integrated out before selecting b. But it gives important insights, especially since the MPI machinery is also intended to be used to generate the underlying event associated with other processes. Assume *e.g.* that we want to produce a hard jet sample, *i.e.* $p_{\perp 1} > p_{\perp \mathrm{min}}$. For a large $p_{\perp \mathrm{min}}$ the steep fall of $\mathrm{d}\sigma/\mathrm{d}p_{\perp}$ ensures that the argument of the exponent is tiny, and so the exponent itself is close to unity and can be neglected. The b and $p_{\perp 1}$ expressions then factorize. The former variable is selected proportional to $\widetilde{\mathcal{O}}(b)$, while the latter is selected according to the conventional differential cross section. Since $\widetilde{\mathcal{O}}(b)$ is more peaked at small b than $\mathcal{P}_{\mathrm{int}}(b)$, it means that hard processes are selected at more central b values than inclusive non-diffractive events. The physics is quite clear: the probability to obtain a hard collision is proportional to the full parton-parton collision rate, $\langle\widetilde{n}_{\mathrm{MPI}}(b)\rangle \propto \widetilde{\mathcal{O}}(b)$, and so it is strongly peaked at small b, while already a single MPI is enough to obtain a non-diffractive event, and so that probability saturates at unity in $\mathcal{P}_{\mathrm{int}}(b)$. The consequence of picking a smaller b in hard processes is that the selection rate for subsequent MPIs, eq. (266), also is larger, thus giving a higher level of underlying activity than that of the full non-diffractive event sample, the "pedestal effect".

While the expression in eq. (265) provides for interpolation between hard and soft events, it is important to note that only the non-diffractive processes, *i.e.* the ones where the hardest interaction is selected by the MPI machinery, involve the full correlation. If one studies a hard process, be it hard QCD jets or something else, then in PYTHIA the selection of process kinematics is done with no reference to MPIs. It is only if and when, after the MPI machinery is invoked, that the $p_{\perp}$ scale of the hard process is used to select a b value that takes into account the Sudakov factor.

Therefore, in the study of hard QCD jets, one should not pick such a low $p_{\perp \mathrm{min}}$ that the Sudakov factor deviates appreciably from unity. In practice, this means that one should have

$p_{\perp\text{min}}$ at least above 20 GeV at LHC energies. If one wants to study jets below that scale, one can as well start out from the full non-diffractive sample. When a hard process is fed into the MPI machinery, however, b is chosen according to eq. (265) in full, *i.e.* including the Sudakov. That is, if by mistake one were to generate LHC jets at or below 10 GeV, the Sudakov would not be used in the $p_\perp$ selection, and thus the cross section would be overestimated, but it would be used in the b selection, and thereby provide the correct underlying-event activity.

So far, we have only considered $2 \rightarrow 2$ QCD processes in the MPI framework, but the list can be extended also to other ones. By default PYTHIA allows other $2 \rightarrow 2$ processes to be included in the Sudakov factor, and thereby also in the MPI generation: jet pairs via s-channel γ^* or t-channel $\gamma^*/Z/W^\pm$ exchange, events with one or two photons, or charmonium or bottomonium recoiling against a jet. Needless to say, these cross sections are much lower than the standard QCD ones, and therefore do not make much of a difference, but nevertheless help provide a richer non-diffractive or underlying-event structure.

Another issue is what upper limit to set for the selection of $p_{\perp 2}$. If studying QCD jets, the ordering $p_{\perp 1} > p_{\perp 2}$ is obvious; anything else would not reproduce the inclusive scattering cross section. But, if the hard process is single Z production, say, then this is not part of the MPI machinery, and so there is no double counting involved by allowing the underlying events to contain jets up to the kinematic limit. (The exception is if weak showers are switched on; then a hard QCD jet can emit a softer Z, and so such topologies could be double counted.) A few options are available, but the default strategy in PYTHIA is to split events into two types. If the final state of the hard process contains only (d, u, s, c or b) quarks, gluons, and photons then $p_{\perp\text{max}}$ is chosen to be the factorization scale for internal processes, and the `scale` value for external Les Houches input. If not, interactions are allowed to go all the way up to the kinematic limit.

6.2.4 Momentum and flavour conservation

As formulated so far, the same PDFs are used for all MPIs. This would allow more momentum to be taken out of a beam than there is, and also favour the repeated collisions of valence quarks that have already reacted. It is here that the ordering of the emissions becomes important. Standard PDFs can indeed be used for the first emission, which is the hardest one and therefore the one most visible and the one that standard PDFs have been tuned to describe. For subsequent emissions, the PDFs can gradually be modified to take into account the effects of the previous ones. An obvious modification is to rescale the x scale such that PDFs do not extend to higher values than left by the previous ones, *i.e.*

$$x_i < x_{i,\text{max}} \equiv X_i = 1 - \sum_{j=1}^{i-1} x_j,\tag{267}$$

but we will also want to consider flavour aspects. The beauty is that these successive modifications, that gradually let the PDFs diverge from the conventional ones, occur at falling $p_\perp$ scales, where individual MPIs become less easily studied, so imperfections do not give large effects. The consecutive reduction of remaining momentum also means that the n_{MPI} distribution, for a fixed b, will fall off faster than the assumed Poissonian. What does not change, fortunately, is the fraction of $n_{\text{MPI}} = 0$ events that have to be thrown away, because that is entirely determined by whether a first MPI can be generated with standard PDFs or not.

To extend the PDF framework, to include not only a simple x rescaling but also flavour counting, it is assumed that quark distributions can be split into a valence and a sea part. In cases where this is not explicit in the PDF parameterizations, it is assumed that the sea is flavour-antiflavour symmetric, so that one can write *e.g.*

$$u(x,Q^2) = u_{\text{val}}(x,Q^2) + u_{\text{sea}}(x,Q^2) = u_{\text{val}}(x,Q^2) + \overline{u}(x,Q^2).\tag{268}$$

The parameterized $u(x,Q^2)$ and $\bar{u}(x,Q^2)$ distributions can then be used to find the relative probability for a kicked-out u quark to be either valence or sea.

For valence quarks two effects should be considered. One is the reduction in content by previous MPIs: if a u valence quark has been kicked out of a proton then only one remains, and if two then none remain. In addition, the constraint from momentum conservation should be included. Together this gives

$$u_{i,\text{val}}(x,Q^2) = \frac{N_{u,\text{val,remain}}}{N_{u,\text{val,original}}} \frac{1}{X_i} u_{\text{val}}\left(\frac{x}{X_i}, Q^2\right), \tag{269}$$

for the u quark in the i'th MPI, and similarly for the d. The $1/X_i$ prefactor ensures that the u_i integrates to the remaining number of valence quarks. The momentum sum is also preserved, except for the downwards rescaling for each kicked-out valence quark. The latter is compensated by a uniform scaling up of the gluon and sea PDFs.

When a sea quark (or antiquark) q_{sea} is kicked out of a hadron, it must leave behind a corresponding antisea parton in the beam remnant, by flavour conservation, which can then participate in another interaction. We can call this a companion antiquark, $\bar{q}_{\text{cmp}}$. In the perturbative approximation the pair comes from a gluon branching $g \to q_{\text{sea}} + \bar{q}_{\text{cmp}}$. This branching often would not be in the perturbative regime, but we choose to make a perturbative ansatz, and also to neglect subsequent perturbative evolution of the q_{cmp} distribution. Even if approximate, this procedure should catch the key feature that a sea quark and its companion should not be expected too far apart in x. Given a selected x_{sea}, the distribution in $x = x_{\text{cmp}} = y - x_{\text{sea}}$ then is

$$q_{\text{cmp}}(x; x_{\text{sea}}) = C \int_0^1 g(y) P_{g \to q_{\text{sea}} \bar{q}_{\text{cmp}}}(z) \delta(x_{\text{sea}} - zy) \, \mathrm{d}z$$

$$= C \frac{g(x_{\text{sea}} + x)}{x_{\text{sea}} + x} P_{g \to q_{\text{sea}} \bar{q}_{\text{cmp}}}\left(\frac{x_{\text{sea}}}{x_{\text{sea}} + x}\right). \tag{270}$$

Here $P_{g \to q\bar{q}}(z)$ is the standard DGLAP branching kernel, $g(y)$ an approximate gluon PDF, and C gives an overall normalization of the companion distribution to unity. Furthermore, an X_i rescaling is necessary as for valence quarks. The addition of a companion quark does break the momentum sum rule, this times upwards, and so is compensated by a scaling down of the gluon and sea PDFs.

In summary, in the downwards evolution, the kinematic limit is respected by a rescaling of x. In addition, the number of remaining valence quarks and new companion quarks is properly normalized. Finally, the momentum sum is preserved by a scaling of gluon and (non-companion) sea quarks. It is interesting to note that the joint PDFs for the first two MPIs behave rather similarly to the Gaunt–Stirling DPS PDFs [238], whereas the PYTHIA approach currently is the only one that explicitly offers triple parton distributions and beyond.

6.2.5 Interleaved and intertwined evolution

So far we have only considered an MPI as a $2 \to 2$ process, but it should be associated with ISR and FSR showers. In particular, ISR needs to take momentum from the beams, and can also change the "original" flavour taken out of the beam during the backwards evolution. This implies a more intricate competition between the MPI systems than already outlined. If all MPIs are first considered, then their number will be maximized, whereas there may be little room left for ISR. If instead ISR is added to each MPI before proceeding to the next, then there will be less room left for MPIs.

Time ordering does not give any clear guidance what is the correct procedure. Incoming high-energy hadrons can be viewed as flat pancakes, such that all MPIs happen simultaneously

at the collision moment, while ISR stretches backwards in time from it, and FSR forwards. But we have no clean way of separating the hard interactions themselves from the virtual ISR cascades that "already" exist in the colliding hadrons.

Instead we choose the same guiding principle as we did when we originally decided to consider MPIs ordered in $p_\perp$: it is most important to get the hardest part of the story "right", and then one has to live with an increasing level of approximation for the softer steps. Since also showers are ordered in (some kind of) $p_\perp$, it is meaningful to choose $p_\perp$ as common evolution scale. Thus the scheme is characterized by one master formula

$$
\begin{aligned}
\frac{\mathrm{d}\mathcal{P}}{\mathrm{d}p_\perp} = {}& \left(\frac{\mathrm{d}\mathcal{P}_{\mathrm{MPI}}}{\mathrm{d}p_\perp} + \sum \frac{\mathrm{d}\mathcal{P}_{\mathrm{ISR}}}{\mathrm{d}p_\perp} + \sum \frac{\mathrm{d}\mathcal{P}_{\mathrm{FSR}}}{\mathrm{d}p_\perp} \right) \\
& \times \exp\left(- \int_{p_\perp}^{p_{\perp\max}} \left(\frac{\mathrm{d}\mathcal{P}_{\mathrm{MPI}}}{\mathrm{d}p'_\perp} + \sum \frac{\mathrm{d}\mathcal{P}_{\mathrm{ISR}}}{\mathrm{d}p'_\perp} + \sum \frac{\mathrm{d}\mathcal{P}_{\mathrm{FSR}}}{\mathrm{d}p'_\perp} \right) \mathrm{d}p'_\perp \right),
\end{aligned}
\tag{271}
$$

that probabilistically determines what the next step will be. Here the ISR sum runs over all incoming partons, two per already produced MPI, the FSR sum runs over all outgoing partons (or dipoles), and $p_{\perp\max}$ is the $p_\perp$ of the previous step. Starting from the hardest interaction, eq. (271) can be used repeatedly to construct a complete parton-level event. The flavour and momentum used by previous MPIs or shower branchings are book kept in accordance with the principles outlined previously, with a few straightforward extensions. For ISR, *e.g.* the x and flavour of the own MPI does not count as used up.

MPIs are not only related to each other by overall momentum and flavour conservation issues, but may be directly interacting with each other. Two such examples are joined interactions and partonic rescattering.

In the former, two partons participating in two separate MPIs may turn out to have a common ancestor when the backwards ISR evolution traces their prehistory. The joined interactions are well known in the context of the forwards evolution of multiparton densities [239, 240]. It can approximately be turned into a backwards evolution probability for a branching $a \to bc$

$$
\mathrm{d}\mathcal{P}_{bc}(x_b, x_c, Q^2) \simeq \frac{\mathrm{d}Q^2}{Q^2} \frac{\alpha_s}{2\pi} \frac{x_a f_a(x_a, Q^2)}{x_b f_b(x_b, Q^2) \, x_c f_c(x_c, Q^2)} z(1-z) P_{a\to bc}(z),
\tag{272}
$$

with $x_a = x_b + x_c$ and $z = x_b/(x_b + x_c)$. The main approximation is that the two-parton differential distribution has been been factorized as $f_{bc}^{(2)}(x_b, x_c, Q^2) \simeq f_b(x_b, Q^2) f_c(x_c, Q^2)$, to put the equation in terms of more familiar quantities.

Just like for the other processes considered, a form factor is given by integration over the relevant Q^2 range and exponentiation. Associating $Q \simeq p_\perp$, joined interactions can be included as a fourth term in eq. (271). But technical complications arise when the kinematics of joined branchings are reconstructed, notably in transverse momentum, and the code to overcome these was never written. One reason is that already the evolution itself showed that joined-interaction effects are small and tend to occur at low $p_\perp$ values [78].

The second intertwining possibility is rescattering, *i.e.* that a parton from one incoming hadron consecutively scatters against two or more partons from the other hadron. The simplest case, $3 \to 3$, *i.e.* one rescattering, has been well studied [241–243]. The conclusion is that it should be less important than two separated $2 \to 2$ processes: $3 \to 3$ and $2 \times (2 \to 2)$ contain the same number of vertices and propagators, but the latter wins by involving one parton density more. The exception could be large $p_\perp$ and x values, but there $2 \to n, n \geq 3$ QCD radiation anyway is expected to be the dominant source of multi-jet events.

For rescattering, a detailed implementation is available as an option in PYTHIA [244], as follows. In order to allow a rescattering then a scattered parton has to be put back into the

PDF, but now as a δ function. A hadron can therefore be characterized by a new PDF

$$f(x,Q^2) \to f_{\text{rescaled}}(x,Q^2) + \sum_i \delta(x - x_i) = f_{\text{un}}(x,Q^2) + f_\delta(x,Q^2), \tag{273}$$

where f_{un} represents the unscattered part of the hadron and f_δ the scattered one. The scattered partons have the same x values as originally picked, in the approximation that small-angle t-channel gluon exchange dominates, but more generally there will be shifts. The sum over delta functions runs over all partons that are available to rescatter, including outgoing states from hard or MPI processes and partons from ISR or FSR branchings. All the partons of this disturbed hadron can scatter, and so there is the possibility for an already extracted parton to scatter again. With the PDF written in this way, the MPI scattering rate can be seen as a sum of four terms, depending on whether the f_{un} or the f_δ is involved on either incoming side. Unfortunately, like for the joined interactions above, the kinematics become quite messy, specifically the propagation of recoils between systems that are partly intertwined but also partly separate.

A third and more dramatic intertwining possibility is that the perturbative cascades grip into each other. An example is the "swing" mechanism, whereby two dipoles in the initial state can reconnect colours, which is a key aspect of the DIPSY generator [245, 246]. An implementation exists in a branch of PYTHIA 8.3 [247], but not yet in the public version.

6.2.6 Spatial parton vertices

While setting spatial production vertices of unstable hadrons and leptons is a standard task (see section 8.1.3), the corresponding task for parton vertices in MPIs (as well as for beam remnants and parton shower) is not. The main issue to tackle is, that as the MPI and shower models are formulated in momentum space only, no obviously correct correlation with an impact-parameter picture exists. The plan is to further develop such an integrated framework, based on matching with dipole calculations on proton Fock states in impact-parameter space [247], but as such information is needed for string interactions (section 7.3) and hadronic rescattering (section 7.4), a basic framework is in place already now.

The basic framework includes four choices for the pp overlap region, from which vertices are sampled randomly. For all model choices, vertices of ISR and FSR partons are smeared relative to their mother by a Gaussian distribution, with a width of $\sigma_{\text{v}}/k_\perp$, where $k_\perp$ is the transverse momentum of the produced parton, and σ_{v} is a parameter to be set by the user.

The four possible choices for the overlap region are:

- The proton profile is a Lorentz-contracted ball of uniform density. This gives an almond-shaped overlap region, similar to heavy-ion collisions, favouring MPIs being displaced perpendicular to the collision plane. This option somewhat collides with impact-parameter selection in the MPI model, as it does not allow any interactions of the impact parameter to be larger than twice the hadron radius.

- The proton profile is a Lorentz-contracted three-dimensional Gaussian (motivated by the proton mass distribution), easily reduced to a two-dimensional one, as the z can be integrated out. The overlap region is taken as the product of the two displaced Gaussians, which is in itself a Gaussian.

- A variation of the above Gaussian scheme, but elongated by a factor $\sqrt{(1+\epsilon)/(1-\epsilon)}$, where ϵ is a parameter determining whether production should be favoured in the collision frame or out of the collision frame.

- Another variation of the Gaussian scheme, but with a modulation factor $1 + \epsilon + \cos(2\phi)$, and ϕ defined with respect to the collision plane.

It should be noted that the models for spatial parton vertices are at a very early stage of development, and subject to change in the future.

6.2.7 Other MPI aspects

There are several topics that concern MPIs, that will be described separately. One such is the issue of colour flow. The colours within each MPI, and its associated ISR and FSR, are initially assigned in the $N_c \to \infty$ limit. This implies that each parton taken out of a hadron, to go into a MPI, leaves its corresponding unique anticolours behind in the beam remnant. With many MPIs involved this gives an unrealistically complicated remnant, and so there is a machinery that attempts to associate an initial-parton colour from one MPI with an initial-parton anticolour from another MPI. Remaining colour lines attach to the remnant partons, see further the beam remnants description. This still allows colour lines to be drawn criss-cross in the event. Colour reconnection (CR) is a mechanism whereby these colour lines may be reconnected, typically in such a way that the total string length is reduced, further described in section 7.2.

In this section we have reasoned around MPIs in the non-diffractive component in hadron-hadron collisions, which is the prime, but not the only, application of the MPI framework. One extension is that photons have a resolved component, where they behave more-or-less like hadrons, and undergo MPIs in a similar manner. Another is that diffraction may be viewed as involving the collision of pomerons with hadrons or with each other, and that also pomerons can be associated with a hadronic structure that allows MPIs to occur. These aspects will be discussed further in their respective context.

A standard task for PYTHIA is to generate one predetermined hard process and then add underlying-event activity to that, which means that most of the time the additional MPI activity will be too soft to give explicitly visible jets. This means that generation efficiency will be low if one is interested in studies of double parton scattering. But, there is a possibility to request two hard scatterings in an event, each of a given type and within given kinematic ranges. While one of the two processes can be selected from the full range of possibilities, the other must be chosen from a list of a dozen process groups. This is not a fundamental limitation, but covers all that we could see a possible application for, and if need be the list could be extended. Furthermore, as a non-standard extension to the Les Houches Accord, it is also possible to feed in external events with two hard processes for further handling in PYTHIA. See section 3.14 for further details.

Since MPIs play such a key role for hadronic event properties, it is important to tune them as well as possible to describe minimum bias, *i.e.* predominantly non-diffractive, and underlying events alike. A number of settings and parameters are available to that end. Of special interest is the $p_{\perp 0}$ parameter, that directly influences important properties such as the multiplicity distribution. Finally, it is worth mentioning that the MPI component normally is the most time-consuming task of the PYTHIA initialization step. In order to prepare the Monte Carlo sampling of the differential cross section, it is necessary to find an upper envelope of it in the $(x_1, x_2, \hat{t})$ phase space. This envelope is based on multichannel sampling, where the relative importance of the channels should be optimized to allow a reasonably high sampling efficiency. The MPI cross section itself also needs to be integrated, as part of the $p_\perp$-evolution formalism. The initialization of non-diffractive events therefore may take around a second, *i.e.* almost two orders more than it takes to generate an LHC event afterwards. If one had to repeat the MPI initialization for each new event, this step would form a bottleneck. That would be the case in diffractive events, where the mass of the diffractive system varies from one event to the next. To this end, diffraction is initialized for a number of logarithmically evenly spaced mass values, and then parameters for intermediate masses are obtained by interpolation. If the incoming beams have varying energies, also non-diffractive events can be

set up for a range of collision energies. Thus initialization may take tens of seconds for the full set of inelastic processes, while the subsequent interpolation time is negligible compared to the event-generation one. If furthermore PYTHIA is initialized for multiple hadron types, the time needed becomes proportionately longer. An option therefore exists to save the MPI initialization data to a file, for reuse in subsequent runs, see section 9.6.2.

6.3 Beam remnants

What is left of a beam particle, after the partons initiating hard interactions and MPI have been removed from it (and showered), is called the beam remnant. By definition, the remnant itself does not participate in any momentum exchanges at scales larger than $\mathcal{O}(1\,\text{GeV})$. Hence, in PYTHIA, it is regarded as a purely non-perturbative object, which does not undergo a parton shower.

The general strategy is to add the minimal number of partons required to conserve the beam particle's quantum numbers (flavour, colour and baryon number), taking into account which valence and sea flavours have been scattered out of it. The remaining beam-particle momentum is then shared amongst those partons, as described below. Note that what is relevant to determining the remnant structure is not which partons initiated Born-level processes (or MPI) at the respective hard-process factorization scale(s), but instead the ones *after* initial-state radiation, at $Q \sim Q_{\text{cutoff}} \sim \mathcal{O}(1\,\text{GeV})$. For brevity, we henceforth refer to these low-scale partons as "initiator partons".

By default, also some "intrinsic transverse momentum" is added for the initiators and the remnant partons. Final momentum conservation is then ensured by rescaling the sampled momenta of the remnant partons appropriately. The procedure is discussed in more detail in [248] and outlined below for hadron beams and the more specialized cases of lepton and photon beams will be discussed in sections 6.5 to 6.7.

6.3.1 Flavour structure

The first step in beam remnant generation is to determine the number and flavours of the remnant partons. This begins by including the remaining valence quarks. For baryons, if two or more valence quarks are present, a randomly selected pair of these is turned into a diquark state. In this case, relative probabilities for different diquark spins are derived within the context of the non-relativistic **SU(6)** model, *i.e.* flavour **SU(3)**$_{\text{uds}}$ times spin **SU(2)**. For instance, a *ud* diquark in a proton remnant is 3/4 spin-0 and 1/4 spin-1, while a *uu* diquark always has spin-1. If the initiator was a gluon, then the remnant is a colour-octet object, which is split into a triplet and an antitriplet, again using **SU(6)** to determine relative weights. For a proton remnant, $P(u + ud_0) = \frac{1}{2}$, $P(u + ud_1) = \frac{1}{6}$, and $P(d + uu_1) = \frac{1}{3}$.

Otherwise, the valence flavours are unambiguous assuming that valence content has been fixed beforehand. As sea quarks are created in pairs, for all sea quarks that have taken out from the beam particle a companion quark with an opposite flavour and colour is added if such have not been already found during partonic evolution. If no other remnants are needed, a gluon (photon) is added to carry the momentum of the hadron (lepton) beam, otherwise no gluons are added as remnant partons unless required to balance for the colour structure. For DIS events, it is also possible to collapse two remnant partons directly into a colour-singlet hadron.

6.3.2 Colour structure

Since the incoming hadrons (or, more generally, the incoming beam particles) are colourless, the combined set of initiator and beam-remnant partons must be colourless, too. In the very

simplest cases, such as when the remnant consists of a single triplet and/or antitriplet colour, there is no ambiguity. But when there are several such charges, the assignment of colour flow in the remnant (roughly, which remnant-parton colours to associate with which initiator-parton colours) is inherently ambiguous, and there is no first-principles solution. PYTHIA contains two distinct models that address this ambiguity, called "old" and "new", based on the time they were developed and implemented. Currently, the "old" one is the default.

The old model [248] is motivated by the way colour flow is treated in parton showers, and extends this to the beam remnant, as follows. Starting from the simplest representation of the colour structure of the valence quarks in the incoming beam particle (a quark-antiquark pair for a meson and a three-quark junction structure for a baryon, simplified to a quark-diquark structure when possible), initiator gluons are attached in random order to one of the valence quarks (selected at random if there are several), and quark-antiquark pairs are added as if they came from gluon splittings. Thus this model captures the qualitative behaviour that is expected from leading-colour QCD.

The new model [249] is motivated by $\mathbf{SU(3)}$ colour algebra, and essentially extends the QCD-based colour-reconnection model to the beam remnant, as follows. First, the set of initiator partons is considered. An $\mathbf{SU(3)}$ product determines the possible overall multiplets that can be formed from those partons. If one assumes they are uncorrelated, the naive probability for the set to be in any of those multiplets would be given simply by state counting. A free parameter allows for the application of an (exponential) weighting factor favouring small multiplets over larger ones. This is intended as a way to mimic correlations due to possible saturation effects which are not otherwise explicitly represented in PYTHIA. Having selected a multiplet for the set of initiator partons, the beam-remnant colour configuration has to be the inverse of that, to conserve the colour-singlet nature of the beam particle. The minimum amount of gluons are added to the beam remnant in order to obtain this colour configuration.

6.3.3 Primordial $k_\perp$

As the hard processes and parton showers in PYTHIA are based on collinear factorization, only the longitudinal momenta are generated during the perturbative treatment. However, some transverse momentum of non-perturbative origin due to Fermi motion of partons inside a hadron is expected. Furthermore, studies on Z-boson transverse-momentum distributions have indicated that a significant amount of partonic $p_\perp$ is required to reproduce these distributions in hadron-hadron collisions. In PYTHIA such partonic transverse momentum is modelled with primordial $k_\perp$ that acts as a proxy for non-perturbative and possibly perturbative initial $p_\perp$.

In PYTHIA the primordial $k_\perp$ is generated from a two-dimensional Gaussian distribution. For hard-process initiators the width of the Gaussian is parameterized as

$$\sigma(Q) = \frac{\sigma_{\text{soft}}Q_{1/2} + \sigma_{\text{hard}}Q}{(Q_{1/2} + Q)} \frac{m}{(m + m_{1/2}y_{\text{damp}})}, \tag{274}$$

where Q is the renormalization scale for the hardest process and $p_\perp$ for subsequent MPIs and m the mass ($\sqrt{\hat{s}}$) of the system. The Q-dependent factor provides an interpolation between a soft scale set by parameter σ_{soft} and a hard scale, set by σ_{hard}, and $Q_{1/2}$ controls the midpoint between these two. The m-dependent factor on the right-hand side in turn provides damping for small-mass and/or large-rapidity systems. Such damping is introduced due to purely technical reasons so the controlling parameters $m_{1/2}$ and $y_{\text{damp}} = (\frac{E}{m})^{r_{\text{red}}}$, where r_{red} controls the of amount rapidity damping, should not have much influence on related observables. For the remnant partons not directly connected to any hard process, the width of the $k_\perp$-distribution is fixed by an another parameter σ_{remn} and does not depend on any scale related to hard

scattering or MPIs. After sampling the $k_\perp$ for each parton in the beam it is inevitable that the total transverse momentum of the beam becomes non-zero. To retrieve the original beam $p_\perp$, the $k_\perp$ of all partons will be rescaled with a common factor in such a way that the net four-momentum of the beam particles will be preserved.

6.3.4 Longitudinal momentum

In addition to the transverse momentum, the remnant partons should also carry the remaining longitudinal momentum of the beam particle, X. As a first step, a momentum fraction $x < X$ is sampled for each remnant parton. In case of valence quarks, the value is sampled according to $(1-x)^a/\sqrt{x}$, where the power a can be adjusted for each parton flavour. Such a distribution approximates the valence quark PDFs around the initial scale $\mathcal{O}(1\,\text{GeV})$ at which the remnants are constructed. For the remaining companion quarks, the momentum fraction is sampled from the distribution defined in eq. (270) which takes into account that the sea quarks are always created in pairs, by definition, from gluon splittings. Gluons (and photons) are only added as remnants if no valence or companion quarks are remaining in the beam. As only one of these will be added as a remnant, it will carry all the remaining beam particle's momentum X.

After the initial momentum fractions have been sampled for each remnant parton, these have be to rescaled to make sure that the total four-momentum is conserved in each event. As now both the initiator and the remnant partons carry also transverse momentum, the longitudinal-momentum fraction of the remnants cannot be simply rescaled with X but some momentum have to be shared between the two beams to balance the event, for details see [248, section 4.4]. In some special cases, such as DIS processes, only one remnant is required and no such balancing can be done. To account for momentum conservation, the final-state parton momenta are then boosted and rotated in such a way that the total four-momentum is conserved for the sampled remnant configuration.

6.4 Hadron-hadron collisions

In section 6.1 we introduced the main event types in hadron-hadron collisions, and how their total and differential cross sections are parameterized in PYTHIA 8.3. Elastic-scattering events are trivial to model, given the $d\sigma/dt$ cross-section expression; there are just two hadrons coming in and the same two coming out, with a momentum transfer t and a randomly-selected φ angle. See section 6.1.2 for the various options available for proton elastic-scattering cross sections, and section 6.1.4 for the less sophisticated expressions used for other hadrons. The subsequent test on MPIs and beam remnants are mainly concerned the non-diffractive component. It has the largest cross section, and especially it is the one where the bulk of hard processes occur, which makes it the most studied one experimentally. In this section we provide some further comments on this event class in section 6.4.1, but in particular describe additional aspects in the description of diffraction in sections 6.4.2 and 6.4.3.

6.4.1 Minimum-bias and related inclusive processes

The inelastic non-diffractive event type is often also called Minimum Bias (MB). Strictly speaking, however, MB refers to the smallest possible trigger bias that allows for the identification of non-empty events in a given experimental context. Depending on the detector acceptance, MB will typically also include contributions from processes that PYTHIA labels as diffractive. Thus, if the aim is to simulate an inclusive sample of "minimum bias" events, usually both diffractive and non-diffractive events must be included, and then subjected to the appropriate experimental trigger requirements.

Other, related, experimental terms are zero bias (*e.g.* based on a bunch-crossing timing trigger, including some *a priori* unknown fraction of genuinely empty events), pileup (essentially also zero bias except in cases where pileup contamination may affect trigger variables such as calorimeter energies), inelastic $\geq N$ events (inelastic events with at least N particles in some given fiducial region), and non-single-diffractive events (typically a "double-sided" MB trigger).

Related to this, note that the distinction between diffractive and non-diffractive processes is not without ambiguity. In experimental contexts, diffraction may be defined in terms of observable "rapidity gaps" with no particle production detected in specific region(s) of the detector, while in theoretical contexts processes that are classified as diffractive typically produce a whole spectrum of gaps with small ones suppressed but not excluded, see section 6.4.2. Conversely, events that are modelled as non-diffractive in origin may produce large rapidity gaps, due to fluctuations in the fragmentation process and/or if colour reconnections — see section 7.2 — are allowed to produce such gaps, and in the transition region there could even be quantum interference between the two categories (not modelled by PYTHIA). Thus, for any given application it is important to phrase experimental measurements in terms of clearly defined physical observables, and consider which MC processes are going to be able to contribute to those.

Usually hard processes, such as jet or gauge-boson production, are assumed to occur within the non-diffractive event class. This is not quite true, since it is possible also for diffractive topologies to contribute to hard cross sections, see section 6.4.3. That contribution typically is of the order of a per cent when modelled or measured experimentally, however, and is neglected by default. This means that the full parton distribution functions (PDFs) are associated with the non-diffractive component. They are used not only for the hard process itself but also for the associated MPI, ISR and FSR activity. See further section 3.12

6.4.2 Diffractive processes

Diffractive event topologies are illustrated in fig. 9 on p. 114, and the differential cross sections are described in section 6.1.3. The choice of diffractive mass(es) and t values sets the overall kinematics of the events, but does not describe the hadronization of the diffractive system. To this end, the Ingelman–Schlein approach is used [250], with details as described further in [199]. In this approach, a pomeron is viewed as a physical particle, akin to a glueball state, with an internal structure and notably with PDFs. Similarly, a reggeon is viewed as a mesonic state, but for the practical handling the two are not distinguished. Single diffraction therefore contains a pomeron-proton subcollision, double diffraction two such, and central diffraction a pomeron-pomeron subcollision. Each such subcollision is assumed to produce particles as in a normal inelastic non-diffractive hadron-hadron collision.

At high energies the modelling on the perturbative level is then given by the MPI machinery, augmented by ISR and FSR. There are a few issues that need to be clarified, however. Notably the MPI collision rate involves a combination of the pomeron-inside-proton flux with the parton-inside-pomeron PDF. What is measured, *e.g.* at HERA, is the convolution of the two, where the absolute normalization of each individually is not known. Historically, the flux normalization was specified, such that then the pomeron PDF does not have to obey the momentum sum rule. This may seem odd, but is in line with some theoretical arguments that the pomeron is not a real particle and therefore is not bound by such constraints. There are a dozen different pomeron PDFs that come with PYTHIA (plus three special-purpose ones), and most of these have a momentum sum of the order of 0.5. It is possible to scale them by a factor, to restore unit normalization. Whether that is done or not, the rescaling of remaining momentum for subsequent MPIs is done as for a normal hadron, however. That is, the normalization matters for the rate of MPI production, but not for the handling of those MPIs that

do occur.

Further, the ordinary non-diffractive MPI rate is related not only to PDFs but also to the normalization with the non-diffractive total cross section σ_{nd}, *cf.* eq. (260) and other MPI expressions. This is an unknown number from first principles, and with the same pomeron-flux-normalization uncertainty as the PDFs, so effectively it can be used to compensate for a non-unit momentum sum. The default value is 10 mb at a collision CM energy of 100 GeV, where it has been tuned (with default PDFs *etc.*) to produce about the same average charge multiplicity as ordinary pp non-diffractive collisions at the same energy. This value could be energy-dependent, *cf.* the pomeron term in eq. (239), but currently the default is a constant value.

Diffraction tends to be peripheral, *i.e.* occur at high-to-intermediate impact parameter for the two protons. That aspect is implicit in the modelling of diffractive cross sections. For the simulation of the pomeron-proton subcollision itself, however, it is rather the impact-parameter distribution of that particular subsystem that should be modelled. That is, it also involves the transverse coordinate-space shape of a pomeron wave function. The outcome of the convolution with a proton wave function could be a different shape than for non-diffractive events, and therefore it can be set separately. The default is a simple Gaussian, for lack of any relevant data. The $p_{\perp 0}$ scale is assumed the same as in non-diffractive events at the same collision energy, but also that is an assumption that could be questioned.

The diffractive mass spectrum extends down to the Δ^+ mass for pp collisions, and obviously a perturbative MPI description would not make sense at such low energies. Instead a separate low-mass description has been implemented. Up to 1 GeV above the hadron mass, the diffractive system is allowed to decay isotropically into a two-hadron state. Above that, a diffractively-excited hadron is modelled as if either a valence quark or a gluon is kicked out from it, along the collision axis with some "primordial $k_\perp$" smearing, *cf.* section 6.3.3.

In the former case this produces a simple string to the leftover remnant, in the latter it gives a hairpin arrangement where a string is stretched from one quark in the remnant, via the gluon, back to the rest of the remnant. The latter topology ought to dominate at higher mass M_X of the diffractive system. Therefore an approximate behaviour like

$$\frac{P_q}{P_g} = \frac{N}{M_X^p} \tag{275}$$

is assumed, with N ($= 5$ by default) and p ($= 1$) as free parameters, and M_X in GeV.

There is a smooth transition between the low-mass non-perturbative and the high-mass perturbative descriptions. The probability for applying the latter is given by [251]

$$P_{\mathrm{pert}} = 1 - \exp\left(-\frac{\max(0, M_X - m_{\min})}{m_{\mathrm{width}}}\right), \tag{276}$$

with $m_{\min}$ and m_{width} free parameters, both by default 10 GeV. Note how P_{pert} vanishes when below $m_{\min}$.

6.4.3 Hard diffraction

The model for hard diffraction is somewhat different from the soft (low- and high-mass) diffraction and it can be applied to any hard process, including *e.g.* high-$p_\perp$ jets and EW bosons. The starting point is again the Ingelman–Schlein picture where these interactions are mediated by a pomeron whose internal structure is given by the diffractive PDFs. It has been observed, however, that this factorization-based approach is broken as the predictions based on the diffractive PDFs determined in diffractive DIS overshoot the hard diffractive data in hadron-hadron collisions roughly by an order of magnitude [252,253]. In the PYTHIA framework this can be naturally explained by having several non-diffractive partonic interactions,

MPIs, in the same hadron-hadron collisions on top of the diffractive process. These may then produce particles that fill up the rapidity gap used to select the diffractive events leading to seemingly a suppressed diffractive cross section. The details of this *dynamical rapidity gap survival model* are presented in [254], together with several data comparisons, and are briefly outlined below.

After a hard process and its kinematics are sampled, the events of diffractive origin are first selected based on relative magnitude of the diffractive, $f_i^{\mathrm{p,D}}$, and non-diffractive, $f_i^{\mathrm{p,ND}}$, PDFs which together form the inclusive (the usual) hadronic PDFs

$$f_i^{\mathrm{p}}(x,Q^2) = f_i^{\mathrm{p,ND}}(x,Q^2) + f_i^{\mathrm{p,D}}(x,Q^2). \tag{277}$$

The diffractive part, in turn, can be defined as a convolution between the pomeron flux $f_{\mathbb{P}}^{\mathrm{p}}$ and pomeron PDF $f_i^{\mathbb{P}}$:

$$f_i^{\mathrm{p}}(x,Q^2) = \int_x^1 \frac{\mathrm{d}x_{\mathbb{P}}}{x_{\mathbb{P}}} f_{\mathbb{P}}^{\mathrm{p}}(x_{\mathbb{P}}) f_i^{\mathbb{P}}(x/x_{\mathbb{P}},Q^2), \tag{278}$$

which can be considered as parton-in-pomeron-in-proton PDFs, typically determined using diffractive DIS data from HERA. After this tentative selection of diffractive events corresponding to the Ingelman–Schlein approach, the pomeron kinematics are sampled and the event is processed further. The essence of the PYTHIA model is then to perform a full parton-level evolution for the original hadron-hadron system and to check whether any MPIs, that would render the event to a non-diffractive one, has occurred. This allows for the generation of a sample where only events without such additional interactions remain and the rapidity gap has survived. It is also possible not to perform such a check and obtain the purely factorization-based result that serves as a baseline for the expected cross-section suppression. Notice, however, that MPIs in the pomeron-hadron system are still allowed as these would not fill up the rapidity gap between the excited hadron and the pomeron remnants. Remarkably, this model relies solely on the MPI model in PYTHIA and does not require any further parameters tuned to data. Yet, it can qualitatively explain the order-of-magnitude difference between the purely factorization-based predictions and Tevatron and LHC data, and reproduces the latest CMS data for diffractive dijets [255] with a good precision. Only single diffraction is currently implemented, and if both beams have been found to emit pomerons, the diffractive side is selected randomly with equal probabilities. It is possible to consider pomeron emissions from one side only which can be useful for non-symmetric collisions.

6.5 Lepton-lepton collisions

Lepton colliders have a reputation for providing the cleanest collisions possible, with $\mathrm{e^+e^-} \to \mathrm{Z} \to \mathrm{f\bar{f}}$ at LEP/SLC providing a prime example, where Z properties could be studied in minute detail. At lower energies, charm and beauty factories have advanced our understanding of the standard model, *e.g.* the weak unitarity triangle(s). The key argument for future lepton colliders often is precision Higgs physics. Nevertheless, lepton colliders also have their challenges, as will be discussed in this section.

6.5.1 Bremsstrahlung and lepton PDFs

A lepton is surrounded by a cloud of virtual photons. In a collision, such as $\mathrm{e^+e^-}$ annihilation, some of those photons survive in the final state as so-called bremsstrahlung, mainly travelling near the incoming lepton directions, and the annihilation energy is reduced correspondingly. Similarly to the traditional PDF evolution in Q^2 of a hadron, one can here start from a low-Q^2 $f_{\mathrm{e}}^{\mathrm{e}}(x,Q_0^2) = \delta(x-1)$ and evolve it with a splitting kernel

$$\mathrm{d}\mathcal{P}_{\mathrm{e}\to\mathrm{e}\gamma} = \frac{\mathrm{d}Q^2}{Q^2} \frac{\alpha_{\mathrm{em}}}{2\pi} \frac{1+z^2}{1-z} \mathrm{d}z, \tag{279}$$

in close analogy with q → qg. The resummed effects of multiple photon emissions are described in PYTHIA by an NLO expression [256] of the approximate shape

$$f_e^e(x, Q^2) \approx \frac{\beta}{2}(1-x)^{\beta/2-1}, \quad \beta = \frac{2\alpha_{em}}{\pi}\left(\ln\frac{Q^2}{m_e^2}-1\right). \quad (280)$$

The form is divergent but integrable for $x \to 1$, *i.e.* the electron tends to keep most of the energy. To handle the numerical precision problems for x very close to unity, where 64-bit double precision would not be sufficient, the (electron) parton distribution is set to zero for $x > 1 - 10^{-10}$, and is rescaled upwards in the range $1 - 10^{-7} < x < 1 - 10^{-10}$, in such a way that the total area under the parton distribution is preserved:

$$\left(f_e^e(x, Q^2)\right)_{mod} = \begin{cases} f_e^e(x, Q^2), & 0 \le x \le 1 - 10^{-7}, \\ \dfrac{1000^{\beta/2}}{1000^{\beta/2}-1} f_e^e(x, Q^2), & 1 - 10^{-7} < x < 1 - 10^{-10}, \\ 0, & x > 1 - 10^{-10}. \end{cases} \quad (281)$$

Turning to the photon flux, the evolution equation eq. (279) is deceptive in that it appears to treat the electron and photon on equal footing. But, there is no resummation of the photon spectrum, as there is for the one-and-only electron, only an increasing number of photons as the evolution continues. The typical kinematics is also different. When we consider $f_e^e(x, Q^2)$, it is for an annihilating $e^\pm$, where $m_e^2 \ll Q^2 \sim s$, and the radiated energy manifests itself in terms of massless photons. For the $f_\gamma^e(x, Q^2)$, it is instead the electron that has to be on mass shell, a requirement that leads to a non-trivial Q_{min}^2, and the photon that is virtual. This gives a PDF like

$$f_\gamma^e(x, Q^2) = \frac{\alpha_{em}}{2\pi}\frac{1+(1-x)^2}{x}\ln\left(\frac{Q^2}{Q_{min}^2}\right), \quad Q_{min}^2 \approx \frac{m_e^2 x^2}{1-x}, \quad (282)$$

which obviously should vanish if $Q^2 \le Q_{min}^2$. In typical physics applications, it is conventional to set $Q^2 = Q_{max}^2 \sim 1$ GeV² to define a beam of quasi-real photons, that then can lead to γp and $\gamma\gamma$ collisions. A photon more virtual than that would rather be considered as the propagator of a deep inelastic scattering event, and one would not use PDF language to describe it. See further section 6.6 and section 6.7.

The above equations for an electron beam can easily be extended to a muon one, simply by replacing m_e by m_μ, and similarly for τ. Neutrinos do not couple to photons and so there is no need to introduce a substructure for them.

Returning to the issue of e^+e^- annihilation, the effects of bremsstrahlung are more easily illustrated if only one photon emission is considered, but from either side, in which case

$$\frac{d\sigma}{dx_\gamma} = \frac{\alpha_{em}}{\pi}\left(\ln\frac{s}{m_e^2}-1\right)\frac{1+(1-x_\gamma)^2}{x_\gamma}\sigma_0(\hat{s}), \quad (283)$$

where x_γ is the photon energy fraction of the beam energy, $\hat{s} = (1-x_\gamma)s$ is the squared reduced hadronic CM energy, and σ_0 is the ordinary annihilation cross section at the reduced energy. For $e^+e^- \to \gamma^* \to f\bar{f}$, where $\sigma_0(\hat{s}) \propto 1/\hat{s} \propto 1/(1-x_\gamma)$, the bremsstrahlung spectrum thus is singular both for $x_\gamma \to 0$ and $x_\gamma \to 1$. The former is a true singularity, corresponding to infinitely soft photons, that fortunately also carry away infinitely little energy from the electron. The latter is cut off by the mass threshold for $f\bar{f}$ production.

If instead the e^+e^- collider is running on a peak in the cross section, like the Z one at LEP 1, and neglecting interference with γ^* for simplicity, then $\sigma_0(\hat{s}) < \sigma_0(s)$. While the soft-photon singularity remains, any non-negligible photon energy will push the Z propagator further off-shell, which leads to a suppression of such photon emissions and of the total Z cross section.

The situation is even more extreme for charm and beauty factories when they run on a narrow ψ or Υ state, where the net effect is a loss of cross section. PYTHIA does not simulate such emissions, however, or indeed the production of onium states by e^+e^- colliders.

Finally, note that leptons can be polarized both transversely and longitudinally, the former by plane polarization in circular rings and the latter by spin rotation thereof. This can lead to non-trivial effects on cross sections, since the standard model distinguishes between left- and right-handed fermions, and is therefore expected to be a main staple at future linear colliders. While PYTHIA 6.4 encoded spin-dependent cross sections for a few common processes, none of these have been ported to PYTHIA 8.3. If Les-Houches event input is used, such effects can be taken into account already at that level, and will not affect the continued handling of the event by PYTHIA.

6.5.2 Beamstrahlung

At potential future linear e^+e^- colliders, the beams will be so tightly collimated that the electrical field of one beam will significantly deflect the individual $e^\pm$ of the other. This acceleration of charges leads to the emission of photons — beamstrahlung. Like bremsstrahlung, it gives a reduced collision energy, a disadvantage that has to be balanced against the gains of a higher luminosity. Beamstrahlung emits real photons and keeps the electrons real as well, so there is no Q^2 dependence but only an x one. The $f_e^e(x)$ spectrum is highly dependent on the beam parameters, and varies *e.g.* between the front and the tail of a bunch. It is therefore in the realm of machine physicists to provide relevant spectra, *e.g.* with the GUINEA-PIG program [257]. Simplified parameterizations are found in the CIRCE program [258].

For e^+e^- annihilation, the beamstrahlung and bremsstrahlung effects must be convoluted. Relevant code for handling such a convolution does not (yet) exist in PYTHIA 8.3. In case of need, a temporary solution is to split the energy remaining after beamstrahlung, but before bremsstrahlung, into small bins that are generated separately and combined in proportion to their respective cross section. This requires an initialization for each bin, but this is not such a big overhead since the MPI bottleneck is absent in e^+e^- annihilation.

6.5.3 Processes

PYTHIA contains many processes initiated by a fermion-antifermion pair, and these can almost all be used both for hadron and lepton colliders. The list includes electroweak processes, top production, Higgs physics, new gauge bosons, supersymmetry, and so on.

Most prominent is $e^+e^- \to \gamma^*/Z \to f\bar{f}$. It has been the main staple of all lepton colliders so far, possibly with the exception of LEP 2. In addition to precision electroweak physics, it has allowed the study of FSR and hadronization under the cleanest conditions that we can hope for. The simplest $\gamma^*/Z \to q\bar{q}$ process produces a single string between the q and $\bar{q}$ endpoints. One order up, $\gamma^*/Z \to q\bar{q}g$ offers access both to α_s and to tests of string topologies, specifically to confirm that a string is drawn from the q via the g to the $\bar{q}$. With four-jet events, mainly $\gamma^*/Z \to q\bar{q}gg$, the non-Abelian nature of QCD could be established. Taken together, the measured particle composition can be used to tune flavour parameters, measured jet rates and correlations to tune showers, and measured particle spectra to tune longitudinal and transverse fragmentation properties.

For LEP 2, instead W^+W^- pair production was the most prominent process, although $\gamma\gamma$ physics contributed at an even higher rate. Apart from electroweak physics, of note is that $e^+e^- \to W^+W^- \to q_1\bar{q}_2q_3\bar{q}_4$ offers a test bed for colour reconnections, further described in section 7.2.

6.6 Lepton-hadron collisions

In lepton-hadron collisions the events are often classified in terms of virtuality of the intermediate photon, Q^2. Events where the virtuality is large, or the mass of exchanged EW boson is large, and the target hadron breaks up are referred to as deep inelastic scattering (DIS). At low virtualities ($Q^2 \lesssim 1$ GeV^2), the events are in the photoproduction region where the photons can either interact directly as unresolved particles or fluctuate into a hadronic state with equal quantum numbers. In PYTHIA 8.3 these two event classes are handled in separate frameworks and the special features of the former class are discussed below. The photoproduction framework is, in turn, introduced in section 6.7.

6.6.1 Parton distribution functions and structure functions

In DIS the intermediate boson scatters off a parton in the target hadron in a relatively clean scattering process where the kinematics characterizing the scattering can be related to the four-momentum of the outgoing lepton. Therefore, such collisions can be used to study the structure of the hadron and the initial-state QCD dynamics. Let P denote the four-momentum of the incoming hadron, k the incoming lepton and k' the scattered lepton. Then it is possible to define the following Lorentz-invariant quantities

$$
\begin{aligned}
Q^2 &= -q^2 = -(k-k')^2 \,, \\
W^2 &= (P+q)^2 \,, \\
x &= \frac{Q^2}{2\,P \cdot q} \,, \\
y &= \frac{P \cdot q}{P \cdot k} \,,
\end{aligned}
\tag{284}
$$

purely based on measured energy and scattering angle of the scattered lepton. In fully inclusive events, where the hadronic final state is integrated out, it is possible then to write down the cross section of such a scattering process in terms of these quantities without making further assumptions on the proton structure

$$
\frac{\mathrm{d}^2\sigma}{\mathrm{d}x\mathrm{d}y} = N^l \left(y^2\,x\,F_1^l(x,Q^2) + (1-y)\,F_2^l(x,Q^2) \mp (y-\frac{y^2}{2})x\,F_3^l(x,Q^2) \right). \tag{285}
$$

The coupling factor N^l is different for neutral- and charged-current DIS and the sign of the last term depends on whether the incoming lepton l is charged or neutral (neutrino) and if it is a particle or an antiparticle. The structure functions $F_i^l(x,Q^2)$ represent the partonic structure of the hadron. In the leading-order parton model [259,260] the structure functions are simply proportional to the sum of the parton distributions $f(x,Q^2)$ but do depend also on beam lepton type. The x can be interpreted as the momentum-energy fraction of the parton with respect to the hadron momentum P and the Q^2 dependency arise from the QCD corrections at higher orders. The goal of PYTHIA 8.3 is, however, to provide fully exclusive events for which the relevant treatment is described next.

6.6.2 Deep inelastic scattering

The DIS framework describes processes where the scattered lepton emits a highly-virtual (point-like) photon or a massive gauge boson that interacts with the constituents of the target hadron breaking it up. As there currently are no models for intermediate photon virtualities ($Q^2 \sim 1$ GeV^2) where high-virtuality, point-like, and low-virtuality hadronic processes contribute to cross sections, the DIS framework provides a reliable description only at sufficiently

large Q^2 where the scattering is purely mediated by a point-like particle. As the resolved-photon contribution fade continuously (roughly as $\sim 1/Q^2$), it is impossible to set a hard cut for such a region. In most applications, however, a limit of $Q^2 > 5$ GeV^2 has turned out to be sufficient to ensure negligible contributions from the hadronic fluctuations. As the model for intermediate virtualities implemented in PYTHIA 6 was based on several parameterizations mimicking the physical picture and turned out to be somewhat fragile with restricted predictability, we have decided to develop a completely new model for such processes that will be implemented in a future PYTHIA 8.3 release.

Hard processes In the LO DIS implemented in PYTHIA, the incoming lepton scatters off a quark from the target hadron by exchanging an EW boson. As described in section 3.2, this includes both neutral- and charged-current processes with charged leptons and neutrinos, and the interference between a virtual photon and the Z boson can be accounted for. The DIS-optimized scale-setting options are listed in section 3.10 and the relevant phase-space cuts in section 3.13. It is also possible to provide the hard process as an input from an ME generator in the LHE format. However, no matching of higher-order processes and the default parton shower has been implemented. As the hard-processes are set up in the collinear approximation, no off-shellness is allowed for the initial lepton line. Thus no radiation should be allowed for the initial-state lepton and no PDFs for the lepton used. The phase-space sampling for DIS is inherited from generic massless $2 \rightarrow 2$ scattering where the initiators are assumed massless but the final-state particles can have finite masses. This is not ideal for DIS, however, since the invariants typically studied in DIS, Bjorken x and Q^2, are often derived from the four-momentum of the scattered lepton. Due to a mismatch in masses, these variables might then not match the internally sampled values which can lead to unphysical configurations such as $x > 1$ when invariants are derived from the scattered lepton. To fix the issue, a new phase-space sampling optimized for t-channel exchange of bosons with (potentially varying) masses will be implemented. Heavy-quark pairs can be produced in two different ways: if the lepton scatters off a heavy quark, a companion will be added by ISR, or the heavy-quark pair can be formed from a gluon splitting by FSR. Similarly, DIS events with more than one jet can be formed via PS emissions but no explicit hard dijet processes have been included. The showers do, however, include matrix-element corrections for the first emissions. As the DIS process is a scattering of a single point-like particle, no MPIs are allowed.

Parton showers Both initial- and final-state radiation from deep-inelastic-scattering processes require a careful treatment of the branching kinematics. If emissions from the hadronic system disturb (via recoil) the lepton line, or *vice versa*, then both the x and the Q^2 distribution are affected by showering. In this case, an intricate recalculation of the hard-scattering cross section after each parton-shower emission is required, making the strategy sub optimal[6]. The natural resolution is to ensure that any recoil due to the branching process is contained either in the hadronic or the leptonic system. For the hadronic system, this is most easily achieved by employing a "local dipole recoil" strategy, in which the kinematic recoil is absorbed by a colour-connected partner. Such a strategy is employed by the DIRE section 4.3 and VINCIA section 4.2 showers, and is an option for the simple-shower methods [80]. To model the QED evolution of the leptonic line, this approach is insufficient, however, and more complex strategies are necessary, or the conservation of x and Q^2 may need to be relaxed, *e.g.* for charged-current DIS events, where electric charge flows from the lepton to the hadron system. The latter is the case in the DIRE shower.

---

[6]Similar concerns apply to any scattering via t-channel colour-singlet exchange, *e.g.* to Higgs production in vector-boson fusion.

Another important aspect of modelling radiation in DIS events is the phase-space boundary for emissions that involve incoming partons. The factorization scale (*i.e.* typically Q^2) gives a natural phase-space boundary when using backward initial-state evolution [52]. However, the kinematic boundary is more accurately given by the invariant mass of the radiating dipole or the invariant mass of the hadronic system (W^2), which are often, and especially for low-x values, significantly larger than Q^2. A natural resolution of this issue is to keep the tight Q^2 constraint for the shower, and use (tree-level) merging to supplement the missing phase-space regions. Another approach is to abandon the DGLAP-based initial-state evolution [6]. In lieu of the latter, hard initial-state emission in the partons shower models of PYTHIA should be considered with caution.

As the DIS events are rather clean, they offer a very good environment to study parton-shower dynamics. For example, since the parton shower produces $p_\perp$ kicks for the initiator via emissions, it can be thought to resemble perturbative evolution of transverse-momentum dependent (TMD) PDFs [261, 262]. Thus one should obtain reasonable $p_\perp$ distributions in Semi–Inclusive DIS (SIDIS) from the parton-shower enabled PYTHIA simulations.

Hadronization   The hadronization of DIS events is analogous to that of hadron-hadron scattering systems. The scattered lepton does not partake in hadronization and since no multi-parton interactions are included in DIS events, no colour reconnection model is employed. At present, no DIS data has been used in the tuning of the hadronization model. The study of spin polarizations and the higher-dimensional structure of the hadron are typically important aspects of DIS analysis. In this context, it should be noted that the PYTHIA 8.3 hadronization model does not by default consider polarization, though external tools to model such effects have been proposed [263].

6.7   Photon-hadron and photon-photon collisions

The possibility to turn a charged lepton into a photon using laser back scattering has been studied, but has not been realized in the current or foreseen colliders. Thus photon-induced collisions are usually studied in colliders with charged beam particles that may emit photons when accelerated to high energies. The shape of the photon flux and the virtuality spectra are, however, different for different beam types but, given an appropriate flux, the photon-induced processes can be treated in a single framework regardless of the original beam configuration. Here we focus on low-virtuality (quasi-real) photons and introduce the current simulation framework in PYTHIA 8.3 for processes involving such effective beams.

6.7.1   Parton distribution functions of resolved photons

In total there are three separate contributions for processes with low-virtuality photons: a photon can interact either as an unresolved particle, it can split perturbatively into quark-antiquark pair, or it can fluctuate into hadronic state non-perturbatively. The two latter contributions, where the partonic constituents act as initiators for hard scattering, can be described with DGLAP-evolved PDFs. As in the case of hadrons, the evolution equation for resolved photons do include a hadron-like component where a non-perturbative ansatz is evolved according the usual QCD DGLAP kernels. In addition to this, however, the evolution equation contains also a point-like component which feeds in more quark-antiquark pairs with increasing evolution scale that may evolve further by QCD splittings. The full evolution equation for resolved photons is

$$\frac{\partial f_i^\gamma(x,Q^2)}{\partial \log(Q^2)} = \frac{\alpha_{\text{em}}(Q^2)}{2\pi} e_i^2 P_{i\gamma}(x) + \frac{\alpha_{\text{s}}(Q^2)}{2\pi} \sum_j \int_x^1 \frac{\mathrm{d}z}{z} P_{ij}(z) f_j^\gamma(\frac{x}{z}, Q^2), \qquad (286)$$

where the $\gamma \to q\bar{q}$ splitting kernel in LO is $P_{i\gamma}(x) = 3(x^2 + (1-x)^2)$ and Q^2 is the factorization scale at which the partonic structure is probed. As in the case of proton PDFs, the parameters related to the non-perturbative ansatz at the initial scale are determined in a global QCD analysis comparing to experimental data. In PYTHIA 8.3, the default set for the resolved photon PDFs is from the CJKL analysis [264] which conveniently provides the hadron-like and point-like parts separately, which can be used for finer classification of the events with resolved photons. No dependence on the photon virtuality is included in these PDFs, but all photons are taken as real with zero virtuality, which is the case also for the LO photon-initiated cross sections currently implemented in PYTHIA 8.3.

6.7.2 Photoproduction

Photoproduction typically refers to processes where a beam lepton emits a low-virtuality (quasi-real) photon that then collides with a hadron from the other beam. The following describes some special features of such collisions. These are not unique to ep colliders – similar processes can take place also in e^+e^-, pp, pA, and AA collisions as will be discussed in the following.

Photon flux and kinematic limits When the emitted photons are quasi-real and almost collinear with the beam leptons, the cross section calculations can be simplified by factorizing the photon flux from the hard perturbatively calculated part. In case of lepton beams, the flux of quasi-real photons can be obtained from the well-known Weizsäcker–Williams [265, 266] or Equivalent Photon Approximation (EPA). The flux differential in photon virtuality Q^2 is

$$f_\gamma^l(x_\gamma, Q^2) = \frac{\alpha_{\mathrm{em}}}{2\pi} \frac{dQ^2}{Q^2} \frac{1 + (1-x_\gamma)^2}{x_\gamma}, \tag{287}$$

where x_γ is the momentum fraction carried by the (almost) collinear photon with respect to the parent lepton. Integration from the minimum allowed virtuality yields the photon-in-lepton PDF in eq. (282). In photoproduction, the upper limit Q^2_{max} is typically of the order 1 GeV^2, depending on the considered experimental setup and detector acceptance. The lower limit is restricted by the requirement of physical kinematics (on-shell leptons) for the $1 \to 2$ splitting and depends on x_γ, the mass of the lepton, m_l, and the energy of the beam in the CM frame, E

$$Q^2_{\mathrm{min}}(x_\gamma) = \frac{2m_l^2 x_\gamma^2}{1 - x_\gamma - m_l^2/E^2 + \sqrt{1 - m_l^2/E^2}\sqrt{(1-x_\gamma)^2 - m_l^2/E^2}} \approx \frac{m_l^2 x_\gamma^2}{1 - x_\gamma}. \tag{288}$$

From a similar consideration, one can find the kinematically allowed upper limit for x_γ

$$x_\gamma^{\mathrm{max}} = \frac{2\left(1 - \frac{Q^2_{\mathrm{max}}}{4E^2} - \frac{m_l^2}{E^2}\right)}{1 + \sqrt{\left(1 + \frac{4m_l^2}{Q^2_{\mathrm{max}}}\right)\left(1 - \frac{m_l^2}{E^2}\right)}}, \tag{289}$$

which typically is very close to unity. The lower limit of x_γ can be derived from the minimum considered W of the photon-hadron system. Similarly, as for hadron-hadron collisions, this should be large enough to justify the perturbative treatment that PYTHIA is largely based on. After the values for x_γ and Q^2 have been sampled from the allowed phase space, the full kinematics for the intermediate photon can be derived. The transverse and longitudinal

momentum, $q_\perp$ and q_z as shown in fig. 12, can be calculated from

$$q_\perp = \sqrt{\frac{\left(1 - x_\gamma - \frac{Q^2}{4E^2}\right)Q^2 - \left(x_\gamma^2 + \frac{Q^2}{E^2}\right)m_l^2}{1 - \frac{m_l^2}{E^2}}}\,, \tag{290}$$

$$q_z = \frac{E\left(x_\gamma + \frac{Q^2}{2E^2}\right)}{\sqrt{1 - \frac{m_l^2}{E^2}}}\,. \tag{291}$$

The azimuthal angle is sampled from a flat distribution and the scattered lepton four-momentum can be obtained simply from $k' = k - q$. It is also possible to provide the photon flux externally in PYTHIA 8.3, but the sampling has been optimized for the form in eq. (287). The kinematics and the allowed phase-space region are independent from the applied flux.
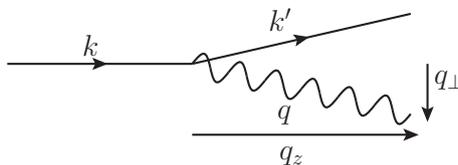
Figure 12: Kinematics of a photon emission.

Direct and resolved photons   If the (quasi-)real photon is the initiator of the hard scattering, *i.e.* an unresolved (or direct) photon, the photon flux acts essentially as a PDF and can be directly applied for sampling of the process kinematics. If the photon has fluctuated into a hadronic state, for which the partonic structure is given by the resolved photon PDFs described above, these PDFs have to be convoluted with the flux to define so-called parton-in-photon-in-lepton PDFs

$$f_i^\gamma(x, Q^2) = \int_x^1 \frac{\mathrm{d}x_\gamma}{x_\gamma} f_\gamma^{\mathrm{p}}(x_\gamma) f_i^\gamma(x/x_\gamma, Q^2)\,, \tag{292}$$

where the photon virtuality has been integrated out and Q^2 refers to the factorization scale at which the resolved photon is probed. Here, it is also assumed that the PDFs are independent of the photon virtuality, though alternatives containing such information exist, see *e.g.* [267]. The flux is also used to sample the intermediate photon kinematics required to reconstruct the full event including the remnants of the resolved photon and the kinematics of the scattered lepton. In PYTHIA 8.3 both of these contributions, direct and resolved, are included and can be generated simultaneously to obtain the correct mixture of the possible contributions for a given process at considered kinematics.

ISR with photon beams   For direct photons, no ISR splittings have been implemented as in these cases the effect from additional QED emissions is typically small. For the resolved photons, however, some additional care needs to be taken when generating ISR due to the extra term in the PDF evolution, see eq. (286), compared to purely hadronic beam particles. As this term feeds in quark-antiquark pairs when evolving forwards with DGLAP, in backwards evolution, relevant for the ISR, this will collapse partons back into the original unresolved photon as illustrated in fig. 13. If such splittings are found during the PS evolution, one can think of these processes being of point-like origin and if not, the partons have originated from the hadron-like part of the PDFs. This dynamical selection of these two contributions have

then further implications for beam remnants and MPIs as discussed below. This is also one of the key differences between the old PYTHIA 6 implementation where such selection was done already when sampling the hard scattering, and no MPIs were allowed for the point-like contribution at any scale.
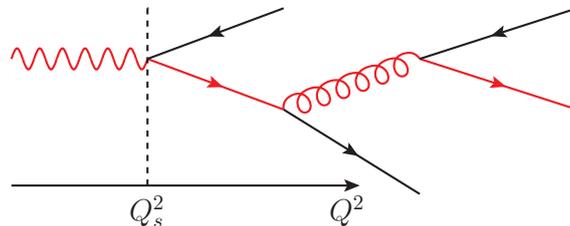
Figure 13: Backwards evolution of a point-like photon that collapses into unresolved photon at the scale Q_s. The hard-process initiator whose splittings are traced back in ISR is highlighted with red colour.

MPIs with resolved photons   Similarly as with resolved hadron beams, the resolved photons may also experience several partonic interactions in each collision. These MPIs are modelled in the same way as for hadrons as described in section 6.2, but some aspects require further attention. The first one follows from the ISR generation discussed above. If the photon has collapsed back to an unresolved state, it can not have further MPIs below the scale at which this splitting has occurred, in fig. 13 this scale is denoted with Q_s. Such an ordering is possible thanks to the interleaved evolution of PS and MPIs, see eq. (271). Another potential difference is related to the screening parameter in semi-hard cross sections from which the MPI probabilities are calculated from. Since the partonic and spatial structure of resolved photons are quite different compared to protons, it would be expected that the value of this parameter should be separately tuned for collisions involving resolved photons. Indeed, first comparisons to HERA data [268] indicate that a somewhat larger screening parameter yielding a lower MPI probability is preferred but the constraints are still rather sparse and would benefit from further measurements of low-$p_\perp$ hadron production. Also, the impact-parameter profile could be modified but this would require more experimental data sensitive to MPIs.

Remnants   Since the PDFs for resolved photons contain both a hadron- and a point-like part, the remnant construction also needs to be adjusted to handle both cases. The main difference to a purely hadronic state is that since the point-like contribution is of a perturbative nature, the collapse back to a pure photon state should also be handled perturbatively, namely with the parton showers. Unlike in PYTHIA 6 the distinction into a point-like and hadron-like part is not done when the (semi-)hard scattering is selected, but the term corresponding to $\gamma \to q\bar{q}$ splitting in ISR algorithm will select the cases where the initiator has originated from a perturbative photon splitting. In cases where there are no MPIs in the event, ending up in such a configuration means that there is no need to add any non-perturbative remnants, as the necessary partons have been added perturbatively by the parton shower as illustrated in fig. 13. If the ISR generation will not end in a $\gamma \to q\bar{q}$ splitting, the resolved photon is taken to be hadron-like, and the remnants will be constructed similarly as for any hadrons. In this case, the valence flavour is sampled based on relative weights derived from the PDFs. The remnant construction becomes more complicated if the initiator is found to be of a point-like origin but the beam photon has encountered additional MPIs before (at scale $Q^2 > Q_s^2$) the resolved state is collapsed into an unresolved one. Then there are several initiators kicked out from the beam, so a single companion cannot make the beam configuration flavour and colour neutral.

In this case the remnant is again constructed as for any hadron, but the primordial $k_\perp$ for the initiator of the hardest process and its companion are derived from the scale Q_s^2 at which the $\gamma \to q\bar{q}$ branching collapsing the photon to an unresolved state has occurred.

Hard processes and diffraction   The hard diffraction with dynamical rapidity-gap survival model introduced in section 6.4.3 has been implemented also for photoproduction. For direct photons, the no-MPI requirement has zero effect since there are no MPIs with unresolved photons. However, as MPIs can still occur with resolved photons, some suppression is expected also for hard-parton initiated processes. Indeed, there are indications that diffractive dijet photoproduction cross sections are suppressed compared to pQCD predictions based on diffractive PDFs for the target proton. The observed suppression factor depends on the applied kinematic cuts and varies between 0.5–0.9 in different analyses [269, 270]. The milder suppression compared to hadron-hadron collisions at the Tevatron and LHC is explained by the presence of the direct component and the smaller invariant mass of the photon-proton system at HERA kinematics which both reduce MPI probability compared to hadronic collisions at higher energies. As demonstrated in [271], the MPI-based model in PYTHIA 8.3 provides a reasonable description for the various HERA data.

Soft QCD processes   Apart from the non-diffractive low-$p_\perp$ $2 \to 2$ scatterings that are generated with the regulated cross section from the MPI framework using photon PDFs, the soft processes with real photons are modelled according to the vector meson dominance (VMD) model. In this model the photon is described as a linear combination of different vector-meson states with prefactors derived from experimental data. In PYTHIA 8.3, the values are taken from the analysis presented in [272] that have also been used in an SaS fit [207] for total and elastic cross sections applied here. The included vector meson states are ρ^0, ω, ϕ^0, and J/ψ but Υ is currently neglected. In the VMD model for elastic and diffractive processes, the incoming photon will first transform into a vector-meson state sampled according to relative weights. Then, the interaction is handled similarly as for any other hadron-hadron case described in section 6.1.5. The elastic scattering process in photoproduction is often referred to as exclusive vector-meson production for which there are nowadays a good amount of data from HERA experiments, see *e.g.* [273–279]. The SaS parameterization tends to provide a good description for low-mass vector-meson production, *e.g.* in case of ρ^0, but underestimates higher-mass states such as the J/ψ by a large margin. This indicates the need for further, possibly pQCD based, modelling for high-scale elastic processes.

6.7.3   Photon-photon collisions

Similarly as photoproduction in ep collisions, the charged-lepton beams in e^+e^- collisions may emit photons that can interact with each other leading to effective photon-photon collisions. If both of the photons have a low virtuality, there are a number of possible combinations that must be accounted for. In the most complex case, where both photons are resolved, the collisions are generated in a similar manner as in hadron-hadron collisions, including parton showers for the initial and final state, beam remnants, and, in particular, MPIs with the same special features as with photoproduction as discussed earlier. If one photon is unresolved and other resolved, the interactions are somewhat simpler, since the unresolved photon scatters off a parton from a resolved photon. In this case, no MPIs can take place and ISR and beam remnants are generated only for the hadron side. Both photons can also interact as unresolved particles when all particles are produced from the outgoing particles through FSR and hadronization, which are relevant also for other possible contributions.

Kinematics The initial phase-space sampling assumes that the incoming photons are collinear with respect to the beam particles. However, as kinematically allowed photons emitted from massive (on-shell) particles will always have a finite virtuality, they will also possess some transverse momentum given by eq. (290). The direction of this $q_\perp$ is not a priori known and is sampled only after the hard process kinematics are determined. Thus the final invariant mass of the photon-photon system, $W_{\gamma\gamma}$, will depend on the virtualities of the photons and their relative azimuthal angle, $\Delta\phi = \phi_1 - \phi_2$. The resulting W can again be derived from the kinematics, giving

$$W_{\gamma\gamma}^2 = 2E_1 E_2 x_{\gamma 1} x_{\gamma 2} - Q_1^2 - Q_2^2 + 2q_{z1}q_{z2} - 2q_{\perp 1}q_{\perp 2}\cos(\Delta\phi), \qquad (293)$$

where x_i are the momentum fractions of the photons with respect to the beam leptons whose CM energies are E_i. To account for the possibly modified W^2 ($= \hat{s}$ for the direct-direct case), the cross section and relevant kinematic variables are recalculated after the virtualities and the direction of the photons are sampled. Typically the changes in the cross section and kinematics are negligible, but are needed in order to preserve the four-momentum of the event. An exception is, however, $2 \rightarrow 1$ processes where it is important to keep the mass of the intermediate particle intact, a prime example being Higgs-boson production, where the photon momentum fractions are modified instead.

Possible final states There are many topics that can be studied in photon-photon collisions and the relative importance of direct and resolved contributions varies by the process and considered kinematics. For example, Higgs production in $\gamma\gamma$ collisions is dominated by the direct-direct contribution but for QCD processes, such as jets or heavy quarks that contribute to the background of Higgs studies, the resolved photons may also have a significant contribution. Another interesting phenomenon is the MPIs in a photon-photon system which can be studied with low-$p_\perp$ hadrons that arise almost completely from resolved-resolved interactions. Also QED processes, such as dilepton production, can be considered to calibrate the photon fluxes as they are not sensitive to QCD effects.

6.7.4 Ultra-peripheral collisions

As briefly mentioned earlier, other charged beam particles, including protons and heavy nuclei, may also emit photons that interact with the other beam or photons emitted by the other beam. When the beam particles do not interact hadronically but stay intact and emit photons that give rise to a hard interaction, the events are referred to as Ultra–Peripheral Collisions (UPCs). Due to the requirement of beam particles with finite size not breaking up, the emitted photons have always a small virtuality and can therefore be handled with the photoproduction framework introduced above. The photon-induced processes where the beam hadron break ups can be simulated by using a PDF set that includes perturbatively generated photons from DGLAP evolution with the usual PYTHIA model for hadron-hadron collisions.

The key difference between photon fluxes from hadrons and charged leptons is that the finite size of the emitting particle needs to be accounted for. For protons, a good approximation is obtained with the electric dipole form factor, giving a Q^2-differential flux of the form

$$f_\gamma^p(x_\gamma, Q^2) = \frac{\alpha_{\mathrm{em}}}{2\pi} \frac{1 + (1 - x_\gamma)^2}{x_\gamma} \frac{\mathrm{d}Q^2}{Q^2} \frac{1}{(1 + Q^2/Q_0^2)^4}, \qquad (294)$$

where $Q_0^2 = 0.71$ GeV^2. Integrating over the possible virtualities will provide the flux derived in [280]. Another flux has been implemented for protons that is based on work by Budnev *et al.* (see [281]). The downside in the latter is that since only a virtuality-integrated form is

provided, there is not enough information to sample the full kinematics of the intermediate photon and the virtuality sampling needs to be turned off. Therefore, this flux is not suited to study observables sensitive to the transverse momentum of the intermediate photon as the $q_\perp$ is set to zero.

For heavy nuclei, it is possible to use form factors and derive the photon flux in a similar manner as for protons. Usually it is more convenient to work in the impact-parameter space since the heavy nuclei have a well-defined size and therefore it is possible to remove events where hadronic interactions dominate the particle production by rejecting events with small impact parameter. As shown in [282], it is possible to derive an analytic form for the flux differential in the impact parameter by assuming a point-like charge distribution. In fact, this provides a good approximation for the flux with a more realistic density profile when considering the region outside of the nucleus relevant for UPCs. Integrating this from the minimum allowed impact-parameter value $b_{\min}$ gives

$$f_\gamma^A(x_\gamma) = \frac{\alpha_{\mathrm{em}} Z^2}{\pi x_\gamma} \left[2\xi K_1(\xi) K_0(\xi) - \xi^2 \left(K_1^2(\xi) - K_0^2(\xi) \right) \right], \qquad (295)$$

where Z is the electric charge (number of protons) of the nuclei A, $\xi = b_{\min} x_\gamma m_N$. As the nuclear beams are typically defined in terms of per-nucleon energy, m_N here also refers to average nucleon mass. A suitable value for $b_{\min}$ is given by the sum of the radii of the colliding nuclei. Such a flux is included in PYTHIA 8.3 but can only be enabled by providing this as a pointer to the Pythia object with a dedicated method. The shape and magnitude of this flux is very different from the flux for charged leptons, and therefore the phase-space sampling must be re-optimized for efficient event generation. A suitable over-estimate is included, but the parameters may have to be re-adjusted for different beam configurations. When using this flux, the virtuality sampling has to be disabled since the allowed virtualities have been essentially integrated over when converting to impact-parameter space by Fourier transform from the momentum space.

The current framework can already be applied to many processes studied in UPCs but have a few limitations as well. In proton-proton collisions it is possible to study both photon-photon and photon-proton collisions with fully reconstructed kinematics, when a Q^2-dependent flux is used. This includes all hard processes initiated by photons or partons and also soft QCD processes apart from central- and double-diffractive events. These allows for the study of minimum-bias photon-proton collisions, inclusive and diffractive jet production, and photon-initiated dilepton production with all different contributions, to name a few. In case of heavy ions, the palette is somewhat more limited due to a Q^2-independent photon flux and lack of model for photon-nucleus collisions, which will be addressed in future releases. In pA collisions, where the flux from the heavy nucleus is amplified by the Z^2 factor so that γp component dominates the cross sections, almost all the same final states can be studied as in proton-proton collisions apart from observables highly sensitive to transverse momentum of the intermediate photon. For QCD observables, the effect from neglected Q^2 dependence will be washed out by the QCD radiation. In AA collisions subsequent photon-nucleon interactions are not modelled, but high-$p_\perp$ observables and direct-photon dominated processes can be generated with reasonable accuracy. Photon-photon interactions can also be considered, with the only limitation being the neglected Q^2 dependence in the kinematics that again has an effect for the $q_\perp$-dependent observables, *e.g.* the acoplanarity of dilepton pairs produced by two direct photons.

6.8 Heavy ion collisions

The Heavy Ion (HI) collider physics community has traditionally not had very close ties to the rest of the High–Energy Physics (HEP) community. This has also been reflected in the event

generator community, where the authors of HI event generators, although they some times make use of *e.g.* the string fragmentation in PYTHIA, did not interact much with the authors of the main general purpose event generators for pp, ep, and e^+e^- collisions. However, with the arrival of the LHC, the situation has changed. Not only are HI and particle physicists now part of the same collaborations, the physics questions being asked are also starting to converge, and typical observables studied in HI collisions are being applied to pp, and *vice versa*. It should therefore not come as a surprise that PYTHIA 8.3 now also has some HI functionality implemented.

There are several ways to study HI collisions in PYTHIA 8.3. In section 6.7.4 we described how to study ultra-peripheral HI collisions, and there is also the possibility to use nuclear PDFs to study some observables. Here, we will concentrate on the modelling of complete exclusive hadronic final states using the so-called ANGANTYR model [283], which is the default way of handling HI collisions in PYTHIA 8.3.

6.8.1 Wounded nucleons

The ANGANTYR model in PYTHIA 8.3 can be said to be the successor of the old FRITIOF program [284] which used string fragmentation to generate final states in HI collisions, and was based on the so-called wounded nucleon model [285]. The basic assumption in the wounded nucleon model is that each nucleon that participates in a HI collisions contributes to the multiplicity of the full final state, according to a multiplicity function $W(y)$ which has a triangular form in rapidity

$$W(y) \propto \frac{1}{2}\left(1 + \frac{y}{y_{\max}}\right), \tag{296}$$

where $y_{\max}$ is the rapidity of the nucleon in the collision rest frame. This would yield the following simple form of the rapidity distribution in an AA, for a given number of wounded nucleons (or *participants*), $N_{\text{part},p}, N_{\text{part},t}$ in the projectile and target nuclei respectively,

$$N(y) = N_{\text{part},p} W(y) + N_{\text{part},t} W(-y). \tag{297}$$

FRITIOF, in its simplest form, used the fact that the distribution of particles of a hadronizing string is flat in rapidity. For each wounded nucleon, a string was stretched out to an endpoint randomly positioned uniformly in rapidity, which then on average reproduces the form in eq. (296). Despite the simplistic nature of the model, FRITIOF was able to provide a fairly good description of collider data at the energies available in the 1980s. In fact, even pp collisions (with $N_{\text{part},p} = N_{\text{part},t} = 1$) were reasonably described.

With the energies achievable at RHIC and LHC, the basically non-perturbative FRITIOF model falls short of reproducing data, and the ANGANTYR model was developed to address these shortcomings.

6.8.2 The ANGANTYR model

In comparison to FRITIOF, the ANGANTYR model introduces two major new ingredients. First, rather than wounded nucleons only resulting in a string stretched out and being hadronized, a full diffractive excitation is generated using the full multiparton interaction machinery of PYTHIA where these are described in terms of a pomeron-proton collision. In addition, a more advanced version of the Glauber simulation is used where special attention is given to the fluctuations in the nucleon wave functions, making it possible to differentiate between different types of Nucleon–Nucleon (NN) subcollisions.

Starting with the new Glauber modelling, we rely on the Good–Walker formalism [206] to connect the different types of NN semi-inclusive cross sections with fluctuations in the wave functions [286].

For a projectile particle with an internal substructure, the elastic scattering eigenstates may differ from the mass eigenstates. In the following we denote the eigenstates of the scattering amplitude T, with $T\Phi_l = t_l \Phi_l$, and the mass eigenstates Ψ_i. The ground state (*e.g.* a nucleon) is denoted Ψ_0.

The mass eigenstates can be written as linear combinations of the scattering eigenstates, $\Psi_i = \sum_l c_{il} \Phi_l$, and a scattering is treated as a measurement, where one of the eigenvalues, t_l, is chosen with probability $|c_{0l}|^2$.

The elastic amplitude for a ground state projectile is directly written as $\langle \Psi_0 | T | \Psi_0 \rangle = \sum_l |c_{0l}|^2 t_l \equiv \langle T \rangle$. The resulting $\langle T \rangle$ is then the expectation value of the amplitude T for the projectile. The elastic cross section then becomes

$$\mathrm{d}\sigma_{\mathrm{el}}/\mathrm{d}^2 b = \langle T(b) \rangle^2 \,. \tag{298}$$

In impact-parameter space, the amplitude depends on b, and the total diffractive-scattering cross section, σ_{diff}, can be written as the sum of transitions to all Φ_l:

$$\mathrm{d}\sigma_{\mathrm{diff}}/\mathrm{d}^2 b = \sum_l \langle \Psi_0 | T | \Phi_l \rangle \langle \Phi_l | T | \Psi_0 \rangle = \langle \Psi_0 | T^2 | \Psi_0 \rangle \,, \tag{299}$$

since Φ_l form a complete set of eigenstates. The cross section for diffractive excitation only, can then be obtained by subtracting the elastic cross section from the above result. It directly becomes the fluctuations in the scattering amplitude:

$$\mathrm{d}\sigma_{\mathrm{diff-tot}}/\mathrm{d}^2 b = \langle T^2 \rangle - \langle T \rangle^2 \,. \tag{300}$$

If both projectile and target are in flux, this leads to a number of options. This is the case for NN collisions. One can have single-diffractive excitation of the projectile (denoted subscript Dp) or the target (denoted subscript Dt), as well as both, *i.e.* double diffraction (denoted subscript DD). With elastic scattering (denoted subscript el), the remaining cross section is then the absorptive or *inelastic non-diffractive* cross section (denoted subscript abs). Denoting averages over projectile and target states by $\langle \cdots \rangle_p$ and $\langle \cdots \rangle_t$ respectively, all the semi-inclusive different NN cross sections can be written as,

$$
\begin{aligned}
\mathrm{d}\sigma_{\mathrm{tot}}/\mathrm{d}^2 b &= \langle 2T(\mathbf{b}) \rangle_{p,t} \,, \\
\mathrm{d}\sigma_{\mathrm{abs}}/\mathrm{d}^2 b &= \left\langle 2T(\mathbf{b}) - T^2(\mathbf{b}) \right\rangle_{p,t} \,, \\
\mathrm{d}\sigma_{\mathrm{el}}/\mathrm{d}^2 b &= \langle T(\mathbf{b}) \rangle_{p,t}^2 \,, \\
\mathrm{d}\sigma_{\mathrm{Dt}}/\mathrm{d}^2 b &= \left\langle \langle T(\mathbf{b}) \rangle_p^2 \right\rangle_t - \langle T(\mathbf{b}) \rangle_{p,t}^2 \,, \\
\mathrm{d}\sigma_{\mathrm{Dp}}/\mathrm{d}^2 b &= \left\langle \langle T(\mathbf{b}) \rangle_t^2 \right\rangle_p - \langle T(\mathbf{b}) \rangle_{p,t}^2 \,, \\
\mathrm{d}\sigma_{\mathrm{DD}}/\mathrm{d}^2 b &= \left\langle T^2(\mathbf{b}) \right\rangle_{p,t} - \left\langle \langle T(\mathbf{b}) \rangle_p^2 \right\rangle_t - \left\langle \langle T(\mathbf{b}) \rangle_t^2 \right\rangle_p + \langle T(\mathbf{b}) \rangle_{p,t}^2 \,.
\end{aligned}
\tag{301}
$$

We note that the diffractive excitation is directly related to fluctuations in the nucleon wave function.

In ANGANTYR, we use these cross sections in the Glauber modelling to determine not only which nucleons have been wounded, but also to differentiate if they were non-diffractively scattered or only diffractively excited. The fluctuations are by default modelled using a varying radius of the nucleons, according to a Gamma function,

$$P(r) = \frac{r^{k-1} e^{-r/r_0}}{\Gamma(k) r_0^k} \,, \tag{302}$$

and in addition, introducing a varying *opacity* of the elastic amplitude, which depends on the radii of the projectile and target nucleons, r_p and r_t,

$$T(\mathbf{b}, r_p, r_t) = T_0(r_p + r_t)\Theta\left(\sqrt{\frac{(r_p + r_t)^2}{2T_0(r_p + r_t)}} - b\right), \tag{303}$$

where

$$T_0(r_p + r_t) = \left(1 - \exp\left(-\pi(r_p + r_t)^2/\sigma_t\right)\right)^\alpha. \tag{304}$$

We then obtain the differential semi-inclusive cross sections in eq. (301) using $\langle\cdots\rangle_i = \int \mathrm{d}r_i P(r_i)(\cdots)$, which gives *e.g.*

$$\left\langle \langle T(\mathbf{b})\rangle_p^2\right\rangle_t = \int P(r_t)\left(\int P(r_p)T(\mathbf{b}, r_p, r_t)\mathrm{d}r_p\right)^2 \mathrm{d}r_t. \tag{305}$$

Three parameters (k, r_0 and σ_t) depend on the NN collision energy, and need to be determined. By default this is done in the ANGANTYR initialization by fitting the integrated total and semi-inclusive NN cross sections to the parameterization in PYTHIA 8.3 (see section 6.1), using a simple genetic algorithm. If needed, the parameters can be specified by the user to avoid the somewhat time-consuming fitting procedure.

The Glauber calculation works as follows. First the 3D positions of the nucleons in the nuclei are modelled using a Woods–Saxon parameterization (by default the parameterizations with a *hard core* from [287, 288] is used). Then, an impact parameter between the nuclei is generated according to a user-specified importance sampling (by default a 2D Gaussian). For each nucleon we then sample the wave function according to eq. (302). This gives us the probability that a projectile nucleon, i, scatters non-diffractively with target nucleon, j, as

$$2T(\mathbf{b}, r_i, r_j) - T^2(\mathbf{b}_{ij}, r_i, r_j), \tag{306}$$

where $\mathbf{b}_{ij}$ is the impact parameter between the nucleons. But, we also want to obtain the probability of diffractive excitation, which involves the fluctuations. We do this by generating an additional radius, r', for each nucleon, thus *sampling* the fluctuations. In this way we obtain four statistically equivalent NN collisions and we can ensure that on the average we obtain the correct integrated non-diffractive and diffractive excitation cross sections, by shuffling the probabilities between the four combinations so for each the probability never exceeds unity, as explained in [283]. It should be noted that this trick does not allow us to determine the correct amount of elastic scattering, but these scattering are of less importance in a Glauber calculation.

In the end of the Glauber modelling, we have a long list of all potential NN subcollisions with an assigned type of interaction. These will now tell us how many, and of which kind of NN events we will generate using the normal pp minimum-bias framework in PYTHIA 8.3, to be merged together into a full HI collision event. The way this is done is as follows.

- Order all non-diffractive subcollisions in the NN impact parameter, b_{ij}, and iterate with increasing b_{ij}.

- If none of the nucleons has been involved in a non-diffractive subcollision with smaller b_{ij}, generate a (primary) non-diffractive subevent.

- If one of the nucleons has been involved in a previous subevent, generate a single-diffraction NN event corresponding to the diffractive excitation of the other nucleon (using a special modification as explained in [283]) and merge this with the corresponding previous subevent.

- If both of the nucleons are already in a generated subevent, do nothing.

When we merge a single diffraction subevent, we only add the diffractively excited subsystem, removing the elastically scattered nucleon. We also take some longitudinal momentum from the remnants of the primary event to ensure momentum-energy conservation.

In a similar way, we go through all double- and single-diffractive subcollisions, and add these to the full HI event. In the end, we take all non-interacting nucleons and collect them into projectile and target nucleus remnants, which each end up as a single entry in the event record with PDG-ID codes of the form 100*ZZZAAA*9, depending on the number of neutrons and protons, which in the PDG standard corresponds to a highly-excited nucleus.

It should be noted that all subevents above are generated on the parton level, which allows us to hadronize them together. This enables us the option to perform string shoving and rope formation (see sections 7.3.1 and 7.3.2) on the full HI partonic state.

The main use of the ANGANTYR model is to generate minimum-bias events. It is however, also possible to generate specific hard processes in HI events. If a hard process is specified by the user, the Glauber modelling will proceed as before, but (at least) one of the non-diffractive primary NN events will be replaced by a specific hard interaction event, and at the same the event will be reweighted by a factor given by

$$N_{\text{ND}} \sigma_{\text{hard}} / \sigma_{\text{ND}}, \tag{307}$$

where N_{ND} is the number of non-diffractive subcollisions. Note that for the specified hard processes, ANGANTYR treats pp, pn, np, and nn subcollisions separately, which is not the case for the minimum bias, where isospin symmetry is assumed.

By default, PYTHIA 8.3 will automatically initialize the ANGANTYR machinery as soon as one of the beams is specified to be a nucleus (using the PDG ID of the form 100*ZZZAAAI*, where I indicates the excitation level). It is possible to use the ANGANTYR machinery also for minimum-bias pp collisions, by setting `HeavyIon:mode = 2`.

Finally, it should be noted that only the most commonly used nuclei are defined by default in PYTHIA 8.3, but a user can easily define further nuclei. Note also that the beam energy of a nucleus is specified by giving the energy per nucleon, following the convention of the field.

7 Hadronization

Hadronization (often also referred to as fragmentation) is the process of turning the final outgoing, coloured partons into colourless hadrons. This transition is non-perturbative, and must be handled by models. In PYTHIA it is based on the Lund string model [289, 290], which is also historically the core of the JETSET/PYTHIA programs. Even though the core methods for string hadronization are identical to previous versions of PYTHIA, the past years have seen significant activity in the area of fragmentation dynamics, guided by the discovery of heavy-ion-like effects in hadronic collisions. In PYTHIA, these efforts have culminated in a multitude of models modifying the original Lund strings in the presence of other strings in an event.

7.1 The Lund String model

Results from lattice QCD support viewing the confining force field between a colour and an anti-colour charge, such as a q$\bar{\text{q}}$ pair, as a flux tube with potential energy increasing linearly with the distance between the charge and the anti-charge. As the partons move apart, energy is transferred from the partons at the ends of the string to the string itself, by $\kappa \approx 1$ GeV/fm.
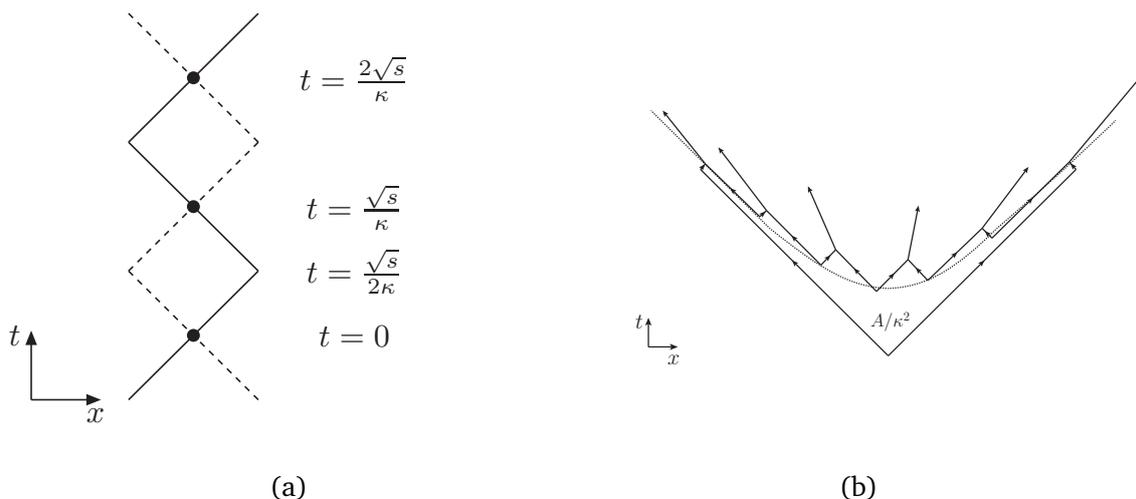
Figure 14: (a) The yoyo picture of a meson, at several steps in time as explained in the text. (b) A quark-antiquark string breaking into hadrons. The original pair is moving outwards along light-like trajectories. New $q\bar{q}$ pairs are produced around a hyperbola in (x, t), and combine into hadrons.

This directly gives rise to the so-called "yoyo" modes of single $q\bar{q}$ dipoles in 1+1 dimensions[7], as illustrated in fig. 14 (a). In the figure, an evolution starts at time $t = 0$, where all the energy is stored in the ends, and none in the string,

$$(E, p_x)_{q\bar{q}} = \frac{1}{2}(\sqrt{s}, \pm\sqrt{s}), \quad E_{\text{string}} = 0. \tag{308}$$

The string reaches its maximal extension at time $t = \sqrt{s}/2\kappa$. Here, all energy has been transferred from the end-points to the string:

$$(E, p_x)_{q\bar{q}} = (0, 0), \quad E_{\text{string}} = \sqrt{s}. \tag{309}$$

At time $t = \sqrt{s}/\kappa$, the string ends are back at their starting point, but with their momenta swapped compared to eq. (308), and finally at $t = 2\sqrt{s}/\kappa$, the string has been through a full period.

In the string picture, yoyo modes like this are identified as mesons, with flavour determined by their quark content (see section 7.1.1). Longer strings will break into hadrons, with new $q\bar{q}$ pairs breaking up the original string. Aligning the string axis of the original string with the x axis, this process is depicted in fig. 14 (b). The $q\bar{q}$ pairs are produced around a hyperbola, and joins together to form the hadrons, depicted as arrows. A hadron produced on the string is then characterized by two adjacent vertices (i and $i-1$), with space-time coordinates (x and t) correlated through the hadron mass (m):

$$m_i^2/\kappa^2 = (x_i - x_{i-1})^2 - (t_i - t_{i-1})^2. \tag{310}$$

In general, a string will break into a state with n hadrons, which in the model is given by the probability [291]:

$$d\mathcal{P} \propto \prod_{i=1}^{n} \left[N d^2 p_i \delta(p_i^2 - m^2) \right] \delta^{(2)}\left(\sum p_i - P_{\text{tot}}\right) \exp(-bA), \tag{311}$$

---

[7]The following convention for spatial coordinates is used. When discussing the 1+1 dimensional string, x is taken as the spatial coordinate. When we move on to discuss 3+1 dimensional strings, the coordinate z is chosen to be the coordinate along the string axis, as this will often coincide with the coordinate along the beam axis, which is normally denoted z.

where A is the area covered by the string before breakup in units of κ, as shown in fig. 14 (b), and b is a parameter. If the string breaking is imagined as an iterative process, the consistency constraint that the same result should be obtained (on average) by fragmenting from the left or the right, one obtains the distribution of momentum fraction (z) of remaining light-cone momentum taken by each hadron as

$$f(z) \propto \frac{(1-z)^a}{z} \exp\left(-\frac{bm^2}{z}\right),\tag{312}$$

where a is a new parameter related to N and b in eq. (311). Once transverse momenta are introduced, the substitution $m^2 \rightarrow m_\perp^2$ is performed, with the "transverse mass" defined by

$$m_\perp^2 = m^2 + p_\perp^2.\tag{313}$$

The resulting form of eq. (312) is known as the *Lund symmetric fragmentation function*. This simple picture of a $q\bar{q}$ system can be extended to topologies including gluons, without introducing new parameters, by viewing the gluon as a kink on the string in the $N_c \rightarrow \infty$ limit, with separate colour and anti-colour indices. A string can as such stretch from *e.g.* the quark end through a number of gluons, and end in the antiquark end [290].

While the default behaviour of PYTHIA is to always use eq. (312) with given values for the parameters a and b, the a parameter can in principle be different for each flavour. This possibility is implemented for s quarks and diquarks. Going from an old flavour i to a new flavour j, the fragmentation function would thus be modified as:

$$f(z) \propto \frac{z^{a_i}}{z}\left(\frac{1-z}{z}\right)^{a_j} \exp\left(-\frac{bm_\perp^2}{z}\right).\tag{314}$$

Finally, the Bowler modification [292] done in the Artru–Mennesier model [293] allows for massive endpoint quarks with mass m_Q. This modified the symmetric fragmentation function, as the areas swept out by massive endpoint quarks is reduced compared to massless ones. Though using this modification is a break with the Lund-string philosophy, it is available as an option, where an effective a term for a discrete mass spectrum [294] is used:

$$f(z) \propto \frac{1}{z^{1+r_Q bm_Q^2}} z^{a_\alpha}\left(\frac{1-z}{z}\right)^{a_\beta} \exp\left(-\frac{bm_\perp^2}{z}\right).\tag{315}$$

A common use case is to enable the Bowler modification for fragmentation for heavy quarks, as it can describe the somewhat harder spectrum better.

The derivation of eq. (312) also gives the probability distribution in proper time (τ) of $q\bar{q}$ breakup vertices, *i.e.* a quantity that can be interpreted as (input to) a hadron production time. In terms of $\Gamma = (\kappa\tau)^2$ it is:

$$\mathcal{P}(\Gamma)\mathrm{d}\Gamma \propto \Gamma^a \exp(-b\Gamma)\mathrm{d}\Gamma.\tag{316}$$

From this distribution it is possible to calculate the average breakup time of a $q\bar{q}$ string:

$$\langle\tau^2\rangle = \frac{1+a}{b\kappa^2}.\tag{317}$$

Default PYTHIA values for a and b give $\langle\tau^2\rangle \approx 2$ fm. The Γ_i values can be defined recursively

$$\Gamma_i = (1-z)\left(\Gamma_{i-1} + \frac{m_\perp^2}{z}\right),\tag{318}$$

with $\Gamma_0 = 0$.

7.1.1 Selection of flavour and transverse momentum

In the previous section, the $q\bar{q}$ pairs in the string breaking were treated as massless and without transverse momenta. If the quark and antiquark has a transverse mass, they can no longer be produced in a single vertex, but must tunnel through a forbidden region of size $m_\perp/\kappa$. The tunnelling probability can be calculated in the WKB approximation, giving [289]:

$$\frac{1}{\kappa}\frac{\mathrm{d}\mathcal{P}}{\mathrm{d}^2p_\perp} \propto \exp(-\pi m_\perp^2/\kappa) = \exp(-\pi m^2/\kappa)\exp(-\pi p_\perp^2/\kappa). \tag{319}$$

Here $m_\perp$ is the transverse mass of the quark, and the factorization of the result allows separation of the generation of m and $p_\perp$.

The relative production of light quarks of different mass, and thus of different flavour, could in principle be obtained directly by inserting u, d and s quark masses in eq. (319). It is, however, not obvious what quark masses to use. Current quark masses lead to too little strangeness suppression, and constituent quark masses lead to too much. Instead, the suppression is viewed as a free parameter, and tuned to LEP data. The current default s suppression relative to u or d types is 0.217, which does not imply unreasonable effective quark masses in eq. (319). Heavier quark flavours are suppressed too heavily to be produced in string breakings, for any reasonable value of their masses.

The generation of $p_\perp$ by eq. (319), can be implemented by giving the quark and antiquark Gaussian $p_\perp$-kicks with $\sigma^2 = \kappa/\pi \approx (0.25\,\mathrm{GeV})^2$. Fits to data have this number higher, around $\sigma = 0.35$ GeV, implying that a large fraction of the $p_\perp$ kick comes from another source, such as soft gluon radiation below the parton shower cutoff.

Besides production of the normal light-quark species, other hadron types can be produced through the same mechanism with a few modifications. Excited mesons are allowed by letting quarks and antiquarks combine to a total spin of either 0 or 1. Considering only pseudoscalar and vector multiplets, the expectation of the relative rate is 1 : 3, while data – at least in the case of $\pi : \rho$ – prefers a ratio about 1. This difference between prediction and data can be explained as a result of differences in the hadronic wave function [295, 296], but this comes at the expense of many free parameters, which have to be tuned to data. Baryons can be produced using eq. (319) as well, by allowing diquark-antidiquark string breakings [297]. Compared to the production of s quarks, this process will be suppressed by a larger (effective) diquark mass. In such an approach, the produced baryon-antibaryon pair will be neighbours along the string, and share two flavours. This simple picture is modified by considering an underlying step-wise mechanism for baryon production, first suggested by Casher, Neuberger and Nussinov [298], and realized in the "popcorn" model [299] in PYTHIA. In the popcorn model, diquarks are generated by first producing a $q\bar{q}$ pair as a vacuum fluctuation on the string, without breaking it. By producing more new $q\bar{q}$ pairs in between, meson production between the baryon-antibaryon pair is allowed. The whole process is illustrated in fig. 15. In principle, several mesons can be produced in between a baryon-antibaryon pair through the popcorn mechanism, but currently only the simplest case of a single meson is implemented in PYTHIA.

While this explanation above suffices for an introduction of the physics behind the model, there are many important implementation details to be faced when going from a "physics level" description of the Lund string to the actual implementation in PYTHIA, which must be able to handle arbitrarily complicated configurations of partons. In the next subsections we outline several of the more specialized features in PYTHIA string fragmentation, and the thought behind the implementation. While some are completely new models on top of the old hadronization framework, others remain the same as even the oldest version of the JETSET and PYTHIA 6.3 programs . Those specific parts of the discussion are therefore largely carried over from the PYTHIA 6.4 manual [14].
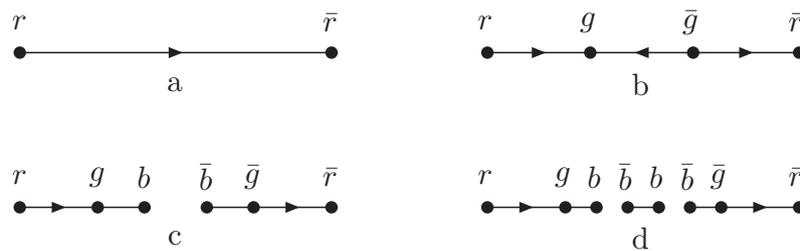
Figure 15: Illustration of step-wise popcorn production of a baryon-antibaryon pair, with a meson in between. In frame a), a string is spanned between a red-antired ($r\bar{r}$) q$\bar{\text{q}}$ pair, with colour flow indicated by the arrow. In frame b), a green-antigreen ($g\bar{g}$) q$\bar{\text{q}}$ pair has appeared as a vacuum fluctuation between them, reversing the colour flow in the central part of the string. In frame c), an additional pair is produced, breaking the string, and in frame d) another breakup produces a meson between the baryon and anti-baryon. Figure from [246].

7.1.2   Joining two jets in q$\bar{\text{q}}$ events

Keeping with the simple picture of a single q$\bar{\text{q}}$ pair, the iterative procedure obtained by successive application of eq. (312), is only valid when the remaining mass of the system, after fragmenting off a hadron, is large. If the algorithm implementing eq. (312) were to start from one end, and create hadrons successively until the other end is reached, the mass of the last hadron would be fully constrained by four-momentum conservation, and would therefore be off-shell.

   The practical route taken in PYTHIA, is to randomly fragment off hadrons from either the q or $\bar{\text{q}}$ end in each step, with z taken to be either the positive or negative light-cone momentum respectively. To wit, if the step is on the q side, z is the remaining $E + p_z$ fraction, and if the step is on the $\bar{\text{q}}$ side, z is the remaining $E - p_z$ fraction. Once the mass of the remaining system has dropped below a certain value, with some smearing to avoid an unphysical sharp cutoff, the remaining system is fragmented into two "final" hadrons, and the chain ends.

7.1.3   Fragmentation of systems with gluons

Most of the preceding discussion has involved the simple system of a single string spanned between a q$\bar{\text{q}}$ pair. While sufficient to explain the basic features of the model and implementation, it is far from covering the complexity in hadronization of multiparton systems. A Lorentz covariant algorithm exists, however, and in this section the machinery employed for this task is outlined, noting that the complete machinery is complicated, and covered in detail in [290, 300].

   The basis of the algorithm is to divide multiparton systems to be fragmented into smaller string pieces, spanned between individual partons. Consider a long string spanned between a q$\bar{\text{q}}$ pair (labelled 1 and n in the following), with a number of gluons in between (labelled $2, ..., n-1$). Such a string will contain $n-1$ separate pieces. The kinematics of those pieces are, as for simple q$\bar{\text{q}}$ strings, determined by the four-momenta of the endpoint partons. In the case of gluons, the four-momentum is shared between the two neighbouring string pieces, each taking half. It must furthermore be assumed that endpoint (anti-)quarks are massless, for the fragmentation algorithm to work. In practise this is done by attaching a fictitious string piece with a massless (anti-)quark to the string end, replacing the massive quark. This string piece in a later step becomes part of the massive hadron produced from the massive quark.

In summary, we have therefore $n-1$ string pieces defined by adjacent[8] four-momentum pairs (j, k), with the parton going towards the q end further indexed with a $+$ and the parton going towards the $\bar{\text{q}}$ end with a $-$. In general, a hadron is now created by taking a step from a region (j_1, k_1) to (j_2, k_2). A step may be taken within just a single region, or between two different regions. The resulting hadron four-momentum can be written as

$$p = \sum_{j=j_1}^{j_2} x_+^{(j)} p_+^{(j)} + \sum_{k=k_1}^{k_2} x_-^{(k)} p_-^{(k)} + p_{x1} \hat{e}_x^{(j_1 k_1)} + p_{y1} \hat{e}_y^{(j_1 k_1)} + p_{x2} \hat{e}_x^{(j_2 k_2)} + p_{y2} \hat{e}_y^{(j_2 k_2)}, \qquad (320)$$

where the four-momentum fraction of $p_\pm^i$ taken by the hadron is denoted $x_\pm^i$, and (p_x, p_y) are the transverse momenta produced at the string breaks according to eq. (319) with $(\hat{e}_x, \hat{e}_y)$ spacelike unit four-vectors normal to the string direction in the respective region.

The only remaining degree of freedom is z, to be determined by eq. (312). The interpretation of z is, however, only well-defined for a step in the initial string regions. But via eq. (318) a z value can be translated into a new $\Gamma = (\kappa\tau)^2$ value, and Γ is well defined across region boundaries. Together with the $p^2 = m^2$ constraint on eq. (320) this is sufficient to find the relevant $x_+^{(j_2)}$ and $x_-^{(k_2)}$ values of the next breakup vertex.

7.1.4 Hadron vertices

While the production vertices of hadrons are impossible to detect experimentally, calculating them still has applications in other parts of the simulation, most notably hadronic rescattering. In this section we describe the space-time picture for $q\bar{q}$ pairs, based on methods developed in [301].

From the linear potential $V(r) = \kappa r$, the equations of motion are

$$\left| \frac{dp_{z,\text{q}/\bar{\text{q}}}}{dt} \right| = \left| \frac{dp_{z,\text{q}/\bar{\text{q}}}}{dz} \right| = \left| \frac{dE_{\text{q}/\bar{\text{q}}}}{dt} \right| = \left| \frac{dE_{\text{q}/\bar{\text{q}}}}{dz} \right| = \kappa. \qquad (321)$$

The sign on each derivative is negative if the distance between the quark is increasing, and positive if the distance is decreasing. After sampling E_{h_i} and p_{h_i} for each hadron, these equations lead to simple relations between the space-time and momentum-energy pictures, $z_{i-1} - z_i = E_{h_i}/\kappa$ and $t_{i-1} - t_i = p_{h_i}/\kappa$, where z_i and t_i denote the space-time coordinates of the ith breakup point (note that $z_{i-1} > z_i$ since points are enumerated from right to left). In the massless approximation, the endpoints are given by $z_{0,n} = t_{0,n} = \pm\sqrt{s}/2\kappa$. This specifies the breakup points, but there is still some ambiguity as to where the hadron itself is produced. The default in PYTHIA 8.3 is the midpoint between the two breakup points, but it is also possible to specify an early or late production vertex at the point where the light-cones from the two quark-antiquark pairs intersect.

A complete knowledge of both the space-time and momentum-energy pictures violates the Heisenberg uncertainty principle. This is compensated for in part by introducing smearing factors for the production vertices, but outgoing hadrons are still treated as having a precise location and momentum. Despite not being a perfectly realistic model, there is no clear systematic bias in this procedure, and any inaccuracies associated with this violation are expected to average out.

There are several further complications to these process. One is more complicated topologies such as those involving gluons or junctions. Another is the fact that the massless approximation is poor for heavy $q\bar{q}$ pairs. For massive quarks, rather than moving along their

---

[8]It is possible to have string regions spanned by non-adjacent pairs as well, created when a gluon loses all its energy to the string. These regions form an integral part of the formalism, and help ensure that string fragmentation is rather insensitive to soft and collinear gluon emissions in the parton-shower stage.
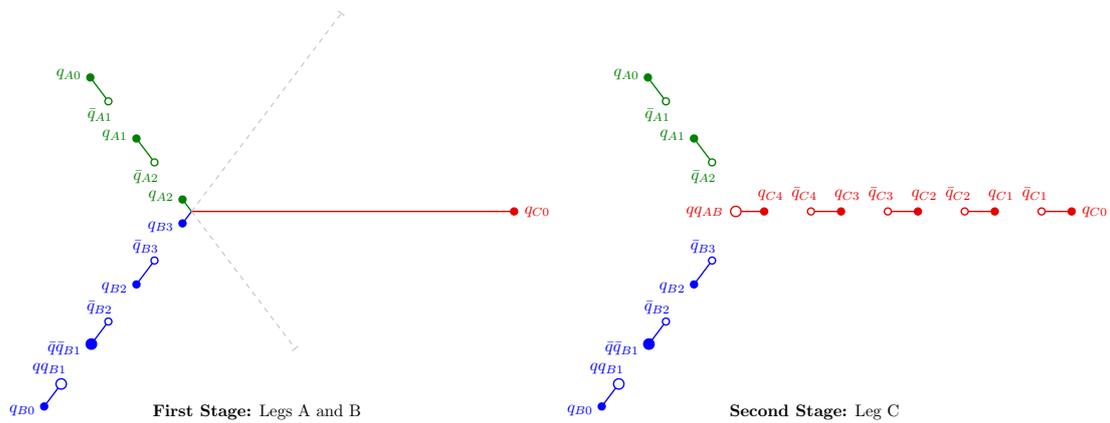
Figure 16: Illustration of the two main stages of junction fragmentation. (left) First, the junction rest frame (JRF) is identified, in which the pull directions of the legs are at 120° to each other. (If no solution is found, the CM of the parton system is used instead.) The two lowest-energy legs (A and B) in this frame are then fragmented from their respective endpoints inwards, towards a fictitious other end which is assigned equal energy and opposite direction, here illustrated by grey dashed lines. This fragmentation stops when any further hadrons would be likely to have negative rapidities along the respective string axes. (right) The two leftover quark endpoints from the previous stage (q_{A2} and q_{B3}) are combined into a diquark (qq_{AB}) that is then used as endpoint for a conventional fragmentation along the last leg, alternating randomly between fragmentation from the q_C end and the qq_{AB} end as usual.

light-cones, the quarks move along hyperbolas $E^2 - p_z^2 = m^2 + p_\perp^2 = m_\perp^2$. Both these issues are addressed in more detail in [301].

7.1.5 Junction topologies

Junction topologies in their simplest form arise when three massless quarks in a colour-singlet state move out from a common production vertex, a textbook example of which is given by a baryon-number-violating super-symmetric decay $\chi^0 \to$ qqq. In that case it is assumed that each of them pull out a string piece, a "leg", to give a Y-shaped topology, where the three legs meet in a common vertex, the junction. This junction is the carrier of the baryon number of the system: the fragmentation of the three legs from the quark ends inwards will each result in a remaining quark near to the junction, and these three will form a baryon around it.

The junction will be at rest in a frame where the pull of the three legs balance each other, which is when the angle between each quark pair is 120°. It is therefore convenient to handle the hadronization in such a frame. There is no first-principles description of junction-string fragmentation. Instead the process is split into a few steps, to make use of the existing string machinery in a credible manner [302], illustrated in fig. 16. First, the two lowest-energy legs are considered separately, each as if it were a q$\bar{q}$ string, with a fictitious $\bar{q}$ in the opposite direction to the q. All fragmentation is from the q end of the respective system, however, and keeps on going until almost all the original q energy is used up, resulting in the situation illustrated in the left-hand pane of fig. 16. At that stage the remaining unmatched two quarks (q_{A2} and q_{B3} in the figure) are combined into a diquark, carrying the unspent energy and momentum. This diquark now forms one end of the remaining string out to the third quark, which can be fragmented as a normal string system, illustrated in the right-hand pane of fig. 16. One criterion that the procedure works, *e.g.* that the fragmentation of the two first legs is

stopped at about the right remaining energy, is that the junction baryon is formed with a low momentum and with minimal directional bias in the junction rest frame. Additional checks are also made to ensure that the final string mass is above the threshold for string fragmentation. Otherwise, repeated attempts are made, starting over with the first two strings.

Unfortunately real-life applications introduce a number of complications. One such is that the pull is more complicated when the endpoints are not massless. Then, in a fraction of the events, there is no analytic solution. Typically this happens when a massive quark is almost at rest in the configurations that come closest to balance, and an approximate balance along these lines may be obtained. An even more complicated case is when a leg is stretched via a number of intermediate gluons between the junction and the endpoint quark, as would be a natural consequence of parton-shower evolution in the $\chi^0 \to$ qqq decay. Then the initial motion of the junction is set by the gluon nearest to it. But often this gluon has low energy and, once that is lost to the drawn-out string, it is the direction of the next-nearest gluon that sets a new net pull. Thus, there is no frame where the junction remains at rest throughout the whole fragmentation process. An effective average pull is then defined for each of the three legs, as a weighted sum of the respective parton momenta, where the weight drops exponentially as the energy sum of partons closer to the junction increases, *cf.* [302].

The absence of an exact solution for the junction rest frame leads to an approximate iterative procedure being used. One of the more common sources of PYTHIA warnings is that this procedure does not converge. If no fix can be found any other way, then ultimately the centre-of-mass frame of the system is taken as the junction rest frame.

Junction fragmentation is not only a topic for exotic physics, but very much part of ordinary QCD hadronic physics. It appears if two valence quarks are kicked out of a baryon beam by the MPI machinery. Since these interactions typically involve colour exchange, two of the ends will stretch to partons from the other incoming beam, unless colour reconnection gives another result. The fragmentation follows the already outlined procedure, which can lead to the beam baryon number being transported in to the central region of the event, *cf.* [248].

Also antijunctions may exist, where the colour lines from three antiquarks meet, and such antijunctions carry a negative baryon number. A string system may contain both a junction and an antijunction, or even multiple of such. The simplest such topology is when one leg connects a junction to an antijunction, leaving two other junction legs to quarks and two antijunction legs to antiquarks. It is here assumed that the total string length (see section 7.2) is smaller for such a topology than for having two simple q$\bar{\text{q}}$ strings, or else the junction pair would annihilate to give the simple string topology, *cf.* [248, 302]. Conversely, when the string length can be reduced, more-or-less parallel q$\bar{\text{q}}$ strings may colour reconnect into junction-antijunction systems, see further section 7.2.2.

To reduce the complexity of multijunction fragmentation, each system is split up into smaller ones that only contain (at most) one junction or antijunction each. Consider *e.g.* a junction-antijunction topology. If the leg connecting the two contains at least one gluon, it can be split up by a replacement g $\to$ q$\bar{\text{q}}$. If not, a small amount of energy can be shuffled from the regular q and $\bar{\text{q}}$ legs into some energy (and momentum) for this connecting leg, so that it can be split.

Another subtlety concerns what spin state to choose for the diquark that is formed at the end of the fragmentation of the two first legs, the one labelled qq_{AB} in fig. 16, which we will call the junction diquark. For conventional (non-junction) fragmentation, empirically one finds that $S = 1$ diquark states are heavily suppressed, interpreted as due to significantly higher masses and smaller binding energies. However, unlike in conventional string breaks, where diquark-antidiquark pairs are formed together in a *single* coherent tunnelling process (modulo fluctuations such as in the popcorn scenarios), the junction diquark is formed by combining the leftovers from two *separate* string breaks; PYTHIA 8 therefore allows for the

$S = 1$ suppression factor for junction diquarks to be set independently of that for conventional diquarks. Moreover, analogously to in the meson sector, it can be set independently for b-, c-, s-, and light-flavoured junction diquarks, where the label always refers to the heaviest of the two constituents.

It is also worth emphasizing that, within the context of the current PYTHIA modelling, junctions represent the sole mechanism for producing baryons containing multiple heavy flavours, such as Ξ_{cc}, Ω_{cc}, Ω_{ccc}, and their b-flavoured relatives. Note, however, that this will still be quite rare; since heavy flavours cannot be produced by string breaks, they can only appear as endpoints, say q_{A0} and q_{B0} in fig. 16. The only possibility to form a double-heavy-flavoured baryon involving these is if there is too little energy in *both* legs A and B for any other string breaks to occur, so that q_{A0} and q_{B0} are combined directly into the junction diquark, which is then doubly-heavy flavoured. We note that, so far, no dedicated emphasis has been placed on developing the heavy-quark aspects of junction fragmentation, though that may change with experimental interest. Predictions should therefore be regarded as tentative.

In summary, the full machinery for junction hadronization is convoluted and not without weaknesses, but overall it serves its purpose, and finds use in several physics contexts.

7.1.6 Small-mass systems

If the invariant mass of the $q\bar{q}$ system is small, a few complications to the fragmentation process can arise. For example, for an $s\bar{s}$ system at 0.9 GeV, the string cannot fragment as there is not enough energy to form an outgoing K $\overline{\text{K}}$ pair, nor can the quarks enter a "yoyo motion" as there is no hadron with compatible mass and flavour content. Furthermore, even if the string can fragment, at low energies the available phase space might be so small that the fragmentation algorithm becomes very inefficient. These situations can occur for instance towards the end of a parton shower by g $\rightarrow$ q$\bar{q}$ branchings or during hadronic rescattering, and are handled using approaches inspired by cluster fragmentation [303].

To improve the efficiency of the algorithm, the first step is to assume that the string will break at only a single point, and a few attempts are made to find possible outgoing two-hadron states. If these attempts fail, next the algorithm tries to form a single hadron from the endpoints, then put that hadron on-shell by transferring momentum to or from a neighbouring string system. If no momentum rearrangement is possible, further attempts are made to find possible two-hadron states, but now only the lightest possible hadrons for the given flavour content are considered. If this still does not work, the string may collapse to the lightest possible hadron given the endpoints, and produce one additional π^0. Finally, if this is not possible either, the last resort is to collapse the string to the lightest possible hadron, and transfer momentum with a neighbouring parton or hadron.

String systems are handled in order of increasing mass relative to the two-body threshold, so normally other systems are still unfragmented when addressing this kind of issue. Especially in (low-energy) hadronic rescattering there may two low-energy strings. Then, when the first string is handled, its collapse may reduce the mass of the other string. In this case, that system may also collapse to a single hadron, which is put on-shell by transferring momentum with a hadron from the previously fragmented string.

7.2 Colour reconnections

In PYTHIA (and other event generators), a simplified set of rules for colour flow is used to determine between which partons confining potentials should arise. In the context of the string model, this determines a unique string topology which sets the stage for the subsequent hadronization.
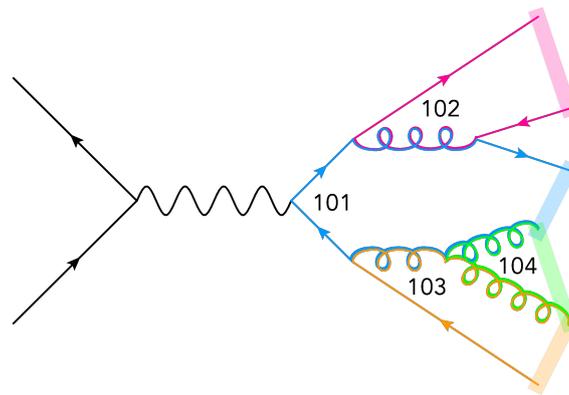
Figure 17: Illustration of LC colour flow in a simple $e^+e^- \to q\bar{q} \otimes$ shower event. The shaded regions represent the resulting unique LC string topology.

Specifically, all perturbative processes (including MPIs, ISR and FSR) are handled in a *leading colour* (LC) limit in which the probability for any two random colours to both be the same vanishes. Formally, this is done by taking the limit $N_c \to \infty$ with $\alpha_s N_c$ kept fixed [55] so that QCD amplitudes retain their $N_c = 3$ normalizations. This accomplishes two things: 1) it eliminates colour-interference effects which are suppressed by powers of $1/N_c^2 \to 0$, and 2) it allows for a particularly simple representation of gluons in colour space, as direct products of a colour and an anticolour, since the weight of the singlet in $N_c \otimes \bar{N}_c = (N_c^2 - 1) \oplus 1$ vanishes as $N_c \to \infty$.

In the LC limit, Feynman-diagram amplitudes in colour space are represented by products of independent "colour lines". Each of these expresses conservation of a distinct colour charge, and is represented by a δ_{ij} connection between partons carrying colours i and j (suitably crossed). We call this an LC dipole connection. Due to the orthogonality of the basis states and the lack of interference in this limit, each such line translates directly to a coherent colour-singlet structure at the colour-summed amplitude-squared level, which is confining at large distances. Thus, each LC dipole emerging from the perturbative stages of the event evolution can be uniquely mapped to a string piece (discussed further, *e.g.* in [53]). We use the term "colour *re*connection" (CR) to refer to any scenario that results in changes relative to this map in defining the starting configuration of hadronizing strings in an event.

A simple illustration of the map between LC dipoles and string pieces, for an $e^+e^- \to \gamma^*/Z \to q\bar{q}q'\bar{q}'gg$ event, is shown in fig. 17. Matching colour (and anticolour) charges are represented by Les Houches colour (and anticolour) tags [304, 305] numbered from 101–104 in this example and indicated by coloured lines in the diagram. In keeping with the $N_c \to \infty$ nature of the LC limit, the number of different tags is not limited to three, and each new tag is distinct from all others. This produces a unique set of colour connections which can be traced to form the LC string topology (shaded regions).

In hadronic collisions, the structure of the beam remnants is also to be modelled, after MPIs have extracted multiple coloured objects from them. Here it is useful define rules on how to equate some of these colours and anticolours with each other, so as to keep the total colour charge of a remnant within reasonable bounds. Note that this would still classify as "colour connection", insofar as it is the initial assignment of remnant colours, although the consequences propagate in from the remnants to the central perturbative interactions. This is discussed further in the section on beam remnants, section 6.3. As used in this section, the term CR applies to models that go beyond this, *i.e.* that allow for departures from the simple colour rules discussed above and/or address ambiguities that are left unresolved by them. CR may be classified as one example of a broader palette of string interactions, with other

examples presented in section 7.3.

Note that, occasionally, "junction" structures (see section 7.1.5) may also be present. Unlike dipole-type δ_{ij} connections, junctions (and antijunctions) represent ϵ_{ijk} structures in colour space; these are explicit $N_c = 3$ structures which have no analogy in the $N_c \to \infty$ limit. In PYTHIA, they can appear in the initial state in proton beams [248], in hard BSM processes (or decays) with baryon number violation [82, 302], and/or as a product of colour reconnections in the final state (in pairs of junctions and antijunctions to conserve overall baryon number) [249]. Due to the added technical complexity of dealing with junction structures, the latter possibility is, however, so far only invoked by the QCD-based CR model, *cf.* section 7.2.2.

Several different scenarios are included in PYTHIA, as described in the following subsections, each with its own motivations and underpinnings. The unifying feature is that these models act only by reassigning colours, with no explicit momentum exchanges between the involved partons. The decisions whether and how to reassign still can depend both on momentum-energy and on space-time relations between partons. Also, the changes at the level of produced hadrons still can be dramatic, due to the changed lengths and orientations of the resulting hadronizing strings.

Historically, CR was first discussed in the context of charmonium production [306–308], notably in weak B decay to J/ψ, *e.g.* $\overline{\text{B}}^0 = b\overline{d} \to W^-c\overline{d} \to s\overline{c}c\overline{d} \to J/\psi\overline{K}^0$. In such decays the c and $\overline{c}$ belong to two separate colour singlets, but ones that overlap in space-time, with the possibility of soft gluon exchange to create the new singlets.

The first large-scale application of CR was in the PYTHIA MPI model of hadronic collisions [13], notably to explain the increasing mean transverse momentum $\langle p_\perp \rangle$ with increasing charged multiplicity n_{ch} observed at CERN's Sp$\overline{\text{p}}$S collider [309]. If all MPIs draw out strings and fragment in the same manner, $\langle p_\perp \rangle(n_{ch})$ would be essentially flat. CR was therefore introduced in such a way that the total string length is reduced. Each further MPI then on the average increases n_{ch} less than the previous one, while giving the same $p_\perp$ from (mini)jet production, resulting in an increasing $\langle p_\perp \rangle(n_{ch})$.

LEP 2 offered a good opportunity to search for CR effects. Specifically, in a process $e^+e^- \to W^+W^- \to q_1\overline{q}_2q_3\overline{q}_4$, CR could lead to the formation of alternative "flipped" singlets $q_1\overline{q}_4$ and $q_3\overline{q}_2$, and correspondingly for more complicated string topologies, formed when parton showers are included. Such CR would be suppressed at the perturbative level, since it would force some $W^\pm$ propagators off the mass shell [310]. This suppression would not apply in the soft region. Based on a combination of results from all four LEP collaborations, the no-CR null hypothesis is excluded at a 99.5% CL [311]. Within the SK I scenario, described below, the best description is obtained for ∼50% of the 189 GeV W^+W^- events being reconnected, in qualitative agreement with predictions.

More recently, Tevatron [312] and LHC [313,314] measurements of the top-quark mass in hadronic top-quark decays brought CR effects on precision observables to the fore again, with several new models geared towards the increased complexity of hadron collisions produced first in PYTHIA 6 [315–317] and later in PYTHIA 8 [249,318]. Hadronic reconstruction of the top-quark mass remains an important impetus for further explorations of CR model space and for the development of systematic and exhaustive ways to constrain modelling ambiguities and parameters experimentally.

The importance of colour algebra versus dynamics differs widely between models. Taking the simple W^+W^- case above, there is a 1/9 probability that $q_1\overline{q}_4$ and $q_3\overline{q}_2$ are singlets purely by colour algebra. But such accidental singlets do not stop $q_1\overline{q}_2$ and $q_3\overline{q}_4$ from still being singlets as well; so nevertheless, a dynamics principle would be needed to decide which singlet set takes precedence when it is time to hadronize. Furthermore, once parton showers are included, the number of colour charges in an event increases, and the possibilities for CR with it. In the extreme limit, a string may be viewed as a chain of (non-perturbative) gluons

infinitesimally closely spaced, such that the string constantly flips colour, so there would be no suppression of CR for lack of nearby matching colours.

In several of the models below the concept of a "string-length" λ plays a prominent dynamics role. It is a measure of how many hadrons of some reference hadronic mass m_0 there are room for (in phase space), if the hadrons are evenly spaced in rapidity along the string. For a simple $q\bar{q}$ string of mass $m_{q\bar{q}}$ one possible definition is $\lambda = \ln(m_{q\bar{q}}^2/m_0^2)$. In principle, λ is well defined also for more complicated string topologies [291], but in practice its construction is too complicated. Instead, approximate expressions are used, like

$$\lambda \approx \sum_{i=0}^{n} \ln\left(1 + \frac{m_{i,i+1}^2}{m_0^2}\right), \quad m_{i,i+1}^2 = (\epsilon_i p_i + \epsilon_{i+1} p_{i+1})^2, \; \epsilon_q = 1, \; \epsilon_g = \frac{1}{2}, \tag{322}$$

for a string $q_0 g_1 g_2 \cdots g_n \bar{q}_{n+1}$, where $\epsilon_g = 1/2$ because gluon momenta are shared between two string pieces. The addition of 1 is to ensure that a low-mass section does not give a negative contribution, and is not always used. More generally, if low masses are common, it probably signals that there is a larger underlying issue, *e.g.* having too low a cut-off for shower evolution.

One may think of λ as the "free energy" of a string system, available for particle production. More specifically, the classical (Nambu–Goto) string action to be minimized is proportional to the worldsheet area spanned by the string motion. Here, λ is then related to the average area spanned before hadron formation. Therefore, it is often assumed that nature prefers a low string length whenever possible.

This principle is significantly violated in the early stages of a collision, however. There the hard interaction and the MPIs turn two incoming beam-hadron states, each with minimal λ, into a combined one of a significantly higher value. Consequently, we discuss here the aftermath, where strings begin to be pulled out and eventually fragment back to hadrons of low λ.

Similarly, some general considerations of the space-time picture are necessary. One is that the spatial evolution of showers need not be traced. That is, parton showers occur at time scales sufficiently shorter than hadronization ones so that, to first approximation, all the final partons can be viewed as emerging from a common vertex. Furthermore, while the branching of a low-mass high-energy parton can be significantly displaced, the daughters will tend to be sufficiently close, by any distance measure, such that CR is unlikely to break them apart. Another issue is how the lifetime of intermediate resonances compares with the CR time. The W, Z, and t have intermediate decay time scales, about an order of magnitude shorter than typical hadronization times. (Whereas the H is much more long lived.) But the two would become more comparable if time is added for the decay products to expand and begin interacting with the environment given by the rest of the pp collision. Ideally, the situation should therefore be simulated dynamically, where different time orderings are possible outcomes, but that would be fraught with uncertainties and is typically not done. Instead, a more common option is to allow only early or only late resonance decays, *i.e.* before or after hadronization. In early decays, all partons can reconnect, while in late decays the resonance decay products cannot.

7.2.1 The MPI-based model

The first CR model implemented in PYTHIA 8, and currently still the default, attempts to reduce λ by a complete merge of the partons of separate MPI systems. The probability for two MPIs to be reconnected this way is a function of the lower $p_\perp$ scale of the two, of the form

$$P_{\text{rec}}(p_\perp) = \frac{(R_{\text{rec}} p_{\perp 0})^2}{(R_{\text{rec}} p_{\perp 0})^2 + p_\perp^2}, \tag{323}$$

where $p_{\perp 0}$ is the parameter introduced in eq. (253) to damp the $p_{\perp} \to 0$ infinity of the QCD $2 \to 2$ cross section, and R_{rec} is a phenomenological parameter. An R_{rec} of order unity would seem reasonable; empirically somewhat larger values are found. The reconnection probability is chosen to be higher for soft systems, reflecting that the latter are described by more extended wave functions, thus having a higher probability to overlap and interact with other systems.

Now consider an event containing n MPIs, which have been generated in order of falling $p_{\perp}$, $p_{\perp 1} > p_{\perp 2} > \ldots > p_{\perp n}$. The reconnections are then done in a two-step procedure, as follows.

First, the MPI systems are tested for reconnection in sequence of increasing $p_{\perp}$, *i.e.* starting with system n. For an arbitrary m, $2 \leq m \leq n$, the reconnection probability $P_m = P_{\text{rec}}(p_{\perp m})$ is used to decide whether system m should be merged with $m - 1$ or not. If not, the same relative probability holds for a merger with $m - 2$, and so on to the top. That is, there is no explicit dependence on the higher $p_{\perp}$ scale, but implicitly there is via the survival probability of not already having been merged with a lower-$p_{\perp}$ system. In total, the probability for m not to merge therefore is $(1 - P_m)^{m-1}$. Note that mergings may cascade: if m is merged with l, $1 < l < m$, then l in its turn may be merged with an even-higher-$p_{\perp}$ system k, $1 \leq k < l$, and then also m counts as merged with k.

Second, once it has been decided which systems should be reconnected, the actual merging is carried out in the opposite direction. That is, first the hardest system is studied, and all colour dipoles (i, k) in it are found, as usual in the $N_c \to \infty$ limit. This includes those to the beam remnants, as defined by the holes of the incoming partons. Then consecutively, each softer system to be merged with it is considered in order of decreasing $p_{\perp}$. For each such system, the gluons j are inserted, in order of decreasing gluon p_T, into the dipole (i, k) that minimizes the increase in the λ measure for the harder system

$$\Delta \lambda = \lambda_{j;ik} \equiv \lambda_{ij} + \lambda_{jk} - \lambda_{ik} = \ln \frac{(p_i \cdot p_j)(p_j \cdot p_k)}{(p_i \cdot p_k)m_0^2} . \tag{324}$$

Note that the first term of eq. (322) is not required here, since an $E_k \to 0$ (for fixed relative angles) would affect all $\lambda_{j;ik}$ the same way and thus not alter the choice of the winning (i, k) dipole. Although gluons dominate, MPIs may also contain quarks. Those $q\bar{q}$ pairs that originate from the splitting of a gluon can be inserted into the higher-$p_{\perp}$ system by the same criterion as would have been used for such a gluon. The (few) other quarks are not affected by the CR procedure, but remain for the beam-remnant handling to address.

The CR procedure is carried out before resonance decays are considered by default, *i.e.* the late decay option introduced above. It is possible to switch to early decays, however.

7.2.2 QCD-based colour reconnections

As discussed in the introduction to section 7.2, during the perturbative stages of the event evolution, LC colour flow is used to keep track of which partons are colour connected to each other. In the LC limit, each colour tag is matched by only a single unique anticolour tag in the event (or a combination of two colour tags, if junctions are present). At the perturbative level, these connections represent LC dipoles/antennae, and they are one-to-one mapped to string pieces at the non-perturbative stage, enforcing colour confinement.

Beyond the LC limit however, there should be a finite probability also for LC-unconnected partons to "accidentally" find themselves in a colour-singlet state, or in some other coherent state with a lower total colour charge than the scalar sum of their individual charges. This

follows from the **SU**(**3**) colour-algebra rules:

$$\mathbf{3} \otimes \overline{\mathbf{3}} = \mathbf{8} \oplus \mathbf{1}\,, \tag{325}$$

$$\mathbf{3} \otimes \mathbf{3} = \mathbf{6} \oplus \overline{\mathbf{3}}\,, \tag{326}$$

$$\mathbf{3} \otimes \mathbf{8} = \mathbf{15} \oplus \overline{\mathbf{6}} \oplus \mathbf{3}\,, \tag{327}$$

$$\mathbf{8} \otimes \mathbf{8} = \mathbf{27} \oplus \mathbf{10} \oplus \overline{\mathbf{10}} \oplus \mathbf{8} \oplus \mathbf{8} \oplus \mathbf{1}\,, \tag{328}$$

where the representations that correspond to a coherent addition of charges (with lower total charge) are highlighted in red. In the LC limit, colour-unconnected quark-antiquark pairs are never allowed to form a singlet; they are always in an overall octet state, while quark-quark, quark-gluon, and gluon-gluon ones are in sextet, quindecuplet, and vigintiseptet states, respectively.

The starting point for the QCD-based CR scheme [249] is that slightly simplified versions of eqs. (325) to (328) can be used to compute probabilities for LC-unconnected partons to *stochastically* enter into coherent states with one another. This does not invalidate the LC colour topology, but it does allow for (potentially many) *other* viable mappings of the same parton system to different string configurations. Optionally, configurations that involve (re)connections between systems with large relative boosts can be excluded if deemed to be in conflict with causality, as discussed further below. The model then chooses between the remaining allowed configurations by selecting the one that minimizes the λ measure, eq. (322). In principle, one could allow fluctuations around this, but that is not currently done in the model.

A characteristic feature of this model is that it provides a qualitatively new mechanism for the creation of baryon-antibaryon pairs, in addition to the conventional mechanism of string breaks to diquark-antidiquark pairs. The $\overline{\mathbf{3}}$ in eq. (326), the $\overline{\mathbf{6}}$ in eq. (327), and the two decuplets in eq. (328) represent colour states that involve colour-epsilon structures. In the context of the string model, these map to string junctions (and antijunctions), around which baryons will form, *cf.* section 7.1.5 and [302]. As a consequence, the effective baryon-to-meson ratio increases with the amount of CR in this model, and hence more active events (*e.g.* with many MPI) will generally exhibit higher baryon fractions. Note that colour conservation implies that the model always creates equal numbers of baryons and antibaryons; these pairs can, however, be well separated in phase space, contrary to the more localized nature of the conventional diquark string breaks. Moreover, the model also allows for the formation of doubly-heavy-flavour baryons such as Ξ_{bc}, a possibility that does not occur within the conventional diquark-type string breaks. In the current formulation of the model, however, no special attention has been devoted to questions specific to heavy quarks, hence this aspect should be considered to be associated with substantial uncertainty.

At the technical level, the model approximates the QCD probabilities expressed by eqs. (325) to (328) by randomly assigning an index between 0–8 to each Les-Houches colour tag, subject to the requirement that gluons must have different colour and anticolour indices. Any parton pairs with matching colour and anticolour indices are then considered to be in relative singlets and are candidates for dipole-type string pieces. (This mimics the representation first proposed in [319].) Stochastically, this reproduces the $\frac{1}{9}$ probability of eq. (325) exactly.

The algorithm starts from the LC topology and considers each index group in turn, working its way down from high to low dipole invariant masses, at each step considering all allowed possibilities and executing a swap if that lowers the total λ measure. Note that $q\bar{q}$ pairs originating directly from $g \rightarrow q\bar{q}$ branchings are also excluded from having the same index. Consequently, quarkonium formation from such pairs is not expected in this model in its current formulation.

If junction-type reconnections are enabled, the algorithm then works its way through three

separate groups of indices: [0,3,6], [1,4,7], and [2,5,8] (chosen so that they are trivial to separate using the modulo 3 operation). Within each of these groups, any partons carrying two different colour indices (say, 0 and 3) are allowed to add coherently to the overall anticolour of the third (say, -6) and enter into corresponding junction-type reconnections if that reduces the λ measure. This enables a decent ($\frac{2}{9}$) approximation to the probability for junction-type reconnections, but does underestimate the true QCD group weights somewhat, see [249]. This procedure (dipole-style reconnections followed by junction reconnections) is iterated until no more favourable reconnections are identified.

In addition to the colour rules, the dipoles also need to be causally connected in order to perform a reconnection. The definition of causally connected dipoles is not exact, and several different options are available. All the time-dilation modes introduce a tunable parameter, which provides a handle on the overall amount of CR.

When the two strings are allowed to reconnect, they will reconnect if it lowers the total string length, as defined by an approximation to the λ measure. Several options for different approximations are available. The λ measure is not well understood, especially for junction structures, and a tunable parameter allows for the enhancement or suppression of junction-type connections to dipole ones. This affects how many baryons are generated by the model. See also the description of junction fragmentation in section 7.1.5.

Although the main objective of the model is to treat reconnections involving large invariant masses, there is of course a tail towards small masses as well. For very low masses $< \mathcal{O}(1\,\mathrm{GeV})$, string fragmentation becomes technically complicated (as each hadron needs to straddle several gluon "kinks"), especially when junctions are involved, and also the approximations made in the λ measure are not particularly reliable. Therefore, reconnections involving string pieces with masses below m_0, *cf.* eq. (322), are excluded from participating in the CR framework. (Technically, partons making up such low-mass systems are treated collectively as a single pseudo-particle for the purpose of reconnections.)

7.2.3 The gluon-move scheme

In the effort to determine the top mass as accurately as possible, CR is one of the major sources of systematic error. To better understand the situation, a range of new models were developed and implemented in [318]. Many of these are crude straw-man models, or applicable only to top decay. They are therefore not integrated as standard options, but may be obtained by using the `ColourReconnectionHooks.h` plugin; see `main29.cc` for an example.

In the late resonance decays approach it is possible to use separate CR models for the underlying event and for the top decay products. Then two collections of gluons are constructed, one with the gluons radiated from the top decay products and the other with the gluons from the rest of the event. Iterating over the former set in random order, a random fraction of the gluons from the top are forced to exchange colours with a gluon from the rest of the event. The latter gluon can be picked according to one of five different criteria, *(i)* at random, *(ii)* giving the smallest invariant mass, *(iii)* giving the largest invariant mass, *(iv)* giving the smallest (with sign) $\Delta\lambda$ value, or *(v)* as *(iv)* but only if $\Delta\lambda < 0$.

For early resonance decays, three possible operations were implemented, swap, move, and flip. The latter two are implemented in the main body of PYTHIA.

The swap model is similar to option *(iv)* above. In it, a random fraction of all final-state gluons are chosen for possible reconnection. For each gluon pair j and m in this set, on dipoles (i,k) and (l,n) respectively, one calculates the difference $\Delta\lambda$ resulting from a swap of the two gluon colours

$$\Delta\lambda(j,m) = \lambda_{m;ik} + \lambda_{j;ln} - \left(\lambda_{j;ik} + \lambda_{m;ln}\right) = \lambda_{im} + \lambda_{mk} + \lambda_{lj} + \lambda_{jn} - \left(\lambda_{ij} + \lambda_{jk} + \lambda_{lm} + \lambda_{mn}\right).$$

(329)

A corresponding reconnection is performed if $\min_{j,m} \Delta\lambda(j,m) \leq \Delta\lambda_{\mathrm{cut}}$. Here, $\Delta\lambda_{\mathrm{cut}} \leq 0$ is a free parameter that relates to the generated CR rate. The procedure is iterated over until no allowed swaps remain.

The closely related move model works as follows. Again a random fraction of all final-state gluons are selected. The change in the string length $\Delta\lambda$ is calculated for each such gluon j on a final-state dipole (i,k), if the gluon is moved to any other final-state dipole (l,n):

$$\Delta\lambda(j, ik \to ln) = \lambda_{j;ln} - \lambda_{j;ik} = \lambda_{lj} + \lambda_{jn} + \lambda_{ik} - \left(\lambda_{ij} + \lambda_{jk} + \lambda_{ln}\right). \tag{330}$$

Here, the minimum is found as $\Delta\lambda_{\min} = \min_{j,l,n} \Delta\lambda(j, ik \to ln)$, and the move is carried out if $\Delta\lambda_{\min} \leq \Delta\lambda_{\mathrm{cut}}$. This is then repeated as long as the latter criterion is fulfilled.

There is some fine print. If a colour-singlet subsystem consists of two gluons only, then it is not allowed to move any of them, since that would result in a colour-singlet gluon. Also, at most as many moves are made as there are gluons, which normally should be enough. A specific gluon may be moved more than once, however. Finally, a gluon directly connected to a junction cannot be moved, and also no gluon can be inserted between it and the junction. This is entirely for practical reasons, but should not be a problem, since junctions are rare in this model.

Neither the swap nor move methods reconnect quarks. That is, a $q\bar{q}$ pair starting out at the opposite ends of a string will remain connected throughout. The intermediate gluons on this string can change, however, and in the move model even the number of such gluons. A flip step can be added after the swap or move ones, to also reconnect quarks. There, two string pieces, (i,k) and (l,n), are connected to (i,n) and (l,k). For any two separate colour-singlet subsystems this gives

$$\Delta\lambda_{\min} = \min_{i,k,l,n} \left[\lambda_{in} + \lambda_{lk} - (\lambda_{ik} + \lambda_{ln})\right]. \tag{331}$$

The system pair with smallest $\Delta\lambda_{\min}$ is allowed to flip, as long as $\Delta\lambda_{\min} \leq \Delta\lambda_{\mathrm{cut}}$. Singlet systems cannot undergo a second flip. Junction topologies are either included or excluded among the allowed flip possibilities. It is also possible to switch on/off move and flip separately.

7.2.4 The SK models

The SK I and SK II models [310, 320] were specifically developed for $e^+e^- \to W^+W^- \to q_1\bar{q}_2q_3\bar{q}_4$ at LEP 2, and work (almost) equally well for an $(\gamma^*/Z)(\gamma^*/Z)$ intermediate state. They are not intended to handle hadronic collisions, however, except in special contexts. The prime example is Higgs decays of the same character as above, $H \to W^+W^-/ZZ$, since the Higgs is so long lived that its decay is decoupled from the rest of the event [321].

The labels I and II refer to the colour-confinement strings being modelled either by analogy with type I or type II superconductors. In the former case the strings are viewed as transversely extended "bags" [322]. The reconnection probability is then related to the space-time-integrated overlap of the W^+ and W^- string pieces. In the latter model, strings are instead assumed to be analogous with vortex lines, where all the topological information is stored in a thin-core region. Then, reconnection only can occur when these cores pass through each other.

The imagined time sequence is as follows. The W^+ and W^- move apart from their common production vertex and decay at some distance. A perturbative parton shower evolves from an original $q\bar{q}$ pair produced at each decay vertex. The typical distance that a virtual parton travels before it branches is shorter than the fragmentation time, by a factor of five or more. Note that a Lorentz-boosted virtual parton gives rise to a string piece inheriting this boost factor. Thus, the branching-to-fragmentation time separation remains in all parts of an event, even if one part started hadronization while another part still is in the shower stage,

if viewed *e.g.* in the event rest frame. To a decent first approximation, each W can therefore be viewed as instantaneously decaying into a string spanned between the partons at the end of the shower. The strings expand transversely and longitudinally, (at most) by the speed of light, and fragment into hadrons and disappear. The W^+ and W^- string pieces may overlap before that happens, however, and then a reconnection may occur in the overlap region.

The reconnection probability in scenario I is proportional to the integrated space-time overlap of the W^+ and W^- strings, with saturation at unit probability. In more detail, first consider a string piece expanding along the $\pm z$ direction. Then the colour field strength is taken to be

$$\Omega(\mathbf{x}, t) = \exp\left\{-(x^2 + y^2)/2r_{\text{had}}^2\right\}\ \theta(t - |\mathbf{x}|)\ \exp\left\{-(t^2 - z^2)/\tau_{\text{frag}}^2\right\}, \tag{332}$$

where $\mathbf{x} = (x, y, z)$. The first factor gives a Gaussian falloff in the transverse directions. The assumed string width $r_{\mathbf{had}} \approx 0.5$ fm corresponds to a hadron radius of $\sqrt{3}\, r_{\mathbf{had}}$. The time retardation factor $\theta(t - |\mathbf{x}|)$ ensures that information on the W decay spreads outwards with the speed of light. The last factor gives the no-fragmentation probability at a given proper time along the string axis, where $\tau_{\text{frag}} \approx 1.5$ fm follows from the default a and b fragmentation-function parameter values. The combined field-strength expression must be appropriately rotated, boosted, and displaced to the respective W decay vertex for each string piece. A complication is that two string pieces from the same W may well overlap in a space-time point, by convoluted string drawings or by imperfections in the modelling at a gluon kink. Then the string field strength $\Omega_{\text{max}}^+(\mathbf{x}, t)$ is defined as the maximum of all the contributing Ω^+'s at the relevant point. From this, the probability for a reconnection to occur is chosen to be

$$\mathcal{P}_{\text{recon}} = 1 - \exp\left(-k_{\text{I}} \int \mathrm{d}^3\mathbf{x}\, \mathrm{d}t\ \Omega_{\text{max}}^+(\mathbf{x}, t)\, \Omega_{\text{max}}^-(\mathbf{x}, t)\right), \tag{333}$$

with k_{I} a free parameter. The integral is approximated by Monte-Carlo methods. The exponentiation $I \to 1 - \exp(-I)$ ensures that the probability saturates at unity. Once a reconnection occurs, however, its space-time location is selected according to the differential probability $\Omega_{\text{max}}^+(\mathbf{x}, t)\, \Omega_{\text{max}}^-(\mathbf{x}, t)$ without any saturation. This defines the string pieces involved, and the new colour singlets are obtained by a flip as described above (dipoles $(i, k) + (l, n) \to (i, n) + (l, k)$).

Reconnections in scenario II can only take place when the cores of two string pieces cross each other. Then the transverse size of strings is irrelevant, which gives a much simpler algorithm than in scenario I. Now a string-piece core at time t is parameterized by a one-parameter set $\mathbf{x}(t, \alpha)$, where $0 \leq \alpha \leq 1$. Two string pieces (i, k) and (l, n) from the W^+ and W^- decays cross when $\mathbf{x}_{(i,k)}^+(t, \alpha^+) = \mathbf{x}_{(l,n)}^-(t, \alpha^-)$. This simple sytem of linear equations always has a unique solution in (t, α^+, α^-), but the solution need not be in the physically allowed domain. Such false solutions are easily rejected. As an example, if there is no shower activity, so that the event only consists of the two $q_1\overline{q}_2$ and $q_3\overline{q}_4$ strings, it is easy to see that these are moving apart from each other already from their creation and will never meet. A solution will nevertheless be found, but with negative t and possibly either or both of the $\alpha^\pm$ outside their allowed range. Further, as in scenario I, neither string piece should have had time to fragment at crossing. This gives two extra suppression factors $\exp\{-\tau^2/\tau_{\text{frag}}^2\}$, with τ the proper lifetime of each string piece at the point of crossing. Only the one that occurs first is chosen if several string crossings are possible. Reconnection is done with a flip, as in scenario I.

In models I and II the string length is not tested, so it may increase. The geometry of the process still tends to favour a reduced λ. For the model variants I′ and II′, a reduced λ is imposed as an additional requirement on allowed reconnections.

7.2.5 Other CR models

It is relevant to remember that many more CR models have been proposed, and several implemented in past PYTHIA versions. Some of these could be resuscitated using the existing colour-reconnection user hook, or an expanded version thereof, should the need arise.

In PYTHIA 6.4, several colour-annealing scenarios were available [315, 323], again primarily intended to be useful for top-mass uncertainty studies in hadronic collisions. They start from the assumption that, at hadronization time, no information from the perturbative colour history of the event is relevant, so all existing colour tags are erased. Instead, what determines how hadronizing strings form between the partons is a minimization of the total potential energy stored in these strings, as represented by the λ measure. The minimization is achieved by an iterative procedure, which unfortunately can be quite time consuming. The scenarios differ by details such as whether closed gluon loops are suppressed or not, or whether only free colour triplets are allowed to initiate string pieces.

Also in PYTHIA 6.4, the GH model [324] offered a simpler option for W^+W^- events, based on colour factors and string length reduction, without any space-time picture.

In the ARIADNE program for e^+e^- and $e^\pm p$, CR was introduced based on λ minimization [319], but CR could occur after each new parton-shower emission, and thereby affect the continued shower evolution. A similar idea is the dipole-swing mechanism for the initial-state evolution of incoming hadrons [245, 247].

When rapidity gaps were found in HERA DIS events, one early alternative to the Ingelman–Schlein pomeron picture [250] was that the gaps were a consequence of CR [325–328]. The Uppsala group has subsequently expanded this soft colour interactions approach to encompass also hadronic events, for topics such as diffraction and other rapidity gaps [329], and charmonium production [330]. One important difference relative to many of the models above is the frequent use of an "area law" [331] rather than the λ measure. The area that a string motion sweeps out is related to its m^2. For a string consisting of several pieces, the total area is defined as $A = \sum_{i=0}^{n} m_{i,i+1}^2$, with masses calculated as in eq. (322). The probability of a reconnection is then $P = R_0 \left[1 - \exp(-b\,\Delta A)\right] = R_0 \left[1 - \exp(-b(A_{\text{old}} - A_{\text{new}}))\right]$. The $R_0 \approx 1/N_c^2$ is an assumed colour-factor suppression, and b is the same as in eq. (312). Note that, had A been defined as the product of masses rather than a sum, then $\ln A$ would have been closely related to λ, and in particular a $\Delta\lambda$ and a ΔA scan would find the same optimal reconnection region, but that is not the case now. The related code is available in some earlier PYTHIA versions.

CR has also been studied in the context of other generators, such as HERWIG [332–334] and SHERPA [335]. It is not possible to address CR in equivalent terms for cluster as for string fragmentation, so there is no straight correspondence, but some basic ideas nevertheless are shared.

7.3 String interactions and collective effects

Heavy-ion collision experiments have for decades studied the possible creation of a Quark–Gluon Plasma (QGP) in high-energy collisions of heavy nuclei. Monte-Carlo simulations of physics processes involving QGP creation, is mostly carried out in designated generators or generator frameworks such as JEWEL [336] or JETSCAPE [337] (both of which in fact use PYTHIA as a hard-process generator). Another approach is to segment individual events into "core" and "corona" parts [338], where the former are treated as QGP, and the latter in vacuum. This is the case for EPOS-LHC [339], which is an independent framework, as well as for other approaches built on top of PYTHIA [340].

PYTHIA has thus, historically, played on a different field than generators focused on the special observables obtained in heavy-ion collisions. Instead, PYTHIA is often used as a generator supplying an initial state, with a focus on the hard process, parton shower, and hadronization

as in lepton collisions, with no QGP produced or assumed. This clear division of tasks was questioned by data from LHC. First in 2010, with the discovery of long-range azimuthal correlations of final-state hadrons in high multiplicity pp collisions, referred to as "the near-side ridge" [341], and later by observations of enhanced production of strange and multi-strange final-state hadrons, incompatible with model fits to LEP data [342–345]. The latter culminated in the observation that not only is the observed strangeness production incompatible with model fits to LEP data, strange/non-strange ratios also increase with multiplicity, and the increase smoothly connects pp with pA and AA collision systems [346]. This clearly meant that PYTHIA could no longer assume that effects traditionally ascribed to QGP formation are only present in heavy-ion collisions. While CR models can account qualitatively for some of the observed effects [347, 348], they are wholly unsuitable in others [349]. Instead of introducing QGP formation into PYTHIA, as the approaches cited above in some sense have already done, the route taken is to expand the Lund string model to its furthest consequence, by allowing interactions between strings in densely populated regions of space. Whether interactions between strings are indeed responsible for all collective effects observed in pp and heavy-ion collisions, is still unknown. The models introduced here should thus clearly be understood as one possibility among several others, however unified by the underlying assumption that QGP is not formed. Furthermore, they are all work in progress at the time of writing, and subject to change. There is no clear demarcation between what constitutes a model of colour reconnection, as introduced above, and models of string interactions. In this manual we have drawn the line between models operating in momentum space (the colour reconnection models) and models operating in real space.

7.3.1 String shoving

In the original formulation of the Lund string model, strings are treated as massless relativistic strings, which presupposes that strings have no transverse extensions. In collisions with many strings occupying the available spatial volume, this approximation breaks down, and strings are allowed to interact with mainly repulsive forces. The realization of this picture is denoted the "string shoving model". While similar ideas were explored analytically already in 1988 [350], the modern version of the string shoving model is formulated to take into account input from lattice QCD, and is based more firmly on the correspondence with a superconductor. This model is rather new at the time of writing [283] and is still being extended with further consequences being explored [351, 352]. The model contains three basic physics components: 1) the string shape, 2) the string transverse width, and 3) the interaction force between two strings.

The transverse shape of the colour-electric field of the flux tube (the string shape) is determined with input from lattice QCD [353], and can be well described by a Gaussian:

$$E(\rho) = N \exp\left(-\frac{\rho^2}{2R}\right), \tag{334}$$

where N is a normalization factor, ρ is the radial coordinate, and R is the string equilibrium radius. The normalization constant is determined by assuming that the field energy per unit length $\int d^2\rho E^2/2$ is a constant fraction (g) of the string tension. This gives $N^2 = 2g\kappa/(\pi R^2)$. The strings expand from their time of formation with infinitesimal width, until they either attain the maximum width R, or until the string's fragmentation proper time, τ_{had}, has been reached. While the equilibrium width of a string can be argued either by lattice considerations or from models, the number is associated with such large uncertainty, that it is in practice kept as a free parameter of the model, with reasonable values between around 0.5 and 1.5 fm. The same holds for the parameter g. The string repulsion force can then be calculated from the energy of the colour-electric field of two overlapping, parallel strings $\int d^2\rho(\vec{E}_1 + \vec{E}_2)^2/2$. If

strings are separated from each other by the transverse distance $d_\perp$, the interaction energy becomes $2g\kappa \exp(-d_\perp^2/(4R^2))$, which gives the interaction force per unit length:

$$f(d_\perp) = \frac{g\kappa d_\perp}{R^2} \exp\left(-\frac{d_\perp^2}{4R^2}\right). \tag{335}$$

The above treatment leading to eq. (335), is made in terms of Abelian fields. As such, anti-parallel strings would attract each other rather than repel. In a non-Abelian theory like QCD, the picture is more complex, leading to repulsion being the dominant mechanism. As an example, consider the case of oppositely oriented triplet fields. One obtains an octet field with probability 8/9, which still leads to a repulsion, and a singlet field with probability 1/9, leading to attraction. Since singlets correspond to the total attenuation of fields, it can further be assumed that singlets are already handled by colour-reconnection mechanisms [352, 352].

The technical implementation is concerned with two further questions, namely calculating $d_\perp$ for two given string pieces, and distributing the resulting pushes in the event. For the former, a suitable Lorentz frame is defined, where a string pair always lies in parallel planes, called the parallel frame [352]. One can then boost a pair of strings from the lab frame to the parallel frame, where the string topology is specified with an opening angle between the two partons in the string ends and a skewness angle between the two strings — both of which are constrained by momentum-energy conservation. The angles can be expressed in terms of pseudorapidity and invariant masses s_{ij} for the string formed by partons i and j:

$$\cosh(\eta) = \frac{s_{14}}{4p_{\perp 1}p_{\perp 4}} + \frac{s_{13}}{4p_{\perp 1}p_{\perp 3}}, \text{ and } \cos(\phi) = \frac{s_{14}}{4p_{\perp 1}p_{\perp 4}} - \frac{s_{13}}{4p_{\perp 1}p_{\perp 3}}. \tag{336}$$

Furthermore, the strings now evolve and interact in the proper time in the parallel frame. Calculating this interaction for every possible string pair is, among other aspects, a computational challenge, and to curb the possibility of running into being an extreme computing resource-consuming program, we for now neglect end-string effects which, for example, have been studied in [290].

The shoving force is distributed to the outgoing hadrons formed after string fragmentation, taking into account that the total applied push is a result of a time evolution. The integrated push $\Delta p_\perp$ is:

$$\Delta p_\perp = \int dt \int dz f(d_\perp(t)), \tag{337}$$

where the integration limits in z are time dependent. Since the time ordering of pushes is important, $\Delta p_\perp$ is split up into several (fixed) small pushes $\delta p_\perp$, according to a probability distribution $P(t)$. The total push is then:

$$\Delta p_\perp = \int dt P(t)\delta p_\perp, \quad \text{with} \quad P(t) = \frac{1}{\delta p_\perp} \int dz f(d_\perp), \tag{338}$$

when $\delta p_\perp$ is small. The pushes can then be ordered in time (in the parallel frame) using the veto algorithm. The resulting procedure corresponds to a time evolution with dynamical time stepping, where steps are large when pushes are small and *vice versa*.

In t,z space, this would look like hadrons flying out along the direction of their original pseudorapidity, even after the pushes are applied, spreading out in a light-cone that extends in such a way that it encloses all the hadrons which receive a share of this generated $\Delta p_\perp$. This distribution of pushes is performed as shown in fig. 18.
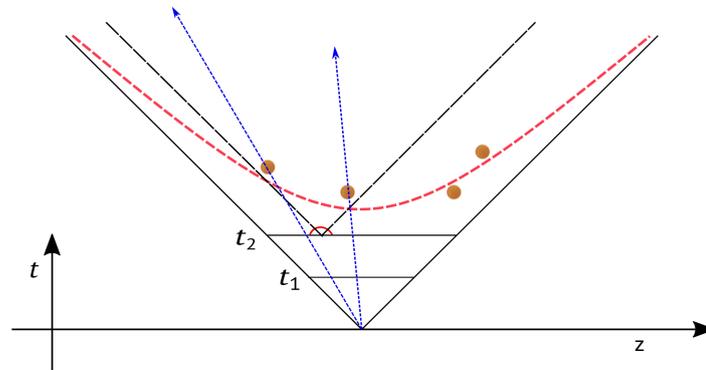
Figure 18: Space-time diagram of a Lund string showing the trajectory of hadrons when they receive their share of $\Delta p_\perp$ resulting from string shoving interactions. The blue lines show the initial pseudorapidity lines for the hadrons formed, the red line implies a $\delta p_\perp$ generated from shoving, and the red dashed line shows the τ_{had}.

7.3.2 Rope hadronization

A simple string drawn between a quark and an antiquark is an **SU(3)** triplet (or anti-triplet depending on direction of colour flow). When several strings overlap with each other at hadronization time, the rope-hadronization model posits that end-point colour charges will act together coherently to form a stronger field — a rope. This possibility was noted in the classic paper by Biro, Nielsen and Knoll from 1984 [354].

The new, stronger field is an **SU(3)** multiplet. According to lattice calculations [355], the energy density (and thus the string tension) scales as the second Casimir operator (C_2) of the rope multiplet. When a rope is formed by ordinary triplet and anti-triplet strings, the net colour charge is obtained from the addition of random coloured triplets and anti-triplets [246, 354, 356]. A resulting multiplet is uniquely characterized by two quantum numbers p and q, with a specific state corresponding to p coherent triplets and q coherent anti-triplets (a normal triplet string is thus characterized as $\{p,q\} = \{1,0\}$). The multiplicity of a multiplet is given by:

$$2N = (p+1)(q+1)(p+q+2). \tag{339}$$

This allows for an iterative addition of multiplets. Starting from a given multiplet $\{p,q\}$, adding a triplet gives the three possible multiplets [246]:

$$\{p+1,q\}, \{p-1,q-1\}, \text{ and } \{p,q-1\}, \tag{340}$$

with weights given by eq. (339). The anti-triplet case is given directly from symmetry. Once it is established which triplets and anti-triplets are overlapping in an event, a random walk procedure can be carried out to find p and q for the rope multiplet. Since the energy density of the rope is proportional to C_2, the relative tension of the multiplet to the triplet can be calculated directly as:

$$\frac{C_2(\{p,q\})}{C_2\{1,0\}} = \frac{1}{4}(p^2 + pq + q^2 + 3p + 3q). \tag{341}$$

When the rope breaks up, it does so in a step-wise manner, one string at a time. By considering the change in available field energy in the transition $\{p, q\} \rightarrow \{p-1, q\}$, neglecting the contribution from the vacuum pressure to the total energy, the energy available in a single string breaking becomes the *effective string tension* $\tilde{\kappa}$:

$$\tilde{\kappa} = (2p-1)\kappa. \tag{342}$$

While the string tension κ does not enter explicitly[9] into the PYTHIA implementation of string hadronization, it does enter implicitly through the parameters of eqs. (312) and (319). From the implicit dependence on κ, transformation rules for all parameters can be defined, given the assumption that the PYTHIA default values of all parameters correspond to $\tilde{\kappa} = \kappa$, as they are tuned to LEP data [357] where there are no overlapping strings, and thus $p = 1$ and $q = 0$. The most important affected parameters [246], are: those involved in suppression of strangeness (ρ), diquark production (ξ), diquark with strange-quark content relative to diquarks without strange quarks (x), the suppression of spin-1 diquarks relative to spin-0 diquarks, and the width of the transverse momentum distribution in string breakings (σ). Letting $h = \tilde{\kappa}/\kappa$, the transformation rules for ρ, x and y are similar:

$$\rho \mapsto \tilde{\rho} = \rho^{1/h}, \ x \mapsto \tilde{x} = x^{1/h}, \ \text{and} \ y \mapsto \tilde{y} = y^{1/h}, \tag{343}$$

while $\sigma \mapsto \tilde{\sigma} = \sigma^{1/h}$. The ξ parameter is more complicated, and transforms like:

$$\xi \mapsto \tilde{\xi} = \tilde{\alpha}\beta \left(\frac{\xi}{\alpha\beta} \right)^{1/h}, \tag{344}$$

with α depending on all the above parameters, and β a free parameter.

7.3.3 The thermal model

The thermal model [358], available as a non-default option, can partly be viewed as an alternative to the rope model, sharing similar objectives. Not all details have been fully developed, so its main purpose is for exploration. One motivation for it is that hadronic $p_\perp$ spectra in low-energy collisions are reasonably well described by an exponential fit

$$\frac{\mathrm{d}\sigma}{\mathrm{d}^2 p_\perp} = N \exp(-m_{\perp\text{had}}/T), \ \text{with} \ m_{\perp\text{had}} = \sqrt{m_{\text{had}}^2 + p_\perp^2}, \tag{345}$$

where N and T are (approximately) common for all hadron types. Another motivation is that local quantum-mechanical fluctuations in the string transverse profile translate into a fluctuating string tension κ, which can broaden the Gaussian $p_\perp$ into an exponential-like form [359]. (Compare with fluctuations in the proton size, which are commonly advocated and used *e.g.* in the ANGANTYR modelling of cross sections [283].) While traditionally T is associated with a temperature, in such a scenario it would rather be derived from κ.

The thermal model is implemented as follows. In each string break the q and $\bar{\text{q}}$ receive opposite and compensating $p_\perp$ values, chosen such that the $p_\perp$ sum of two adjacent string breaks precisely gives an $\exp(-p_\perp/T)$ spectrum. Starting from a known flavour in one string break, the next flavour and the resulting intermediate hadron is chosen among all possibilities according to a relative weight $\exp(-m_{\perp\text{had}}/T)$. Assuming the production of two hadrons with different masses m_1 and m_2, this approach then implies the same production rate for $p_\perp \gg m_1, m_2$, but more suppression of the heavier hadron at low $p_\perp$. Thus, there is less production of heavier states, but they come with a larger $\langle p_\perp \rangle$, which is as intended.

---

[9]The string tension does enter explicitly into the vertex positions in section 7.1.4, but the effect of rope formation has so far not been taken into account for hadron vertices.

There is some fine print, like that each particle should be weighted by the number of spin states, that flavour-diagonal mesons can mix, that baryons need **SU(6)** symmetry factors, that baryons receive a free overall normalization factor with respect to mesons, and so on. The number of flavour-related free parameters still is significantly reduced relative to the ordinary string fragmentation.

Overall the particle composition comes out reasonably well, with some excess of the heavier baryons. This is in contrast to the normal string fragmentation, where it is difficult to produce enough of these particles. The larger $\langle p_\perp \rangle$ for heavier particles also improves agreement with pp data, but resonance decays act to dilute the effects, so $\langle p_\perp \rangle (m_{\mathrm{had}})$ still does not rise quite fast enough.

Another issue is what happens when several strings are close packed. In the rope model, this leads to a higher κ, in quantised steps. An alternative is to assume a continuously increasing κ as each string is squeezed into a decreasing effective area. Such an option is implemented as part of the thermal model, but can also be applied to the default Gaussian one. In this approach, the T or κ parameter is rescaled by a power of the effective number of strings in the neighbourhood of a new hadron. Therefore a trial average step along the string is made before a new hadron is produced, giving a likely hadron rapidity and $p_\perp$. Then, one may count the number of strings crossing that rapidity, as a simple measure of string density. A smooth damping is applied for particles produced at larger $p_\perp$, which are likely to be produced in minijets sticking out from the denser-populated central region. The close-packing enhancement can be used *e.g.* to increase strangeness production in high-multiplicity pp events, similar to the rope model, but it has not been as extensively compared with data.

7.4 Hadronic rescattering

After hadrons have been produced, outgoing hadrons can interact in secondary collisions. This rescattering can be relevant when studying collective effects, but can lead to a significant slow-down of PYTHIA, and is therefore not on by default. It is enabled by setting `HadronLevel:Rescatter = on`. Here, we will outline the rescattering algorithm, then summarize some notable effects of rescattering of which the average user should be aware. A more detailed discussion of the rescattering framework is given in [213] in the context of pp collisions, while [360] discusses physics results for pA and AA collisions.

There are two aspects to the rescattering algorithm: first, describing how two hadrons interact with each other in their rest frame; and second, describing the evolution of the event as a whole.

Consider two hadrons in their rest frame, with CM energy $\sqrt{s}$ and impact parameter b. We assume that the probability of an interaction occurring is a function of b and the total cross section σ_{tot}. The cross section generally depends on $\sqrt{s}$ and the specific hadron species, as described in section 6.1.5. There is no solid theory for how P depends of b, so two different models are implemented in PYTHIA 8.3. The default is a Gaussian dependency,

$$P(b) = P_0 e^{-b^2/b_0^2}, \tag{346}$$

where P_0 is referred to as the opacity, a free parameter that is 0.9 by default, and the characteristic length scale is

$$b_0 = \sqrt{\frac{\sigma_{\mathrm{tot}}}{P_0 \pi}}. \tag{347}$$

The alternative model is a disk model,

$$P(b) = P_0 \Theta(b - b_0), \tag{348}$$

where Θ is the Heaviside step function. For $P_0 = 1$, this corresponds to the black-disk model used by most existing hadronic rescattering frameworks. The two models are normalized such that if b is chosen uniformly on a disk with radius much larger than b_0, then both models will give the same interaction probability. In practice, rescattering is more likely in dense regions where b tends to be biased towards lower values, so the narrower distributions like the black disk will lead to more rescattering activity. If it is determined that the hadrons should interact, the interaction time is defined as the time of closest approach in their rest frame.

The algorithm for performing rescattering for the whole event proceeds as follows:

1. Start with an event right after hadronization.

2. For each hadron pair, test whether they could interact, using the probability P defined above.

3. If a pair could potentially interact, record the interaction time for that pair in a time-ordered list.

4. Choose the earliest interaction in the list where participants have not already interacted, and simulate the collision. Which process to simulate is chosen with probabilities proportional to the partial cross sections for each process.

5. Check whether the newly produced hadrons can interact with existing ones, and if so, add the interaction times for those pairs to that list.

6. Continue picking interactions from the list until there are no more potential rescatterings.

Short-lived hadrons can also decay during the rescattering phase. To model this, the decay times of those hadrons are recorded in the list together with rescattering interaction times, and the decay occurs when it is chosen in step 3 above, if it has not already rescattered.

Enabling rescattering has a few consequences for the shape of events. First, rescattering increases charged multiplicity, since only processes with two incoming hadrons are allowed, but inelastic processes can produce more than two outgoing ones. For PYTHIA 8.307 with default parameters and pp at 13 TeV, this can be compensated by setting `MultipartonInteraction:pT0Ref = 2.345`. For beams such as pPb and PbPb, other values might restore the multiplicity, but a more thorough retune is necessary in order to simultaneously obtain the correct multiplicity in all three cases. In such a retune, it would also be relevant to include other effects such as ropes (section 7.3.2) and shoving (section 7.3.1). For now, the user is advised to assume that rescattering will change the charged multiplicity.

Similarly, hadron composition will change. Baryon number in particular is reduced in rescattering through annihilation processes. For example, the process $p\overline{p} \to \pi^+\pi^-\pi^0$ is possible, but not the reverse. Another way the composition changes is through resonance production, $e.g.$ $\pi K \to K^*$, but be aware that this resonance production is not easily detectable in experiment; for a resonance production followed by a decay, $\pi K \to K^* \to \pi K$, the invariant mass of the outgoing system is the same as for the incoming one. In other words, this process produces a K^* that is visible in the event record, without necessarily changing the observable πK mass spectrum.

A particular consequence of the increased multiplicity is that each hadron will on average have lower $p_\perp$, which could affect $e.g.$ spectra that are sensitive to $p_\perp$ cuts. At the same time, the mean $p_\perp$ for particular hadron species may increase. This is the case for example with protons, which will move slower than pions with similar $p_\perp$, and will therefore receive a push from behind. This phenomenon is referred to as the "pion wind".

Rescattering has been shown to give rise to some collective effects, in particular azimuthal flow in PbPb collisions. PYTHIA 8.3/ANGANTYR with rescattering provides a good description of elliptic flow coefficients at large multiplicities, and a more modest contribution at low multiplicities. It can also lead to some jet modifications, but with the aforementioned $p_\perp$ shift, it is not clear how to interpret these modifications. See [360] for further discussion.

7.5 Bose–Einstein effects

Ideally, coloured partons could be formed into colourless final-state hadrons using amplitude-based quantum mechanics, but because these transitions are non-perturbative, phenomenological models of hadronization are employed. Due to the probabilistic nature of these hadronization models, coherence in final-state particles cannot be directly described. A classic example of such final state coherence is Bose–Einstein effects, where correlations arise between identical bosons in an event from the symmetrization of the production amplitude. While these correlations are expected to have a negligible impact for most measurements in pp collisions, Bose–Einstein effects[10] have been observed in minimum bias pp and p$\bar{\text{p}}$ data [362–365], as well as e$^+$e$^-$data [366–369]. Additionally, some precision measurements such as W-mass determination using hadronic final states may be sensitive to Bose–Einstein effects [370].

Assuming a geometric picture with a Gaussian distribution of production vertices in space-time, two-particle correlations of identical bosons are enhanced by a unitless factor of,

$$f_2(Q) = 1 + \lambda e^{-Q^2 Q_0^{-2}}, \tag{349}$$

with respect to a final state with no coherence effects [371]. Here, Q^2 is $(p_i - p_j)^2$ where p_i and p_j are the four-momenta of the two particles, λ is the incoherence parameter, and Q_0 is a reference Q related to the radius of the particle source as $r \equiv \hbar/Q_0$. The incoherence parameter is limited between 0 where there is no effect, and 1 with a maximal effect.

For a high multiplicity event with multiple two-particle correlations, the event weight can be naively approximated as the product of $f_2(Q)$ for each particle pair. Note, this is a slight overestimate of the event weight for most event topologies. These event weights cannot modify the overall normalization, as this would increase the cross section for the final state. If the weights are normalized to unity, the total cross section is not modified, but the multiplicity distribution will be shifted to higher multiplicities. Neither of these behaviours is desirable, as both cross sections and multiplicity distributions are already well described without Bose–Einstein effects.

Instead, in PYTHIA Bose–Einstein effects are introduced by shifting the momenta inside particle pairs. Assuming the distribution of Q for particle pairs is given by flat phase space, then solving,

$$\int_0^Q \mathrm{d}q \, \frac{q^2}{\sqrt{q^2 + 4m^2}} = \int_0^{Q'} \mathrm{d}q \, f_2(q) \frac{q^2}{\sqrt{q^2 + 4m^2}}, \tag{350}$$

for Q' determines the new Q value needed to produce an enhancement of $f_2(Q)$ for that particle pair with individual particle mass m. The three-momentum for the two particles can then be shifted by,

$$\Delta \vec{p}_{i,j} = c(\vec{p}_i - \vec{p}_j), \tag{351}$$

where $\vec{p}_i' = \vec{p}_i + \Delta \vec{p}_{i,j} \, \vec{p}_j' = \vec{p}_j - \Delta \vec{p}_{i,j}$, and the constant coefficient c is determined from setting $Q'^2 = (p_i' - p_j')^2$. Because events may have more than one particle pair, the total shift for a given particle is then,

$$\vec{p}_i' = \vec{p}_i + \sum_{j \neq i} \Delta \vec{p}_{i,j}, \tag{352}$$

---

[10]Within the heavy-ion and astrophysics communities these effects are oftentimes discussed in the context of Hanbury-Brown–Twiss interferometry [361].

where the sign of $\Delta\vec{p}_{i,j}$ is such that the three-momenta of the event is conserved.

Effectively, this shifting of momentum corresponds to pulling particle pairs closer together, and while three-momenta is conserved throughout this process, energy conservation is violated and the total energy of the event is reduced. The form of $f_2(Q)$ from eq. (349) arises from integrating the pair symmetrization term $1+\cos(\Delta x \cdot \Delta p)$ over a Gaussian distribution of production vertices in space-time. Consequently, any source distribution other than a Gaussian will result in an oscillatory behaviour of $f_2(Q)$, with alternating values of $f_2(Q) > 1$, where $Q' < Q$ results in the pair pulled together, and $f_2(Q) < 1$, where $Q' > Q$ results in the pair separated apart. With the appropriate damping of this oscillatory behaviour, for a given particle configuration, a form of $f_2(Q)$ can be found where both conservation of three-momenta and energy is achieved. Some pairs at low Q are pulled together, reducing the net energy, while other pairs at middle Q are separated apart, increasing the net energy.

To achieve this behaviour, a form of $f_2(Q)$ is selected to have one oscillation before damping. The ansatz of the BE_{32} algorithm [372],

$$f_2(Q) = \left[1 + \lambda e^{-Q^2 Q_0^{-2}}\right]\left[1 + \alpha\lambda e^{-Q^2 Q_0^{-2}/9}\left(1 - e^{-Q^2 Q_0^{-2}/4}\right)\right], \qquad (353)$$

is chosen for $\alpha < 0$, where the new second factor effectively models the initial minimum of the oscillation as a smeared Gaussian [14]. This form does not have any deep physical meaning, but provides the necessary first oscillation while maintaining the initial Gaussian distribution form, and has the limiting behaviour of $f_2(0) = 1 + \lambda$. The factor α is iteratively determined per event after calculating all relevant p_i', such that energy is still conserved even after three-momentum shifting is performed for each relevant particle. Consequently, at least two particle pairs must be present for Bose–Einstein effects to be introduced.

Bose–Einstein correlations are performed after hadronization but prior to particle decays, and are not included by default. Effects may be switched on or off for different particle groupings: pions with π^0, π^+, and π^- pairs; kaons with K_S^0, K_L^0, K^+, and K^- pairs; and eta mesons with η and η' pairs. Many of these particle species are produced not only from primary hadronization, but also from the decay of short-lived particles. Consequently, a configurable minimum decay width can be set so that any particles with a larger width are decayed prior to the application of Bose–Einstein effects. The default minimum decay width is set at 0.02 GeV so that both ρ and K^* mesons are decayed before correlations are introduced. Both the shifted and unshifted versions of particles are kept in the event record for bookkeeping purposes. All shifted particles are assigned a status of 99 and are set as the children of their unshifted entry.

7.6 Deuteron production

The deuteron, D, is a bound proton and neutron state, which, similar to Bose–Einstein effects (see section 7.5), must be formed after hadronization. Understanding deuteron production in the context of collider-based experiments can help efforts in modelling nuclei formation and reduce prediction uncertainties when searching for possible dark-matter induced excesses in cosmic ray flux ratios [373]. In heavy-ion physics, formation of loosely bound systems are often used to determine the chemical freeze-out temperature in statistical hadronization models [374]. In PYTHIA 8.3, two deuteron formation models are available, the coalescence model [375, 376] and the more sophisticated Dal–Raklev model [377]. Both models are implemented through the same configurable framework, with the Dal–Raklev model set as the default configuration. All deuteron production is switched off unless explicitly requested by the user. Note that while the discussion here is for the deuteron, anti-deuteron production is also performed following the exact same method, but with all particles swapped to antiparticles.

In the coalescence model, all possible p and n pair combinations are determined. For each pair the magnitude of their three-momenta difference,

$$k = \sqrt{(\vec{p}_i - \vec{p}_j)^2}\,, \tag{354}$$

is calculated in the rest frame of the pair, where $\vec{p}_i$ and $\vec{p}_j$ are the two three-momenta of the pair. If k is less than some cutoff value c_0, the pair is bound into a deuteron, otherwise the nucleons remain unmodified. Spatial separation, in addition to momentum separation, could also be considered, although this has not been implemented in any of the models described here. The ordering of testing pairs for binding is randomized, and after each successful binding, any remaining pairs containing one of the bound nucleons are no longer considered for binding. For the coalescence model, this implies that the binding cross section is flat as a function of k. If there are two unique pairs each with $k < c_0$, both pairs have an equal probability of being bound, even if one k is smaller than the other.

After a nucleon pair is selected for binding, a deuteron is formed. In principle, the three-momentum of this deuteron could be calculated as $\vec{p}_i + \vec{p}_j$, and while three-momentum for the event would be conserved, energy would not. Instead, an isotropic decay into a deuteron and photon is performed in the rest frame of the pair. Because the primary process for deuteron formation at the low momentum differences of the coalescence model is radiative capture, $pn \to \gamma D$, this provides a reasonable approximation of the process and conserves both energy and momentum. While spin correlations could be considered, these typically are negligible after boosting the deuteron and photon into the laboratory frame.

The Dal–Raklev model expands upon the coalescence model by considering the following formation channels, other than just $pn \to D\gamma$.

- $pn \to \gamma D$
- $pn \to \pi^- \pi^+ D$
- $pp \to \pi^+ D$
- $pp \to \pi^+ \pi^0 D$
- $pn \to \pi^0 D$
- $pn \to \pi^0 \pi^0 D$
- $nn \to \pi^- D$
- $nn \to \pi^- \pi^0 D$

Channels can be removed, modified, or added. Each channel must have a two-body initial state and an n-body final state where $n > 1$ and at least one of the outgoing particles is a deuteron. For each of these channels the kinematics of the final state are determined by an isotropic decay in the rest frame of the initial state pair. Additionally, the binding cross section is no longer considered as a uniform distribution up to a cutoff parameter, but is instead determined from fits of differential nucleon-scattering data from a number of experiments [377].

Four cross-section parameterizations are available. For each channel, one of the following parameterizations must be selected, and the necessary coefficients c_i provided.

1. The coalescence model parameterization as previously described is a step function with two parameters, the cutoff parameter c_0 and a normalization parameter c_1. The normalization allows channels using this parameterization to be used in combination with other channels.

$$\frac{d\sigma(k)}{dk} = c_1 \Theta(c_0 - k)\,. \tag{355}$$

2. The $pn \to \gamma D$ cross-section differential in k can be parameterized by a polynomial below a cutoff of c_0, and with an exponential above. Due to Runge's phenomenon, the polynomial is fixed to its value at $k = 0.1$ GeV for $k < 0.1$ GeV.

$$\frac{d\sigma(k)}{dk} = \begin{cases} d\sigma(0.1 \text{ GeV})/dk\,, & \text{for } k < 0.1 \text{ GeV}\,, \\ \sum_{i=1}^{12} c_i k^{i-2}\,, & \text{for } 0.1 \text{ GeV} \geq k < c_0\,, \\ e^{-c_{13}k - c_{14}k^2}\,, & \text{otherwise}\,. \end{cases} \tag{356}$$

3. The two-body final states with a pion and deuteron are parameterized using a cross section differential in q, the momentum magnitude of the pion in the nucleon-pair rest frame, divided by the mass of the pion. Because the final state is two-body, the pion momentum magnitude is already known *a priori*.

$$\frac{d\sigma(q)}{dq} = \frac{c_0 q^{c_1}}{(c_2 - e^{c_3 q})^2 + c_4}. \tag{357}$$

In the default Dal–Raklev model, the $pn \to \pi^0 D$, $pn \to \pi^+ D$, and $pn \to \pi^- D$ channels use this parameterization.

4. The cross sections for the three-body final states with pions are differential in k and are parameterized with,

$$\frac{d\sigma(k)}{dk} = \sum_{i=0} \frac{c_{5i} k^{c_{5i+1}}}{(c_{5i+2} - e^{c_{5i+3} k})^2 + c_{5i+4}}, \tag{358}$$

where the number of coefficients is variable but must be a multiple of 5. The default $pn \to \pi^- \pi^+ D$, $pn \to \pi^0 \pi^0 D$, $pn \to \pi^+ \pi^0 D$, and $nn \to \pi^- \pi^0 D$ channels use this parameterization.

Not only does the shape of the differential cross sections matter, but also the normalization, as this determines the relative rates between the channels. In the Dal–Raklev model the γD channel dominates at low k. For $k > 1$ GeV the πD channels dominate except between roughly 1 and 2 GeV where the $\pi\pi D$ channels dominate. An additional unitless normalization scale can be configured to increase or decrease the total deuteron production cross section. A number of fits for this normalization constant have been made using various data sets from the LHC, with the default normalization set from differential 7 TeV ALICE data [377].

8 Particles and decays

There are several ways to classify unstable particles, and in PYTHIA at least three classifications are useful:

- by lifetime, specifically for coloured particles whether above or below the hadronization time;

- by if the partial and total widths of a particle are perturbatively calculable, such as for the μ, γ^*/Z, $W^\pm$, top, Higgs bosons, and most BSM particles, or not, such as for hadrons;

- by if a particle is part of the hard process, and cannot be produced anywhere else, such as in parton showers or hadronization, *e.g.* Z, $W^\pm$, top, and Higgs bosons.

These classifications are necessary to understand how particles are technically treated.

In PYTHIA a distinction is made between the following technical representations of particle states: resonances with an average lifetime shorter than the hadronization scale; particles with an average lifetime comparable to or longer than the hadronization scale; and partons that carry colour charge and must be hadronized. By default, any state with a nominal mass above 20 GeV is considered as a resonance, *e.g.* γ^*/Z, $W^\pm$, top, Higgs bosons, and most BSM states such as sfermions and gauginos. However, some light hypothetical weakly interacting or stable states, such as the gravitino, are also by default considered as resonances to ensure a full treatment of angular correlations in their decays. All remaining states without colour charge, primarily leptons and hadrons, are treated as particles, while quarks and gluons are

considered as partons. There are some exceptions like colour-octet onia, which are treated as both partons carrying colour charge, and particles that decay after hadronization.

Resonance states are sequentially decayed during the hard process, see section 3.11 for details. For example, in the hard process $gg \to H \to Z[\to \mu^+\mu^-]Z[\to \mu^+\mu^-]$ the decay of the Higgs into Z bosons is first performed, followed by the decays of the γ^*Z resonances into muon pairs. The cross sections calculated for hard processes with resonances depend upon the available decay channels of the resonances; closing decay channels will reduce the cross section for the process while opening decay channels will increase the cross section. Consequently, when using the cross section calculated for a resonance produced in a hard process, the available branching fractions of the resonance are already included in the cross section. In most cases, angular correlations are included in the decay of the resonance. In some cases, mixed decays of the resonances are needed, *e.g.* $gg \to H \to Z[\to \mu^+\mu^-]Z[\to e^+e^-]$. In this example, both the muon and electron channels could be left open. However, in some cases this might result in inefficient generation of the required final state. Consequently, a special class is available in PYTHIA, `ResonanceDecayFilterHooks`, which can be used to select specific final states from the resonance decays.

Lighter states such as the J/ψ, which can be produced by the hard processes of section 3.3, are technically treated by PYTHIA as particles and not resonances. This is because the J/ψ can be produced in both hadronization and particle decays, where the cross section of these J/ψ production mechanisms is not known *a priori*. The reduced cross section of the J/ψ due to closed decay channels can only be determined after the generation of full events, including J/ψ production from the hard process, hadronization, and particle decays. Similarly, states that are only produced in the hadronization and particle decays, *e.g.* the ρ, are also considered particles and not resonances. An important exception to the treatment of resonances is the production of weak bosons in the parton shower, see section 4.1.4. Here, while closing the decay channels of the weak bosons will modify the hard-process cross section, the decays of the weak bosons in the parton shower will still remain inclusive. The decay channels of the weak bosons in the parton shower can be selected using the special IDs 93 and 94 for the Z and W, respectively. However, changing these decay channels will not affect the hard-process cross section and must be book kept carefully by the user.

8.1 Particle properties

For all states, a number of properties must be defined. Each state is uniquely identified by its PDG ID, or when a PDG convention is not available, a PYTHIA specific numbering convention, *i.e.* for BSM and colour-octet onium states. For each state a human readable name is stored, as well as an antiparticle name when relevant. The quantum numbers for each state must be defined: the spin, electric charge, and colour charge. Note that the spin information is duplicated for hadrons, where the spin can also be determined from the PDG ID. The experimentally observable properties of the state are also specified including a nominal mass, a nominal width, allowed limits of this width, and a nominal proper lifetime. Additionally, a number of decay related options can be specified including whether the state may decay, if the width is perturbatively calculable, and if the width should be forced to be rescaled. Each state may also have a list of decay channels which determine how the state is decayed. Each channel is configured with a flag specifying if the channel is available for the particle/antiparticle state, a branching ratio, a mode specifying a possible matrix element for the decay, and a list of the decay products.

8.1.1 Masses

The default masses for most particles in PYTHIA are taken from experimental observation as summarized by the PDG [378]. There are three exceptions: quarks and diquarks, unobserved or poorly studied hadrons, and hypothetical BSM particles. For hypothetical particles, *e.g.* BSM Higgs bosons, hidden valley hadrons, or fourth generation fermions, reasonable defaults are provided, see section 10.1.2 for details.

Due to ill-defined quark masses, two types of quark masses are used in PYTHIA, kinematic and running. The kinematic masses are those defined in the PYTHIA particle database, and are used when generating phase space. For example, in the process $gg \rightarrow c\bar{c}$, the kinematic mass of the c quark is used. Similarly, the $g \rightarrow q\bar{q}$ splittings of the parton shower use these kinematic masses. While these quark masses can be changed, their default values have been carefully chosen following a number of considerations [303]. Modifying these default values can lead to unintended consequences across all aspects of PYTHIA including hard process generation, the parton shower, hadronization, and even particle decays. Consequently, care should be taken when changing these quark masses from their default values.

Running quark masses are used when calculating mass-dependent couplings, which include couplings of the quarks to SM and BSM Higgs bosons. The running masses for the quarks are calculated at one loop in the $\overline{\text{MS}}$ normalization scheme using,

$$m(Q) = m_0 \left(\frac{\ln(Q_0/\Lambda)}{\ln(Q/\Lambda)} \right)^{12/(33-n_{\text{f}})}, \tag{359}$$

where m_0 is the input mass at the reference scale Q_0, and n_{f} is the number of active flavours in calculating α_s. For the light quarks, Q_0 is set at 2 GeV, while for the c, b, and t quarks Q_0 is set at m_0. The input masses can be configured with the parameters `ParticleData:mXRun`, where X is the quark name, *i.e.* either d, u, s c, b, or t, to be put equal to the $\overline{\text{MS}}$ mass of the quark. The reference value of α_s used in calculating Λ is defined at the scale m_Z and set with the parameter `ParticleData:alphaSvalueMRun`.

The default masses of unobserved hadrons and diquarks have been set using the constituent mass model from PYTHIA 6 [14, 379], which considers the spin-spin couplings of the quark combinations. The semi-empirical formula for a hadron mass is given by,

$$m = m_0 + \sum_i m_i + k\, m_{\text{d}}^2 \sum_{i<j} \frac{\mathcal{S}_{ij}}{m_i\, m_j}, \tag{360}$$

where the terms m_0 and k are determined from known hadron masses and depend upon the multiplet of the hadron, m_i are the constituent quark masses, and $\mathcal{S}_{ij}$ are the spin-spin interactions for each quark-pair combination. The constituent quark masses are taken from PYTHIA 6 as 0.325 GeV for the u and d, 0.5 GeV for the s, 1.6 GeV for the c, and 5 GeV for the b. Since the t does not form narrow bound states, the t constituent mass is not needed.

For mesons and diquarks, there is only one quark pair, given by q_1 and q_2. For diquarks and meson multiplets with orbital angular momentum $L = 0$, the spin-spin term for $S = 0$ states is $\mathcal{S}_{12} = -3$, while for the $S = 1$ states this term is $\mathcal{S}_{12} = 1$. For both pseudoscalar and vector mesons, m_0 is set to 0 GeV and k is fitted to be 0.16 GeV. For the excited multiplets with $L = 1$, the spin-spin terms vanish with k set to 0 GeV and m_0 fitted to be 0.45 GeV, 0.5 GeV, 0.55 GeV, and 0.6 GeV for scalars, $S = 0$ axial-vectors, $S = 1$ axial-vectors, and tensors, respectively. The masses of diquarks are calculated using the same k value as for baryons, 0.048 GeV, and $m_0 = 0.077$ GeV which is two-thirds the baryon m_0 value.

There are three possible combinations for baryons, and the spin-spin terms depend not only upon the spin of the baryon, but also the quark composition. For $S = \frac{1}{2}$ baryons the

spin-spin term is given by,

$$\sum_{i<j} \frac{\mathcal{S}_{ij}}{m_i \, m_j} = \frac{1}{m_1 m_2} - \frac{2}{m_1 m_3} - \frac{2}{m_2 m_3}, \tag{361}$$

if there are either two identical flavours, q_1 and q_2, or all the quark flavours are different and the two lighter quarks are in an anti-symmetric spin state. For this anti-symmetric case q_3 is the heaviest quark, while q_1 and q_2 are the two lighter quarks. When all the quarks are all different flavours and the light quark pair is symmetric, the spin-spin term is given by

$$\sum_{i<j} \frac{\mathcal{S}_{ij}}{m_i \, m_j} = -\frac{3}{m_2 m_3}, \tag{362}$$

where q_2 and q_3 are the two lighter quarks when relevant. For the $S = \frac{3}{2}$ baryons, the spin-spin term is given by,

$$\sum_{i<j} \frac{\mathcal{S}_{ij}}{m_i \, m_j} = \frac{1}{m_1 m_2} + \frac{1}{m_1 m_3} + \frac{1}{m_2 m_3}, \tag{363}$$

where the ordering of the quarks does not matter. For all baryons, the fitted parameters are set as $m_0 = 0.11$ GeV and $k = 0.048$ GeV. The default masses for a number of baryons in PYTHIA are calculated using these factors and eq. (360). These baryons include the double and triple-heavy Ξ and Ω baryons.

8.1.2 Widths

Widths are relevant for sampling the masses of both resonances and particles. For particles, widths are fixed when sampling a particle mass except for the case of hadronic rescattering, see sections 7.4 and 8.2.3 for further details. The parameter `ParticleData:modeBreitWigner` determines what type of distribution is used to select particle masses. Note that this parameter is set for all particle species; it is not possible to choose different mass shapes on a species-by-species basis. For a value of 0 the fixed on-shell particle mass is used, while for 1 a non-relativistic Breit–Wigner is used,

$$\mathcal{P}(m) \, dm \propto \frac{1}{(m-m_0)^2 + \Gamma^2/4} \, dm. \tag{364}$$

By setting a value of 2 a mass dependent width can be included,

$$\Gamma(m) = \Gamma_0 \sqrt{\frac{m^2 - m_{\text{thr}}^2}{m_0^2 - m_{\text{thr}}^2}}, \tag{365}$$

where m is the selected mass, m_0 is the on-shell mass, and m_{thr} is the average threshold mass. The threshold mass is the sum of the on-shell masses for the decay products, and is consequently channel dependent. However, to decouple mass selection and decay, the mass threshold is taken as the average mass threshold for all decay channels, weighted by branching fraction.

A relativistic Breit–Wigner can also be selected,

$$\mathcal{P}(m^2) \, dm^2 \propto \frac{1}{(m^2 - m_0^2)^2 + m_0^2 \Gamma^2} \, dm^2, \tag{366}$$

with the option 3, where a fixed Γ is used just as for option 1. The relativistic Breit–Wigner can also be used with the mass dependent width of eq. (365) with option 4. For all mass selection

options, relativistic or otherwise, the mass distribution is truncated via the `NN::mMin` and `NN:mMax` parameters set for each particle species; here, `NN` is the given particle species ID. The default mass shape in PYTHIA is option 4, a mass-dependent relativistic Breit–Wigner.

For particles with broad mass distributions that are not treated as resonances, the mass selection models above can distort the particle branching ratios. Regardless of the selected mass of a particle, all decay channels, even those with an on-shell mass threshold above the selected mass, are considered. Only after the masses for the decay products are sampled, are channels eliminated if not kinematically available. In this way, decay channels can remain open if there are downward fluctuations in the selected masses of the decay products. However, if the mass distribution for a particle is truncated at a lower mass, decay channels with lower mass thresholds may be enhanced. A good example is the ρ^0 which, as a broad low-mass resonance, has any number of non-perturbative and threshold effects. The mass distribution for the ρ^0 is limited by the rare e^+e^- decay channel. However, truncating the mass distribution at this mass threshold can result in oversampling the lighter decay channels. Consequently, the mass threshold for the $\pi\pi$ channel is used instead for the default low-mass truncation of the ρ^0 mass distribution.

For resonances, widths are sampled using the relativistic Breit–Wigner of eq. (366), but with a number of options available for determining the partial widths of the resonance at a given mass. This calculation method can be set by the user with the `NN:meMode` parameter when defining the decay channels for a particle. The default value is 0, where the partial width is calculated perturbatively for the resonance if already available in PYTHIA. If a given width is not available via a perturbative calculation, then this width is set to zero. However, a number of alternative partial width calculations are available.

- `NN:meMode = 100`: The partial width is defined as the branching fraction for that decay channel, multiplied by the total width. This method results in mass-independent widths and does not account for mass-threshold effects, which may result in issues when a resonance is produced significantly off-shell with a mass below the on-shell mass. When this occurs, it is possible that no decay channels remain kinematically open, and the resonance can no longer be decayed. However, it is also possible that downward mass fluctuations may occur in the masses of the decay products, allowing some channels. Consequently, all decay channels are considered whenever a resonance is decayed, even if the on-shell masses of the products kinematically exclude the channel.

- `NN:meMode = 101`: The partial widths are calculated in the same fashion as for `NN:meMode = 100`, but are now set to zero if the sum of the on-shell masses for the decay products is not kinematically allowed at the mass for which the partial width is being calculated. Consequently, the total width becomes mass dependent through the introduction of step functions at the kinematic limits for each decay channel.

- `NN:meMode = 102`: This method builds upon the method of `NN:meMode = 101` but uses a smooth threshold factor, rather than a step function. For two-body decays the partial width is multiplied by the factor,

$$\beta = \sqrt{(1 - m_1^2/m^2 - m_2^2/m^2)^2 - 4m_1^2 m_2^2/m^4}\,, \tag{367}$$

where m_i are the masses of the decay products and m is the selected mass of the decaying resonance. While this correctly includes the phase-space suppression for an isotropic two-body decay, any channel specific modifications due to the matrix element for the decay are not included. For higher multiplicity decays, a less sophisticated factor of,

$$\beta = \sqrt{1 - \sum_i m_i/m}\,, \tag{368}$$

is used which roughly approximates the phase-space suppression. For this method, the branching ratio for each decay channel should be provided without a phase-space suppression factor, otherwise phase-space suppression for that channel will be double counted. When using this method, the branching fractions for the resonance as calculated by PYTHIA will not match those provided by the user.

- `NN:meMode = 103`: The phase-space suppression of `NN:meMode = 102` is used, but the branching fraction for the channel is divided by the β factor calculated at the on-shell mass of the resonance, β_0^{-1}. Consequently, the branching fractions no longer need to be adjusted for phase-space suppression, and the branching fractions calculated by PYTHIA will match those provided by the user. However, in some cases β_0^{-1} can be very large if a channel is very near threshold for the nominal mass of the resonance. The parameter `ResonanceWidths:minThreshold` defines the minimum allowed β_0 and limits the correction for resonance masses well above the on-shell mass.

Note that it is possible to mix and match partial width calculation methods for a given resonance, *i.e.* some decay channels may have their partial width calculated perturbatively, while the methods outlined above are used for others.

8.1.3 Lifetimes

While the lifetime of a particle is inversely related to its width, decoupling the lifetime and width of a particle is oftentimes useful for practical purposes. Consequently, both the lifetime and the width of a particle species can be specified independently in PYTHIA. The lifetime is given as the nominal proper lifetime multiplied by the speed of light, $c\tau_0$, and has units of millimetres. For particles with a non-zero lifetime, a lifetime is sampled according to an exponential decay,

$$\mathcal{P}(\tau)\mathrm{d}\tau \propto \exp(-\tau/\tau_0)\mathrm{d}\tau\,, \tag{369}$$

where the τ_0 used here is not calculated from the width, but rather specified independently. When the hadronic-rescattering framework is enabled and the independently provided τ_0 is zero, the nominal proper lifetime is automatically calculated using the width, if the particle species has at least a single available decay channel. See section 7.4 for details. Similarly, missing lifetimes are calculated when vertex positions and rapid hadron decays are enabled in the hadronization. For resonances, τ_0 is automatically determined from the calculated width of the resonance. However, in some cases very long lifetimes are necessary, which could result in such narrow widths that the calculation of the cross section becomes numerically unstable. Here, the width and lifetime for a resonance can be made independent by setting the flag `NN:tauCalc = false` for that resonance. This can be particularly useful when scanning lifetime space for BSM resonances.

After the lifetime for a particle or resonance is selected, the decay vertex position is calculated as,

$$x_{\mathrm{dec}} = x_{\mathrm{pro}} + \tau\,\frac{p}{m}\,, \tag{370}$$

where m is the mass of the particle, p the momentum, and x_{pro} the production-vertex position that may be either the primary interaction point or from some previous decay. This treatment of the decay vertex assumes all particles travel without interaction, including no magnetic fields or interactions with detector materials. Consequently, decay chains can be stopped to allow the subsequent decays of the particles to be handed to a detector simulation. A number of criteria for stopping decays is available. Particles with a specified minimum nominal lifetime can be stopped from decaying with the flag `ParticleDecays:limitTau0`. Similarly, particles with a selected lifetime greater than a configurable minimum lifetime can be set stable

with the `ParticleDecays:limitTau` flag. Particles can also be limited from decaying geometrically, either within a sphere with `ParticleDecays:limitRadius`, or within a cylinder with `ParticleDecays:limitCylinder`.

8.2 Decays

Table 2: Available matrix element modes for particle decays. Here, V is a vector meson, P a pseudoscalar, H a generic hadron, X any non-partonic initial state, and A and B any non-partonic final states.

force	process	eq.	meMode
any	$X \to q\bar{q}$ or $X \to gg$	none	91
any	$X \to q\bar{q}A$, where A is a colour singlet	none	93
any	$X \to q\bar{q}\ldots$	none	42 - 80
any	$H \to AB$	(372)	3 - 7
strong	$V \to \pi^+\pi^-\pi^0$, where V is an isoscalar	(376)	1
strong	$P \to PV[\to PP]$	(377)	2
strong	$P \to \gamma V[\to PP]$	(378)	2
strong	$V \to ggg$ or $V \to \gamma gg$	(379)	92
EM	$H \to A\gamma^*[\to \ell^+\ell^-]$	(380), (381)	11
EM	$H \to q\bar{q}\gamma^*[\to \ell^+\ell^-] \to A\ell^+\ell^-$	(380), (381)	11
EM	$H \to AB\ldots\gamma^*[\ell^+\ell^-]$	(380), (381)	12
EM	$H \to q\bar{q}\gamma^*[\to \ell^+\ell^-] \to AB\ell^+\ell^-$	(380), (381)	12
EM	$H \to \gamma^*[\to \ell^+\ell^-]\gamma^*[\to \ell^+\ell^-]$	(380), (381)	13
weak	$H \to \bar{\nu}_\ell\ell^- A$	(382)	22/23
weak	$H \to \bar{\nu}_\ell\ell^- q\bar{q}$	(382)	22/23
weak	$X \to q\bar{q}A$, where A is a colour singlet	(382)	94
weak	$H \to \bar{\nu}_\ell\ell^- AB\ldots$	(383)	22/23
weak	$H \to q\bar{q}q\bar{q}$	(384)	22, 23
weak	$\ell^- \to \nu_\ell A\ldots$	(384)	21
weak	$\ell^- \to \bar{\nu}_\ell\ell^-\ell^+\nu_\ell$	(382)	22/23
weak	$H \to \gamma q\bar{q}$	(385)	31

Particle decays might at first appear to be one of the simpler components of PYTHIA, given the clear factorization between the production and decay of particles. The masses, widths, and decay channels for most particles can be set directly to experimentally observed values, and typically do not require sophisticated calculations. Once this information is provided, a particle can be decayed by randomly selecting a decay channel with a weight proportional to its branching fraction, and then distributing the products of the selected channel according to phase space. However, there are a number of complications which require modifications to this initial approach.

The technical generation of phase space for decays with more than three products can be non-trivial to perform efficiently, and requires the use of specialized algorithms such as the M-generator or RAMBO, which are introduced in section 2.2.4. After phase-space generation, a matrix-element weight can be applied to ensure the correct kinematic distribution, given the nature of the decay. For particles with non-zero spin, spin effects can change the kinematic distribution not only for a single decay, but also between correlated decays. Finally, additional photons need to be probabilistically included in radiative decays.

All these complications assume the decay channel is exclusive, *i.e.* the number and type of decay products is fixed. For many decays, such as those of charm and bottom hadrons, this is not the case. A full list of the available decays are provided in table 2. About 40% of decay channels in PYTHIA have dedicated matrix elements, corresponding to 50% of decays when weighted by branching fraction. The remainder of this section describes these decays.

8.2.1 Hadron decays with parton showers

The decays of many particles are not known in an exclusive hadronic form but instead, the relative rates between exclusive partonic channels is known. Consequently, it is necessary to evolve these exclusive partonic decays into final state hadrons. In PYTHIA there are two mechanisms for this evolution. In the first method, the partons are passed to the timelike parton shower of section 4, followed by the hadronization of section 7.1. This method is used for $b\bar{b}$ states, and typically the parton shower does not significantly modify the decay. By default, the partons produced in the decay are distributed uniformly in phase space, with the notable exception of `NN:meMode = 92` detailed in section 8.2.4 and `NN:meMode = 94` detailed in section 8.2.6.

A number of parton and colour configurations are available for this type of inclusive decay via the parton shower as follows. Here, c_i is used to indicate a colour index and $\bar{c}_i$ anti-colour index.

- $q\bar{q}$: The quark carries c_1 the antiquark $\bar{c}_1$. This type of decay is set with `NN:meMode = 91`. Examples of decays using this matrix element mode are $\Upsilon \to q\bar{q}$. Hidden valley hadrons also heavily utilize this decay.

- gg: The first gluon carries c_1 and $\bar{c}_2$ while the second gluon carries c_2 and $\bar{c}_1$. This decay is also specified by `NN:meMode = 91` and is primarily used for quarkonia, *e.g.* $\eta_b \to gg$.

- ggg: The first gluon carries c_1 and $\bar{c}_2$, the second c_2 and $\bar{c}_3$, and the third c_3 and $\bar{c}_1$. This configuration is intended for the decays of quarkonia, *e.g.* Upsilon $\to ggg$, and set with `NN:meMode = 92`.

- $gg\gamma$: The first gluon carries c_1 and $\bar{c}_2$ and the second c_2 and $\bar{c}_1$. This decay is also intended for quarkonium decays, *e.g.* $\Upsilon \to \gamma gg$ and is set with `NN:meMode = 92`.

- $q\bar{q}X$: This is the same as the colour-singlet $q\bar{q}$ decay mode, except with an additional colour singlet X, and is selected with `NN:meMode = 93` for flat phase space, and `NN:meMode = 94` for a weak decay.

For all of these decays, the ordering of the partons as passed to PYTHIA does not matter.

8.2.2 Inclusive hadron decays

The second method for inclusive hadronic decays is to first determine hadrons from the partons and then distribute these hadrons in phase space. This method is used primarily for multibody decays of hadrons such as the D and B mesons, where only a few channels are known experimentally. The flavours for a channel can then be dynamically built from the initial partonic content of a weak decay. For this type of decay, either one or two parton pairs can be specified in the decay, in addition to any non-parton particles. Here, a parton is either a quark or diquark. The number of final particles is determined from a Poisson distribution with a mean of,

$$\lambda = \frac{n_{\text{known}} + n_{\text{spec}}}{2} + \frac{n_{\text{partons}}}{4} + \rho_{\text{mult}} \ln(m_{\text{diff}}/m_{\text{mult}}), \tag{371}$$

where n_{known} is the number of non-partonic particles in the specified decay, n_{spec} is the number of spectator partons, and n_{partons} is the number of partons. Here, the spectator partons are those partons that do not participate in the partonic weak decay. The mass m_{diff} is the difference between the decaying particle mass and the sum of the nominal decay-product masses. A reference mass m_{mult} is set by the parameter `ParticleDecays:multRefMass` and can be used to tune the average decay multiplicity. An additional factor, ρ_{mult}, also determines the average decay multiplicity and is set via the parameter `ParticleDecays:multIncrease` for all relevant matrix-element modes except `NN:meMode = 23` where `ParticleDecays:-multIncreaseWeak` is used instead. See section 8.2.6 for further details.

The method for selecting the final hadrons is as follows.

1. The multiplicity is selected according to eq. (371) and is required to be less than 10. A minimum multiplicity can be required by setting the `NN:meMode` between 42 and 50, where the minimum multiplicity is given by `meMode - 40`. Alternatively, the multiplicity can be fixed by setting the `NN:meMode` between 62 and 70. Here the multiplicity is calculated as `meMode - 60`.

2. The number of hadrons to form is the difference between the selected multiplicity from the previous step, and the number of non-parton particles in the decay.

3. One of the partons is selected at random and a new parton and hadron is formed, following the flavour selection of section 7.1.1.

4. The previous step is repeated until the number of remaining hadrons to select is the same as the number of parton pairs.

5. The remaining parton pairs are formed into hadrons.

6. If there are two pairs, they may be reshuffled, as determined by the probability `ParticleDecays:colRearrange`, *i.e.* for a value of 0 the pairs will never be reshuffled but for a value of 1 they will always be reshuffled.

7. If the mass of the final decay products is less than the decaying particle, the hadron selection is kept, otherwise the process begins again with step 1.

This model is very similar to the hadronization model, but the momenta of the hadrons is now just determined with phase space. For most decays this approximation is valid as the decay-product momenta should be very low and on average reproduce the correct kinematic behaviour. While the flavour selection is the same as for hadronization, the mass constraint of step 7 will typically bias decays to the lighter pseudoscalar mesons, particularly for high multiplicity decays.

For these types of inclusive decays, the special particle ID 82 can be used to randomly select a light flavour pair, *i.e.* $u\overline{u}$, $d\overline{d}$, or $s\overline{s}$. The suppression of selecting an $s\overline{s}$ pair with respect to $u\overline{u}$ and $d\overline{d}$ is configured by the parameter `StringFlav:probStoUD` which is also used in the flavour selection of the hadronization algorithm of section 7.1.1. When specifying decays with this ID, the channel should be given as an 82 -82 pair, where the ordering does not matter. A similar ID is 83 which is the same as 82, but intended for decays that proceed through a gluon loop. Since this loop will increase the average multiplicity of the decay, eq. (371) is modified by adding an additional constant specified by the parameter `ParticleDecays:multGoffset`. The primary decay of the J/ψ into three gluons, as well as many of the other onium states, use this special ID.

For some particles, exclusive decays must be specified in addition to inclusive decays. Matrix-element modes are provided in PYTHIA to prevent double counting the exclusive decays

in the inclusive decays. An `NN:meMode` between 52 and 60 reproduces the same behaviour as an `NN:meMode` between 42 and 50, but will exclude any generated final state that matches a non-partonic decay channel. An example of such a decay is $\eta_c \to q\bar{q}$. Similarly, if `NN:meMode` is between 72 and 80, the behaviour for `meModes` between 42 and 50 is reproduced, but again excluding any generated final state that matches a non-partonic decay channel.

8.2.3 Variable-width hadrons

For standard particle decays, the probability used to select a decay channel is calculated using a fixed branching ratio, independent of the decaying particle mass. The hadronic rescattering framework (*cf.* section 7.4), however, includes mass-dependent partial widths for two-body decays of hadrons. For hadrons included in the rescattering framework, decay channels are picked using these partial widths. The partial width for the decay of a hadron resonance H into particles A and B, $H \to AB$, is given by,

$$\Gamma_{H \to AB}(m) = \Gamma_0 \frac{m_0}{m} \frac{\Phi(2l+1,m)}{\Phi(2l+1,m_0)} \frac{1.2}{1.0 + 0.2\frac{\Phi(2l,m)}{\Phi(2l,m_0)}} \, . \tag{372}$$

Here, Γ_0 is the nominal partial width of the decaying hadron at its nominal mass m_0, set from experiment. The angular momentum of the two-body decay is given by l. In PYTHIA, this angular momentum is specified by the user as $l = \text{meMode} - 3$. At high masses the final multiplicative factor regulates the partial width. Similar to resonance production, see section 8.1.2, these partial widths define not only the branching fractions of the hadron but also production.

The phase space is given by

$$\Phi(l,m) = \int dm_A \int dm_B \, q^l(m, m_A, m_B) BW(m_A) BW(m_B) \, , \tag{373}$$

where $q(m, m_A, m_B)$ is the magnitude of the A and B momentum in the centre-of-mass frame,

$$q(m, m_A, m_B) = \frac{\sqrt{(m^2 - (m_A + m_B)^2)(m^2 - (m_A - m_B)^2)}}{2m} \, . \tag{374}$$

Finally, the mass distribution for each of the two decay products is given by a Breit–Wigner,

$$BW(m) = \frac{1}{2\pi} \frac{\Gamma(m)}{(m^2 - m_0^2)^2 + \frac{1}{4}\Gamma^2(m)} \, . \tag{375}$$

While this mass distribution does include a mass-dependent width, phase-space considerations ensure these mass-dependent widths can be evaluated recursively from the lowest mass particle to the highest. Note that performing decays with variable partial widths only affects the branching ratios of the decay channels, and not the angular distribution of the decay products. By default, a number of hadrons are decayed using variable partial widths in PYTHIA. This includes many of the excited mesons as well as a number of the baryons. For technical reasons, variable partial width decays are never performed for the ρ or f_2 mesons.

8.2.4 Strong decays

Most decays proceeding via the strong force in PYTHIA are modelled with pure phase space. However, there are four special cases that are generated according to matrix elements: isoscalar vector mesons decaying into pseudoscalar mesons, pseudoscalar mesons decaying into a pseudoscalar and vector mesons, pseudoscalar mesons decaying into a photon and vector meson, and vector mesons decaying into a three gluon final state.

The ω meson decays predominantly into a three-pion final state of $\pi^+\pi^-\pi^0$. This decay can be modelled using the isobar model [380], where the decay proceeds via the intermediate $\rho^0\pi^0$ or $\rho^\pm\pi^\mp$ state. The matrix element for this decay is given by

$$|\mathcal{M}|^2 \propto \left[(m_1 m_2 m_3)^2 - (m_1 p_2 p_3)^2 - (m_2 p_1 p_3)^2 - (m_3 p_1 p_2)^2 + 2(p_1 p_2)(p_1 p_3)(p_2 p_3)\right]|\mathcal{F}|^2 , \tag{376}$$

where m_i and p_i are the mass and momentum of decay product i. Here, π^+ corresponds to $i = 1$, π^- to $i = 2$, and π^0 to $i = 3$. The function $\mathcal{F}$ includes possible final-state interactions of the pions, and depends upon the full kinematics of the decay. When no final-state interactions are present, $\mathcal{F} = 1$, which corresponds to P-wave distributed phase space. In PYTHIA, this assumption of no final-state interactions is made. However, there is experimental evidence that final-state interactions could play an important role in this decay [381].

The ϕ meson is also an isoscalar like the ω meson and has a non-negligible branching to the $\rho^0\pi^0$ and $\rho^\pm\pi^\mp$ channels, where the larger ϕ mass provides sufficient phase space for a ρ resonance. However, a contact $\pi^+\pi^-\pi^0$ decay, without the ρ-resonance structure, is also possible [382], and is described by the same matrix element as for the ω meson. The ρ^0 itself can also decay into a $\pi^+\pi^-\pi^0$ final state described by this matrix element, although this decay channel is heavily suppressed due to phase space. For both the ϕ and ρ^0 mesons, no final-state interactions are considered in these decay channels. The matrix element of eq. (376) can be selected by setting NN:meMode = 1.

In the decay chain $P_0 \to P_1 V_2[\to P_3 P_4]$, where P is a pseudoscalar meson and V a vector meson, the decay products P_3 and P_4 are distributed in the rest frame of V_2 according to $\cos^2\theta$, where θ is the angle between P_0 and P_3. The corresponding matrix element, is given by

$$|\mathcal{M}|^2 \propto (p_0 p_2)(p_2 p_3) - m_2^2(p_0 p_3) , \tag{377}$$

where again i specifies the particle in the decay chain, m_i is the mass of that particle, and p_i is the momentum. Similarly, for the decay chain $P_0 \to \gamma V_2[\to P_3 P_4]$, the distribution of P_3 and P_4 is now given by $\sin^2\theta$ in the rest frame of V_2. The matrix element for this decay is,

$$|\mathcal{M}|^2 \propto m_2^2\left[2(p_2 p_3)(p_0 p_2)(p_0 p_3) - m^2(p_2 p_3)^2 - m_2^2(p_0 p_3)^2 - m_3^2(p_0 p_2)^2 + (m m_2 m_3)^2\right]. \tag{378}$$

While these two matrix elements are relevant for all appropriately produced vector-meson decays into a pseudoscalar-meson pair, in practice the relevant vector-meson decay channels are: $\rho \to \pi\pi$, $\omega \to \pi^+\pi^-$, $K^* \to K\pi$, $\phi \to KK$, $\phi \to \pi^+\pi^-$, and $D^* \to D\pi$. Note that when the vector meson is not produced in the decay chain $P_0 \to P_1/\gamma V_2$, these matrix elements are not used. As an example, in the decay chain $D \to \pi K^*[\to K\pi]$, the decay products of the K^* are distributed according to eq. (377). To use these matrix elements, NN:meMode = 2 must be set.

For the decays of vector-like onium states into a partonic final state of gluons, $V_0 \to g_0 g_1 g_2$, or gluons and a photon, $V_0 \to \gamma_0 g_1 g_2$, the matrix element,

$$|\mathcal{M}|^2 \propto \left(\frac{1-x_1}{x_2 x_3}\right)^2 + \left(\frac{1-x_2}{x_1 x_3}\right)^2 + \left(\frac{1-x_3}{x_1 x_2}\right)^2 , \tag{379}$$

is used. Here, x_i is twice the energy of particle i divided by the mass of the decayer in the rest frame of the decayer, $2E_i/m$. For the two gluon and photon decay, the two-gluon system is required to have a minimum mass configured by the parameter StringFragmentation:stopMass to ensure that the system can properly hadronize. This matrix element is set using meMode = 92 as is done for the partonic decays $\Upsilon \to ggg$ and $\Upsilon \to \gamma gg$. Because eq. (379) is symmetric, ordering of the decay products when configuring PYTHIA does not matter.

8.2.5 Electromagnetic decays

The electromagnetic decay $\pi^0 \to \gamma^*[\to e^+e^-]\gamma$ can be generated with a factorized approach. To begin, the γ^* mass is selected, using the decay matrix element integrated over the solid angle, but still dependent upon the γ^* mass, m_1.

$$|\mathcal{M}|^2 \propto \frac{1}{m_1^2}\left(1 + \frac{2m_2^2}{m_1^2}\right)\sqrt{1 - \frac{4m_2^2}{m_1^2}}\left(1 - \frac{m_1^2}{(m-m_\text{max})^2}\right)^3 \frac{1}{(m_{\rho^0}^2 - m_1^2)^2 + m_{\rho^0}^2\Gamma_{\rho^0}^2}. \tag{380}$$

The subscript i is 0 for the π^0, 1 for the virtual γ^*, 2 for the e^+, 3 for the e^-, and 4 for the real γ; the mass for each particle is given by m_i and m_max is the maximum mass of the off-shell photon, $i.e.$ $m_\text{max} = m_4 = 0$ for this decay channel of the π^0. The final factor of this expression is the VMD propagator for the ρ^0, where m_{ρ^0} is the mass of the ρ^0, and Γ_{ρ^0} the width. This propagator is negligible for any decaying particle with a mass far from the ρ^0 mass, which includes the case of the π^0. Next, after the γ^* mass is selected, the two-body decay of $\pi^0 \to \gamma^*\gamma$ is performed. Finally, the angular distribution of the e^+e^- pair is generated according to,

$$|\mathcal{M}|^2 \propto (m_1^2 - 2m_2^2)\left[(qp_2)^2 + (qp_3)^2\right] + 4m_2^2\left[(qp_2)(qp_3) + (qp_2)^2 + (qp_3)^2\right], \tag{381}$$

where p_i is the momentum of the corresponding particle with index i, and $q = p_0 - p_1$. For efficiency and simplicity, this angular distribution is generated in the rest frame of the decaying particle, which if highly boosted, can result in minor numerical induced violations in momentum-energy conservation. Consequently, the momentum of the final lepton is calculated as $p_3 = p_1 - p_2$ in the laboratory frame.

The matrix element for this decay channel is also valid for similar processes where a lepton pair, $\ell^+\ell^-$, is produced via an off-shell photon. Such decay channels include $\eta \to \ell^+\ell^-\gamma$, $\omega \to \ell^+\ell^-\pi^0$, $\phi \to \ell^+\ell^-\eta$, $B \to \ell^+\ell^- K/K^*$, $B_s^0 \to \ell^+\ell^-\phi$, and $\Sigma^0 \to \ell^+\ell^-\Lambda^0$. This matrix element can also be used for the final state $\ell^+\ell^- q\bar{q}$. In this particular case, the $q\bar{q}$ is converted into a single hadron, following the inclusive decay selection of section 8.2.2 but with the multiplicity of the decay fixed to three. In all the cases described above, the matrix element for these decay channels is set with `NN:meMode = 11`.

The form of eq. (380) and (381) are also approximately valid for decay channels with the final state $\gamma^*[\ell^+\ell^-]AB\ldots$, where there are two or more decay products in addition to the lepton pair. Such decays include $\eta \to \ell^+\ell^-\pi^+\pi^-$, $K_S^0 \to \ell^+\ell^-\pi^+\pi^-$, $B^0 \to \ell^+\ell^-\pi^0\pi^0$, and $B^+ \to \ell^+\ell^- u\bar{s}$. For this type of decay channel, eq. (380) is still used to select the mass, but with $m_\text{max} = m_A + m_B + \ldots$, and eq. (381) is used without modification. The phase-space generation, after selecting m_1, is now performed as a decay with multiplicity $n-1 > 2$, where n is the final multiplicity of the decay. Setting `NN:meMode = 12` selects this matrix element. If A and B are replaced with a $q\bar{q}$ final state, the system is collapsed down into two hadrons, with the flavour selection again performed using the inclusive decay algorithm but with the multiplicity fixed at four.

Finally, these matrix elements are also used to approximate $\gamma^*[\ell^+\ell^-]\gamma^*[\ell^+\ell^-]$ decay channels. Following the same numbering convention, the mass of the first off-shell photon, m_1 is selected using eq. (380) where $m_\text{max} = m_5 + m_6$, $i.e.$ twice the mass of the second lepton flavour. Then, the mass of the second off-shell photon, m_3, is selected again with eq. (380) but using indexing $i-2$ and setting $m_\text{max} = m_2 + m_3$. After performing the two-body decay of the $\gamma^*\gamma^*$ system, the angular distributions for the two lepton pairs are generated independently using eq. (381). This type of decay channel is specified with `NN:meMode = 13` and can be used for decays such as $\pi^0 \to e^+e^-e^+e^-$. The technical implementation for all decays using eqs. (380) and (381) require that the lepton pair should always be set as the final two decay products when defining these decay channels.

8.2.6 Weak decays

The helicity averaged matrix element for the t-channel weak scattering of fermions, $f_0 f_1 \to f_2 f_3$, is,

$$|\mathcal{M}|^2 \propto (p_0 p_1)(p_2 p_{\text{rem}}), \tag{382}$$

where p_i is the momentum of particle with index i and $p_{\text{rem}} = \sum_{i=3} p_i$ is the sum of the remaining momenta, which here is just p_3. By crossing symmetry, this matrix element can also be used for weak decays. An example is the fully leptonic decay of the τ lepton, $\tau^- \to \bar{\nu}_\ell \ell^- \bar{\nu}_\tau$. The particle ordering determines the corresponding i for each particle in eq. (382), *e.g.* $i = 1$ for the anti-neutrino and $i = 2$ for the charged lepton. This matrix element can also be used to approximate semi-leptonic decays of D and B mesons, *e.g.* $D^0 \to \ell^+ \nu_\ell \pi^-$ or $B^0 \to \nu_\ell \ell^+ \pi^-$, where the final fermion pair is collapsed into a single hadron. In this example, the ordering of the neutrino and charged anti-lepton is swapped between the two decays. This is because for D-meson decays, the partonic f_0 is a c quark, while for the B-meson this is a b antiquark. Similarly, this matrix element can be used for the semi-leptonic decays of baryons, *e.g.* $n \to \bar{\nu} e^- p$, or the leptonic decays of charged leptons, *e.g.* $\mu^- \to \bar{\nu}_e e^- \nu_\mu$. When not using the sophisticated τ decays of section 8.2.8, this helicity averaged matrix element can also be used for the leptonic decays of the τ.

Semi-leptonic decays can also be specified with their partonic content, *e.g.* $D^0 \to \ell^+ \nu_\ell d\bar{u}$ or $B^0 \to \nu_\ell \ell^+ d\bar{u}$, where the ordering of the quarks does not matter. Similarly, baryon decays of this nature like $\Xi_c^0 \to e^+ \nu_e s(sd)_0$, can be decayed using this matrix element where one of the partons is a diquark, *i.e.* $(sd)_0$. When partonic content is specified, the parton system is collapsed to a single hadron following the flavour-selection rules of section 7.1.1. The matrix element of eq. (382) is used for all the decays described above by setting either `NN:meMode = 22` or `NN:meMode = 23`. For these types of decays there is no difference between these two matrix-element modes. The only technical requirement for these decays is that the first two particles of the decay are the neutrino/charged-lepton pair, followed by either a hadron or a parton pair, where ordering of the partons does not matter. In some cases it is convenient to use the special particle ID 81 to act as a place holder for the spectator quark or diquark, which is then automatically replaced with the correct spectator flavour. For baryons, an ambiguity can arise in this selection where the spin of the diquark cannot always be determined uniquely. For the example decay of the Ξ_c^0 given here, the spectator flavour can either be $(sd)_0$ or $(sd)_1$, while the automatic flavour will always select the $(sd)_0$ diquark.

In some cases, semi-leptonic decays with more than one final-state hadron are needed, *e.g.* $D^0 \to e^+ \nu_e K^0 \pi^-$. The additional hadrons can be physically interpreted as being produced from the fragmentation of the spectator parton, resulting in hadrons with a significantly softer momentum than the hadron containing the spectator quarks. This softer momentum is modelled by taking the product of eq. (382) and an exponential damping factor,

$$|\mathcal{M}_{\text{damp}}|^2 \propto |\mathcal{M}|^2 \prod_{i=4} e^{-|p_i|^2/\sigma_{\text{soft}}^2}, \tag{383}$$

which is calculated in the rest frame of the decay, where the product is taken over all hadrons following the spectator hadron with momentum magnitude $|p_i|$. Here, $|\mathcal{M}|^2$ is calculated with eq. (382) and σ_{soft} is the damping term which can be configured by the user with the parameter `ParticleDecays:sigmaSoft`. A single damping parameter is used for all decays and is expected to fall within the range $0.2 - 2$, where a smaller value increases the damping. For semi-leptonic decays with two or more final state hadrons, this matrix element can be used by setting `NN:meMode = 22` or `NN:meMode = 23`. Again, there is no difference between these two matrix-element modes for decays of this type. As before, the ordering of the decay as passed to PYTHIA matters. The neutrino/charged-lepton pair must be specified first, in

the correct order as discussed above, followed by the hadron containing the spectator quark, followed by any remaining hadrons, which will then have their momentum damped.

The matrix element for weak decays into purely hadronic final states, where the decay is defined only by partonic content, is approximated by PYTHIA. An example of this class of decay is $B^0 \rightarrow u\bar{d}\bar{c}d$ which will result in a final state with a D meson. The partonic content should be set as $q_1q_2q_3q_4$ where q_1 and q_2 are colour connected, and either q_3 or q_4 is the spectator quark/diquark. The special particle code 81 can be used here to automatically determine the spectator flavour. Just like for the partonic semi-leptonic decays, the final two partons are collapsed into a single hadron following the flavour-selection rules of section 7.1.1. The first two partons are then fragmented into multiple hadrons, following the method of section 8.2.2.

When `NN:meMode = 22` is used, the mean number of final particles in the decay is calculated with eq. (371) using the ρ_{mult} parameter `ParticleDecays:multIncrease`. When `NN:meMode = 23` is used instead, the mean number of final particles is calculated using `ParticleDecays:multIncreaseWeak`. The former parameter is intended, although not required, to be smaller than the latter, since in weak decays only the mass of the off-shell W boson is available to the fragmenting partonic system, and not the entire parent mass. Additionally, for `NN:meMode = 23` a minimum of three final particles are required in the decay after flavour selection. After the final particles are determined for each decay, the matrix element,

$$|\mathcal{M}|^2 \propto \frac{2E_1}{m}\left(3 - \frac{4E_1}{m}\right), \tag{384}$$

is used where m is the mass of the decaying hadron and E_1 is the energy of the hadron containing the spectator quark, in the rest frame of the decay. This matrix element can also be used for hadronic τ decays when the sophisticated treatment is not needed by specifying `NN:meMode = 21`. Here, the first decay product should always be the ν_τ, which increases the energy of the neutrino with respect to flat phase space.

Partonic radiative decays via the weak force are roughly approximated with the matrix element,

$$|\mathcal{M}|^2 \propto \left(\frac{2E_1}{m}\right)^3, \tag{385}$$

where m is the mass of the decaying hadron and E_1 is the energy of the photon in the rest frame of the decay. Effectively, this increases the photon energy with respect to flat phase space. The partonic content for these decays should be set as a photon, the spectator quark, and the flavour-changing quark, e.g. $B^0 \rightarrow d\bar{s}\gamma$ where d is the spectator quark. Unlike the previous weak decays, where the spectator system is collapsed to a single hadron, the spectator system is fragmented into multiple hadrons following the inclusive selection of section 8.2.2. However, the multiplicity for the decay is selected with a geometric distribution,

$$P(n) = \left(1 - \frac{1}{2}\right)^{n-1} \frac{1}{2}, \tag{386}$$

rather than a Poisson distribution, where a minimum multiplicity of 2 and a maximum multiplicity of 10 is required. This type of decay is specified by setting `NN:meMode = 31`, and the decay products can be assigned in an arbitrary order.

In all the decays above, the matrix element is applied to the final particles of the decay, not the partonic content. In some cases it is useful to apply the matrix element to the partonic content of the decay, and then perform a full parton shower followed by hadronization, using the parton-shower method of section 8.2.1. Specifying `NN:meMode = 94` does this, where the matrix element of eq. (382) is used to distribute the phase space of the partons from the decay.

In addition to the weak decays described above, B systems may mix prior to decay. This mixing is controlled by the flag `ParticleDecays:mixB` and has a probability of,

$$\mathcal{P} = \sin^2\left(\frac{x\tau}{2\tau_0}\right), \tag{387}$$

where τ is the selected lifetime of the particle, and τ_0 the nominal proper lifetime. The mixing parameter x is set with `ParticleDecays:xBdMix` and `ParticleDecays:xBsMix` for the B_d and B_s systems, respectively.

8.2.7 Helicity decays

A generic helicity-density formalism is available in PYTHIA which can be used for τ decays as well as muon decays in lepton-flavour violating production. External tools have also used this framework for heavy-neutral-lepton decays. The weight for an n-body decay of an arbitrary particle is given by,

$$\mathcal{W} = \rho_{\lambda_0\lambda_0'}\mathcal{M}_{\lambda_0;\lambda_1\dots\lambda_n}\mathcal{M}^*_{\lambda_0';\lambda_1'\dots\lambda_n'}\prod_{i=1,n}\mathcal{D}^{(i)}_{\lambda_i\lambda_i'}. \tag{388}$$

The decaying particle is given index 0 and the decay products are assigned indices i through n. The helicity for each particle is given by λ_i and summations are performed over each repeated helicity index. The helicity density matrix for the decaying particle is given by ρ, while the decay matrix for each decay product is given by $\mathcal{D}$. The helicity matrix element for the decay is $\mathcal{M}$ and depends upon the helicity of the decaying particle as well as the decay products.

For a particle produced from a $2 \rightarrow n$ hard process, the helicity-density matrix for an outgoing particle with index i is given by,

$$\rho^{(i)}_{\lambda_i\lambda_i'} = \rho^{(1)}_{\kappa_1\kappa_1'}\rho^{(2)}_{\kappa_2\kappa_2'}\mathcal{M}_{\kappa_1\kappa_2;\lambda_1\dots\lambda_n}\mathcal{M}^*_{\kappa_1'\kappa_2';\lambda_1'\dots\lambda_n'}\prod_{j\neq i}\mathcal{D}^{(j)}_{\lambda_j\lambda_j'}, \tag{389}$$

where $\rho^{(1,2)}$ are the helicity-density matrices for the incoming particles, $\mathcal{M}$ is the helicity matrix element for the process, and $\kappa_{1,2}$ are the helicities of the incoming particles. For incoming two-helicity-state beam particles with a known longitudinal polarization $\mathcal{P}_z$ the helicity-density matrix is diagonal with elements $(1 \pm \mathcal{P}_z)/2$.

Before any particles are decayed in a given sequence, all decay matrices in eq. (388) and eq. (389), $\mathcal{D}$, are initialized to the identity matrix. In a $2 \rightarrow n$ process, a first outgoing particle is randomly selected and decayed using a helicity-density matrix determined with (389). The decay matrix for this first decay is calculated as

$$\mathcal{D}^{(0)}_{\lambda_0\lambda_0'} = \mathcal{M}_{\lambda_0;\lambda_1\dots\lambda_n}\mathcal{M}^*_{\lambda_0';\lambda_1'\dots\lambda_n'}\prod_{i=1,n}\mathcal{D}^{(i)}_{\lambda_i\lambda_i'}. \tag{390}$$

After the full decay tree for this first particle is determined, the remaining particles for the $2 \rightarrow n$ process are then randomly selected and decayed using the helicity-density matrix of (389) with the updated decay matrices for the already decayed outgoing particles.

When a particle from the hard process is selected for decay, the full decay tree of that particle is performed. A single branch of the decay tree is followed until a final stable particle is reached. The helicity-density matrices for particles produced from decays are calculated with,

$$\rho^{(i)}_{\lambda_i\lambda_i'} = \rho^{(0)}_{\lambda_0\lambda_0'}\mathcal{M}_{\lambda_0;\lambda_1\dots\lambda_n}\mathcal{M}^*_{\lambda_0';\lambda_1'\dots\lambda_n'}\prod_{j\neq i}\mathcal{D}^{(j)}_{\lambda_j\lambda_j'}, \tag{391}$$

where the $\rho^{(0)}$ is the helicity-density matrix of the parent particle. The algorithm then calculates the decay matrix for the last particle decayed with eq. (390) and the next undecayed

Table 3: Summary of available τ^- decay models in PYTHIA. The ν_τ is omitted from the decay products for brevity and charge conjugation is implied for τ^+ decays.

mult.	ref.	meMode	decays
2		1521	π^-, K^-
3		1531	$e^- \bar{\nu}_e, \mu^- \bar{\nu}_\mu$
	[385]	1532	$\pi^0 \pi^-, K^0 K^-, \eta K^-$
	[386]	1533	$\pi^- \bar{K}^0, \pi^0 K^-$
4	[387]	1541	$\pi^0 \pi^0 \pi^-, \pi^- \pi^- \pi^+$
	[388]	1542	$K^- \pi^- K^+, K^0 \pi^- \bar{K}^0, K_S^0 \pi^- K_S^0, K_L^0 \pi^- K_L^0, K_S^0 \pi^- K_L^0,$ $K^- \pi^0 K^0, \pi^0 \pi^0 K^-, K^- \pi^- \pi^+, \pi^- \bar{K}^0 \pi^0$
	[389]	1543	$\pi^0 \pi^0 \pi^+, \pi^- \pi^- \pi^+, K^- \pi^- K^+, K^0 \pi^- \bar{K}^0, K^- \pi^0 K^0,$ $\pi^0 \pi^0 K^-, K^- \pi^- \pi^+, \pi^- \bar{K}^0 \pi^0, \pi^- \pi^0 \eta$
	[383]	1544	$\gamma \pi^0 \pi^-$
5	[390]	1551	$\pi^0 \pi^- \pi^- \pi^+, \pi^0 \pi^0 \pi^0 \pi^-$
6	[391]	1561	$\pi^0 \pi^0 \pi^- \pi^- \pi^+, \pi^0 \pi^0 \pi^0 \pi^0 \pi^-, \pi^- \pi^- \pi^- \pi^+ \pi^+$

branch of the decay tree is traversed until all branches of the decay tree have been decayed. In this way the decays of the outgoing particles from the hard process are correlated. As implemented in PYTHIA, this full recursion is not necessary since the implemented τ decays are typically provided with stable final-state particles.

The hard-process generation of PYTHIA uses unpolarized matrix elements to generate the phase space of the hard process, and so dedicated $2 \to n$ helicity matrix elements are needed to determine the helicity density matrix after phase-space generation. For τ decays a number of helicity matrix elements are available. Correlated decays from γ, Z, Z$'^0$, γ/Z/Z$'^0$, neutral Higgs bosons, and t-channel $\gamma\gamma \to \ell\ell$ production are provided. Single τ decays from W, W$'$, charged Higgs bosons, and B/D decays are also provided. For all these production mechanisms the relevant parameters that can be configured for the unpolarized production mechanisms are also used in the helicity matrix elements. This includes the axial and vector couplings for the new gauge bosons, as well as the parity of the Higgs bosons. When a particle used in the helicity decay framework is provided from outside of PYTHIA, the `SPINUP` digit is interpreted as the helicity of the particle in the laboratory frame. A number of options can be configured to fine tune the helicity treatment of τ decays in PYTHIA.

8.2.8 Tau decays

While unpolarized simplified models of τ decays are available in PYTHIA, see section 8.2.6, dedicated models which use the helicity-density framework are available. These models are based on those provided in TAUOLA [383], and are available for all decay channels with branching fractions greater than 0.04%, including up to six-body tau decays. The general helicity-density matrix for these decays used in eq. (388) is given by

$$\mathcal{M} \propto \bar{u}_{\nu_\tau} \gamma_\mu (1 - \gamma^5) u_\tau J^\mu, \tag{392}$$

where only the current J^μ needs to be specified. Here, u and $\bar{u}$ are Dirac spinors, γ^μ are the Dirac matrices, and the Weyl basis as adopted in HELAS [384] is used throughout.

Here, a brief description of the available τ decays is provided; more details can be found in [392] with a summary given in table 3. Note that the ordering of the particles matters, and whenever numerical indices are used, 0 is the decaying τ^- while nu_τ has index 1. For two-body decays into a neutrino and pseudoscalar meson, $\mathrm{tau}^- \to \nu_\tau P$, the hadronic current is given by,

$$J^\mu \propto p_2^\mu.\tag{393}$$

The current for the fully leptonic three-body decay $\tau^- \to \nu_\tau \ell^- \bar\nu_\ell$ is

$$J^\mu = \bar u_2 \gamma^\mu (1 - \gamma^5) \nu_3.\tag{394}$$

Three-body decays with hadronic states can proceed via vector and scalar currents,

$$\begin{aligned}
J^\mu \propto {}& \frac{c_v}{\sum_i w_{vi}} \bigg((p_3 - p_2)^\mu \sum_i w_{vi} BW_p(m_2, m_3, s_2, m_{vi}, \Gamma_{vi}) \\
& - s_1 (p_2 + p_3)^\mu \sum_i \frac{w_{vi} BW_p(m_2, m_3, s_2, m_{vi}, \Gamma_{vi})}{m_{vi}^2} \bigg) \\
& + \frac{c_s}{\sum_j w_{sj}} (p_2 + p_3)^\mu \sum_j w_{sj} BW_s(m_2, m_3, s_2, m_{sj}, \Gamma_{sj}),
\end{aligned}$$

where w_s, v_i are complex weights for each vector and scalar current, $c_{s,v}$ are the scalar and vector couplings, and BW_p is a P-wave Breit–Wigner. The final state determines the relevant couplings and weights to use. The general form of the hadronic current for four-body decays is given by,

$$\begin{aligned}
J^\mu \propto {}& \bigg(g^{\mu\nu} - \frac{q^\mu q^\nu}{s_1} \bigg) \bigg((F_3 - F_2) p_2 + (F_1 - F_3) p_3 + (F_2 - F_1) p_4 \bigg)^\mu \\
& + F_4 q^\mu + i F_5 \epsilon^\mu (p_2, p_3, p_4),
\end{aligned}$$

where each F_i is a model specific form factor and ϵ is the permutation operator.

The hadronic current for the decay $\tau^- \to \nu_\tau \gamma \pi^0, \pi^-$ is given by [383]

$$\begin{aligned}
J^\mu \propto {}& F(s_1, \vec m_\rho, \vec\Gamma_\rho, \vec w_\rho) F(0, \vec m_\rho, \vec G_\rho, \vec w_\rho) F(s_4, \vec m_\omega, \vec G_\omega, \vec w_\omega) \\
& \bigg(\varepsilon_2^\mu \big(m_{\pi^-}^2 p_{4\nu} p_2^\nu - p_{3\nu} p_2^\nu (p_{4\nu} p_3^\nu - p_{4\nu} p_2^\nu) \big) \\
& - p_3^\mu \big((p_{3\nu} \varepsilon_2^\nu)(p_{4\nu} p_2^\nu) - (p_{4\nu} \varepsilon_2^\nu)(p_{3\nu} p_2^\nu) \big) \\
& - p_2^\mu \big((p_{3\nu} \varepsilon_2^\nu)(p_{4\nu} p_3^\nu) - (p_{4\nu} \varepsilon_2^\nu)(m_{\pi^-}^2 + p_{3\nu} p_2^\nu) \big) \bigg),
\end{aligned}$$

where F is a sum over the possible vector currents including ρ and ω resonances. The five-body decays depend on sub-currents for each allowed resonance [390, 393],

$$\begin{aligned}
J^\mu_{\pi^0 \pi^0 \pi^0 \pi^-} &\propto J^\mu_{0, a_1 \to \rho\pi} + J^\mu_{0, a_1 \to \sigma\pi}, \\
J^\mu_{\pi^0 \pi^- \pi^- \pi^+} &\propto J^\mu_{-, a_1 \to \rho\pi} + J^\mu_{-, a_1 \to \sigma\pi} + J^\mu_{-, \omega \to \rho\pi},
\end{aligned}$$

and are based in the Novosibirsk model. The six-body decay model [391] can be written as a summation of a and b-type currents,

$$J^\mu \propto \sum J_a^\mu + \sum J_b^\mu,\tag{395}$$

where each term is one of the possible final state permutations. The a-type currents proceed through a $a_1 \to \omega\rho$ resonance structure, while the b-type proceed via a $a_1 \to \sigma a_1 [\to \rho\pi]$ structure.

Part III

Using PYTHIA 8.3

PYTHIA 8.3 provides comprehensive choices for modelling all kinds of physics effects in collision experiments, as can be seen from the bulk of this manual. It is often not necessary to know the details of all components to start using the program to calculate useful quantities, however. The descriptions provided in this section should allow a new user to set up and use PYTHIA for most standard model and new physics processes, using default settings for showers, MPIs, and hadronization that have been tested to work at the LEP and LHC experiments. By extension, it should also be useful in many other contexts. All settings corresponding to particular physics models or to changing the "tunes" (*i.e.* parameter fitting for showers, MPIs, and hadronization) are documented in the HTML online manual, which is also distributed in the `share/Pythia8/htmldoc/` directory of the released source code. One can begin browsing from the `Welcome.html` home page of that directory.

PYTHIA is under constant and active development. Therefore, any specific detail of this article can become obsolete soon after it is released. We therefore urge users seeking specific information to:

- Make sure to read the most recent version of this manuscript, in conjunction with the most recent code version. Some information, which may have been correct when the manuscript was obtained, may be outdated when being read.

- Consult the HTML manual, which always contains specific settings and reasonable defaults for all physics processes, as well as suggestions for analyses. It also contains a detailed change-log documenting updates between code versions.

- Use the examples distributed with the working version of PYTHIA for inspiration. Examples are kept up to date, and should always correspond to the program version downloaded.

Past and present code versions, documentation, some relevant presentations, and more can be found at the PYTHIA website:

<div align="center">

https://www.pythia.org/

</div>

It is continuously kept up to date.

In section 9 we will describe the logic behind using PYTHIA as a library to write a stand-alone analysis, and in section section 10 we describe interfacing to external programs.

9 Using PYTHIA stand-alone

The default way of using PYTHIA, is to use it as a C++ library, and write "`main`" programs performing the desired simulation tasks. This can be done completely stand-alone, as PYTHIA in principle contains everything needed for a complete physics analysis. Several such example `main`'s are shipped with PYTHIA in the `examples/` sub-directory. In the following we will describe and exemplify how such user code can be written, and then go on to give more advanced use cases, covering deeper interactions with the simulation than allowed from an example `main`.

9.1 Installation

The latest version of PYTHIA 8.3 (as well as older versions) can be downloaded from https://www.pythia.org/ as a gzipped tarball `pythia83XX.tgz`. On Unix, Linux, or MacOS systems this can be unzipped with

```
tar -xvfz pythia8307.tgz
```

(On Windows systems, we recommend to install a virtual machine running Linux, *cf. e.g.*, this tutorial.) The simplest installation can be made using the standard commands

```
./configure
make
```

Configuration options (especially for linking against external libraries) can be found by typing

```
./configure --help
```

Details can also be found in the `README` file distributed with PYTHIA 8.3. If an install location is specified with `--prefix`, then `make install` will copy libraries, headers, and shared documentation to that location in the standard Unix/Linux hierarchy. Details of the configuration can be accessed either via the generated `Makefile.inc` file or the `pythia8-config` script in the `bin` directory.

Most users would then change to the `examples/` sub-directory, find a suitable example to use as a template for their analysis, modify the desired parts, and compile and run the examples (say, `main01`) by:

```
make main01
./main01
```

It is, however, also possible to compile and run PYTHIA programs outside the `examples/` directory. Three environment variables could be potentially useful, providing the paths to the compiled libraries, and to the settings and particle properties databases,

```
PYTHIA8PATH = <set to head Pythia directory>
PYTHIA8DATA = $PYTHIA8PATH/share/Pythia8/xmldoc
LD_LIBRARY_PATH = $PYTHIA8PATH/lib:$LD_LIBRARY_PATH
```

9.2 Program setup

The simplest PYTHIA 8.3 user code comprises three main sections — initialization, the event loop, and final statistics. A skeleton of a simple program is as below. Note that the skeleton program should compile but not produce any reasonable output, as no reasonable settings are read in.

```
#include "Pythia8/Pythia.h" // access to Pythia objects.
using namespace Pythia8;    // allow simplified notation.

void main() {

 // --- Initialization ---

  Pythia pythia;      // Define Pythia object.
  Event& event = pythia.event; // quick access to current event.

  // Read in settings
  pythia.readString("..."); //  line by line...
  pythia.readFile("cardfile.cmnd");   // or via file.

  // Define histograms, external links,
  // local variables etc. here. E.g.
  int maxEvents = 1000; // The number of events to run.

  pythia.init();   // Initialize

  // --- The event loop ---

  for(int iEvent = 0; iEvent < maxEvents; iEvent++){

    // Generate next event;
    // Produce the next event, returns true on success.
    if(!pythia.next()) {
      // Any error handling goes here.
    }

    // Analyse event; fill histograms etc.

  } // End event loop.

  // ---  Calculate final statistics ---
  pythia.stat();

  // Print histograms etc.

  return;
}
```

9.3 Settings

The internal PYTHIA 8.3 event generation is divided into three steps:

- Process level, dealing with the hard process.

- Parton level, dealing with showers, MPIs, colour reconnection, and beam remnants.

- Hadron level, dealing with hadronization and further decays of the particles produced.

Naturally, there are specific settings to control each of these levels. Aside from this, there are several classes of settings to address output during initialization and generation of each event. In the following, we give an overview of how these may be used. However, the reader should consult the online manual (also accessible from `share/Pythia8/htmldoc/-Welcome.html` distributed with the release) for a full listing of all available settings and options. Note that all possible setting keys are indexed, and can be searched via the `Search Docs` box in the upper-right corner of the page.

It is possible to run PYTHIA 8.3 entirely with the default settings. The only minimal user input required is the choice of production process. As a default, the incoming beams are both protons with a centre-of-mass energy of 14 TeV with the parton distribution function set to the NNPDF2.3 QCD+QED LO $\alpha_s(M_Z) = 0.130$ one [229]. Furthermore, initial- and final-state radiation is turned on, using the internal PYTHIA simple shower. MPIs and hadronization are both on by default, and all unstable hadrons with $c\tau_0 < 1000$ mm are decayed to stable ones. The default tune is the Monash 2013 one [357], see section 9.9.2.

PYTHIA 8.3 collects settings performing related functions into groups (*e.g.* overarching parton-level settings are named `PartonLevel:*`). Input strings for changing settings have the form

```
settingGroup:nameOfSetting = value
```

For example, decays of all resonances can be turned off by setting

```
ProcessLevel:resonanceDecays = off
```

PYTHIA 8.3 supports four different types of settings:

- `flag` is a boolean `true` or `false`. Acceptable input alternatives include `on/off`, `yes/no`, and `1/0`.

- `mode` is an integer switch enumerating either available options or a wider range of values. Acceptable values are integers.

- `parm` is a real number parameter.

- `word` is a character string. It cannot contain single or double quotation marks, or curly braces, *i.e.* { }.

It is further possible to have a vector of each of these types. If necessary, users can define their own settings that can then be used in their code.

The user can read in settings in one of two ways: either line-by-line with `pythia.readString()` calls inside the user C++ code, or by providing a plain-text file that is read at run time. The latter has the advantage of not requiring a recompilation every time a change is made. It is triggered by

```
pythia.readFile("cardfile.cmnd");
```

inside the code.

All settings have reasonable default values enabled, and can furthermore be defined with maximal and/or minimal values beyond which they cannot be changed. These can be studied in the online manual under the respective parameter. A parameter can be forced outside the allowed bound by using the keyword `force`, for example:

```
PhaseSpace:pTHatMinDiverge force= 0.1
```

will force the parameter `PhaseSpace:pTHatMinDiverge`, which usually has a minimal value of 0.5 GeV, to 0.1 GeV. *The `force` keyword should be used with extreme caution!* The boundaries are there for a reason, and breaking them can make the program unstable or invalidate the physics model.

If nothing else is mentioned explicitly, dimensional parameters have units of GeV for energy, momentum, and mass, and mm for length and time, with the speed of light $c = 1$ implicit. Internal cross sections are book kept in mb, but communication with other programs may require conversion from/to other units.

9.3.1 Beams and PDFs

The incoming beams are set by providing the PDG code of the incoming particles to `Beams:idA` and `Beams:idB` (the default for both is proton *i.e.* 2212). For example, a p$\bar{\text{p}}$ collision can be set by changing the value of `Beams:idB` to

```
Beams:idB = -2212
```

An e$^+$e$^-$ collision can be set by `idA = 11` and `idB = -11`. Currently available beams include protons (2212), neutrons (2112), pions ($\pm 211, 111$), most other light hadrons (but not necessarily all combinations of them), electrons (11), muons (13), photons (22), and several heavy-ion species. The collision energy can then be set by

```
Beams:eCM = 2000.
```

Units of GeV are implicit, as already mentioned. For heavy-ion collisions, this is the energy per nucleon-nucleon collision, as per the usual heavy-ion conventions.

By default, collisions are assumed to be in the CM frame. Other options can be set with `Beams:frameType`. Using option `Beams:frameType = 2` the beam energies can be set separately and *e.g.* a HERA-like beam configuration can be obtained with

```
Beams:frameType = 2
Beams:idA = 2212
Beams:eA = 920.
Beams:idB = -11
Beams:eB = 27.5
```

Furthermore, the beams do not need to be back-to-back but option `Beams:frameType = 3` allows for setting also some transverse momentum for the beams. A particularly useful setting to automatically set beam information when using external LHE files (see section 10.1.1 for details) is

```
Beams:frameType = 4
```

It is also possible to specify a simple Gaussian spread of incoming beam momentum and of the interaction vertex position. These can be set by `Beams:allowMomentumSpread` and `Beams:allowVertexSpread` and their accompanying parameters in the x, y, and z directions for each beam.

The applied proton PDF set can be selected with setting `PDF:pSet` which is also applied for antiprotons and neutrons via isospin symmetry. By default, this sets PDFs to be the same for beam *A* and *B* but it is also possible to set the PDFs for beam *B* separately using option `PDF:pSetB`. The internal PDF sets can be selected by setting an integer value for the above options, *e.g.* the current default is set with

```
PDF:pSet = 13
```

To use LHAPDF grids instead, PYTHIA needs either be linked to the LHAPDF library or one can use the internal implementation for the LHAPDF grid interpolation, see see section 10.1.4. In the first case, the set is defined with a string `LHAPDF6:set/member`, *e.g.*

```
PDF:pSet = LHAPDF6:NNPDF23_lo_as_0130_qed/0
```

which would correspond to the current default above. Also LHAPDF version 5 is supported and enabled with keyword `LHAPDF5:set/member`. The internal interpolation for the LHAPDF 6 format is enabled with `LHAGrid1:filename` and with this, the default PDF can be obtained with

```
PDF:pSet = LHAGrid1:NNPDF23_lo_as_0130_qed_0000.dat
```

The grid file should be located in the folder `share/Pythia8/xmldoc` or an absolute path should be provided. These settings change the PDF used throughout the program, including hard-process generation, MPIs, and ISR. To keep the underlying event description intact, one can also change the PDFs only for the hard processes by setting `PDF:useHard = on` and selecting the hard PDFs with `PDF:pHardSet`. All the above options can be used to select the PDFs for hard processes and one can also include nuclear modifications for these with `PDF:useHardNPDFA = on` or `PDF:useHardNPDFB = on`. Similarly, one can select PDFs for other beam types including pions, pomerons, photons, and leptons, see the [online manual](#) and section 3.12 for further details.

9.3.2 Process selection

The minimal initialization information required by PYTHIA 8.3 to generate events is which process(es) are to be run. This is done by turning on the relevant flags. For example, to generate a $gg \rightarrow q\bar{q}$ hard process, set

```
HardQCD:gg2qqbar = on
```

A full list of internally defined processes is available in appendix A.

It is possible to turn on more than one process at a time. PYTHIA 8.3 will then generate events for each process in proportion to their cross sections. Some extra switches are also available for processes that are often grouped together, *e.g.*

```
HardQCD:all = on
```

will turn on all QCD $2 \rightarrow 2$ quark/gluon production processes. Since these processes are divergent in the $p_\perp \rightarrow 0$ limit, it is necessary to introduce a lower transverse-momentum cutoff `PhaseSpace:pTHatMin`. Note that such a parton-level cut does not directly translate into a cut on jet properties, since intermediate parton showers, MPIs, hadronization effects, and jet finders will distort the original simple process. Further details are available in sections 3.1 and 3.13.

Several choices of renormalization and factorization scale are available. For $2 \rightarrow 2$ processes,
these can be set via `SigmaProcess:renormScale2` and `SigmaProcess:factorScale2`, respectively. The default for the renormalization scale is the geometric mean of the squared

transverse masses of the two outgoing particles. The default of factorization scale is set at the smaller of the two squared transverse masses. The possible options are listed in section 3.10.

9.3.3 Soft processes

The bulk of the total cross section in high-energy hadronic collisions is not associated with a visible hard process. A reasonably complete and consistent description of these relevant processes is instead obtained with

```
SoftQCD:all = on
```

This includes elastic, single- and double-diffractive, and non-diffractive processes, which alternatively could be switched on individually. The inelastic processes, *i.e.* the diffractive and non-diffractive ones, include a modelling of MPIs, which does include a tail of high-$p_\perp$ processes. Thus `HardQCD:all` becomes a subset of the `SoftQCD:all` total cross section, and one should not mix `SoftQCD` and `HardQCD` processes. Colour screening ensures that the hard processes here are damped appropriately at low $p_\perp$ values, as described in section 6.2.

At very low collision energies the perturbative processes are gradually phased out and only truly soft processes remain. This occurs *e.g.* in hadronic rescattering, or in the final stages of the evolution of a cosmic-ray cascade in the atmosphere. To simulate low-energy collisions directly, use

```
LowEnergyQCD:all = on
```

or related `LowEnergyQCD:*` flags to turn on only a subset of the available processes. These are assumed to be accurate below 10 GeV. It is possible to simultaneously turn on both `LowEnergyQCD:*` and `SoftQCD:*` processes, in which case a mix of the two is used at intermediate energies.

A number of other processes are available, including numerous non-QCD processes which may not be applicable for proton beams. See appendix A.1 for a complete list of included standard-model processes, and appendix A.2 for a list of BSM processes.

9.3.4 Parton- and hadron-level settings

The primary switches for parton showers are

```
PartonLevel:ISR = on|off
PartonLevel:FSR = on|off
```

PYTHIA 8.3 has two other showers available, aside from the "simple showers". The choice of shower model can be performed with

```
PartonShowers:model = 1|2|3
```

where the default (1) corresponds to the "old" simple shower, (2) corresponds to VINCIA, and (3) to the DIRE shower.
Finally, the primary switch for hadronization is

```
HadronLevel:all = on|off
```

9.3.5 Particle data

All known information regarding particles (mass, charge, decay width, branching fractions, *etc.*) is stored within the `ParticleData` class. Each particle has the following basic properties:

- `id` holds the PDG identity number of the particle.

- `name` is a string containing the name of the particle. Particle and antiparticle names are stored separately, with `void` returned when no antiparticle exists.

- `spinType` in the form of an integer equal to $(2s + 1)$.

- `chargeType` is three times the electric charge.

- `colType` is the colour representation (`0`: uncoloured, `(-1)1`: (anti-) triplet, `2`: octet, `(-3)3`: (anti-) sextet).

- `m0` is the nominal mass in GeV.

- `mWidth` is the Breit–Wigner width in GeV.

- `mMin, mMax` are the limits for mass generated by the Breit–Wigner.

- `tau0` is the proper lifetime in mm.

- `mayDecay` sets whether the particle is allowed to decay.

- `isVisible` sets whether the particle is to be considered visible by the detector.

Other than these, there are a few special properties related to external decays which can be found in the [online manual](). Any property of a particle can be changed by setting:

```
NN:Property = value
```

where `NN` is the PDG ID of the particle.

The next critical piece of information for a particle is its decay table. The decay table is comprised of decay modes (or decay channels), each of which has the following properties:

- `onMode` sets whether this decay channel is open where 0 is off, 1 is on, 2 on for the particle but not for the antiparticle, and 3 is on for the antiparticle but not for particle.

- `bRatio` sets the branching ratio for the channel.

- `meMode` sets how this decay is handled, in particular whether internal matrix element reweighting is available to account for mass or angular correlations. The default is 0 and corresponds to flat phase space. See table 2 for available matrix-element modes for particles and section 8.1.2 for available matrix-element modes for resonances.

- `multiplicity` sets the number of daughters, the maximum allowed is eight.

- `product(i)` is an array that holds the PDG IDs of the daughter particles; empty slots are set to zero.

Several shortcuts exist to quickly set up the decay table of a particle. For example, deleting the existing decay table to start anew can be done by using the following.

```
NN:oneChannel = onMode bRatio meMode product1 product2 ...
NN:addChannel = onMode bRatio meMode product1 product2 ...
```

Branching fractions are automatically rescaled such that the sum is one. Certain modes can be turned on or off based on the identity of the products by using the following shortcuts

```
NN:offIfAny = product1 product2 ...
NN:onIfAny = product1 product2 ...
NN:onPosIfAny = product1 product2 ...
NN:onNegIfAny = product1 product2 ...
```

This turns on the mode if any of the products in the list matches one in the `product(i)` array. Note that `onPos...` (`onNeg...`) above means that setting only applies to the decays of the (anti)particle. Further shortcuts (to select based on matching all products *etc.*) can be found in the [online manual](#).

Adding new particles can be done either by directly calling `ParticleData::addParticle` from the program or by using the SLHA interface with a `QNUMBERS` block (*cf.* section 10.1.2).

9.4 Analysis of generated event

A generated "event" is essentially a list of particles — initial, final, or intermediate — that are generated sequentially based on probabilistic calculations. A user will mostly be interested in studying kinematic variables constructed from the momenta of initial or final-state particles. The following three classes will be useful in constructing such variables. The full list of available classes and methods for analysing an event is available in the [online manual](#).

9.4.1 The `Vec4` class

The `Vec4` class is designed to hold the four-momentum (or indeed any other four-vector quantity that may be needed) of the particles in the collision event. Some useful methods are

- `px(), py(), pz(), e()` to access the individual components.

- `mCalc()` for calculated mass $\sqrt{E^2 - p_x^2 - p_y^2 - p_z^2}$.

- `pT()` and `pAbs()` for the transverse momentum and the absolute value of the three-momentum, respectively.

- `theta(), eta(), phi()` for the polar and azimuthal angles, rapidity and pseudorapidity, respectively.

- `rot(double theta, double phi)` to rotate the three-momentum.

- `bst(const Vec4& p)` and `bstback(const Vec4& p)` to boost the current vector by $\vec{\beta} = \pm\frac{\vec{p}}{E}$.

9.4.2 The `Particle` class

The `Particle` class forms the fundamental particle unit, multiples of which are assembled in the form of an "event". Each `Particle` has the following properties:

- `id()` for the PDG code.

- `status()` for the status of the current particle (initial, final, stable, or intermediate *etc.*, see the [online manual](#) for the full status codes). For most users, the only relevant check is, if the number is greater than zero, which denotes a stable, final-state particle. This can also be determined directly by asking `isFinal()`.

- p() returns a four-vector whereas px(), py(), pz(), e() can be used directly to access components.

- mother1(), mother2() refer to the indices of the first and last mother, with several special rules. motherList() returns a vector of all the mother indices, circumventing the need to know these rules.

- daughter1(), daughter2() refer to the indices of the first and the last daughter, with several special rules (all contiguous indices in between are daughters of said particle). daughterList() returns a vector of all the daughter indices, circumventing the need to know these rules

- vProd() for the production four-vertex.

- tau() is the lab-frame lifetime in mm/c.

9.4.3 The Event class

Finally, we come to the main result of the program which is held in a class called Event, representing a collision event. It contains a dynamic array (vector) of particles along with helper methods that are useful to extract information from the array. A single Pythia instance contains two Events, called process and event. The first of these, process, contains only the hard process whereas the second event contains the full history of the collision event. The user usually does not need to manually add or remove particles from either of these arrays. The individual particles can be accessed simply by using their index in the event (*e.g.* pythia.event[i]). All methods corresponding to the particle then can be accessed *e.g.* pythia.event[i].phi() accesses the azimuthal angle φ. Some useful methods beyond those given for individual particles are:

- detaAbs(int i1, int i2) and dphiAbs(int i1, int i2) to obtain $\Delta\eta$ and $\Delta\varphi$ between two particles in the event.

- REtaPhi(int i1, int i2) for the R distance between two particles.

with more given in the online manual.
Several useful functions that take an Event as in input are available to the user to construct important quantities, *e.g.* SlowJet is a sequential clustering algorithm that can be used to form jets from final-state particles whereas Sphericity and Thrust classes calculate these inclusive variable.

9.5 Program output

The most basic level of output that can be requested is a listing of the full event (when inside the event loop), which is done simply by

```
pythia.event.list()
```

A printout of the statistics, *i.e.* number of tried and accepted events, as well as the number of events produced for each process and the resulting cross section can be obtained by using

```
pythia.stat()
```

For hard processes with *e.g.* $p_\perp$ cuts, the cross section must be calculated by Monte-Carlo integration. This is done automatically as events are generated. After generation, the total cross section and its statistical error can be accessed by calling respectively:

```
pythia.info.sigmaGen()
pythia.info.sigmaErr()
```

Pythia also provides rudimentary built-in histogramming via the `Hist` class. The main methods of interest are

- `Hist(string title, int numberOfBins, double xMin, double xMax, bool logX)` the constructor, defining a histogram.

- `fill(double value, double weight = 1.0)` to fill the histogram with an optional weight.

- `table(string fileName, bool printOverUnder = false, bool xMidBin = true)` to output the histogram as a table.

Piping the histogram object directly to the standard output (`std::cout << myhist;`) will also give a rudimentary ASCII output of the histogram. There are methods that will also generate PYTHON PYPLOT code for cleaner graphical representations.

9.5.1 Messages, warnings, and errors

PYTHIA 8.3 provides four basic levels of diagnostic output that are available in the `Info` class. All such generated output is provided in a summary at the end of each run and can be useful as a sanity check or as debug information. The main categories are:

Abort means that something went seriously wrong, either in initialization or generation. In the former case, event generation cannot begin. In the latter case, the event is flawed, and should be skipped. In either case the respective method `Pythia::init()` or `Pythia::next` will return `false`, to allow the user to react. There are occasions where an abort may be deliberate, such as when a file of Les-Houches events is read and the end of the file is reached.

Error typically means that something went wrong during event generation, but the program will backup and try again. In cases where this is not possible, a separate **Abort** will be issued. A typical run can issue several errors, without it being a problem, unless the program aborts. If encountering unusually many errors, it can be a good idea to check if any run parameters are set to unreasonable values, making a calculation unable to converge. The user can set the maximum number of errors to allow before the entire run is aborted via the `Main:timesAllowErrors` parameter.

Warning is less severe. Typically the program will try again with a good chance of success. Usually no action needs to be taken by the user.

Message represents informative outputs that confirm *e.g.* reading of an external file. Verbosity of messages can be set separately for each module that provides this function (*e.g.* `SLHA:verbose` can be set to zero for a silent read.)

9.6 Advanced settings examples

The example use cases given above, are enough for performing simple tasks with PYTHIA 8.3. In most cases, however, when a user wants to apply specialized built-in physics capabilities, the application is more complicated, generally scaling with the complexity of the required tasks. For this purpose, PYTHIA 8.3 ships with a large number of examples (in the `examples/` sub-directory), intended to showcase various applications.

In this section we provide a thorough explanation of two such advanced use cases, to highlight the versatility of the distributed code. Settings for matching and merging are presented in section 9.6.1, while section 9.6.2 discusses options for changing the beam configuration on an event-by-event basis.

9.6.1 Matching and merging settings

PYTHIA 8.3 offers implementations of a large variety of matching and merging schemes. This allows both flexibility, but crucially also cross-checks of the results of combining fixed-order perturbative calculations with the event-generator machinery.

POWHEG matching   allows for the combination of specialized next-to-leading order calculations with PYTHIA 8.3. To facilitate the matching, PYTHIA 8.3 offers vetoed parton showers via so-called `PowhegHooks`. This tool is available for the default showers as well as VINCIA[11] and prevents the over counting of emissions. It can be enabled with the setting:

```
POWHEG:veto = 0|1
```

Since it is not strictly guaranteed that the first shower emission can be considered the hardest emission according to the POWHEG criteria, the number of emissions to be subjected to vetoed showering can be adjusted by:

```
POWHEG:vetoCount = value
```

Furthermore, vetoed showering only needs to be applied to Born-type configurations, which can be tagged by the minimal number of partons in the process:

```
POWHEG:nFinal = value
```

Vetoed showering relies on comparing the hardness of an emission to an allowed maximal hardness. The definition of "hardness" is determined by:

```
POWHEG:pTdef = 0|1|2
```

Values other than 1 are discouraged. The definition of the "maximal hardness" can be adjusted with:

```
POWHEG:pThard = 0|1|2
```

where values other than 0 only serve testing purposes. Finally, the setting:

```
POWHEG:pTemt = 0|1|2
```

---

[11]For VINCIA, `PowhegHooks` should be swapped for `PowhegHooksVincia`. All settings listed here retain their importance with VINCIA. More details can be found in [189, appendix A].

determines for which sets of particles the hardness comparison should be applied, with a value of 0 strongly recommended. A few further, more advanced, settings are listed in the online manual.

MC@NLO matching   employs shower-specific fixed-order calculations, which handle the overlap between shower and fixed-order calculation by explicit subtraction. When interfacing these calculations, it is paramount to guarantee consistency of settings between the fixed-order calculation and the parton shower, for any aspects that might have an impact at the NLO level. No new settings need be introduced in PYTHIA. The relevant settings to produce consistent results depend on the shower and the MC@NLO provider. When using MC@NLO inputs with PYTHIA's simple showers, a minimal set of consistent settings is:

```
SpaceShower:pTmaxMatch = 1
SpaceShower:pTmaxFudge = 1
TimeShower:pTmaxMatch = 1
TimeShower:pTmaxFudge = 1
SpaceShower:MEcorrections = off
TimeShower:MEcorrections = off
TimeShower:globalRecoil = on
TimeShower:weightGluonToQuark = 1
```

Please refer to the online manual for further details.

CKKW-L merging   allows for the combination of several multi-jet tree-level fixed-order calculations with each other and the wider PYTHIA 8.3 environment. For example, calculations of Drell–Yan lepton-pair production at hadron colliders in association with zero, one, two, or more additional jets can be combined. In this context, "additional jets" refers to further QCD partons, as well as W and Z bosons, in the case of simple showers and DIRE.

The inputs for multi-jet merging need to be regularized to avoid soft/collinear configurations. The regularization cut also acts as the criterion to distinguish between fixed-order and parton-shower phase space regions — the so-called merging scale. If the input events are regulated by a $k_\perp$ cut, the following flag can be used to interpret the merging scale in terms of the $k_\perp$ definition:

```
Merging:doKTMerging = on|off
```

For the simple showers, the merging scale definition may also be set in terms of the shower evolution variable $p_\perp$ by setting:

```
Merging:doPTLundMerging = on|off
```

It must be emphasized that this option is naturally not available within VINCIA's merging. The simple shower also offers further built-in merging scale definitions, and the option to supply a pointer to a user-defined `MergingHooks` class to implement new merging scale definitions. The value of the merging scale separating fixed-order and parton-shower regions must be specified via the parameter:

```
Merging:TMS = value
```

The merging further requires the definition of the "process" through the string:

```
Merging:Process = value
```

where `value` should identify the particles of the lowest-multiplicity process partaking in the merging. The process is used under the assumption that each event will contain exactly the specified particles, and potentially further particles that are considered as additional radiation. Looser process definitions are possible through the use of "particle containers", and the "guess" option, see the online manual. Finally, the number of additional jets must be set via:

```
Merging:nJetMax = nJets
```

Other settings are documented in the online manual.

Sector merging The VINCIA antenna shower in PYTHIA 8.3 comes with its own implementation of the CKKW-L merging algorithm, which differs from the one implemented for the simple showers. The main difference is that VINCIA's sector showers are maximally bijective, *i.e.* possess a minimal number of possible histories that lead to any given multi-parton configuration. As such, they are specifically designed for merging with high-multiplicity matrix elements for which the complexity grows factorially with the number of possible shower histories, *cf.* section 5.4.

Sector merging may be enabled by using VINCIA with its sector shower option turned on[12] and switching on merging:

```
PartonShowers:model = 2
Vincia:sectorShower = on
Merging:doMerging = on
```

By default, it is then assumed that the merging scale is defined in terms of VINCIA's evolution variable, *cf.* section 4.2. Other definitions (such as a $k_\perp$ regularization) may be used via the appropriate settings listed for the simple showers above.

While the merging-scale value and the number of additional jets must be set in exactly the same way as listed for the simple showers above, an important difference pertains to the syntax of the process definition. Different to the `Merging:Process` setting in the default merging implementation, the whole string must be encased in curly braces when using VINCIA:

```
Merging:Process = { value }
```

In addition, particles must be specified one at a time and be separated by a white-space character. The initial and final state should be separated by > and exactly two initial-state particles must be specified. It must be emphasized that a process string in the "default" syntax cannot be processed by VINCIA and will lead to an abort.

More advanced settings can be found in the online manual.

UMEPS merging extends CKKW-L tree-level merging, by ensuring that inclusive cross sections for n additional jets are not changed by the inclusion of calculations for $m > n$ extra jets. This is achieved by introducing subtractions that act to remove the effect of higher-multiplicity events from lower-multiplicity inclusive cross sections. As an extension to CKKW-L, UMEPS shares the settings of the former. Beyond these settings, the different stages of UMEPS merging can be invoked by:

---

[12]The sector shower flag is listed only for completeness — sector showers are switched on in VINCIA by default.

```
Merging:doUMEPSTree = on|off
```

which yield CKKW-L-reweighted tree-level results (up to small differences in Sudakov reweighting), and by:

```
Merging:doUMEPSSubt = on|off
```

which produce the necessary subtractions. Depending on the example main program, these two stages may directly be mixed internally, so that only the first setting may be necessary.

NLO merging extends the leading-order merging machinery of PYTHIA 8.3 with (externally generated) next-to-leading order QCD event samples. As an extension of LO machinery, NLO merging inherits many of the settings of LO merging. The result of NLO merging is an inclusive calculation that recovers NLO QCD accuracy for inclusive cross sections with $n \leq n_{\mathrm{NLO}}$ additional partons, and LO (QCD) accuracy for inclusive cross sections with $n_{\mathrm{NLO}} < m \leq n_{\mathrm{LO}}$ jets. The maximal number of jets for which NLO samples are available ($n \leq n_{\mathrm{NLO}}$) has to be set by using:

```
Merging:nJetMaxNLO = value
```

PYTHIA 8.3 offers two NLO merging schemes as part of its core code: NL3 and UNLOPS. Other NLO merging schemes (such as the FxFx scheme) can be embedded with the help of `UserHooks`. NL3 merging is a straight-forward extension of CKKW-L, and mixes augmented CKKW-L-reweighted tree-level events with events from NLO samples. The reweighted LO stage is enabled by using the flag:

```
Merging:doNL3Tree = on|off
```

while the processing of NLO samples requires setting the switch:

```
Merging:doNL3Loop = on|off
```

Typically, NLO input samples contain not only NLO corrections, but tree-level contributions as well. If this is the case, then explicit removal of tree-level contributions from the NLO sample is necessary to avoid double counting. This subtraction is enabled by using the flag:

```
Merging:doNL3Subt = on|off
```

Note that this subtraction is not related to any of the UMEPS subtractions, but rather a necessity due to the structure of available inputs. NL3 only supports the use of the `Merging:-doPTLundMerging` merging scale definition.

UNLOPS merging is an extension of UMEPS that — like UMEPS at leading order — ensures that NLO inclusive cross sections are exactly retained, with the help of unitarity subtractions. Due to this, UNLOPS merging proceeds in four phases. The reweighting of tree-level inputs is enabled by:

```
Merging:doUNLOPSTree = on|off
```

while the processing of NLO samples is produced when using:

```
Merging:doUNLOPSLoop = on|off
```

Both of the former stages should then be accompanied by subtractions to ensure the correctness of the inclusive cross section. Subtractive leading-order samples are produced when using

```
Merging:doUNLOPSSubt = on|off
```

while subtractive NLO events are enabled by:

```
Merging:doUNLOPSSubtNLO = on|off
```

Depending on the example main program, these two tree-level-dependent stages, as well as the two NLO-dependent stages, may directly be mixed internally, so that only the first two settings may be necessary in practice. UNLOPS supports the use of the `Merging:doPTLundMerging` merging-scale definition natively. Other merging-scale definitions (embedded by custom `MergingHooks` classes) can be enabled by setting

```
Merging:unlopsTMSdefinition = value
```

to a non-zero `value`.

9.6.2 Variable energies and beam particles

By default, the beam configuration is initialized at one specified energy. In some cases, however, one may need to generate events across a range of energies. In PYTHIA 8.3, this feature is enabled by setting

```
Beams:allowVariableEnergy = on
```

When this is enabled, the MPI machinery for `SoftQCD` will be initialized at a grid of energies ranging from 10 GeV up to the maximum energy specified by `Beams:eCM`. This way, interpolation can be used to efficiently find the relevant coefficients at each particular energy. (The `LowEnergyQCD` code is intended for energies below 10 GeV where MPIs are irrelevant, and no specific initialization is needed.) Events can then be generated using one of the variant `Pythia::next` methods below, corresponding to the kinematics setup specified by `Beams:frameType`. In other cases, it is also necessary to change the beam particle types on an event-by-event basis. One example of a relevant use case is hadronic cascades in a medium like the Earth's atmosphere or a particle detector. A number of settings must be explicitly switched on to enable this feature:

```
SoftQCD:all = on
LowEnergyQCD:all = on
Beams:allowVariableEnergy = on
Beams:allowIDAswitch = on
```

This will initialize the MPI machinery for a set of some 20 different common hadrons. To switch beam configurations, use one or more of the following variants of the `Pythia::set` methods

```
pythia.setBeamIDs( idA, idB = 0)
pythia.setKinematics( eCM)
pythia.setKinematics( eA, eB)
pythia.setKinematics( pxA, pyA, pzA, pxB, pyB, pzB)
pythia.setKinematics( pAin, pBin)
```

that match the `Beams:frameType` set. After calling these methods, all subsequent events called with `next` will use the updated configuration, unless the `set` call was unsuccessful. The first method preserves the kinematics of the previous event, modulo the change of masses. In this framework, currently only $p/n/\overline{p}/\overline{n}$ is supported for `idB`. An optional parameter `procType` can be passed to `next`, and is used to generate an event of a particular type, such as non-diffractive or single-diffractive on a specified side.

For applications such as cascades in a medium, the decision whether a variable-type interaction should occur or not must be based on the relevant cross section. To this end, the parameterizations outlined in section 6.1.4 and section 6.1.5 can be accessed by using the

```
pythia.getSigmaTotal( idA, idB, eCMAB, mixLoHi = 0)
pythia.getSigmaPartial( idA, idB, eCMAB, procType, mixLoHi = 0)
```

methods. Here, the default `mixLoHi = 0` gives a smooth interpolation between the low-and high-energy descriptions.

Typically, the MPI initialization is the slowest step also in a normal LHC run setup, and with variable particles and energies it will take several minutes. It is possible to speed up the initialization process by saving the MPI parameterizations to disk. This is done using the `MultipartonInteractions:reuseInit` option, which can take the following values:

- 0 (default): MPIs are reinitialized every time.

- 1: MPIs are reinitialized and the parameterization is saved to disk.

- 2: The MPI parameterization is loaded from disk. If the data file does not exist, initialization fails.

- 3: The MPI parameterization is loaded if the file exists, otherwise it is reinitialized and saved to disk.

When using non-zero values, the file name `MultipartonInteractions:initFile` to save/load from must be specified.

9.7 Advanced usage

Often, the user might want to use PYTHIA to simulate physics effects that are not already implemented in the standard release. We therefore provide several ways of extending PYTHIA capabilities. The event generation process can be interrupted at various points (*e.g.* after hard scattering, after first branching in the parton shower, *etc.*) by using "user hooks". These can be used to reweight (or veto) events and change distributions accordingly. Additionally, any extra production process or decay of a new particle can be implemented by inheriting from PYTHIA classes that provide cross section calculation or decay width calculation machinery. We refer to any processes implemented this way as "semi-internal". Finally, when extending capabilities, one may wish to have run-time user-input information in the same way as PYTHIA settings. We therefore provide some placeholder settings as well as methods to add custom settings keys that can be used to accompany any new functionality.

9.7.1 User-defined settings

Should the user require additional settings be provided via a card file, some spares have been made available following the same schema as the normal PYTHIA settings: three each ($N = 1$, 2, 3) of boolean flags via `Main:spareFlagN`; integer modes via `Main:spareModeN`; floating point parameters via `Main:spareParmN`; and strings via `Main:spareWordN`. These can all be set in the card file and interpreted by the user to suit their needs. To add completely new settings keywords, the user can use corresponding methods in the `Settings` class, *e.g.*

```
addFlag(string key, bool default)
```

to add a boolean, *i.e.* a `Flag`, and

```
addParm(string key, double default, bool hasMin, bool hasMax,
double min, double max)
```

to add a double-precision parameter. For further fine-grained control or using the comma-separated vector type settings, we advise the user to refer to the methods documented in the `Settings Scheme` section of the [online manual](#).

9.7.2 User hooks

User hooks are placeholders where the user can interrupt normal PYTHIA program flow to customize behaviour. The behaviour of the hook (*i.e.* the position in program flow where it is designed to interrupt) is set by functions of the type `canVetoX` where `X` indicates one of the pre-defined locations. An accompanying `doVetoX` is then executed during every instance of the `X`. A user defines a hook by creating a class inheriting from `UserHooks`, overriding one or more of the hook methods, and passing an object of that class to a `Pythia` instance:

```
pythia.setUserHooksPtr(make_shared<MyUserHooksClass>());
```

It is also possible to add more than one `UserHooks` object as follows:

```
pythia.addUserHooksPtr(make_shared<AnotherUserHooksClass>());
```

Note, however, that this may give rise to ambiguities if several objects have overridden the same hook function. For the standard `doVetoX` functions, `X` will be vetoed if any of the objects veto, while for some hook methods for which it is not possible to deduce a reasonable combination. In the latter case, PYTHIA will issue a warning during initialization.

PYTHIA provides user hooks for ten cases: interruption while switching between main-generation levels (*e.g.* process to parton); during parton-level evolutions based on p_T or after a step; vetoes for ISR or FSR emissions; to modify cross section or phase space sampling; after resonance decays; to modify shower scales; to allow colour reconnections; to enhance certain rare splittings (*e.g.* $g \rightarrow b\bar{b}$); and finally, to modify hadronization. The details of each of these hooks can be found in the [online manual](#). Using these hooks to modify parton level emissions (*e.g.* to match matrix-element contributions from different orders) is discussed in section 9.8. Here we discuss a simple case of modifying a resonance decay (*e.g.* to select certain kinematics or decay modes). The `Pythia::process` event contains the hard scattering process and decay of resonances produced in the hard scattering. Defining:

```
bool MyHook::canVetoResonanceDecays() {
  // By default returns false.
  // Set to true to run following method
  // after each resonance decay
  return true;
}

bool MyHook::doVetoResonanceDecays(Event& process){
  // Look through the process to check
  // for desired characteristics.
  // Return false to accept the event.
  // Return true to veto the event.
  return false;
}
```

this method can be used *e.g.* with an LHE file (an LHE event is always stored in the `process` before migrating it to the full `event`) that already has decayed resonances that are decayed again by PYTHIA.

9.7.3 Semi-internal processes and resonances

While PYTHIA provides a large number of models, built-in production processes and resonances, it is oftentimes necessary to either modify existing processes or add new ones. The class structure provided by PYTHIA can be easily inherited from to include new processes. Any new particles produced can be implemented as new resonances.

For a resonance, there are three levels of methods used when calculating decay widths in various channels. The first, `initConstants()` is run once per resonance and can be used to set couplings or any other properties that do not depend on kinematics. The second `calcPreFac()` has access to the kinematic configuration (masses of particles and phase-space variables), whereas the third, `calcWidth()` has access to all information and usually contains a case-wise calculation of the decay width in all channels. When there is no flavour-dependent factor in the calculation, `calcPreFac()` can be used to set the internal variable `widNow` (inherited from `ResonanceWidths`) which serves as the calculated width for a given channel. The example program `main22` provides a working example of a new resonance.

```
class NewResonance : public ResonanceWidths {

public:

  // Constructor.
  NewResonance(int idResIn) {initBasic(idResIn);}

private:

  // Locally stored properties and couplings.
  double coupling1, coupling2;
```

```
  // Initialize constants.
  virtual void initConstants();

  // Calculate various common kinematic factors
  // for the current mass.
  virtual void calcPreFac(bool = false);

  // Calculate width for each channel.
  virtual void calcWidth(bool = false);

};
```

Once the resonance is set up, it can be added to the PYTHIA particle data table before initializing using

```
ResonanceWidths* newResonance = new NewResonance(pid);
// Where pid is the integer PDG id.
pythia.setResonancePtr(newResonance);
```

This will automatically call the relevant width calculation functions on initialization and calculate the total width of the particle based on all open channels. The use may have to set up the decay table (*i.e.* a list of open channels) using the commands in section 9.3.5 if the particle is not part of the PDG standard [35].

Once all new particles are set up, production modes can be set up by inheriting from PYTHIA's SigmaProcess class and its derivatives. For $2 \to 1$, use SigmaProcess, for $2 \to 2$ use Sigma2Process, and use Sigma3Process for $2 \to 3$. All relevant kinematic variables are already set up and will be filled event by event based on PYTHIA's phase-space generator. The production process should be set up before calling init() in the main Pythia class. The kind of incoming particles needed for the production are set by the return value of inFlux(); options are qqbar, qqbarSame (same flavour $q\bar{q}$), qg (qg and $\bar{q}g$), ffbar (includes quarks $q\bar{q}$ and leptons $\ell\bar{\ell}$), gg (gluons), and a few more.

```
class Sigma1qqbar2NewResonance : public SigmaProcess {

public:

  // Constructor.
  Sigma1qqbar2NewResonance() {}

  // Initialize process.
  virtual void initProc();

  // Calculate flavour-independent parts of cross section.
  virtual void sigmaKin();

  // Evaluate sigmaHat(sHat).
  // Assumed flavour-independent so simple.
  virtual double sigmaHat() {return sigma;}
```

```
    // Select flavour, colour and anticolour.
    virtual void setIdColAcol();

    // Info on the subprocess.
    virtual string name() const {return "q qbar -> NewResonance";}
    virtual int    code() const {return 10000;}
    virtual string inFlux() const {return "qqbarSame";}
    virtual int    resonanceA() const {return 1000025;}

    // Set internally shared variables (like couplings)
    // as protected or private
    ...
  }
```

Similar to the new resonance-width calculation, there are three progressive methods that can be used to optimize running time. First, `initProc()` is called once per run and can be used to set constants or couplings based on input parameters. Second, `sigmaKin()` can be used to set up kinematic factors for unresolved processes that do not rely on flavour information of the incoming and outgoing states. Third, `sigmaHat()` can be used to calculate the full contribution of the phase-space point, which is returned as a double-precision floating-point number. The return value should be the value of $\frac{d\sigma}{dt}$ for $2 \to 2$ and $|\mathcal{M}|^2$ for $2 \to 3$ processes, respectively. If the user wishes to input the matrix element squared instead of $\frac{d\sigma}{dt}$, then they should also override `bool convertM2()` to return `true` (see `include/Pythia8/SigmaProcess.h` for full class definition and explanatory comments). Finally, an important step before the process is usable is to set the incoming and outgoing colours (*i.e.* colour topology) and flavours where necessary.

```
  void Sigma1qqbar2NewResonance::setIdColAcol() {

    // Flavours simply to be copied from incoming
    // quark ids i.e. id1, id2
    setId( id1, id2, idNew);

    // Colour flow topologies. Swap when antiquarks.
    // Say NewResonance is an octet
    // col1, acol1, col2, acol2, colRes, acolRes
    setColAcol( 1, 0, 0, 2, 1, 2);
    if (id1 < 0) swapColAcol();

  }
```

The new process can now be added to the PYTHIA process array by declaring:

```
    SigmaProcess* sigma1Res = new Sigma1qqbar2NewResonance();
    pythia.setSigmaPtr(sigma1Res);
```

9.7.4 Multithreading

In most cases, events are generated independently of each other. This means that in principle, event generation can easily be split across multiple threads in order to speed up generation.

In practice, each `Pythia` object contains an internal state, and is therefore not thread-safe. A straightforward workaround is to create multiple `Pythia` objects, each initialized with its own random seed using `Random:seed` and `Random:setSeedon`.

Starting from PYTHIA 8.307, the `PythiaParallel` class provides a framework for doing this. This class is intended to provide a lightweight solution to easily enabling parallelism for simple studies. Objects of this class are constructed and initialized similarly to normal `Pythia` objects, but rather than having a `next()` method that generates a single event, it provides the `run` method, which generates a number of events in parallel. The way this works is that the `PythiaParallel` object creates and keeps track of a number of `Pythia` sub-objects. These sub-objects create events in parallel. Whenever an event is generated, the `Pythia` object that generated it is passed to the user so that the resulting event can be analysed. This process then continues until a required number of events has been generated, as specified by the `Main:numberOfEvents` setting. The following snippet gives an example of how to generate events using this class.

```
#include "Pythia8/Pythia.h"
// PythiaParallel.h must be included explicitly.
#include "Pythia8/PythiaParallel.h"
using namespace Pythia8;

void main() {
  // The PythiaParallel object is created
  // and initialized as normal.
  PythiaParallel pythia;
  pythia.readString("SoftQCD:nonDiffractive = on");
  pythia.readString("Main:numberOfEvents = 10000");
  pythia.init();

  // Example: plot charged multiplicity
  Hist nCh("Charged multiplicity", 100, -0.5, 399.5);

  // This defines the callback that will analyse events.
  function<void(Pythia& pythiaNow)> callback =
    [&nCh](Pythia& pythiaNow) {
    int nChNow = 0;
    for (int i = 0; i < pythiaNow.event.size(); ++i)
      if (pythiaNow.event[i].isFinal() &&
        pythiaNow.event[i].isCharged()) nChNow += 1;
    nCh.fill(nChNow);
  };

  // Generate events in parallel, using
  // the specified callback for analysis.
  pythia.run(callback);

  // Print histogram.
  cout << nCh;
}
```

In this example, the `callback` is defined via an anonymous function (also known as a lambda
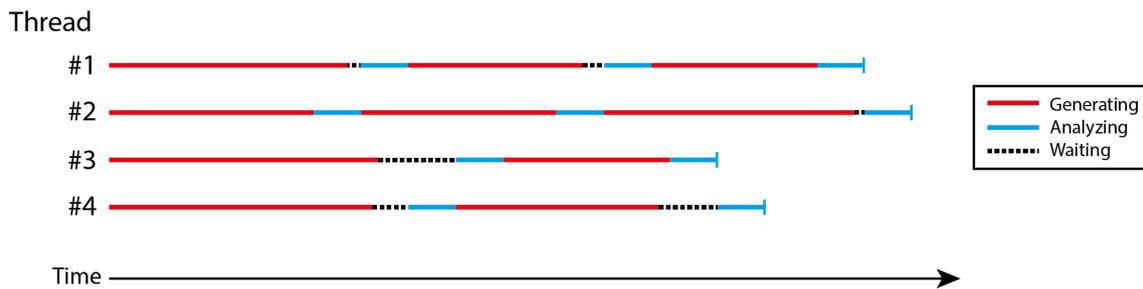
Thread

Figure 19: Illustration of how 10 events are generated and processed in parallel by four threads. The red lines indicate that a thread is generating an event. The blue lines indicate that a thread is analysing the event. Note that two threads are not allowed to analyse at the same time; the dashed black lines indicate that a thread is done generating the event, and is waiting for another thread to finish its analysis.

function). It would also be possible to define it as a named function, *e.g.* with signature

```
void callback(Pythia& pythiaNow)  ...
```

The advantage of using an anonymous function is that it can directly access local variables such as the `nCh` histogram, which is captured by reference according to the `[&nCh]` specifier. It is not necessary to actually save this anonymous function in the `callback` local variable. They can instead be passed directly to `run`, which would make the structure of the code more similar to running with `Pythia::next`. Further examples on using the `PythiaParallel` class are included with the PYTHIA 8.307 distribution.

By default, the framework tries to identify the number of available hardware threads and use the maximum degree of parallelism. Alternatively, the number of threads can be fixed using the `Parallelism:numThreads` setting. This can be useful in order to limit the computational resources spent on generation, and is mandatory on systems where the number of threads cannot be detected.

Although event generation is done in parallel, the analysis is synchronized by default so that only one event is processed at the same time. In the example above, this ensures that it is not possible for two threads to simultaneously write to the `nCh` histogram. An illustration of this is shown in fig. 19. Usually, the analysis is much faster than the actual event generation, and this does not have a significant impact on the run time. However, if the analysis is slow or if the number of threads is very large, the different threads may spend a non-negligible amount of time waiting for other threads to finish the processing. In this case, the run time can be improved by setting `Parallelism:processAsync = on`, which will cause the generated events to be also processed in parallel. It is then up to the user to ensure mutually exclusive access to thread-unsafe resources such as histograms.

It is also possible to use external libraries to perform naive parallelization. Several examples using OPENMP are available in the PYTHIA distribution.

9.8 Event weight handling

By default, PYTHIA produces unweighted events. This means that every event produced by the generator represents an equal share of the total cross section of the chosen processes. However, some settings and functionalities require the use of weighted events. Weighted events no longer represent equal shares of the total cross section, but are augmented with a corrective event weight that needs to be taken into account when filling histograms of physics

observables.

Event weights are useful in different scenarios, which are listed below. One of the key advantages is the use of parameter or setting variations. Rather than regenerating events for different settings and choices of parameters, a vector of corrective event weights can be included in the event generation, reducing the total computation time significantly compared to the generation of separate event samples.

A detailed list of available event weights, related settings and how to access them is available in the online manual. In the following, we provide an overview of process-specific weights and how they can be accessed. Furthermore, we describe the automated-variation weights and the weight container, which collects all weights in a common structure.

9.8.1 Overview of process specific weights

PYTHIA collects the available event weights in a single nominal event weight, which is accessible through the `Info::weight()` function. In a usual setting, this weight is set to 1 and thus uninteresting. Several functionalities and settings lead to a modification of this weight though, in which case the weight must be included when filling histograms.

- Biased phase-space point selection allows for the reduction of statistical fluctuations for specific kinematic configurations. The corrective weight needs to be included to ensure that the overall distributions are not changed.

- If Les-Houches events are used as input, some strategies allow for negative weights, which will be included in the event weight and need to be taken into account. For the strategies 4 and −4, the event weight has units pb, and is converted to mb upon output.

- For heavy-ion collisions, PYTHIA allows a Gaussian sampling of the impact parameter space, leading to weighted events.

- In rare cases, the initial over-estimate of the differential cross section might for specific phase-space points lie below the correct differential cross section. In these cases, a weight above 1 is provided to compensate for this violation.

- Enhanced parton-shower emissions (*cf.* section 4.1.5) need to be corrected for with a weight to ensure that the distributions remain unchanged when improving the statistical relevance of rare emissions.

- Multi-jet merging requires event weights to account for Sudakov factors and the running of coupling parameters. For the leading-order merging schemes CKKW-L and UMEPS, these are by default included in `Info::weight()`. For the next-to-leading-order multi-jet merging schemes NL3 and UNLOPS, the merging weight needs to be included and is available from `Info::mergingWeightNLO()`.

9.8.2 Automatic weight variations

In addition to the nominal weight, additional weights can be provided to take into account variations of settings and parameters. Filling histograms with these respective weights allows for an estimation of the corresponding distributions without rerunning PYTHIA. Additional variation weights are available from the parton shower, multi-jet merging and external LHEF input.

The parton shower currently allows for renormalization-scale variations and non-singular term variations in both initial- and final-state radiation, discussed further in section 4.1.5. Besides, it allows for the variation of PDF members of LHAPDF 6 families. Details on the

usage of these variations can be found in the `online manual`. The physics background is described in [20].

The multi-jet merging schemes CKKW-L, UMEPS, NL3, and UNLOPS also allow for renormalization-scale variations. Furthermore, variations of the UNLOPS merging scheme itself are available. For details, see [185] and the `online manual`. The renormalization-scale variations in the merging are automatically combined with corresponding variations from LHEF input and the parton shower.

With the availability of variation weights from different sources, PYTHIA 8.3 introduces a common structure, the weight container, to make all these weights available to the user. This structure is also used for writing available event weights to HEPMC output. The naming conventions are based on [394, p. 162]. If multi-jet merging is activated, combined weights for renormalization-scale variations in LHEF input, parton shower, and merging are included. Custom weights from LHEF input or the parton shower are presented with a prefix to emphasize that further processing or combining might be necessary. While HEPMC output automatically contains these variation weights, the user can access them directly through the following methods:

```
int Info::numberOfWeights()
string Info::weightNameByIndex(int i)
double Info::weightValueByIndex(int i)
vector<string> Info::weigthNameVector()
vector<double> Info::weightValueVector()
```

The first entry of this weight vector, or correspondingly the weight with index 0 is the nominal weight, including the weights from all the above-mentioned sources.

9.9 Tuning PYTHIA

By default, PYTHIA 8.3 operates using a particular set of run-time parameters that determine the behaviour of the physics models. A set of parameters that is chosen based on a comparison of PYTHIA 8.3 predictions to data is generically named a "tune". As the name suggests, the procedure to obtain a tune is similar to adjusting the pegs on a stringed instrument to achieve a certain sound. However, there is no universal agreement on what constitutes a good tune in contrast to a good sound. The goal of tuning is to find an "optimal" set of physics parameters, p^*, that minimizes the difference between the experimental data and the simulated data from the event generator. In practice, this difference is defined as follows:

$$\chi^2_{\mathrm{MC}}(p, \vec{w}) = \sum_{\mathcal{O} \in \mathcal{S}_{\mathcal{O}}} \sum_{b \in \mathcal{O}} w_{\mathcal{O}_b} \frac{(\mathrm{MC}_b(p) - \mathcal{O}_b)^2}{\Delta \mathrm{MC}_b(p)^2 + \Delta \mathcal{O}_b^2} \,, \tag{396}$$

where $\mathcal{S}_{\mathcal{O}}$ is the set of observables used in the tune, $b \in \mathcal{O}$ denotes the bins in a certain observable $\mathcal{O}$, and $\vec{w}$ is a vector of weights $w_{\mathcal{O}_b}$ for each bin of each observable. The Δs are the uncertainties on the simulated data and the observable. The weights $w_{\mathcal{O}_b} \geq 0$ reflect how much an observable contributes to the tune, *i.e.* if $w_{\mathcal{O}_b} = 0$ for some $\mathcal{O}_b$, then this observable bin will not influence the tuning of p, whereas if $\vec{w}_{\mathcal{O}_b} = 1$ then all data is treated equally. The choice of $\mathcal{S}_{\mathcal{O}}$ and $w_{\mathcal{O}_b}$ determines a unique tune, and these choices are driven by both theoretical and experimental considerations. The variable in eq. (396) is called a "chi-squared", but due to the presence of weights there is no guarantee that it will have the properties of a proper χ^2 distribution.

9.9.1 Comments on the tuning procedure

There are several aspects of this problem that make it non-trivial. One is that the model, which is a mixture of theoretically- and phenomenologically-grounded sub-models, does not describe all data equally well. Related to this is that there is no systematic method for predicting *a priori* where the model will fail — the sub-models are often deeply entwined and are not factorizable, despite our original intention that they should be. If one had a numerical estimate of the uncertainty coming from a certain model (not just its sensitivity to parameter variations, but an estimate of where it fails), then that could be included in ΔMC_b, which usually includes only the uncertainty arising from a finite number of simulation runs. Another issue is how the tune will be applied — will it be used for generic simulations with lowest-order matrix elements or with matched and merged predictions with higher-order matrix elements?

There is more than one way to attack these problems. One, as suggested above, is to set $\vec{w}_{\mathcal{O}} = 0$ for some set of observables or analyses. Such a decision is made at the beginning of tuning when one decides which data is most relevant, *e.g.* Tevatron data or LHC data from a lower-energy run, minimum bias or inclusive jet data, *etc.* Data can also be removed from the tune when it becomes obvious they do not fall within the envelope of the model predictions. In practice, it is sometimes found that even problematic data should be included, with reduced significance, to improve the overall quality of the tune. This can be accomplished by adjusting some of the $\vec{w}_{\mathcal{O}}$ values to emphasize or de-emphasize certain datasets. Obviously, such *a posteriori* manipulation of data is subject to bias and abuse. However, one should remember that tuning is not hypothesis testing — we do not allow for the possibility that the model is ruled out. To illustrate two different, but not exhaustive, approaches, we will describe the MONASH 2013 and the ATLAS A14 tuning exercises.

9.9.2 The default PYTHIA 8.3 tuning: MONASH 2013

The MONASH 2013 tune is currently the default one. It was performed using data from HEP-DATA and the PDG. It was aimed at non-diffractive, high momentum-transfer collisions using the leading-order matrix elements coded in PYTHIA. It started from the hypothesis that hadronization was independent of the environment, and the related physics parameters could be best constrained using e^+e^- data, particularly from LEP (for most observables) and SLD (for b-hadron specific observables). Any modifications to hadronization predictions from the breakup of the proton, for example, would be handled by explicit models that modified the initial conditions, but not the mechanism of hadronization. Once the observables were selected, all with either $\vec{w}_{\mathcal{O}} = 0$ or 1,; several inconsistent values of particle yields were adjusted based on common sense. Physics parameters related to final state radiation, hadronization, and particle decays were selected using eq. (396) as a guide, but without an explicit global minimization. An additional *ad hoc* "theory uncertainty" of 5% was added per bin of each histogram used in the tune to prevent overfitting. These parameters were then frozen as a particular `eetune`. The tuning of the remaining parameters, specific to hadronic collisions, began with a choice of PDF, which is an integral part of any such tune. The central tune of the NNPDF2.3 PDF set was selected, as it was being used in many other theory calculations at the time. In particular, the choice was leading order with a value of α_s closer to that found in the `eetune`. The tuning of initial-state parameters, such as those related to initial-state radiation, beam remnants, and multiparton interactions, proceeded in a similar fashion using LHC data at the highest energy available. Scaling of the multiparton-interaction parameter was obtained by including Tevatron data. Again, at no point was a global optimization of parameters made based on minimization of a χ^2.

9.9.3 The ATLAS A14 tune

The ATLAS A14 tune took a different approach. First, it took the basic MONASH 2013 parameters as a starting point, with the goal of optimizing parameters for LHC physics studies. It relied heavily on the PROFESSOR [395] framework. The observables were selected and weighted to emphasize high-$p_\perp$ radiation and some top-quark observables. It was designed to be used for BSM physics searches, where precision was not the main goal. To that end, it minimized eq. (396) for ten parameters, but in an iterative process to select weights that produced a "reasonable" fit. Using PROFESSOR, it also produced eigentunes that could be used as alternative tunes to study sensitivity to the PYTHIA parameters. However, because of the inclusion of weights, and since the fit residuals do not appear to be χ^2 distributed, an *ad hoc* criterion was used to determine these variations. In the process of selecting data, many observables were included that are obviously highly correlated. However, those correlations were not reported consistently by the experiments. As a result, some of the observables have a hidden weight.

9.9.4 Automatic tuning approaches

Automatic tuning approaches can be helpful to circumvent some of the challenges that manual tuning entails, like subjectivity based on expert knowledge of models, parameters, constraints, and data, and challenges due to a high amount of data sets and parameters to be taken into account. Automatic tuning aims at simplifying the tuning procedure and making it more systematic, which is especially helpful when many parameters are to be tuned.

A brute force grid-based tuning approach is usually prohibited due to the high computational cost of generator runs, especially if many parameters are to be tuned. To circumvent this problem, one can use iterative optimization approaches, which can take time due to the serial running with different parameters, but focus well on relevant regions in parameter space. Alternatively, one can attempt to parameterize the generator response, and to then optimize based on an interpolation. After an initial generator run, which can be trivially parallelized, the actual optimization based on the interpolation is much more straight forward.

As outlined in [396], an iterated Bayesian optimization approach can be employed for event generator tuning. A χ^2 value for different parameter values is obtained, and all information is used to find the next set of parameters iteratively. This approach thus goes beyond local gradient-based optimization, balancing exploration and exploitation.

The PROFESSOR toolkit employs a parameterization approach. After an initial parallelized MC event-generator run, the generator response is parameterized using a polynomial function. A χ^2 optimization is then performed based on this interpolation. This approach allows for several parameters, but becomes prohibitive if the parameter space becomes too large. It is then beneficial to tune in successive steps based on model and data knowledge.

There are multiple efforts in improving the PROFESSOR tuning approach. The AUTOTUNES method [397] employs PROFESSOR, and goes beyond by automatically identifying subsets of correlated parameters that can be optimized successively. The weights are chosen correspondingly to constrain sub-tunes by the most relevant experimental data. The APPRENTICE method [398] goes beyond PROFESSOR by allowing for more general interpolations, a larger variety of optimization methods, and automated setting of weights.

Automatic tuning methods can be very useful, and are helpful when many parameters are to be optimized based on a large amount of experimental data. In combination with expert knowledge about the tuned models and the experimental data, pitfalls can be avoided, like too strong constraints due to single well-measured distributions or unphysical tuning results.

10 Interfacing to external programs

In most realistic use-cases, PYTHIA 8.3 is not used stand-alone, but rather as part of a large software stack capable of providing everything from calculation of Feynman rules from a Lagrangian, to detector simulation and analysis, including interfaces between all those steps. Technically, PYTHIA 8.3 is a C++ library, and only the users' technical proficiency limits the ways the program can be interfaced to other code, thus a manual section describing external interfaces, will by definition be incomplete. For practical purposes, however, PYTHIA 8.3 comes with a number of interfaces pre-written, and several more with an official or unofficial "blessing" by the developers. These are interfaces which should in general work, and where the PYTHIA 8.3 developers will at take some responsibility for helping users when setting up. Those interfaces are described here, along with an explanation of how PYTHIA 8.3 is expected to interact with them. The section is sub-divided in four. In section 10.1 we describe file-based or run-time based interfaces to external providers of *input* to PYTHIA 8.3, be it external matrix elements, PDFs, or random numbers. In section 10.2 we describe the most often used output formats, such as HEPMC events or ROOT "n-tuples". In section 10.3 we describe run-time interfacing with the analysis tools RIVET and FASTJET, and finally in section 10.4 the use of PYTHIA 8.3 through the PYTHON interface and on multicore HPC architectures is discussed.

10.1 Generation tools

Several file-based or run-time interfaces exist. For file-based interfaces, generation steps must be run in a strict sequence. For run-time interfaces, we take a PYTHIA-centric view, *i.e.* that PYTHIA controls the overall event generation (unless stated otherwise).

10.1.1 Les Houches Accord and Les Houches Event File functionality

The Les Houches Accord (LHA) format [304] allows a factorized event generation chain and is one of the most long-lived and successful interface agreements in particle physics. Using the LHA format, complex perturbative calculations can be factored out from the rest of the event generation chain, and performed by specialized tools. The basic idea of LHA is a run-time interface between two generator codes: the "fixed-order generator" stores the collision setup and cross-section information in memory for the "event generator" to read upon initialization (see table 4). At generation time, the individual phase-space points used in the fixed-order generator are stored in memory for the event generator to read and process further, *cf.* table 5 for the format definition. Originally, the in-memory structures were FORTRAN common blocks (called HEPRUP for initialization and HEPEUP for event information). This original format is still used in modern applications, *e.g.* the interfaces to MADGRAPH or POWHEG BOX discussed below. An example of another in-memory structure is discussed in section 10.1.3.

Although desirable from a computing perspective, run-time interfaces require programming language-specific in-memory representations. The Les Houches event file (LHEF) format [305] is a text-file-based update and extension of LHA, such that no run-time interface is necessary, making the results somewhat more portable. Les Houches Event files provide pre-tabulation and storage of phase-space points, thus enabling the reuse of computationally expensive results.

The LHEF format defines XML-like "tag" structures to store information. As such, all relevant information in a LHEF file is enclosed in:

```
<LesHouchesEvents version="v"> ... </LesHouchesEvents>.
```

The `version` can be $v = 1.0$ [305] or $v = 3.0$ [394].

The `HEPRUP` initialization information of the LHA is mirrored by a text block bracketed with `<init>` ... `</init>`, while the `HEPEUP` event information is captured in a text block enclosed in an `<event>` ... `</event>` tag. Auxiliary information pertaining to all events can also be stored in a block bracketed with a `<header>` ... `</header>` tag. The content of each tag may contain further tags, see tables 6 and 7 for a list of all recognized tags.

A basic example `<header>` block is

```
<header>
Some auxiliary information that
...
is not parsed.
</header>
```

Such a header would be compliant with all versions of the format. Additional tags may appear in later versions (v. 3) of the format, as shown in the example below

```
<header>
Some auxiliary information that
...
is not parsed.
<initrwgt>
<weightgroup type="alphasVariation">
<weight id="A"> nominal alphas </weight>
<weight id="B"> decreased alphas </weight>
<weight id="C"> increased alphas </weight>
</weightgroup>
</initrwgt>
</header>
```

In this particular example, PYTHIA 8.3 will be instructed to expect each event to contain a `<rwgt>` block that contains three `<wgt>` entries.

In a slight extension of the accord, PYTHIA 8.3 will also parse the parts of the `<header>` block that are enclosed in `<slha>` .. `</slha>` as if the block contained an SLHA file. See section 10.1.2 for a description of SLHA files.

The `<init>` block is a mandatory part of any LHE file. A basic example will contain the two beam-particle identifiers, their two energies in GeV, two PDF-author-group identifiers, two PDF-set identifiers, and weighting information, followed, in a separate line, by cross section, statistical error, and unit weight information, followed by an integer process label:

```
<init>
2212 2212 0.4E+04 0.4E+04 -1 -1 21100 21100 -4 1
0.50109086E+02 0.89185414E-01 0.50109093E+02 1234
</init>
```

Nowadays, the most common weighting-strategy information (given by −4 in the example) allows for both positive and negative event weights, where the average weight gives the cross section of the generated events. In later versions of the format, the optional `generator` tag may also be included:

```
<init>
2212 2212 0.4E+04 0.4E+04 -1 -1 21100 21100 -4 1
0.50109086E+02 0.89185414E-01 0.50109093E+02 1234
<generator name="SomeGen1" version="1.2.3"> some additional
comments </generator>
<generator name="SomeGen2" version="a.x.3"> some other comments
</generator> </init>
```

This tag mainly serves to convey information, and does not affect the file processing through PYTHIA 8.3.

The initialization information is then complemented with a large list of `<event>` blocks containing the phase-space points. It should be noted that PYTHIA 8.3 supports an arbitrary list of attributes of the `<event>` tag, and further allows "custom" additions enclosed in `<event>` tags:

- The identifier `#pdf` at the start of a line means the line contains information on PDFs. For example, the line
  `#pdf      1    -1   0.11   0.3   100   0.5   0.3`
  will lead to reading/setting the values: ID(particle extracted from beam "A") = 1, ID(particle extracted from beam "B") = −1; momentum fraction of particle extracted from beam A $x_A = 0.11$, momentum fraction of particle extracted from beam B $x_B = 0.3$; factorization scale $\mu_F = 100$ GeV; value of the parton distribution for beam A $f_A(x_A, \mu_F) = 0.5$; and value of the parton distribution for beam B $f_B(x_B, \mu_F) = 0.3$.

- `<event>` tags are allowed to enclose two hard-scattering events, as is *e.g.* needed when interfacing to external double-parton scattering codes.

- In the latter case, the identifier `#scaleShowers` at the start of a line leads to the two subsequent floating-point values being interpreted as parton-shower starting scales for the first and second hard scattering enclosed by `<event>` ... `</event>`, respectively.

- Omitting the incoming particles in the content of the `<event>` tag can be permissible when interfacing with PYTHIA 8.3 to perform only hadronization of resonance-decay systems.

- The event attributes `npLO` and `npNLO` are parsed, and employed when interfacing to MADGRAPH5_AMC@NLO.

A simple event compliant with both versions of the standard will contain information about the number N of particles in the event, the process label, the "scale", and QED and QCD coupling strengths, followed by N lines containing particle information:

```
<event>
4 1234 5.0 300.0 7.861651E-03 1.084400E-01
2 -1 0 0 101 0 0.000E+00 0.000E+00 3.016E+02 3.016E+02 0.000E+00
0.  9.
-2 -1 0 0 0 102 0.000E+00 0.000E+00 -2.964E+02 2.964E+02
0.000E+00 0.  9.
6 1 1 2 101 0 -1.358E+02 -1.671E+02 1.128E+02 3.000E+02 1.756E+02
0.  9.
-6 1 1 2 0 102 1.358E+02 1.671E+02 -1.076E+02 2.980E+02 1.756E+02
0.  9.
</event>
```

For each particle, its identity, status, pair of mothers, pair of colours, momentum, mass, production vertex, and spin are required information. In version 3.0 of the standard, further information may be added to an `<event>`. A more involved example is:

```
<event type="undecayed_born_level_ttbar">
4 1234 5.0 300.0 7.861651E-03 1.084400E-01
2 -1 0 0 101 0 0.000E+00 0.000E+00 3.016E+02 3.016E+02 0.000E+00
0. 9.
-2 -1 0 0 0 102 0.000E+00 0.000E+00 -2.964E+02 2.964E+02
0.000E+00 0. 9.
6 1 1 2 101 0 -1.358E+02 -1.671E+02 1.128E+02 3.000E+02 1.756E+02
0. 9.
-6 1 1 2 0 102 1.358E+02 1.671E+02 -1.076E+02 2.980E+02 1.756E+02
0. 9.
<rwgt>
<wgt id="A"> 5.0 </wgt>
<wgt id="B"> 4.5 </wgt>
<wgt id="C"> 5.5 </wgt>
</rwgt>
<weights> 1.0 0.7 1.3 </weights>
<scales muf="175.0" mur="175.0" mups="300.0" scale_3="1.0"
scale_4="1.0">
content is not parsed
</scales>
</event>
```

This event contains three auxiliary event weights in the "detailed format", as well as three additional event weights in the "compressed format". These different ways to transmit event weights do typically not appear together. The "detailed format" has become much more widely used. The example above further contains auxiliary scale information through the `scales` tag. This feature can be used to *e.g.* transfer multiple shower starting scales to PYTHIA 8.3. Starting scales for individual particles in the event can be set by including a `scales` attribute ending with `_iPos`, where `iPos` is the position of the particle (in the `<event>`) in question. This functionality is used for MLM jet matching with MADGRAPH, and for MC@NLO Δ matching using MADGRAPH5_AMC@NLO. At present, PYTHIA 8.3 does not support the use of sets of events enclosed in `<eventgroup>`. Such events sets were originally proposed in [399] to collect events that require correlated post-processing. Since the latter is not possible in PYTHIA 8.3, event files containing `<eventgroup>` tags will be treated as if the `<eventgroup>` tag was not present.

Finally, note that PYTHIA 8.3 will perform momentum-conservation checks on each input `<event>`. If inconsistencies (*e.g.* due to rounding errors) are found, then actions will be taken to repair the event. This entails enforcing the correct value of particle rest masses, and ensuring that the incoming momentum matches the outgoing momentum.

Table 4: The information defining the LHA initialization interface (in the `HEPRUP` common block). The suffix `UP` can be read as "user process". At most 100 user processes are allowed. See [304] for details.

block name	description
`IDBMUP(2)`	pair of two integer values defining the PDG IDs of the colliding beams
`EBMUP(2)`	pair of two floating-point values listing the energies of the two colliding beams in GeV
`PDFGUP(2)`	pair of two integer values defining the author group of the PDF fit used as the PDF for the colliding beams
`PDFSUP(2)`	pair of two integer values defining the PDF set used to extract particles from the colliding hadron beams
`IDWTUP`	signed integer value determining how the event weights should be interpreted
`NPRUP`	integer value defining the number of different user processes
`XSECUP(NPRUP)`	list of `NPRUP` double values giving the cross sections (in units of pb) of the individual user processes
`XERRUP(NPRUP)`	list of `NPRUP` double values giving the statistical errors associated with the individual user processes
`XMAXUP(NPRUP)`	list of `NPRUP` double values giving the maximum weight encountered in generating the cross section of the user process
`LPRUP(NPRUP)`	list of `NPRUP` integer identifiers for the user processes; the identifiers will also feature in the in-memory representation of the phase-space point

Table 5: The information defining the LHA event information (in the `HEPEUP` common block). At most 500 particles are allowed. See [304] for details.

block name	description
NUP	number of particle entries in the event
IDPRUP	identifier of the user process for this event
XWGTUP	event weight
SCALUP	scale of the event in GeV
AQEDUP	value of the QED coupling for this event
AQCDUP	value of the QCD coupling for this event
IDUP(NUP)	list of NUP integer values defining the PDG IDs of the individual particles
ISTUP(NUP)	list of NUP integer values defining the status (initial state, final state, or resonance) of the individual particles
MOTHUP(2,NUP)	pair of two lists of NUP integer values defining the mothers of the particles
ICOLUP(2,NUP)	pair of two lists of NUP integer values defining the $N_c \to \infty$ colour (anticolour) flow indices of the particles
PUP(5,NUP)	five lists of NUP double values giving the lab-frame momentum of the particle (P_x, P_y, P_z, E, M) in GeV
VTIMUP(NUP)	list of NUP double values giving the invariant lifetime $c\tau$ (distance from production to decay) in mm
SPINUP(NUP)	cosine of the angle between the spin vector of the particle and the three-momentum of the decaying particle, specified in the lab frame

Table 6: Allowed tags in the `<header>` and `<init>` blocks of a Les-Houches event file.

tag name	description
`<header>`	the tag starting the header block, a completely empty header block is allowed
`<initrwgt>`	optional tag detailing the auxiliary events in the "detailed LHEF v3.0 format"; the following two tags have to be enclosed in this tag
`<weightgroup>`	optional tag defining a group of event weights in the "detailed LHEF v3.0 format"; this group will contain several instances of the following tag
`<weight id="`*name*`">`	optional tag defining a particular auxiliary event weight; PYTHIA 8.3 expects each event to contain a `<wgt>` (see table 7) with id=*name* for a `<weight>` with id=*name*
`<init>`	the tag starting the cross section information and initialization block
`<generator>`	optional tag to transfer information about the generator and generator version used to produce the event sample

Table 7: Allowed tags in the `<event>` block of a Les-Houches event file.

tag name	description
`<event>`	the tag starting the event block; an arbitrary number of attributes is allowed
`<rwgt>`	optional tag enclosing a set of event weights in the "detailed LHEF v3.0 format", see next tag
`<wgt id="`*name*`">`	optional tag transmitting the floating-point value of a unique auxiliary event weight as content; the id=*name* should mirror one of the `<weight>` tags of the `<initrwgt>` block (see table 6)
`<weights>`	optional tag containing an array of floating-point values for a set of auxiliary event weights in the "compressed LHEF v3.0 format"
`<scales`	optional tag allowing additional scale information stored as attributes of the tag

10.1.2 SLHA

The SUSY Les Houches accord format! [400, 401] was designed as a plain-text interface between supersymmetric spectrum generators, decay packages, and event generators. However, it has since been generalized to contain information for any new physics model, *cf. e.g.* [402].

The current SUSY implementation in PYTHIA is fully general with support for flavour- and R-parity violation. The physical mass basis for each class of new particles (squarks, sleptons, charginos, and neutralinos, as well as Higgses) is ordered by mass alone. We refer the reader to the original SLHA2 documentation [401] for the full list of supersymmetric parameters supported by SLHA2. Here we give a summary of how new parameters can be passed to PYTHIA, and the modifications made to extend SLHA2 support to be able to read up to 3-dimensional matrix input.

An SLHA file contains a number of pre-formatted "blocks". The three main blocks most often used for passing information about new particles are QNUMBERS, MASS, and DECAY. As an example, we show here how a new spin-1 particle in a colour-octet representation ("heavy gluon") and a new fermion ("heavy quark") can be defined in SLHA [402]. All characters following a # symbol are ignored as a comment, except the first two words after the particle ID code are assumed to be the name of the particle and, optionally, its antiparticle.

```
BLOCK QNUMBERS 9000021 # HeavyGluon
1 0 # 3 times electric charge
2 3 # number of spin states (2S+1)
3 8 # colour rep (1:singlet, 3:triplet, 8:octet, 6:sextet)
4 0 # Particle/Antiparticle distinction (0=own anti)

BLOCK QNUMBERS 9000006 # HeavyQuark HeavyQuarkbar
1 0 # 3 times electric charge
2 2 # number of spin states (2S+1)
3 3 # colour rep (1:singlet, 3:triplet, 8:octet, 6:sextet)
4 1 # Particle/Antiparticle distinction (0=own anti)
```

Note that many of the particle ID codes below 3 million, and several above it, are already in use in PYTHIA (*e.g.* for hadrons, SM particles, and the MSSM particle spectrum). To avoid conflicts, it is strongly advised to only use codes above 3 million for new BSM particles, and to

check in the particle data table that the codes are not already in use. See also the PDG list of standard particle ID codes [35, sec. 45]. Finally, note that PYTHIA is only able to handle colour singlets, triplets, octets, and sextets.

The mass block [400] contains the mass of the physical particles and is simply a list containing the particle ID code and its mass.

```
BLOCK MASS
9000021 1000.  # HeavyGluon
9000006 450.   # HeavyQuark
```

Note that some matrix-element generators export their complete list of particle masses in this block, including also those of SM particles, which may not agree with PYTHIA's internal values. This can wreak havoc in unintended places, *e.g.* by overwriting PYTHIA's constituent-quark masses by far smaller current-quark masses. Therefore, for particles with ID codes less than one million, PYTHIA normally ignores SLHA input for any particle whose default mass in PYTHIA is smaller than `SLHA:minMassSM = 100` GeV. This allows SLHA input to modify top and Higgs-boson properties, but not those of Z, W, and lighter particles.

Separate `DECAY` blocks [400] can be used to specify decay tables for both new and existing particles. (See further sections 2.3.3 and 3.11 for more on PYTHIA's modelling of resonance production and decays.) The sum of all branching fractions is normalized to one when read in. If a certain decay channel is needed for determining the total width, but is not desired to be generated in the context of a given run, this can be done by setting the branching fraction negative. Each line containing a branching ratio should also contain the number of daughter particles, followed by the ID codes of the daughters. Note that only a single decay table should be provided for each particle type; PYTHIA does not accept separate decay tables for antiparticles. However, if different open decay modes are required for a particle and its antiparticle, this can be accomplished by using the PYTHIA `ParticleData` settings `NN:onIfPos` and `NN:onIfNeg` which are allowed to override the initial SLHA settings if `SLHA:allowUserOverride = true`.

```
# PID Width
DECAY 9000021 0.01
# BR NDA ID1 ID2
0.67 2 9000006 -9000006
0.33 2 6 -6
```

When the SLHA interface is used to modify particle data, the m_{min} and m_{max} limits used in PYTHIA's Breit–Wigner sampling (see section 2.3.3) default to $m_0 \pm \min(5\Gamma_0, m_0/2)$. The m_{min} value is further required to also be above the sum of on-shell masses for the lightest decay channel. The default values can be modified by the user, if so desired.

The default Breit–Wigner treatment for decay tables imported via the SLHA interface is the simple `NN:meMode = 100` one with constant branching fractions, but this can also be changed if desired. The phase-space sampling is isotropic, since the SLHA tables do not convey any differential information. It is up to the user to ensure that the final behaviour is consistent with what is desired and/or to apply suitable post-facto reweightings. Plotting the generator-level resonance and decay-product mass distributions and *e.g.* mass differences, effective branching fractions, *etc.*, may be of assistance to validate the program's behaviour for a given application.

Note, finally, that the default in PYTHIA is to ignore SLHA input for all SM particles except top quarks and Higgs bosons; this protects PYTHIA's more sophisticated modelling of *e.g.* Z

and W decays (as well as its definitions of quarks, hadrons, and leptons), *cf.* section 3.11, from being unintentionally overridden by the simpler SLHA treatment. Similar to the above, this choice can be changed by the user if desired, though care must be taken not to corrupt PYTHIA's hadron or light-quark particle data.

Finally, we describe how user-defined blocks may be accessed via the SLHA class [82]. All unknown, *i.e.* user-defined blocks that can be stored in arrays of up to 3 dimensions are read in via the test SLHA file and saved under the name following the `BLOCK` keyword. Depending on the dimensions of the box, one of these methods can be used to access relevant information. This functionality can be used with *e.g.* the semi-internal processes described in section 9.7.3 to use SLHA files to read complex parameter information. Using the `slhaPtr` object available to all production processes inheriting from the `SigmaProcess` class, a block with `blockName` can be accessed using one of the following.

```
# Single value
bool slhaPtr->getEntry(string blockName, double& value);

# 1D array
bool slhaPtr->getEntry(string blockName, int index,
                                    double& value);

# 2D array
bool slhaPtr->getEntry(string blockName, int index1,
                          int index2, double& value);

# 3D array
bool slhaPtr->getEntry(string blockName, int index1, int index2,
                        int index3, double& value);
```

10.1.3 LHAHDF5

In addition to plain-text based ASCII LHEF, PYTHIA 8.3 now also supports Les-Houches event input via the HDF5 data format, which some matrix-element generation frameworks, such as SHERPA [403], support as an alternative to LHEF event output.

The HDF5 format is an open-source binary data format, organized like a database within a single file. It allows for heterogeneous data storage, which is more compressed than ASCII files. Being indexed in an efficient way, it enables the possibility of data slicing, *i.e.* the reading of data subsets instead of the entire data at once. The HDF5 format is thus well suited for storing large numbers of LHA phase-space points in a more efficient way than text-based file formats, allowing for massively parallelized simultaneous access to a single event file [404].

The LHAHDF5 reader uses the HighFive header library to interface HDF5. Moreover, the HDF5 library tools must be installed and an MPI compiler, such as that shipped with MPICH, is needed. To use the LHAHDF5 reader with PYTHIA 8.3, an example configuration command is therefore given by:

```
./configure --with-mpich[=path] --with-hdf5[=path]
--with-highfive[=path]
```

As a relatively new event file format, the LHAHDF5 standard is still undergoing active development. PYTHIA 8.3 internally uses a three-digit numbering scheme to distinguish different LHAHDF5 versions, characterized as follows:

`0.1.0` The event file contains an index group, in which the indices of the particles in a single event are stored. The indices refer to the particle group. Weight variations are not supported and event weights are stored as a single floating-point number in the event group.

`0.2.0` The event file does not contain an index group. Weight variations are not supported, and event weights are stored as a single floating-point number in the event group.

`1.0.0` The event file does not contain an index group. Weight variations are supported, and event weights are stored in a (possibly one-dimensional) array in the event group.

Currently, not all event files may have their version number stored. Therefore, the version can be specified in the PYTHIA input file using *e.g.* `LHAHDF5:version = 0.2.0`. If a version number is present in the event file that is used, the user input will be ignored and the one in the event file is used instead.

10.1.4 LHAPDF

The LHAPDF package is the community standard for providing external parton distribution functions to event generators. Two versions of LHAPDF are supported by PYTHIA 8.3, version 5 [405], a legacy FORTRAN version, and version 6 [406], with a more performant modern C++ implementation. The use of LHAPDF 5 is discouraged and will be fully removed in the future, but is currently kept to provide PDFs for resolved photons that are not currently available in LHAPDF 6. Both versions act as interpolators and extrapolators, for x and Q^2 PDF grids provided by fitting groups. The LHAPDF libraries do not perform DGLAP evolution, and are restricted in x and Q^2 to the grids provided by each PDF set.

Support for LHAPDF can be enabled during PYTHIA 8.3 configuration by,

```
./configure --with-lhapdf5[=path] --with-lhapdf6[=path]
```

where the `path` can optionally be provided. If the executable `lhapdf-config` is available, the LHAPDF path will be automatically extracted. Plugin libraries are generated along with the `Pythia` library which are then loaded at run time when LHAPDF sets are requested by the user. With this interface, it is technically possible to simultaneously use both an LHAPDF 5 and LHAPDF 6 PDF, but this is strongly discouraged. For all PDFs, proton or otherwise, LHAPDF sets can be selected via setting the relevant configuration key to the value `LHAPDF5:set/member` or `LHAPDF6:set/member`, where `set` is the name of the PDF set to use and `member` is the numerical member of that set. If `member` is not supplied, the nominal member is assumed. The example `main52` demonstrates this syntax, while the example `main51` shows how PDF classes can be used independently of a main `Pythia` instance.

Every LHAPDF set has a range of validity, given by the minimum and maximum x and Q^2 values of the grids provided. By default, PYTHIA 8.3 freezes these PDF sets at all boundaries for the set, *i.e.* for $x < x_{min}$ the PDF value is fixed at x_{min} and for $Q < Q_{min}$ the PDF value is fixed at Q_{min}. It is possible to enable extrapolation below x_{min} by setting the `PDF:extrapolate` flag. This flag applies universally to all PDF sets, both internal and external. Extrapolation should be enabled with care, as the extrapolation is PDF set and LHAPDF version dependent, and in many cases may return nonsensical results. Note that extrapolation for the remaining boundaries, x_{max}, Q_{min}, and Q_{max}, is never performed. These values are always frozen at the limits of validity.

The standardized `LHAGrid1` format used by LHAPDF 6 allows for PYTHIA 8.3 to use grids from LHAPDF 6 sets without requiring the LHAPDF 6 library. Simple cubic interpolation is

performed in $\ln(x)$ and $\ln(Q^2)$, where all Q^2 sub-grids must have the same x-value structure. When less than four Q^2 sub-grids are available, linear interpolation is used instead. All relevant PDF sets can use this interpolation by setting the relevant PDF configuration key to the value `LHAGrid1:file`, where `file` is the full name of the PDF set file. If `file` begins with /, then an absolute file path is used, otherwise the `file` is assumed to be in the `share/Pythia8/pdfdata` directory.

10.1.5 POWHEG

A large number of processes utilizing the POWHEG method (positive weight hardest emission generator) [150, 151, 155] are available via the POWHEG BOX package [178]. The physics behind the matching and merging of the hard processes generated by this package with the PYTHIA parton shower is detailed in section 5. Here, technical details on how results from POWHEG BOX matrix elements may be technically interfaced with PYTHIA are given.

The POWHEG BOX package uses a common FORTRAN code structure, which is then duplicated with process-specific modifications in individual matrix elements, *e.g.* `dijets` which produces NLO dijet events. These individual matrix elements are then compiled to create executables which when run, take input cards from the user and produce LHEF output, see section 10.1.1 for details on this format. This output file can then be directly read into PYTHIA via the `Beams:LHEF` setting. Direct POWHEG BOX input, without correctly setting up matching, will result in double counting of emissions. A special `UserHooks` class, `PowhegHooks` in `Pythia8Plugins`, provides a common interface for appropriately matching POWHEG BOX output with the PYTHIA parton shower. In `main31` a full example is given, demonstrating how dijet events produced from the `dijets` POWHEG BOX matrix element can be correctly passed through PYTHIA to produce full events.

In some cases, particularly within large experimental frameworks, users may wish to directly access the FORTRAN common blocks of a POWHEG BOX executable, passing the event by memory to PYTHIA, rather than through LHEF output. By default, POWHEG BOX builds only executables. However, it is possible to modify the `Makefile` via the command,

```
sed -i "s/F77= gfortran/F77= gfortran -rdynamic -fPIE -fPIC
-pie/g" Makefile
```

so that the executables can also be used as shared libraries. When modified accordingly, these executables can be linked against PYTHIA interface code to produce libraries that can be loaded directly by PYTHIA at run time. Run-time loading, rather than dynamic linking, is used so that multiple POWHEG BOX processes can be accessed by a single `Pythia` instance, without creating symbol collisions between executables that have common names for global functions and variables.

After appropriately modifying the relevant POWHEG BOX `Makefiles` and compiling executables that can also be used as shared libraries, the PYTHIA interface libraries must be created. This can be configured with PYTHIA via,

```
./configure --with-powheg-bin=path
```

where `path` is the directory containing the POWHEG BOX executables. When building PYTHIA, a plugin library for each POWHEG BOX in the provided directory will automatically be created. These plugin libraries can then be used via the `PowhegProcs` class provided in `Pythia8Plugins` as demonstrated in the example `main33`. The program flow is as follows,

```
Pythia pythia; // Create a Pythia instance.
PowhegProcs hvq(&pythia, "hvq"); // Load the "hvq" plugin
library.
hvq.readString("configure here"); // Configure the "hvq" plugin
hvq.init(); // Initialize the plugins.
pythia.init(); // Initialize Pythia.
```

where the heavy-quark process `hvq` has been loaded and configured. It is also possible to include another process,

```
PowhegProcs dijet(&pythia, "dijet", "dijetrun");
```

where the additional argument is needed to ensure that the integration grids from the first process are not overwritten by the second process.

When using the `PowhegProcs` method for interfacing with POWHEG BOX a `PowhegHooks` instance is automatically created and passed to the main `Pythia` instance. The settings for this matching hook must be set by the user through either the `readString` or `readFile` methods of the `Pythia` instance. In many cases, sensible default values are set, but some settings are process dependent and must be correctly configured by the user, *i.e.* `POWHEG:nFinal`.

10.1.6 MADGRAPH5_AMC@NLO

MADGRAPH5_AMC@NLO [153] is a hard process generator, similar to POWHEG BOX, but rather than relying upon individually implemented processes, it can automatically generate arbitrary processes up to NLO. There are a number of ways through which MADGRAPH5_AMC@NLO can be interfaced with PYTHIA.

1. MADGRAPH and AMC@NLO themselves can interface with PYTHIA and pass generated hard processes through PYTHIA to produce full events, all within the MADGRAPH5_AMC@NLO machinery.

2. LHEF output from MADGRAPH5_AMC@NLO can be passed to PYTHIA 8.3, see section 10.1.1 for details on reading LHEF input.

3. Source code for matrix-element libraries, inheriting from the internal `SigmaProcess` class in PYTHIA, can be generated by MADGRAPH.

4. The MADGRAPH5_AMC@NLO executable can be called from within PYTHIA via the `LHAupMadgraph` class.

5. Matrix-element plugins for the DIRE and VINCIA parton showers can be generated by MADGRAPH, compiled, and then loaded at run time.

The latter three methods are covered in more detail below. In all cases, it is important that appropriate matching and merging, see section 5, is configured to ensure there is no double counting between the generated hard process and the remainder of the event produced by PYTHIA 8.3.

Semi-internal processes can be passed to PYTHIA 8.3 via inheriting from the `SigmaProcess` class. The primary method of this class is `sigmaHat` where the exact definition depends upon the final-state multiplicity of the process. Phase-space generation can be handled by PYTHIA 8.3 for $2 \rightarrow 1$, $2 \rightarrow 2$, and $2 \rightarrow 3$ processes, although the $2 \rightarrow 3$ phase-space sampler is not particularly sophisticated. When necessary, users can provide custom external

phase-space samplers. Consequently, while $2 \to n$ processes can be externally supplied, phase-space generation must also be implemented by the user for $n > 3$. A full example is given in the example `main22` but the general syntax is,

```
SigmaProcess* userSigma = new UserSigma();
pythia.setSigmaPtr(userSigma);
```

where `UserSigma` is a user-defined process inheriting from `SigmaProcess`.

Semi-internal process source code can be generated from within the MADGRAPH PYTHON interface as follows.

```
import model model_name
generate mg5_process_syntax
add process mg5_process_syntax
output pythia8 [path_to_pythia]
```

A directory containing the output for the process is placed in the PYTHIA 8.3 source directory specified by `path_to_pythia` and an example is placed in the `examples` directory.

It is also possible to call MADGRAPH from within PYTHIA 8.3 via the `LHAupMadgraph` class provided in `Pythia8Plugins`.

```
shared_ptr<LHAupMadgraph> madgraph =
    make_shared<LHAupMadgraph>(&pythia, true, "madgraphrun",
exe);
madgraph->readString("generate mg5_process_syntax");
pythia.setLHAupPtr(madgraph);
```

This interface generates the relevant MADGRAPH configuration cards, and then runs the MADGRAPH executable, specified by `exe`, to produce LHEF output that is then read in by PYTHIA 8.3. An attempt is made to automatically set up matching and merging, but this process should always be validated by the user. Random-number sequences are automatically handled, based on the PYTHIA 8.3 random-number generator. Whenever the LHEF input is exhausted, a new call is made to the MADGRAPH executable and a new LHEF output is generated.

Finally, it is possible to use MADGRAPH to generate matrix-element plugins for use in the DIRE and VINCIA parton showers. A number of these plugins are already provided with the PYTHIA 8.3 distribution in the `plugins/mg5mes` directory. To enable this plugin support, configure PYTHIA 8.3 with

```
./configure --with-mg5mes[=path]
```

where the path to the matrix-element plugin source-code directories can optionally be specified. A plugin library for each directory in the path will be built, which can then be loaded at run time. Just as for POWHEG BOX, run-time loading of the matrix elements allows for multiple plugins to be used with the same instance of `Pythia`. For DIRE and VINCIA, the plugin library to be used can be specified with the settings `Dire:MEplugin` and `Vincia:MEplugin` respectively.

New matrix-element plugin libraries can be generated by using the `generate` command in the `plugins/mg5mes` directory. In its simplest form, the user just needs to specify the process,

```
./generate --process="mg5_process_syntax"
```

but may also specify a model to use, as well as the output directory. Advanced usage is also possible where a custom MADGRAPH card is passed by the user, or the interactive mode of MADGRAPH is enabled. Note that this feature requires the use of DOCKER to download and run a container with a custom version of MADGRAPH.

The most common interface to MADGRAPH5_AMC@NLO is through text files in LHEF format, *cf.* section 10.1.1. For easy interfacing between MADGRAPH5_AMC@NLO and PYTHIA 8.3, some custom additions to the file format are employed:

- The event attributes `npLO` and `npNLO` are used to set the number of particles at lowest order for events with leading-order and next-to-leading order cross sections, respectively. For the former, `npLO` amounts to a simple final-state particle count. For the latter, `npNLO` gives the number of final-state particles necessary to define the scattering at Born level. It is assumed that `npLO`$\geq 0 \rightarrow$ `npNLO`< 0 and `npNLO`$\geq 0 \rightarrow$ `npLO`< 0, meaning that these attributes also act to signal if an event is a leading-order or next-to-leading-order contribution.

- Several mechanisms to set the parton shower starting scales for individual particles exist. These rely on attributes of the `<scales>` tag defined in the LHEF 3.0 format.

- For the case of MLM matching, the parton-shower starting scale information is also used to signal whether a particle should not be considered for the MLM jet matching procedure. Particles that have been assigned a starting scale $\mu > 2E_{\mathrm{CM}}$ will be considered exempt from the MLM jet matching criterion.

MADGRAPH5_AMC@NLO further incorporates provisions for automatic NLO+PS matched calculations within the MC@NLO approach. The interface between AMC@NLO and PYTHIA 8.3 typically relies on phase-space points transmitted via LHEF. However, for special matching tasks, it is possible to invoke PYTHIA 8.3 from within AMC@NLO. This is the case for the MC@NLO $-\Delta$ matching prescription. The relevant FORTRAN code, wrapping PYTHIA 8.3 functionality, is shipped within MADGRAPH5_AMC@NLO. PYTHIA 8.3 can be set up for use within MADGRAPH5_AMC@NLO by setting the configuration flag `Merging:-runtimeAMCATNLOInterface`. This then allows MADGRAPH5_AMC@NLO direct access to select parts of PYTHIA 8.3's internal merging machinery, to *e.g.* enable the extraction of Sudakov form factors. A more detailed introduction may only be relevant to experts in MADGRAPH5_AMC@NLO, and may be found in the `online manual`.

10.1.7 HELACONIA

While PYTHIA 8.3 has a complete collection of expandable quarkonia processes, see section 3.3, it is sometimes necessary to generate quarkonia states at higher orders or with additional final-state partons. Previous versions of MADGRAPH were able to produce arbitrary tree-level quarkonia processes via MADONIA [407], but the current version of MADGRAPH no longer has this ability to generate bound heavy-quark resonances. However, the standalone HELACONIA [408] package is able to provide the same functionality of the MADONIA package, and beyond.

The program flow of HELACONIA is very similar to that of MADGRAPH. A PYTHON interface is used to generate source code which is then compiled and run to produce LHEF output. This output can then be provided to PYTHIA 8.3 to produce full events with parton showers, underlying event, and particle decays. The HELACONIA syntax is modelled after the MADGRAPH syntax, and consequently, the interface is similar. Unlike MADGRAPH, HELACONIA is not able to produce semi-internal matrix elements inheriting from the `SigmaProcess` class. Instead, HELACONIA can be interfaced either by directly providing LHEF output to PYTHIA 8.3, or using the `LHAupHelaconia` class provided in `Pythia8Plugins`.

The `LHAupHelaconia` interface is very similar to that of `LHAupMadgraph`,

```
shared_ptr<LHAupHelaconia> helaconia =
  make_shared<LHAupHelaconia>(&pythia, true, "helaconiarun",
exe);
helaconia->readString("generate ho_process_syntax");
pythia.setLHAupPtr(helaconia);
```

where `ho_process_syntax` is the HELACONIA equivalent for the MADGRAPH process syntax. The HELACONIA executable must be available via the string `exe`. Every time a PYTHIA 8.3 event is generated, the plugin checks if an event is available from an LHEF file generated by HELACONIA. If not, it will automatically run another batch of events. Random-number seeds and sampling are consistently handled in the same way as for `LHAupMadgraph`.

10.1.8 EvtGen

For many experimental collaborations, particularly those specializing in B-physics, more detailed hadron-decay models are needed than those provided by default in PYTHIA 8.3. The EVTGEN [409] package specializes in B-hadron decays, including sophisticated models, spin correlations, and the ability to implement new models. To include spin correlations EVTGEN does not just decay a single particle at a time, but instead performs the entire decay tree for each given initial particle. Consequently, decays from EVTGEN cannot be included in PYTHIA 8.3 via the provided `DecayHandler` class, called during the decay stage of the hadron level, but must rather be performed after full event generation. Such an interface for EVTGEN is supplied by the class `EvtGenDecays` provided in `Pythia8Plugins`.

In B-physics, particularly at hadron colliders, one oftentimes wishes to produce a large sample of events where each event contains one or more rare signal decays, *e.g.* $B_s^0 \to \mu^+\mu^-$. The first step, of course, is to generate an event with at least one signal particle candidate, while the second step is to force the signal decay for one of these candidates. The weight for an event containing one candidate with a forced signal decay is simply the branching fraction for the signal decay. However, when multiple candidates are present, the event weight becomes slightly more complex, requiring non-trivial bookkeeping. Consequently, the `EvtGenDecays` class in PYTHIA 8.3 provides a generalized mechanism by which to force signal decays for given particle species, while still providing an appropriate event weight.

Signal particle candidates, c_i, do not all need to be the same particle species. Here, a particle species differentiates not only between particle types, *e.g.* B_s^0 and τ^+, but also between particles and antiparticles, *e.g.* τ^+ and τ^-. Additionally, the signal decay for a candidate, with branching fraction $\mathcal{B}_{sig}(c_i)$, can include multiple channels. Consequently, arbitrarily complex signal decays can be forced. As an example, events can be required to contain one or more of the following decays: $\tau^+ \to \bar{\nu}_\tau \pi^+$, $\tau^+ \to \bar{\nu}_\tau \pi^0 \pi^+$, $B_s^0 \to \mu^+\mu^-$, and $\tau^- \to \tau_\nu \pi^- \pi^- \pi^- \pi^+ \pi^+$. Here, assuming equal production of the three particle species (which is almost certainly not the case), the decay $\tau^+ \to \bar{\nu}_\tau \pi^0 \pi^+$ of the four signal decays will be the most commonly forced decay. Following this notation, the event weighting is performed as follows.

1. An event is generated and all n signal particle candidates, c_i, are found. If there are no candidates, $n = 0$, then an event weight, $\mathcal{W}_{event}$, of 0 is returned.

2. If $n > 0$ then a candidate c_i is randomly chosen with probability

$$P(c_i) = \frac{\mathcal{B}_{sig}(c_i)}{\sum_{j=1}^{m} \left(1 - \mathcal{B}_{sig}(c_j)\right)}, \tag{397}$$

where $\mathcal{B}_{\text{sig}}(c_i)$ is the signal branching fraction for each candidate c_i.

3. A channel is selected for the chosen candidate c_i from one of the signal channels contributing to $\mathcal{B}_{\text{sig}}(c_i)$.

4. Channels for all remaining candidates are selected, using all allowed decay channels, not just the signal channels.

5. The number of candidates with a selected signal channel, m, is determined. The channel selection for the candidates is then kept with probability $1/m$. If the channel selection is rejected, the algorithm returns to step 2 and a new set of channels is selected.

6. All candidates are decayed via their selected channel and

$$\mathcal{W}_{\text{event}} = 1 - \prod_{i=1}^{n} \left(1 - \mathcal{B}_{\text{sig}}(c_i) \right), \tag{398}$$

is calculated as the event weight.

An unweighted sample of events can be obtained by randomly selecting events, each with probability $\mathcal{W}_{\text{event}}/\mathcal{W}_{\text{max}}$. The maximum possible event weight, $\mathcal{W}_{\text{max}}$, can be determined by the maximum weight from a sufficiently large sample of events.

To use EVTGEN in PYTHIA 8.3, configure PYTHIA 8.3 with

```
./configure --with-evtgen[=path]
```

where `path` optionally provides the path to the EVTGEN installation. Note that EVTGEN itself also links against PYTHIA 8.3, so in some cases it might be necessary to reconfigure PYTHIA 8.3 after installation of EVTGEN. A full example using EVTGEN is provided in `main48`. The general syntax is,

```
EvtGenDecays evtgen(&pythia, dec, pdl);
pythia.next();
evtgen->decay();
```

where `dec` and `pdl` provide the paths to the EVTGEN decay and particle data files.

10.1.9 External random-number generators

When including PYTHIA in a larger software framework, using a single random-number generator across all components is oftentimes required to ensure reproducible results. Consequently, an external random-number-generator pointer may be passed for use by a given PYTHIA instance.

```
pythia.setRndmEnginePtr(rng)
```

Here, `rng` is a pointer to an instance of a user-defined random number generator derived from the `RndmEngine` class. The only method that must be implemented by the user is `flat` which should return a uniform distribution between 0 and 1. The example below implements a linear congruential generator with a configurable seed, modulus, multiplier, and increment.

```
class RandomLCG : public RndmEngine {
public:
  long int seed{1}, m{2147483648}, a{1103515245}, c{12345};

  // The only method that needs to be implemented.
  double flat() {
    seed = (a * seed + c) % m;
    return double(seed)/m;
  }
};
```

Typically, the `RndmEngine` class can be used to wrap some other random number generator. An example of this is the `MixMadRndm` class which is a wrapper for an implementation of the MIXMAX algorithm [11].

```
#include "Pythia8Plugins/MixMax.h"
MixMaxRndm rng(0, 0, 0, 123);
pythia.setRndmEnginePtr(&rng);
```

The argument to the generator constructor is four seed values. While this functionality of providing an external random number generator is useful, it should be treated with care. Some pseudo-random-number generators implemented in standard packages are not sufficient for large scale generation, *e.g.* the CLHEP implementation of the RANLUX algorithm. Consequently, when possible, the default random number generator in PYTHIA, based on the RANMAR implementation of the Marsaglia-Zaman algorithm [9], is recommended and sufficient for most physics purposes, see section 2.2.1.

10.2 Output formats

PYTHIA comes with a set of example main programs, and in most of these the analysis of the produced event is performed directly in the code there. It is also possible to output the events to be analyzed by interfacing to external programs and code. For this purpose PYTHIA can communicate its events with different output formats as described in this subsection.

10.2.1 HEPMC versions 2 and 3

The standard format for communicating fully generated events is called HEPMC [410, 411] and defines a set of C++ classes to describe an event and all particles therein. Internally, the particles are connected by vertex objects using pointers.

The latest version of the HEPMC code is not yet adopted by all LHC collaboration and PYTHIA therefore has support for both version 2 (2.06 and later) and version 3. The interface as such is provided at the header file level using `Pythia8Plugins/HepMC2.h` or `Pythia8Plugins/HepMC3.h`, and the PYTHIA code itself does not have any dependencies on these. This means that the PYTHIA (shared) library can be built independently of which version of HEPMC should be used. However, if one wishes to use the example main programs that show how to use HEPMC[13] the configuring of PYTHIA must be done according to

```
./configure --with-hepmc3=/path/to/hepmc/installation
```

---

[13] The example main programs can be found in the online manual under `Getting Started → Examples by Keyword`, search for `Hepmc`.

or

```
./configure --with-hepmc2=/path/to/hepmc/installation
```

Besides the particles, other information will also be transferred to the HEPMC format, such as cross sections, parton density information, and different weights (see section 9.8). Note, however, that not all information in the `Pythia8::Event` is preserved in the HEPMC output. Notably, the status codes for particles in HEPMC are only set to 1 (final state particle), 2 (decayed standard model hadron or τ or μ), 4 (incoming beam), or a number in the range 11–200 (generator dependent status of an intermediate particle, given by the absolute value of the corresponding PYTHIA status code).

10.2.2   Histograms with the YODA package

Even though the built-in histogram package might suffice for the most basic use cases, such as one-dimensional histograms, most users require more advanced capabilities. Since PYTHIA 8.3 is not a statistics or plotting package, we refer the user to external programs. For slightly more advanced use cases, we recommend interfacing to the YODA[14] histogram package. If installed, PYTHIA 8.3 can be configured with `--with-yoda=/path/to/yoda`, which allows the user to create `Makefile` recipes with access to YODA histograms easily. The YODA package will then be accessible in PYTHIA 8.3 as any other C++ library can be accessed. Questions regarding the YODA histogram package should be addressed to the YODA authors.

10.2.3   Interfacing with ROOT

For more advanced analyses, many users prefer the ROOT [412] package. PYTHIA 8.3 provides several possibilities to interface with ROOT, version 6 or higher. Use cases can roughly be grouped into three categories:

1. Using ROOT as a histogram package inside PYTHIA 8.3.

2. Using PYTHIA 8.3 to generate ROOT events or "n-tuples", which can be post-processed by ROOT.

3. Steering PYTHIA 8.3 from inside a ROOT-based framework.

We will here briefly cover the first two use cases, but refer the user to the ROOT documentation for using the PYTHIA 8.3 interface in ROOT, where it is extensively documented.

The simplest use case is of the first category which, from a technical point of view, is not too different from using any other C++ library along with PYTHIA 8.3. In the example `main91`, it is shown how to declare a ROOT `TApplication` environment and ROOT `TH1F` histograms, to be filled by PYTHIA 8.3, and displayed on screen. The crucial part is the `Makefile` recipe. If PYTHIA 8.3 is configured `--with-root`, convenient variables pointing to the ROOT libraries and the `root-config` script can be used as shown, to compile a `main` program with the necessary linking to ROOT libraries. The generated histograms can then be saved to a `.root` file for later access.

Most users already familiar with ROOT, would rather store event files generated with PYTHIA 8.3 (so-called "n-tuples") on disk, which can then be post-processed with a ROOT-centric analysis framework, often with auxiliary packages, provided by a large experiment. In such cases, examples `main92` and `main93` can be of inspiration. The `main92` example shows how to store full events into a ROOT `TTree`. For most realistic use cases, this is not very

---

[14]See https://yoda.hepforge.org/.

practical, as such files will quickly grow large, containing a significant amount of information which is of little relevance to the user. The `main93` program provides a more streamlined interface. In the header file `main93.h`, two classes `RootTrack` and `RootEvent` are defined. Those classes define what information about each track (including track-level cuts, *e.g.* desired acceptance) as well as each event, should be stored in an output ROOT-file. If PYTHIA 8.3 has been configured with ROOT, the `main93` example can be run with an input `.cmnd` file with the flag `Main:writeRoot = on`, and the desired information will be stored. If changes are made to the header file, `main93` must be recompiled. For both `main92` and `main93`, the compilation recipe in the `Makefile` is the most difficult part to set up, as both require generation of compiled and linked ROOT dictionary libraries with CINT. A user wishing to go beyond simple extensions of the given examples are encouraged to study the existing `Makefile` recipes, as well as the ROOT documentation on `Linkdef.h`. It is kindly requested that queries about ROOT dictionary library generation are directed to the ROOT authors.

10.3 Analysis tools

The tools included in PYTHIA are normally enough for doing simple analyses of the generated events, but for more complicated analyses, or if direct comparison with data is wanted, the user needs to interface to external tools. Here we describe some of these interfaces.

10.3.1 RIVET versions 2 and 3

The RIVET package [413, 414] is probably the most convenient way of comparing event-generator models to experimental data. The program includes a large collection of experimental analyses encoded (usually by the experiments themselves) in C++ classes that read HEPMC input and produce YODA files that can be plotted together with the experimental data points (also provided by the experiments through HEPDATA [415, 416]).

Since RIVET only needs HEPMC input, the only thing needed for PYTHIA is to write the events to a HEPMC file (see section 10.2.1) or a *pipe* (which is recommended to avoid creating unnecessarily large files), and have RIVET take this as input. Assuming a main PYTHIA program, `mymain-hepmc`, that simply writes HEPMC to the standard output, the commands to do this are

```
mkfifo hepmc-pipe
./mymain-hepmc > hepmc-pipe &
rivet -a SomeAnalysis hepmc-pipe
rivet-mkhtml Rivet.yoda
```

where the last command will produce formatted web pages in the `rivet-plots` subdirectory, with the comparisons to data.

In PYTHIA there is also a more direct way of calling RIVET from within a main program provided. This uses the header file `Pythia8Plugin/Pythia8Rivet.h` and provides simple shortcuts as shown in some of the provided example main programs.[15] To enable this, PYTHIA must be configured using

```
./configure --with-rivet=/path/to/rivet/installation
```

together with the corresponding `--with` for the version of HEPMC that RIVET was configured with.

---

[15]The example main programs can be found in the on-line manual under `Getting Started → Examples by Keyword`, search for `Rivet`.

PYTHIA currently supports direct linking with both versions 2 and 3 of RIVET. The additional features in the later version includes the possibility of using different weights (see section 9.8), several heavy-ion specific features [417], and to provide options to the analyses. Support for version 2 of RIVET will likely be dropped in the future.

10.3.2 FASTJET

The `fjcore` code is distributed together with the PYTHIA 8.3 code by permission from the authors. There is also an interface that inputs PYTHIA 8.3 events into the full FASTJET library, for access to a wider set of methods, but then FASTJET must be linked by using

```
./configure --with-fastjet=/path/to/fastjet/installation
--with-fastjetlib=/path/to/
fastjet/library
```

Among PYTHIA 8.3 example main programs, `main71.cc` shows in the case of W plus jet production, how the FASTJET package can be used for analysis of the final state, and `main80.cc` performs CKKW-L merging with a merging scale defined in $k_\perp$, with `main80.cmnd` and LHE files as input. Also, `main72.cc` compares QCD jet finding in `SlowJet` and FASTJET, using the header file `FastJet3.h` present in the directory `Pythia8Plugins` contributed by Gavin Salam [418].

10.4 Computing environments

PYTHIA has been developed as a C++ library to write and compile programs to execute standalone on a generic *nix operating system on a generic computer. However, we address here the rise in popularity of PYTHON as a development language and a powerful tool in machine-learning applications.

10.4.1 PYTHON interface

To meet the growing requirements of a large user base, PYTHIA includes a flexible PYTHON interface to most frequently used classes, and thus allows a user to write a PYTHIA `main` program entirely in PYTHON. This provides the user direct access to the wealth of analysis and visualization tools, available through PYTHON libraries, all at run time. A number of PYTHON examples are provided, each a direct translation of their corresponding C++ counterpart. The interface is generated with BINDER using the PYBIND11 template library. The specific version of BINDER and PYBIND11 needed to generate the interface is provided through a small DOCKER container.

The default interface is a simplified one, with only the core PYTHIA functionality available. This interface is a trade off between usability and remaining light weight. The top level `Pythia` class is available, as well as all relevant `Event`, `ParticleData`, and analysis tool related classes. An important feature of the interface is that it is bi-directional, derived classes in PYTHON can be passed back to PYTHIA. This is useful, for example, to create a `UserHooks` derived class (see section 9.7.2). All user interface classes, typically passed to the main `Pythia` object via pointers in the standard C++ code, are available through the simplified interface.

A full PYTHON interface can also be generated by the user. Only DOCKER is required to enable the generation of a new PYTHON interface to PYTHIA. The following generates the full interface.

```
cd plugins/python
./generate --full
```

It is also possible to generate a user-defined interface which is tailored to a specific use case via the flag `--user=FILE` instead. Here, FILE is a BINDER configuration file specified by the user. Note that whenever changes are made to the PYTHIA C++ headers, the PYTHON interface must be generated again, whether simplified, full, or user defined.

Installation of the PYTHON interface requires the `Python.h` header to be available. The `python-config` script can be used to find the relevant paths when configuring PYTHIA. An example configuration for PYTHIA with PYTHON 3.6 could then be:

```
./configure --with-python-config=python3.6-config
```

This would configure PYTHIA to be built with the default interface, using PYTHON 3.6. After configuring, the compiled PYTHIA module is available in the `lib/` directory under the top level PYTHIA directory. The PYTHON installation must have that directory made available, *e.g.* by setting:

```
export PYTHONPATH=$(PWD)/lib/:$PYTHONPATH
```

from the top level PYTHIA directory. After compiling with `make`, the PYTHON interface should be available. The following example loads the PYTHIA PYTHON module and prints the internal documentation which includes the available classes, as well as some of the not-so-obvious features.

```
>>> import pythia8
>>> help(pythia8)
```

One of the main reasons for the PYTHON interface is the fast development of a standalone `main` program in PYTHON rather than C++, allowing for an environment of external tools, which the user might be more familiar with. As an example of such a program, consider the short PYTHON script below, which will run PYTHIA to produce a `numpy` histogram containing the distribution of charged hadron multiplicity at mid-pseudorapidity in proton collisions at LHC energies.

```python
# Wrapper around numpy histogram to allow fill functionality.
import numpy as np
class HistoFiller(object):
    def __init__(self, bins):
        self.bins = bins
        self.hist, edges = np.histogram([], bins=bins,
                                        weights=[])
        self.widths = []
        for i in range(len(edges)-1):
            self.widths.append(edges[i+1] - edges[i])

    def fill(self, val, w=1.0):
        hist, edges = np.histogram(val, bins=self.bins,
                                   weights=w)
        self.hist+=hist
```

```
    def get(self):
        scale = 1./sum(self.hist)
        return [h/w*scale for h,w in zip(self.hist,self.widths)],
                [np.sqrt(h)*scale for h in self.hist]

# Set up Pythia and declare histogram.
import pythia8
pythia = pythia8.Pythia()
pythia.readString("SoftQCD:all = on")
pythia.init()
mult = HistoFiller([3.*x for x in range(20)])

# Event loop. Find particles and fill histogram.
for iEvent in range(1000000):
    if not pythia.next(): continue
    nCharged = 0
    for p in pythia.event:
        if p.isFinal() and p.isHadron() and p.isCharged():
            nCharged += 1
    mult.fill(nCharged)

# Plot the histogram using the matplotlib library.
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_subplot(111)
y, ye = mult.get()
ax.errorbar(mult.xvals,y,xerr=[w/2. for w in mult.widths],
yerr=ye, drawstyle='steps-mid',fmt='-',color='black')
ax.set_xlabel(r'$dN_{ch}/d\eta$')
ax.set_ylabel(r'$P(dN_{ch}/d\eta)$')
plt.show()
```

Part IV

Summary and Outlook

Our goal in writing this manual was to provide reference material for users and developers of PYTHIA 8.3. We provided some basic content that we considered mandatory, such as defining what *is* an event generator, how does our code structure reflect the physics, and what sorts of numerical methods we use in the program. This is covered in part I. The core of the manual, provided in part II, describes in detail the phenomenon that is simulated and our assumptions and approximations. Parton showers and hadronic or nuclear physics are covered in more detail because these have been the arenas of more recent development. Other topics are covered more liberally in the HEP literature, and we hope to have provided enough outside references. What is somewhat new compared to other PYTHIA manuals is part III, dedicated to the user of PYTHIA 8.3. Our aim was not to give the user an easy way to skip the description of physics, but to facilitate the use of the program in real analyses and investigations. This part of the manual is the most pragmatic, but also the one most susceptible to acronyms, initialisms, and jargon. It is also the most technical in describing our and others' computer code.

This manual is a snapshot of an evolving entity. Within a short period of our concluding statements, new developments will arise that are not covered in this manual. We hope this continues, even as we pass the torch to the next generation of PYTHIA authors and contributors.

James D. Bjorken ("BJ" to his generation) wrote of the "tyranny" of Monte Carlo in a short paragraph of a larger editorial on the future of particle physics in 1992 [419]. He lamented the fact that Monte-Carlo predictions were taken as the truth, event though most of the prediction was a black-box. Had he read this manual in 2022, we hope he would understand that the authors have provided a code that is more democratic, and allows users to liberally test ideas, but within well-defined boundaries. As such, there is no *single* PYTHIA prediction to compare to data.

Acknowledgements

A large number of people should be thanked for their contributions to the PYTHIA 8.3 event generator.

First of all, Bo Andersson and Gösta Gustafson are the originators of the Lund model, and have strongly influenced the development of both early code versions and also recent model additions. Hans-Uno Bengtsson should furthermore be acknowledged as the originator of the PYTHIA program.

Some made contributions dating way back in the programs' history, others more recently. While praise for the contributions should go to the contributors, blame for mistakes made in modifications to the original code, or failure to keep it up-to-date, should rest with the core authors.

Former authors

Former PYTHIA 8 authors, who are no longer active in the field, are: Stefan Ask, Jesper Roy Christiansen, Richard Corke, Nadine Fischer, and Christine O. Rasmussen. The merging of VINCIA into PYTHIA 8.3 brought with it further significant author contributions from Helen Brooks in particular.

Further contributions

The program has received many smaller and larger contributions and bug reports over time, from users to numerous to mention here. They are mentioned in the online update notes as the bug fixes go in, and are all gratefully acknowledged.

In particular, contributions from the following should be mentioned: Baptiste Cabouat for developing and implementing the initial-final dipole approach, Silvia Ferreres-Solé for implementing the space-time hadronic production points in string fragmentation, and Tomas Kasemets for implementation of new proton PDFs.

Code contributions from the following collaborators and users are also gratefully acknowledged: O. Alvestad, S. Baker, B. Bellenot, R. Brun, A. Buckley, M. Cacciari, L. Carloni, S. Carrazza, R. Ciesielski, V. Hirschi, N. Hod, H. Hoeth, J. Huston, M. Kirsanov, A. Larkoski, B. Lloyd, J. Lopez-Villarejo, O. Mattelaer, M. Montull, A. Morsch, A. Naumann, S. Navin, P. Newman, M. Ritzmann, J. Rojo, G. Salam, K. Savvidy, G. Savvidy, A. Singh, G. Soyez, M. Sutton, R. Thorne, and G. Watt. We also thank J. Altmann and T. Garnett for correction of typos in this manuscript.

Finally, vigilant code tests of PYTHIA releases by Mikhail Kirsanov, Dimitri Konstantinov, and Vittorio Zecca are gratefully acknowledged.

Financial support

Appendices

A   Full list of internal processes

A.1   Standard model processes

Table 8: List of internal soft QCD processes, see section 6.1 for details and references.

process	internal name	code
	`SoftQCD:all`	
$AB \to X$	`SoftQCD:nonDiffractive`	101
$AB \to AB$	`SoftQCD:elastic`	102
$AB \to XB$	`SoftQCD:singleDiffractiveXB`	103
$AB \to AX$	`SoftQCD:singleDiffractiveAX`	104
$AB \to X_1 X_2$	`SoftQCD:doubleDiffractive`	105
$AB \to AXB$	`SoftQCD:centralDiffractive`	106
	`SoftQCD:singleDiffractive`	104, 103
	`SoftQCD:inelastic`	101, 103, 104, 105, 106

Table 9: List of internal hard QCD processes, see section 3.1 for details.

process	internal name	code	refs.
	`HardQCD:all`		
$gg \to gg$	`HardQCD:gg2gg`	111	[420–422]
$gg \to q\overline{q}$	`HardQCD:gg2qqbar`	112	[420–422]
$qg \to qg$	`HardQCD:qg2qg`	113	[420–422]
$qq' \to qq'$	`HardQCD:qq2qq`	114	[420–423]
$q\overline{q} \to gg$	`HardQCD:qqbar2gg`	115	[420–422]
$q\overline{q} \to q'\overline{q}'$	`HardQCD:qqbar2qqbarNew`	116	[420–423]
$gg \to c\overline{c}$	`HardQCD:gg2ccbar`	121	[424]
$q\overline{q} \to c\overline{c}$	`HardQCD:qqbar2ccbar`	122	[424]
$gg \to b\overline{b}$	`HardQCD:gg2bbbar`	123	[424]
$q\overline{q} \to b\overline{b}$	`HardQCD:qqbar2bbbar`	124	[424]
$gg \to ggg$	`HardQCD:gg2ggg`	131	[425]
$q\overline{q} \to ggg$	`HardQCD:qqbar2ggg`	132	[425]
$qg \to qgg$	`HardQCD:qg2qgg`	133	[425]
$qq' \to qq'g$	`HardQCD:qq2qqgDiff`	134	[425]
$qq \to qqg$	`HardQCD:qq2qqgSame`	135	[425]
$q\overline{q} \to q'\overline{q}'g$	`HardQCD:qqbar2qqbargDiff`	136	[425]
$q\overline{q} \to q\overline{q}g$	`HardQCD:qqbar2qqbargSame`	137	[425]
$gg \to q\overline{q}g$	`HardQCD:gg2qqbarg`	138	[425]
$qg \to qq'\overline{q}'$	`HardQCD:qg2qqqbarDiff`	139	[425]
$qg \to qq\overline{q}$	`HardQCD:qg2qqqbarSame`	140	[425]

Table 10: List of internal low-energy QCD processes, see section 6.1.5 for details.

process	internal name	code	refs.
	`LowEnergyQCD:all`		
$AB \to X$	`LowEnergyQCD:nonDiffractive`	151	[213]
$AB \to AB$	`LowEnergyQCD:elastic`	152	[213]
$AB \to XB$	`LowEnergyQCD:singleDiffractiveXB`	153	[213]
$AB \to AX$	`LowEnergyQCD:singleDiffractiveAX`	154	[213]
$AB \to X_1 X_2$	`LowEnergyQCD:doubleDiffractive`	155	[213]
$NN \to N^* N$	`LowEnergyQCD:excitation`	157	[213]
$B\bar{B} \to X$	`LowEnergyQCD:annihilation`	158	[213]
$AB \to R$	`LowEnergyQCD:resonant`	159	[213]

Table 11: List of internal photon-collision processes, the second code in parenthesis is used to separate photons from beam A and beam B when both are possible, see section 3.2 for details.

process	internal name	code	refs.
	`PhotonCollision:all`		[434]
$\gamma\gamma \to q\bar{q}$	`PhotonCollision:gmgm2qqbar`	261	
$\gamma\gamma \to c\bar{c}$	`PhotonCollision:gmgm2ccbar`	262	
$\gamma\gamma \to b\bar{b}$	`PhotonCollision:gmgm2bbbar`	263	
$\gamma\gamma \to e^+e^-$	`PhotonCollision:gmgm2ee`	264	
$\gamma\gamma \to \mu^+\mu^-$	`PhotonCollision:gmgm2mumu`	265	
$\gamma\gamma \to \tau^+\tau^-$	`PhotonCollision:gmgm2tautau`	266	
	`PhotonParton:all`		
$g\gamma \to q\bar{q}$	`PhotonParton:ggm2qqbar`	271 (281)	[435]
$g\gamma \to c\bar{c}$	`PhotonParton:ggm2ccbar`	272 (282)	[436]
$g\gamma \to b\bar{b}$	`PhotonParton:ggm2bbbar`	273 (283)	[436]
$q\gamma \to qg$	`PhotonParton:qgm2qg`	274 (284)	[435]
$q\gamma \to q\gamma$	`PhotonParton:qgm2qgm`	275 (285)	[435]

Table 12: List of internal weak-boson processes, see section 3.2 for details.

process	internal name	code	refs.
	`PromptPhoton:all`		
$qg \to q\gamma$	`PromptPhoton:qg2qgamma`	201	[422, 426]
$q\bar{q} \to g\gamma$	`PromptPhoton:qqbar2ggamma`	202	[422, 426]
$gg \to g\gamma$	`PromptPhoton:gg2ggamma`	203	[427–429]
$q\bar{q} \to \gamma\gamma$	`PromptPhoton:ffbar2gammagamma`	204	[428]
$gg \to \gamma\gamma$	`PromptPhoton:gg2gammagamma`	205	[427–429]
	`WeakBosonExchange:all`		[430]
$ff' \to ff'$	`WeakBosonExchange:ff2ff(t:gmZ)`	211	
$f_1 f_2 \to f_3 f_4$	`WeakBosonExchange:ff2ff(t:W)`	212	
	`WeakSingleBoson:all`		[423]
$f\bar{f} \to \gamma^*/Z$	`WeakSingleBoson:ffbar2gmZ`	221	
$ff' \to W^{\pm}$	`WeakSingleBoson:ffbar2W`	222	
$f\bar{f} \to \gamma^* \to f'\bar{f}'$	`WeakSingleBoson:ffbar2ffbar(s:gm)`	223	[422, 423]
$f\bar{f} \to \gamma^*/Z \to f'\bar{f}'$	`WeakSingleBoson:ffbar2ffbar(s:gmZ)`	224	[422, 423]
$f_1 \bar{f}_2 \to W^{\pm} \to f_3 \bar{f}_4$	`WeakSingleBoson:ffbar2ffbar(s:W)`	225	[422, 423]
	`WeakDoubleBoson:all`		
$f\bar{f}' \to \gamma^*/Z\,\gamma^*/Z$	`WeakDoubleBoson:ffbar2gmZgmZ`	231	[25, 423]
$f\bar{f}' \to Z\,W^{\pm}$	`WeakDoubleBoson:ffbar2ZW`	232	[25, 423]
$f\bar{f} \to W^+ W^-$	`WeakDoubleBoson:ffbar2WW`	233	[423, 431]
	`WeakBosonAndParton:all`		
$q\bar{q} \to \gamma^*/Z\,g$	`WeakBosonAndParton:qqbar2gmZg`	241	[423]
$qg \to \gamma^*/Z\,q$	`WeakBosonAndParton:qg2gmZq`	242	[423]
$f\bar{f} \to \gamma^*/Z\,\gamma$	`WeakBosonAndParton:ffbar2gmZgm`	243	[423]
$f\gamma \to \gamma^*/Z\,f$	`WeakBosonAndParton:fgm2gmZf`	244	[432]
$q\bar{q} \to W^{\pm}\,g$	`WeakBosonAndParton:qqbar2Wg`	251	[423]
$qg \to W^{\pm}\,q$	`WeakBosonAndParton:qg2Wq`	252	[423]
$f\bar{f} \to W^{\pm}\,\gamma$	`WeakBosonAndParton:ffbar2Wgm`	253	[423, 433]
$f\gamma \to W^{\pm}\,f$	`WeakBosonAndParton:fgm2Wf`	254	[432]

Table 13: List of internal onia production processes, see section 3.3 for details. Here only charmonium channels are listed. The same channels are available for bottomonium by substituting $c\bar{c}$, Charmonium: and ccbar with $b\bar{b}$, Bottommonium: and bbbar respectively, and by increasing the code numbers with 100.

process	internal name	code
	Onia:all	
	Onia:all(3S1)	
	Onia:all(3PJ)	
	Onia:all(3DJ)	
	Charmoniun:all	
$gg \to \left\vert c\bar{c}(^3S_1)[^3S_1^{(1)}]\right\rangle g$	Charmonium:gg2ccbar(3S1)[3S1(1)]g	401
$gg \to \left\vert c\bar{c}(^3S_1)[^3S_1^{(1)}]\right\rangle \gamma$	Charmonium:gg2ccbar(3S1)[3S1(1)]gm	441
$gg \to \left\vert c\bar{c}(^3S_1)[^3S_1^{(8)}]\right\rangle g$	Charmonium:gg2ccbar(3S1)[3S1(8)]g	402
$qg \to \left\vert c\bar{c}(^3S_1)[^3S_1^{(8)}]\right\rangle q$	Charmonium:qg2ccbar(3S1)[3S1(8)]q	403
$q\bar{q} \to \left\vert c\bar{c}(^3S_1)[^3S_1^{(8)}]\right\rangle q$	Charmonium:qqbar2ccbar(3S1)[3S1(8)]g	404
$gg \to \left\vert c\bar{c}(^3S_1)[^1S_0^{(8)}]\right\rangle g$	Charmonium:gg2ccbar(3S1)[1S0(8)]g	405
$qg \to \left\vert c\bar{c}(^3S_1)[^1S_0^{(8)}]\right\rangle q$	Charmonium:qg2ccbar(3S1)[1S0(8)]q	406
$q\bar{q} \to \left\vert c\bar{c}(^3S_1)[^1S_0^{(8)}]\right\rangle q$	Charmonium:qqbar2ccbar(3S1)[1S0(8)]g	407
$gg \to \left\vert c\bar{c}(^3S_1)[^3P_J^{(8)}]\right\rangle g$	Charmonium:gg2ccbar(3S1)[3PJ(8)]g	408
$qg \to \left\vert c\bar{c}(^3S_1)[^3P_J^{(8)}]\right\rangle q$	Charmonium:qg2ccbar(3S1)[3PJ(8)]q	409
$q\bar{q} \to \left\vert c\bar{c}(^3S_1)[^3P_J^{(8)}]\right\rangle q$	Charmonium:qqbar2ccbar(3S1)[3PJ(8)]g	410
$gg \to \left\vert c\bar{c}(^3P_J)[^3P_J^{(1)}]\right\rangle g$	Charmonium:gg2ccbar(3PJ)[3PJ(1)]g	411
$qg \to \left\vert c\bar{c}(^3P_J)[^3P_J^{(1)}]\right\rangle q$	Charmonium:qg2ccbar(3PJ)[3PJ(1)]q	412
$q\bar{q} \to \left\vert c\bar{c}(^3P_J)[^3P_J^{(1)}]\right\rangle q$	Charmonium:qqbar2ccbar(3PJ)[3PJ(1)]g	413
$gg \to \left\vert c\bar{c}(^3P_J)[^3S_1^{(8)}]\right\rangle g$	Charmonium:gg2ccbar(3PJ)[3S1(8)]g	414
$qg \to \left\vert c\bar{c}(^3P_J)[^3S_1^{(8)}]\right\rangle q$	Charmonium:qg2ccbar(3PJ)[3S1(8)]q	415
$q\bar{q} \to \left\vert c\bar{c}(^3P_J)[^3S_1^{(8)}]\right\rangle q$	Charmonium:qqbar2ccbar(3PJ)[3S1(8)]g	416
$gg \to \left\vert c\bar{c}(^3D_J)[^3D_J^{(1)}]\right\rangle g$	Charmonium:gg2ccbar(3DJ)[3DJ(1)]g	417
$gg \to \left\vert c\bar{c}(^3D_J)[^3P_J^{(8)}]\right\rangle g$	Charmonium:gg2ccbar(3DJ)[3PJ(8)]g	418
$qg \to \left\vert c\bar{c}(^3D_J)[^3P_J^{(8)}]\right\rangle q$	Charmonium:qg2ccbar(3DJ)[3PJ(8)]q	419
$q\bar{q} \to \left\vert c\bar{c}(^3D_J)[^3P_J^{(8)}]\right\rangle q$	Charmonium:qqbar2ccbar(3DJ)[3PJ(8)]g	420
$gg \to \left\vert c\bar{c}(^3S_1)[^3S_1^{(1)}]\right\rangle\left\vert c\bar{c}(^3S_1)[^3S_1^{(1)}]\right\rangle$		
	Charmonium:gg2doubleccbar(3S1)[3S1(1)]	421
$q\bar{q} \to \left\vert c\bar{c}(^3S_1)[^3S_1^{(1)}]\right\rangle\left\vert c\bar{c}(^3S_1)[^3S_1^{(1)}]\right\rangle$		
	Charmonium:qqbar2doubleccbar(3S1)[3S1(1)]	422

Table 14: List of internal top-production processes and production of fourth-generation fermions. Expressions are from [14].

process	internal name	code
	`Top:all`	
$gg \rightarrow t\bar{t}$	`Top:gg2ttbar`	601
$q\bar{q} \rightarrow t\bar{t}$	`Top:qqbar2ttbar`	602
$qq \rightarrow tq$	`Top:qq2tq(t:W)`	603
$f\bar{f} \rightarrow \gamma/Z \rightarrow t\bar{t}$	`Top:ffbar2ttbar(s:gmZ)`	604
$f\bar{f} \rightarrow W^{\pm} \rightarrow t\bar{q}$	`Top:ffbar2tqbar(s:W)`	605
$\gamma\gamma \rightarrow t\bar{t}$	`Top:gmgm2ttbar`	606
$g\gamma \rightarrow t\bar{t}$	`Top:ggm2ttbar`	607
	`FourthBottom:all`	
$gg \rightarrow b'\bar{b}'$	`FourthBottom:gg2bPrimebPrimebar`	
$q\bar{q} \rightarrow b'\bar{b}'$	`FourthBottom:qqbar2bPrimebPrimebar`	801
$f\bar{f} \rightarrow b'q$ (t-channel W)	`FourthBottom:qq2bPrimeq(t:W)`	803
$f\bar{f} \rightarrow b'\bar{b}'$ (s-channel γ/Z)	`FourthBottom:ffbar2bPrimebPrimebar(s:gmZ)`	804
$f\bar{f}' \rightarrow b'\bar{q}$ (s-channel W)	`FourthBottom:ffbar2bPrimeqbar(s:W)`	805
$f\bar{f}' \rightarrow b'\bar{t}$ (s-channel W)	`FourthBottom:ffbar2bPrimetbar(s:W)`	806
	`FourthTop:all`	
$gg \rightarrow t'\bar{t}'$	`FourthTop:gg2tPrimetPrimebar`	821
$q\bar{q} \rightarrow t'\bar{t}'$	`FourthTop:qqbar2tPrimetPrimebar`	822
$f\bar{f} \rightarrow b'q$ (t-channel W)	`FourthTop:qq2tPrimeq(t:W)`	823
$f\bar{f} \rightarrow t'\bar{t}'$ (s-channel γ/Z)	`FourthTop:ffbar2tPrimetPrimebar(s:gmZ)`	824
$f\bar{f}' \rightarrow t'\bar{q}$ (s-channel W)	`FourthTop:ffbar2tPrimeqbar(s:W)`	825
$f\bar{f}' \rightarrow t'\bar{b}'$ (s-channel W)	`FourthPair:ffbar2tPrimebPrimebar(s:W)`	841
$f\bar{f}' \rightarrow \tau'\bar{\nu}'$ (s-channel W)	`FourthPair:ffbar2tauPrimenuPrimebar(s:W)`	842

Table 15: List of internal SM-Higgs production processes. See section 3.5 for details.

process	internal name	code
	`HiggsSM:all`	
$\bar{f}f \to H_{SM}$	`HiggsSM:ffbar2H`	901
$gg \to H_{SM}$	`HiggsSM:gg2H`	902
$\gamma\gamma \to H_{SM}$	`HiggsSM:gmgm2H`	903
$\bar{f}f \to H_{SM}Z$	`HiggsSM:ffbar2HZ`	904
$\bar{f}f \to H_{SM}W$	`HiggsSM:ffbar2HW`	905
$\bar{f}f \to H_{SM}\bar{f}f$ (ZBF)	`HiggsSM:ff2Hff(t:ZZ)`	906
$\bar{f}f \to H_{SM}\bar{f}f$ (WBF)	`HiggsSM:ff2Hff(t:WW)`	907
$gg \to H_{SM}t\bar{t}$	`HiggsSM:gg2Httbar`	908
$q\bar{q} \to H_{SM}t\bar{t}$	`HiggsSM:qqbar2Httbar`	909
$qg \to H_{SM}q$	`HiggsSM:qg2Hq`	911
$gg \to H_{SM}b\bar{b}$	`HiggsSM:gg2Hbbbar`	912
$q\bar{q} \to H_{SM}b\bar{b}$	`HiggsSM:qqbar2Hbbbar`	913
$gg \to H_{SM}g$	`HiggsSM:gg2Hg(l:t)`	914
$qg \to H_{SM}q$	`HiggsSM:qg2Hq(l:t)`	915
$q\bar{q} \to H_{SM}g$	`HiggsSM:qqbar2Hg(l:t)`	916

A.2 Beyond-the-Standard-Model processes

Table 16: List of internal SUSY particle production processes. Expressions from [82, 437,438]. Particular flavour states can be selected using `IdA` and `idB`, see section 3.6 or the online manual for details.

process	internal name
	`SUSY:all`
$gg \to \tilde{g}\tilde{g}$	`SUSY:gg2gluinogluino`
$q\bar{q} \to \tilde{g}\tilde{g}$	`SUSY:qqbar2gluinogluino`
$qg \to \tilde{q}\tilde{g}$	`SUSY:qg2squarkgluino`
$gg \to \tilde{q}_i\tilde{q}_j^*$	`SUSY:gg2squarkantisquark`
$q\bar{q} \to \tilde{q}_i\tilde{q}_j^*$	`SUSY:qqbar2squarkantisquark`
$q\bar{q} \to \tilde{q}_i\tilde{q}_j^*$ (No EW)	`SUSY:qqbar2squarkantisquark:onlyQCD`
$q\bar{q} \to \tilde{q}_i\tilde{q}_j^*$	`SUSY:qqbar2squarkantisquark`
$qq \to \tilde{q}_i\tilde{q}_j$	`SUSY:qq2squarksquark`
$qq \to \tilde{q}_i\tilde{q}_j$ (No EW)	`SUSY:qq2squarksquark:onlyQCD`
$q\bar{q} \to \tilde{\chi}_i^0\tilde{\chi}_j^0$	`SUSY:qqbar2chi0chi0`
$q\bar{q} \to \tilde{\chi}_i^\pm\tilde{\chi}_j^0$	`SUSY:qqbar2chi+-chi0`
$q\bar{q} \to \tilde{\chi}_i^\pm\tilde{\chi}_j^\mp$	`SUSY:qqbar2chi+chi-`
$qg \to \tilde{q}\tilde{\chi}_i^0$	`SUSY:qg2chi0squark`
$qg \to \tilde{q}\tilde{\chi}_i^\pm$	`SUSY:qg2chi+-squark`
$q\bar{q} \to \tilde{\chi}_i^0\tilde{g}$	`SUSY:qqbar2chi0gluino`
$q\bar{q} \to \tilde{\chi}_i^\pm\tilde{g}$	`SUSY:qqbar2chi+-gluino`
$\bar{f}f \to \tilde{\ell}_i\tilde{\ell}_j^*$	`SUSY:qqbar2sleptonantislepton`
$q_iq_j \to \tilde{q}_k^*$	`SUSY:qq2antisquark`

Table 17: List of internal BSM-Higgs production processes. See section 3.5 for details. Expressions from [439, 440].

process	internal name	code
	HiggsBSM:all	
	(replace H1 with H2 or A3)	
	HiggsBSM:allH1	
$\bar{f}f \to H_1(H_2,A_3)$	HiggsBSM:ffbar2H1	1001, 1021, 1041
$gg \to H_1(H_2,A_3)$	HiggsBSM:gg2H1	1002, 1022, 1042
$\gamma\gamma \to H_1(H_2,A_3)$	HiggsBSM:gmgm2H1	1003, 1023, 1043
$\bar{f}f \to H_1(H_2,A_3)Z$	HiggsBSM:ffbar2H1Z	1004, 1024, 1044
$\bar{f}f \to H_1(H_2,A_3)W$	HiggsBSM:ffbar2H1W	1005, 1025, 1045
$\bar{f}f \to H_1(H_2,A_3)\bar{f}f$ (ZBF)	HiggsBSM:ff2H1ff(t:ZZ)	1006, 1026, 1046
$\bar{f}f \to H_1(H_2,A_3)\bar{f}f$ (WBF)	HiggsBSM:ff2H1ff(t:WW)	1007, 1027, 1047
$gg \to H_1(H_2,A_3)t\bar{t}$	HiggsBSM:gg2H1ttbar	1008, 1028, 1048
$q\bar{q} \to H_1(H_2,A_3)t\bar{t}$	HiggsBSM:qqbar2H1ttbar	1009, 1029, 1049
	HiggsBSM:allH+-	
$\bar{f}f \to H^\pm$	HiggsBSM:ffbar2H+-	1061
$bg \to H^\pm$	HiggsBSM:bg2H+-t	1062
	HiggsBSM:allHpair	
$\bar{f}f \to A_3H_1$	HiggsBSM:ffbar2A3H1	1081
$\bar{f}f \to A_3H_2$	HiggsBSM:ffbar2A3H2	1082
$\bar{f}f \to H^\pm H_1$	HiggsBSM:ffbar2H+-H1	1083
$\bar{f}f \to H^\pm H_2$	HiggsBSM:ffbar2H+-H2	1084
$\bar{f}f \to H^\pm A_3$	HiggsBSM:ffbar2H+H-	1085
$qg \to H_1(H_2,A_3)q$	HiggsBSM:qg2H1q	1011, 1031, 1051
$gg \to H_1(H_2,A_3)b\bar{b}$	HiggsBSM:gg2H1bbbar	1012, 1032, 1052
$q\bar{q} \to H_1(H_2,A_3)b\bar{b}$	HiggsBSM:qqbar2H1bbbar	1013, 1033, 1053
$gg \to H_1(H_2,A_3)g$	HiggsBSM:gg2H1g(l:t)	1014, 1034, 1054
$qg \to H_1(H_2,A_3)q$	HiggsBSM:qg2H1q(l:t)	1015, 1035, 1055
$q\bar{q} \to H_1(H_2,A_3)g$	HiggsBSM:qqbar2H1g(l:t)	1016, 1036, 1056

Table 18: List of internal processes for dark matter. See section 3.8 and [441] for details.

process	internal name	code
$gg \to \chi\bar{\chi}$	DM:gg2S2XX	6011
$gg \to \chi\bar{\chi}j$	DM:gg2S2XXj	6012
$\bar{f}f \to \chi\bar{\chi}$	DM:ffbar2Zp2XX	6001
$\bar{f}f \to \chi\bar{\chi}$	DM:ffbar2Zp2XXj	6002
$\bar{f}f \to \chi\bar{\chi}j$	DM:qg2Zp2XXj	6003
$\bar{f}f \to Z'H$	DM:ffbar2ZpH	6004
$q\bar{q} \to \Psi\bar{\Psi}$	DM:qqbar2DY	6020

Table 19: List of internal processes mediated by new gauge bosons or leptoquarks. See section 3.9.

process	internal name	code	refs.
$f\bar{f} \to \gamma/Z/Z'$	`NewGaugeBoson:ffbar2gmZZprime`	3001	[442]
$f\bar{f} \to W'$	`NewGaugeBoson:ffbar2Wprime`	3021	[442]
$f\bar{f} \to R_0$	`NewGaugeBoson:ffbar2R0`	3041	[442]
	`LeftRightSymmmetry:all`		[443]
$f\bar{f} \to Z_R$	`LeftRightSymmmetry:ffbar2ZR`	3101	
$f\bar{f} \to W'$	`LeftRightSymmmetry:ffbar2WR`	3102	
$\ell\bar{\ell} \to H_L$	`LeftRightSymmmetry:ll2HL`	3121	
$\ell\gamma \to H_L e$	`LeftRightSymmmetry:lgm2HLe`	3122	
$\ell\gamma \to H_L \mu$	`LeftRightSymmmetry:lgm2HLmu`	3123	
$\ell\gamma \to H_L \tau$	`LeftRightSymmmetry:lgm2HLtau`	3124	
$f\bar{f} \to f\bar{f}H_L$	`LeftRightSymmmetry:ff2HLff`	3125	
$f\bar{f} \to H_L H_L$	`LeftRightSymmmetry:ffbar2HLHL`	3126	
$\ell\bar{\ell} \to H_R$	`LeftRightSymmmetry:ll2HR`	3141	
$\ell\gamma \to H_R e$	`LeftRightSymmmetry:lgm2HRe`	3142	
$\ell\gamma \to H_R \mu$	`LeftRightSymmmetry:lgm2HRmu`	3143	
$\ell\gamma \to H_R \tau$	`LeftRightSymmmetry:lgm2HRtau`	3144	
$f\bar{f} \to f\bar{f}H_R$	`LeftRightSymmmetry:ff2HRff`	3145	
$f\bar{f} \to H_R H_R$	`LeftRightSymmmetry:ffbar2HRHR`	3146	
	`LeptoQuark:all`		
$q\ell \to S$	`LeptoQuark:ql2LQ`	3201	[444]
$qg \to \ell S$	`LeptoQuark:qg2LQl`	3202	[444]
$gg \to SS^*$	`LeptoQuark:gg2LQLQbar`	3203	[444]
$f\bar{f} \to SS^*$	`LeptoQuark:qqbar2LQLQbar`	3204	[444]

Table 20: List of internal processes for excited fermions. See section 3.9.

process	internal name	code	refs.
	ExcitedFermion:all		[445, 446]
$dg \to d^*$	ExcitedFermion:dg2dStar	4001	[445, 446]
$ug \to u^*$	ExcitedFermion:ug2uStar	4002	[445, 446]
$sg \to s^*$	ExcitedFermion:sg2sStar	4003	[445, 446]
$cg \to c^*$	ExcitedFermion:cg2cStar	4004	[445, 446]
$bg \to b^*$	ExcitedFermion:bg2bStar	4005	[445, 446]
$e\gamma \to e^*$	ExcitedFermion:egm2eStar	4011	[445, 446]
$\mu\gamma \to \mu^*$	ExcitedFermion:mugm2muStar	4013	[445, 446]
$\tau\gamma \to \tau^*$	ExcitedFermion:taugm2tauStar	4015	[445, 446]
$qq \to d^*q$	ExcitedFermion:qq2dStarq	4021	[445, 446]
$qq \to u^*q$	ExcitedFermion:qq2uStarq	4022	[445, 446]
$qq \to s^*q$	ExcitedFermion:qq2sStarq	4023	[445, 446]
$qq \to c^*q$	ExcitedFermion:qq2cStarq	4024	[445, 446]
$qq \to b^*q$	ExcitedFermion:qq2bStarq	4025	[445, 446]
$q\bar{q} \to e^*e$	ExcitedFermion:qqbar2eStare	4031	[445, 446]
$q\bar{q} \to \nu_e^*\nu_e$	ExcitedFermion:qqbar2nueStarnue	4032	[445, 446]
$q\bar{q} \to \mu^*\mu$	ExcitedFermion:qqbar2muStarmu	4033	[445, 446]
$q\bar{q} \to \nu_\mu^*\nu_\mu$	ExcitedFermion:qqbar2numuStarnumu	4034	[445, 446]
$q\bar{q} \to \tau^*\tau$	ExcitedFermion:qqbar2tauStartau	4035	[445, 446]
$q\bar{q} \to \nu_\tau^*\nu_\tau$	ExcitedFermion:qqbar2nutauStarnutau	4036	[445, 446]
$q\bar{q} \to e^*e^*$	ExcitedFermion:qqbar2eStareStar	4051	[445, 446]
$q\bar{q} \to \nu_e^*\nu_e^*$	ExcitedFermion:qqbar2nueStarnueStar	4052	[445, 446]
$q\bar{q} \to \mu^*\mu^*$	ExcitedFermion:qqbar2muStarmuStar	4053	[445, 446]
$q\bar{q} \to \nu_\mu^*\nu_\mu^*$	ExcitedFermion:qqbar2numuStarnumuStar	4054	[445, 446]
$q\bar{q} \to \tau^*\tau^*$	ExcitedFermion:qqbar2tauStartauStar	4055	[445, 446]
$q\bar{q} \to \nu_\tau^*nu_\tau^*$	ExcitedFermion:qqbar2nutauStarnutauStar	4056	[445, 446]

Table 21: List of internal processes for Randall–Sundrum resonances. See section 3.9 and [447, 448] for details.

process	internal name	code
	ExtraDimensionsG*:all	
$gg \to G^*$	ExtraDimensionsG*:gg2G*	5001
$f\bar{f} \to G^*$	ExtraDimensionsG*:ffbar2G*	5002
$gg \to G^*g$	ExtraDimensionsG*:gg2G*g	5003
$gq \to G^*q$	ExtraDimensionsG*:qg2G*q	5004
$q\bar{q} \to G^*g$	ExtraDimensionsG*:qqbar2G*g	5005
$q\bar{q} \to G_{KK}g$	ExtraDimensionsG*:qqbar2KKgluon*	5006

Table 22: List of internal processes for TeV^{-1}-sized extra dimensions. See section 3.9 for details, expressions from [449].

process	internal name	code
$\bar{f}f \to d\bar{d}$	ExtraDimensionsTEV:ffbar2ddbar	5061
$\bar{f}f \to u\bar{u}$	ExtraDimensionsTEV:ffbar2uubar	5062
$\bar{f}f \to s\bar{s}$	ExtraDimensionsTEV:ffbar2ssbar	5063
$\bar{f}f \to c\bar{c}$	ExtraDimensionsTEV:ffbar2ccbar	5064
$\bar{f}f \to b\bar{b}$	ExtraDimensionsTEV:ffbar2bbbar	5065
$\bar{f}f \to t\bar{t}$	ExtraDimensionsTEV:ffbar2ttbar	5066
$\bar{f}f \to e^+e^-$	ExtraDimensionsTEV:ffbar2e+e-	5071
$\bar{f}f \to \nu_e\bar{\nu}_e$	ExtraDimensionsTEV:ffbar2nuenuebar	5072
$\bar{f}f \to \mu^+\mu^-$	ExtraDimensionsTEV:ffbar2mu+mu-	5073
$\bar{f}f \to \nu_\mu\bar{\nu}_\mu$	ExtraDimensionsTEV:ffbar2numunumubar	5074
$\bar{f}f \to \tau^+\tau^-$	ExtraDimensionsTEV:ffbar2tau+tau-	5076
$\bar{f}f \to \nu_\tau\bar{\nu}_\tau$	ExtraDimensionsTEV:ffbar2nutaunutaubar	5076

Table 23: List of internal processes for large extra dimensions. See section 3.9 for details.

process	internal name	code	refs.
	ExtraDimensionsLED:monojet		[450, 451]
$gg \to Gg$	ExtraDimensionsLED:gg2Gg	5021	
$gq \to Gq$	ExtraDimensionsLED:qg2Gq	5022	
$q\bar{q} \to Gg$	ExtraDimensionsLED:qqbar2Gg	5023	
$\bar{f}f \to GZ$	ExtraDimensionsLED:ffbar2GZ	5024	[450]
$\bar{f}f \to G\gamma$	ExtraDimensionsLED:ffbar2Ggamma	5025	[450]
$\bar{f}f \to \gamma\gamma$	ExtraDimensionsLED:ffbar2gammagamma	5026	[450]
$gg \to \gamma\gamma$	ExtraDimensionsLED:gg2gammagamma	5027	[450]
$\bar{f}f \to \ell\bar{\ell}$	ExtraDimensionsLED:ffbar2llbar	5028	[450]
$gg \to \ell\bar{\ell}$	ExtraDimensionsLED:gg2llbar	5029	[450]
	ExtraDimensionsLED:dijets		[450]
$gg \to gg$	ExtraDimensionsLED:gg2DJgg	5030	
$gg \to q\bar{q}$	ExtraDimensionsLED:gg2DJqqbar	5031	
$qg \to qg$	ExtraDimensionsLED:qg2DJqg	5032	
$qq \to qq$	ExtraDimensionsLED:qq2DJqq	5033	
$q\bar{q} \to gg$	ExtraDimensionsLED:qqbar2DJgg	5034	
$q\bar{q} \to q'\bar{q}'$	ExtraDimensionsLED:qqbar2DJqqbarNew	5035	

Table 24: List of internal processes for unparticles. Expressions from [452, 453], see section 3.9.

process	internal name	code
	`ExtraDimensionsUnpart:monojet`	
$gg \to Ug$	`ExtraDimensionsUnpart:gg2Ug`	5045
$gq \to Uq$	`ExtraDimensionsUnpart:qg2Uq`	5046
$q\bar{q} \to Ug$	`ExtraDimensionsUnpart:qqbar2Ug`	5047
$f\bar{f} \to UZ$	`ExtraDimensionsUnpart:ffbar2UZ`	5041
$f\bar{f} \to U\gamma$	`ExtraDimensionsUnpart:ffbar2Ugamma`	5042
$f\bar{f} \to U \to \gamma\gamma$	`ExtraDimensionsUnpart:ffbar2gammagamma`	5043
$gg \to U \to \gamma\gamma$	`ExtraDimensionsUnpart:gg2gammagamma`	5044
$f\bar{f} \to U \to \ell\bar{\ell}$	`ExtraDimensionsUnpart:ffbar2llbar`	5048
$gg \to U \to \ell\bar{\ell}$	`ExtraDimensionsUnpart:gg2llbar`	5049

Table 25: List of internal hidden valley processes, see section 3.7 and [38, 39] for details.

process	internal name	code
	`HiddenValley:all`	
$gg \to d_\nu \bar{d}_\nu$	`HiddenValley:gg2DvDvbar`	4901
$gg \to u_\nu \bar{u}_\nu$	`HiddenValley:gg2UvUvbar`	4902
$gg \to s_\nu \bar{s}_\nu$	`HiddenValley:gg2SvSvbar`	4903
$gg \to c_\nu \bar{c}_\nu$	`HiddenValley:gg2CvCvbar`	4904
$gg \to b_\nu \bar{b}_\nu$	`HiddenValley:gg2BvBvbar`	4905
$gg \to t_\nu \bar{t}_\nu$	`HiddenValley:gg2TvTvbar`	4906
$q\bar{q} \to d_\nu \bar{d}_\nu$	`HiddenValley:qqbar2DvDvbar`	4911
$q\bar{q} \to u_\nu \bar{u}_\nu$	`HiddenValley:qqbar2UvUvbar`	4912
$q\bar{q} \to s_\nu \bar{s}_\nu$	`HiddenValley:qqbar2SvSvbar`	4913
$q\bar{q} \to c_\nu \bar{c}_\nu$	`HiddenValley:qqbar2CvCvbar`	4914
$q\bar{q} \to b_\nu \bar{b}_\nu$	`HiddenValley:qqbar2BvBvbar`	4915
$q\bar{q} \to t_\nu \bar{t}_\nu$	`HiddenValley:qqbar2TvTvbar`	4916
$f\bar{f} \to d_\nu \bar{d}_\nu$	`HiddenValley:ffbar2DvDvbar`	4921
$f\bar{f} \to u_\nu \bar{u}_\nu$	`HiddenValley:ffbar2UvUvbar`	4922
$f\bar{f} \to s_\nu \bar{s}_\nu$	`HiddenValley:ffbar2SvSvbar`	4923
$f\bar{f} \to c_\nu \bar{c}_\nu$	`HiddenValley:ffbar2CvCvbar`	4924
$f\bar{f} \to b_\nu \bar{b}_\nu$	`HiddenValley:ffbar2BvBvbar`	4925
$f\bar{f} \to t_\nu \bar{t}_\nu$	`HiddenValley:ffbar2TvTvbar`	4926
$f\bar{f} \to e_\nu \bar{e}_\nu$	`HiddenValley:ffbar2EvEvbar`	4931
$f\bar{f} \to \mu_\nu \bar{\mu}_\nu$	`HiddenValley:ffbar2MUvMUvbar`	4932
$f\bar{f} \to \tau_\nu \bar{\tau}_\nu$	`HiddenValley:ffbar2TAUvTAUvbar`	4933
$f\bar{f} \to \nu_{\tau\nu} \bar{\nu}_{\tau\nu}$	`HiddenValley:ffbar2nuEvnuEvbar`	4934
$f\bar{f} \to \nu_{\tau\nu} \bar{\nu}_{\tau\nu}$	`HiddenValley:ffbar2nuMUvnuMUvbar`	4935
$f\bar{f} \to \nu_{\tau\nu} \bar{\nu}_{\tau\nu}$	`HiddenValley:ffbar2nuTAUvnuTAUvbar`	4936
$f\bar{f} \to Z_\nu$	`HiddenValley:ffbar2Zv`	4941

List of acronyms

Below follows a list of standard acronyms used throughout the text, with a reference to the page where the acronym is first introduced.

References

[1] T. Sjöstrand et al., *An introduction to PYTHIA 8.2*, Comput. Phys. Commun. **191**, 159 (2015), doi:10.1016/j.cpc.2015.01.024.

[2] T. Sjöstrand, *The Pythia event generator: Past, present and future*, Comput. Phys. Commun. **246**, 106910 (2020), doi:10.1016/j.cpc.2019.106910.

[3] W. Bartel et al., *Experimental study of jets in electron-positron annihilation*, Phys. Lett. B **101**, 129 (1981), doi:10.1016/0370-2693(81)90505-0.

[4] L. Lönnblad, *Ariadne version 4 - A program for simulation of QDC cascades implementing the colour dipole model*, Comput. Phys. Commun. **71**, 15 (1992), doi:10.1016/0010-4655(92)90068-A.

[5] G. Ingelman, A. Edin and J. Rathsman, *LEPTO 6.5 - A Monte Carlo generator for deep inelastic lepton-nucleon scattering*, Comput. Phys. Commun. **101**, 108 (1997), doi:10.1016/S0010-4655(96)00157-9.

[6] H. Kharraziha and L. Lönnblad, *The linked dipole chain Monte Carlo*, J. High Energy Phys. **03**, 006 (1998), doi:10.1088/1126-6708/1998/03/006.

[7] B. Nilsson-Almqvist and E. Stenlund, *Interactions between hadrons and nuclei: The Lund Monte Carlo - FRITIOF version 1.6 -*, Comput. Phys. Commun. **43**, 387 (1987), doi:10.1016/0010-4655(87)90056-7.

[8] F. James and L. Moneta, *Review of high-quality random number generators*, Comput. Softw. Big Sci. **4**, 2 (2020), doi:10.1007/s41781-019-0034-3.

[9] G. Marsaglia, B. Narasimhan and A. Zaman, *A random number generator for PC's*, Comput. Phys. Commun. **60**, 345 (1990), doi:10.1016/0010-4655(90)90033-W.

[10] M. Lüscher, *A portable high-quality random number generator for lattice field theory simulations*, Comput. Phys. Commun. **79**, 100 (1994), doi:10.1016/0010-4655(94)90232-1.

[11] K. G. Savvidy, *The MIXMAX random number generator*, Comput. Phys. Commun. **196**, 161 (2015), doi:10.1016/j.cpc.2015.06.003.

[12] R. Kleiss and R. Pittau, *Weight optimization in multichannel Monte Carlo*, Comput. Phys. Commun. **83**, 141 (1994), doi:10.1016/0010-4655(94)90043-4.

[13] T. Sjöstrand and M. van Zijl, *A multiple-interaction model for the event structure in hadron collisions*, Phys. Rev. D **36**, 2019 (1987), doi:10.1103/PhysRevD.36.2019.

[14] T. Sjöstrand, S. Mrenna and P. Skands, *PYTHIA 6.4 physics and manual*, J. High Energy Phys. **05**, 026 (2006), doi:10.1088/1126-6708/2006/05/026.

[15] L. Lönnblad, *Fooling around with the Sudakov veto algorithm*, Eur. Phys. J. C **73**, 2350 (2013), doi:10.1140/epjc/s10052-013-2350-9.

[16] S. Plätzer and M. Sjödahl, *The Sudakov veto algorithm reloaded*, Eur. Phys. J. Plus **127**, 26 (2012), doi:10.1140/epjp/i2012-12026-x.

[17] R. Kleiss and R. Verheyen, *Competing Sudakov veto algorithms*, Eur. Phys. J. C **76**, 359 (2016), doi:10.1140/epjc/s10052-016-4231-5.

[18] P. A. W. Lewis and G. S. Shedler, *Simulation of nonhomogeneous poisson processes by thinning*, Nav. Res. Logist. **26**, 403 (1979), doi:10.1002/nav.3800260304.

[19] L. Devroye, *Non-uniform random variate generation*, Springer, New York, US, ISBN 9781461386438 (2013), doi:10.1007/978-1-4613-8643-8.

[20] S. Mrenna and P. Skands, *Automated parton-shower variations in Pythia 8*, Phys. Rev. D **94**, 074005 (2016), doi:10.1103/PhysRevD.94.074005.

[21] F. James, *Monte-Carlo phase space*, CERN Acad. Train. Lect. **23**, CERN-68-15 (1968), doi:10.5170/CERN-1968-015.

[22] R. Kleiss, W. J. Stirling and S. D. Ellis, *A new Monte Carlo treatment of multiparticle phase space at high energies*, Comput. Phys. Commun. **40**, 359 (1986), doi:10.1016/0010-4655(86)90119-0.

[23] H. Brooks, P. Skands and R. Verheyen, *Interleaved resonance decays and electroweak radiation in Vincia*, SciPost Phys. **12**, 101 (2022), doi:10.21468/SciPostPhys.12.3.101.

[24] E. Norrbin and T. Sjöstrand, *Production and hadronization of heavy quarks*, Eur. Phys. J. C **17**, 137 (2000), doi:10.1007/s100520000460.

[25] J. F. Gunion and Z. Kunszt, *Lepton correlations in gauge-boson pair production and decay*, Phys. Rev. D **33**, 665 (1986), doi:10.1103/PhysRevD.33.665.

[26] G. T. Bodwin, E. Braaten and G. P. Lepage, *Rigorous QCD analysis of inclusive annihilation and production of heavy quarkonium*, Phys. Rev. D **51**, 1125 (1995), doi:10.1103/PhysRevD.51.1125.

[27] R. Baier and R. Rückl, *Hadronic collisions: A quarkonium factory*, Z. Phys. C **19**, 251 (1983), doi:10.1007/BF01572254.

[28] R. Gastmans, W. Troost and T. T. Wu, *Cross sections for gluon + gluon → heavy quarkonium + gluon*, Phys. Lett. B **184**, 257 (1987), doi:10.1016/0370-2693(87)90578-8.

[29] P. Cho and A. K. Leibovich, *Color-octet quarkonia production. II*, Phys. Rev. D **53**, 6203 (1996), doi:10.1103/PhysRevD.53.6203.

[30] F. Yuan, C.-F. Qiao and K.-T. Chao, *D-wave heavy quarkonium production in fixed target experiments*, Phys. Rev. D **59**, 014009 (1998), doi:10.1103/PhysRevD.59.014009.

[31] B. Humpert and P. Méry, $\psi\psi$ *production at collider energies*, Z. Phys. C **20**, 83 (1983), doi:10.1007/BF01577721.

[32] C.-F. Qiao, J/ψ *pair production at the Fermilab Tevatron*, Phys. Rev. D **66**, 057504 (2002), doi:10.1103/PhysRevD.66.057504.

[33] P. Nason et al., *Bottom production*, arXiv:hep-ph/0003142.

[34] M. Bargiotti and V. Vagnoni, *Heavy quarkonia sector in PYTHIA 6.324: Tuning, validation and perspectives at LHC(b)* (2007).

[35] P. A. Zyla et al., *Review of particle physics*, Prog. Theor. Exp. Phys. 083C01 (2020), doi:10.1093/ptep/ptaa104.

[36] R. Aaij et al., *Study of J/ψ production in jets*, Phys. Rev. Lett. **118**, 192001 (2017), doi:10.1103/PhysRevLett.118.192001.

[37] A. M. Sirunyan et al., *Study of J/ψ meson production inside jets in pp collisions at $\sqrt{s} = 8$ TeV*, Phys. Lett. B **804**, 135409 (2020), doi:10.1016/j.physletb.2020.135409.

[38] L. Carloni and T. Sjöstrand, *Visible effects of invisible hidden valley radiation*, J. High Energy Phys. **09**, 105 (2010), doi:10.1007/JHEP09(2010)105.

[39] L. Carloni, J. Rathsman and T. Sjöstrand, *Discerning secluded sector gauge structures*, J. High Energy Phys. **04**, 091 (2011), doi:10.1007/JHEP04(2011)091.

[40] S. Dittmaier et al., *Handbook of LHC Higgs cross sections: 1. Inclusive observables*, CERN Yellow Rep.: Monogr., CERN-2011-002 (2011), doi:10.5170/CERN-2011-002.

[41] S. C. Park, H. S. Song and J. Song, *Z boson pair production at CERN LHC in a stabilized Randall-Sundrum scenario*, Phys. Rev. D **65**, 075008 (2002), doi:10.1103/PhysRevD.65.075008.

[42] K. Kovařík, P. M. Nadolsky and D. E. Soper, *Hadronic structure in high-energy collisions*, Rev. Mod. Phys. **92**, 045003 (2020), doi:10.1103/RevModPhys.92.045003.

[43] V. N. Gribov and L. N. Lipatov, *Deep inelastic* ep *scattering in perturbation theory*, Sov. J. Nucl. Phys. **15**, 438 (1972).

[44] Y. L. Dokshitzer, *Calculation of the structure functions for deep inelastic scattering and* e^+e^- *annihilation by perturbation theory in quantum chromodynamics*, Sov. Phys. J. Exp. Theor. Phys. **46**, 641 (1977).

[45] G. Altarelli and G. Parisi, *Asymptotic freedom in parton language*, Nucl. Phys. B **126**, 298 (1977), doi:10.1016/0550-3213(77)90384-4.

[46] M. Glück, E. Reya and I. Schienbein, *Pionic parton distributions revisited*, Eur. Phys. J. C **10**, 313 (1999), doi:10.1007/s100529900124.

[47] M. Glück, E. Reya and A. Vogt, *Pionic parton distributions*, Z. Phys. C **53**, 651 (1992), doi:10.1007/BF01559743.

[48] M. Glück, E. Reya and M. Stratmann, *Mesonic parton densities derived from constituent quark model constraints*, Eur. Phys. J. C **2**, 159 (1998), doi:10.1007/s100520050130.

[49] T. Sjöstrand and M. Utheim, *Hadron interactions for arbitrary energies and species, with applications to cosmic rays*, Eur. Phys. J. C **82**, 21 (2022), doi:10.1140/epjc/s10052-021-09953-5.

[50] V. Bertone, R. Gauld and J. Rojo, *Neutrino telescopes as QCD microscopes*, J. High Energy Phys. **01**, 217 (2019), doi:10.1007/JHEP01(2019)217.

[51] V. V. Sudakov, *Vertex parts at very high-energies in quantum electrodynamics*, Sov. Phys. J. Exp. Theor. Phys. **3**, 65 (1956).

[52] T. Sjöstrand, *A model for initial state parton showers*, Phys. Lett. B **157**, 321 (1985), doi:10.1016/0370-2693(85)90674-4.

[53] G. Gustafson, *Dual description of a confined colour field*, Phys. Lett. B **175**, 453 (1986), doi:10.1016/0370-2693(86)90622-2.

[54] G. Gustafson and U. Pettersson, *Dipole formulation of QCD cascades*, Nucl. Phys. B **306**, 746 (1988), doi:10.1016/0550-3213(88)90441-5.

[55] G. 't. Hooft, *A planar diagram theory for strong interactions*, Nucl. Phys. B **72**, 461 (1974), doi:10.1016/0550-3213(74)90154-0.

[56] S. Höche, D. Reichelt and F. Siegert, *Momentum conservation and unitarity in parton showers and NLL resummation*, J. High Energy Phys. **01**, 118 (2018), doi:10.1007/JHEP01(2018)118.

[57] N. Baberuxki, C. T. Preuss, D. Reichelt and S. Schumann, *Resummed predictions for jet-resolution scales in multijet production in e^+e^- annihilation*, J. High Energy Phys. **04**, 112 (2020), doi:10.1007/JHEP04(2020)112.

[58] M. Dasgupta, F. A. Dreyer, K. Hamilton, P. F. Monni and G. P. Salam, *Logarithmic accuracy of parton showers: A fixed-order study*, J. High Energy Phys. **09**, 033 (2018), doi:10.1007/JHEP09(2018)033.

[59] K. Hamilton, R. Medves, G. P. Salam, L. Scyboz and G. Soyez, *Colour and logarithmic accuracy in final-state parton showers*, J. High Energy Phys. **03**, 041 (2021), doi:10.1007/JHEP03(2021)041.

[60] Z. Nagy and D. E. Soper, *Summations of large logarithms by parton showers*, Phys. Rev. D **104**, 054049 (2021), doi:10.1103/PhysRevD.104.054049.

[61] Z. Nagy and D. E. Soper, *Summations by parton showers of large logarithms in electron-positron annihilation*, arXiv:2011.04777.

[62] M. Dasgupta, F. A. Dreyer, K. Hamilton, P. F. Monni, G. P. Salam and G. Soyez, *Parton showers beyond leading logarithmic accuracy*, Phys. Rev. Lett. **125**, 052002 (2020), doi:10.1103/PhysRevLett.125.052002.

[63] J. R. Forshaw, J. Holguin and S. Plätzer, *Building a consistent parton shower*, J. High Energy Phys. **09**, 014 (2020), doi:10.1007/JHEP09(2020)014.

[64] W. T. Giele, D. A. Kosower and P. Z. Skands, *Higher-order corrections to timelike jets*, Phys. Rev. D **84**, 054003 (2011), doi:10.1103/PhysRevD.84.054003.

[65] S. Plätzer and M. Sjödahl, *Subleading N_c improved parton showers*, J. High Energy Phys. **07**, 042 (2012), doi:10.1007/JHEP07(2012)042.

[66] S. Plätzer, M. Sjodahl and J. Thorén, *Color matrix element corrections for parton showers*, J. High Energy Phys. **11**, 009 (2018), doi:10.1007/JHEP11(2018)009.

[67] J. Bellm, *Colour rearrangement for dipole showers*, Eur. Phys. J. C **78**, 601 (2018), doi:10.1140/epjc/s10052-018-6070-z.

[68] J. Holguin, J. R. Forshaw and S. Plätzer, *Improvements on dipole shower colour*, Eur. Phys. J. C **81**, 364 (2021), doi:10.1140/epjc/s10052-021-09145-1.

[69] J. Isaacson and S. Prestel, *Stochastically sampling color configurations*, Phys. Rev. D **99**, 014021 (2019), doi:10.1103/PhysRevD.99.014021.

[70] S. Höche and D. Reichelt, *Numerical resummation at subleading color in the strongly ordered soft gluon limit*, Phys. Rev. D **104**, 034006 (2021), doi:10.1103/PhysRevD.104.034006.

[71] Z. Nagy and D. E. Soper, *Parton shower evolution with subleading color*, J. High Energy Phys. **06**, 044 (2012), doi:10.1007/JHEP06(2012)044.

[72] Z. Nagy and D. E. Soper, *Effects of subleading color in a parton shower*, J. High Energy Phys. **07**, 119 (2015), doi:10.1007/JHEP07(2015)119.

[73] J. R. Forshaw, J. Holguin and S. Plätzer, *Parton branching at amplitude level*, J. High Energy Phys. **08**, 145 (2019), doi:10.1007/JHEP08(2019)145.

[74] M. De Angelis, J. R. Forshaw and S. Plätzer, *Resummation and simulation of soft gluon effects beyond leading color*, Phys. Rev. Lett. **126**, 112001 (2021), doi:10.1103/PhysRevLett.126.112001.

[75] M. Bengtsson and T. Sjöstrand, *Coherent parton showers versus matrix elements - Implications of PETRA/PEP data*, Phys. Lett. B **185**, 435 (1987), doi:10.1016/0370-2693(87)91031-8.

[76] M. Bengtsson and T. Sjöstrand, *A comparative study of coherent and non-coherent parton shower evolution*, Nucl. Phys. B **289**, 810 (1987), doi:10.1016/0550-3213(87)90407-X.

[77] E. Norrbin and T. Sjöstrand, *QCD radiation off heavy particles*, Nucl. Phys. B **603**, 297 (2001), doi:10.1016/S0550-3213(01)00099-2.

[78] T. Sjöstrand and P. Z. Skands, *Transverse-momentum-ordered showers and interleaved multiple interactions*, Eur. Phys. J. C **39**, 129 (2005), doi:10.1140/epjc/s2004-02084-y.

[79] R. Corke and T. Sjöstrand, *Interleaved parton showers and tuning prospects*, J. High Energy Phys. **03**, 032 (2011), doi:10.1007/JHEP03(2011)032.

[80] B. Cabouat and T. Sjöstrand, *Some dipole shower studies*, Eur. Phys. J. C **78**, 226 (2018), doi:10.1140/epjc/s10052-018-5645-z.

[81] G. Miu and T. Sjöstrand, W *production in an improved parton-shower approach*, Phys. Lett. B **449**, 313 (1999), doi:10.1016/S0370-2693(99)00068-4.

[82] N. Desai and P. Z. Skands, *Supersymmetry and generic BSM models in Pythia 8*, Eur. Phys. J. C **72**, 2238 (2012), doi:10.1140/epjc/s10052-012-2238-0.

[83] J. R. Christiansen and T. Sjöstrand, *Weak gauge boson radiation in parton showers*, J. High Energy Phys. **04**, 115 (2014), doi:10.1007/JHEP04(2014)115.

[84] B. Andersson, G. Gustafson and J. Samuelsson, *The linked dipole chain model for DIS*, Nucl. Phys. B **467**, 443 (1996), doi:10.1016/0550-3213(96)00114-9.

[85] S. Catani, B. R. Webber and G. Marchesini, *QCD coherent branching and semi-inclusive processes at large x*, Nucl. Phys. B **349**, 635 (1991), doi:10.1016/0550-3213(91)90390-J.

[86] G. Gustafson, *Multiplicity distributions in QCD cascades*, Nucl. Phys. B **392**, 251 (1993), doi:10.1016/0550-3213(93)90203-2.

[87] S. Catani and M. H. Seymour, *A general algorithm for calculating jet cross sections in NLO QCD*, Nucl. Phys. B **485**, 291 (1997), doi:10.1016/S0550-3213(96)00589-5.

[88] H. Brooks and P. Skands, *Coherent showers in decays of colored resonances*, Phys. Rev. D **100**, 076006 (2019), doi:10.1103/PhysRevD.100.076006.

[89] T. Plehn, D. Rainwater and P. Skands, *Squark and gluino production with jets*, Phys. Lett. B **645**, 217 (2007), doi:10.1016/j.physletb.2006.12.009.

[90] R. Corke and T. Sjöstrand, *Improved parton showers at large transverse momenta*, Eur. Phys. J. C **69**, 1 (2010), doi:10.1140/epjc/s10052-010-1409-0.

[91] R. K. Ellis, G. Marchesini and B. R. Webber, *Soft radiation in parton-parton scattering*, Nucl. Phys. B **286**, 643 (1987), doi:10.1016/0550-3213(87)90456-1.

[92] B. Andersson, G. Gustafson and T. Sjöstrand, *How to find the gluon jets in* e^+e^- *annihilation*, Phys. Lett. B **94**, 211 (1980), doi:10.1016/0370-2693(80)90861-8.

[93] Y. I. Azimov, Y. L. Dokshitzer, V. A. Khoze and S. I. Troyan, *The string effect and QCD coherence*, Phys. Lett. B **165**, 147 (1985), doi:10.1016/0370-2693(85)90709-9.

[94] B. R. Webber, *Monte Carlo simulation of hard hadronic processes*, Annu. Rev. Nucl. Part. Sci. **36**, 253 (1986), doi:10.1146/annurev.ns.36.120186.001345.

[95] P. Ernström and L. Lönnblad, *Generating heavy quarkonia in a perturbative QCD cascade*, Z. Phys. C **75**, 51 (1997), doi:10.1007/s002880050446.

[96] L. Hartgring, E. Laenen and P. Skands, *Antenna showers with one-loop matrix elements*, J. High Energy Phys. **10**, 127 (2013), doi:10.1007/JHEP10(2013)127.

[97] J. J. Lopez-Villarejo and P. Skands, *Efficient matrix-element matching with sector showers*, J. High Energy Phys. **11**, 150 (2011), doi:10.1007/JHEP11(2011)150.

[98] S. Höche, S. Schumann and F. Siegert, *Hard photon production and matrix-element parton-shower merging*, Phys. Rev. D **81**, 034026 (2010), doi:10.1103/PhysRevD.81.034026.

[99] M. Ritzmann, D. A. Kosower and P. Skands, *Antenna showers with hadronic initial states*, Phys. Lett. B **718**, 1345 (2013), doi:10.1016/j.physletb.2012.12.003.

[100] H. Brooks, C. T. Preuss and P. Skands, *Sector showers for hadron collisions*, J. High Energy Phys. **07**, 032 (2020), doi:10.1007/JHEP07(2020)032.

[101] R. Kleiss and R. Verheyen, *Final-state QED multipole radiation in antenna parton showers*, J. High Energy Phys. **11**, 182 (2017), doi:10.1007/JHEP11(2017)182.

[102] P. Skands and R. Verheyen, *Multipole photon radiation in the Vincia parton shower*, Phys. Lett. B **811**, 135878 (2020), doi:10.1016/j.physletb.2020.135878.

[103] D. R. Yennie, S. C. Frautschi and H. Suura, *The infrared divergence phenomena and high-energy processes*, Ann. Phys. **13**, 379 (1961), doi:10.1016/0003-4916(61)90151-8.

[104] A. Gehrmann-De Ridder, M. Ritzmann and P. Skands, *Timelike dipole-antenna showers with massive fermions*, Phys. Rev. D **85**, 014013 (2012), doi:10.1103/PhysRevD.85.014013.

[105] R. Kleiss and R. Verheyen, *Collinear electroweak radiation in antenna parton showers*, Eur. Phys. J. C **80**, 980 (2020), doi:10.1140/epjc/s10052-020-08510-w.

[106] A. J. Larkoski, J. J. Lopez-Villarejo and P. Skands, *Helicity-dependent showers and matching with VINCIA*, Phys. Rev. D **87**, 054033 (2013), doi:10.1103/PhysRevD.87.054033.

[107] N. Fischer, A. Lifson and P. Skands, *Helicity antenna showers for hadron colliders*, Eur. Phys. J. C **77**, 719 (2017), doi:10.1140/epjc/s10052-017-5306-7.

[108] D. A. Kosower, *Antenna factorization of gauge-theory amplitudes*, Phys. Rev. D **57**, 5410 (1998), doi:10.1103/PhysRevD.57.5410.

[109] W. T. Giele, D. A. Kosower and P. Z. Skands, *A simple shower and matching algorithm*, Phys. Rev. D **78**, 014026 (2008), doi:10.1103/PhysRevD.78.014026.

[110] S. Catani and M. H. Seymour, *The dipole formalism for the calculation of QCD jet cross sections at next-to-leading order*, Phys. Lett. B **378**, 287 (1996), doi:10.1016/0370-2693(96)00425-X.

[111] S. Catani, S. Dittmaier, M. H. Seymour and Z. Trócsányi, *The dipole formalism for next-to-leading order QCD calculations with massive partons*, Nucl. Phys. B **627**, 189 (2002), doi:10.1016/S0550-3213(02)00098-6.

[112] N. Fischer, S. Prestel, M. Ritzmann and P. Skands, *VINCIA for hadron colliders*, Eur. Phys. J. C **76**, 589 (2016), doi:10.1140/epjc/s10052-016-4429-6.

[113] R. J. Verheyen, *Electroweak effects in antenna parton showers*, PhD thesis (2020).

[114] S. Schumann and F. Krauss, *A parton shower algorithm based on Catani-Seymour dipole factorisation*, J. High Energy Phys. **03**, 038 (2008), doi:10.1088/1126-6708/2008/03/038.

[115] M. Dinsdale, M. Ternick and S. Weinzierl, *Parton showers from the dipole formalism*, Phys. Rev. D **76**, 094003 (2007), doi:10.1103/PhysRevD.76.094003.

[116] S. Plätzer and S. Gieseke, *Coherent parton showers with local recoils*, J. High Energy Phys. **01**, 024 (2011), doi:10.1007/JHEP01(2011)024.

[117] S. Höche and S. Prestel, *The midpoint between dipole and parton showers*, Eur. Phys. J. C **75**, 461 (2015), doi:10.1140/epjc/s10052-015-3684-2.

[118] A. Daleo, T. Gehrmann and D. Maître, *Antenna subtraction with hadronic initial states*, J. High Energy Phys. **04**, 016 (2007), doi:10.1088/1126-6708/2007/04/016.

[119] D. A. Kosower, *Antenna factorization in strongly ordered limits*, Phys. Rev. D **71**, 045016 (2005), doi:10.1103/PhysRevD.71.045016.

[120] A. J. Larkoski and M. E. Peskin, *Spin-dependent antenna splitting functions*, Phys. Rev. D **81**, 054010 (2010), doi:10.1103/PhysRevD.81.054010.

[121] A. J. Larkoski and M. E. Peskin, *Antenna splitting functions for massive particles*, Phys. Rev. D **84**, 034034 (2011), doi:10.1103/PhysRevD.84.034034.

[122] J. M. Campbell, M. A. Cullen and E. W. N. Glover, *Four jet event shapes in electron-positron annihilation*, Eur. Phys. J. C **9**, 245 (1999), doi:10.1007/s100529900034.

[123] A. Gehrmann-De Ridder, T. Gehrmann and E. W. N. Glover, *Infrared structure of* e^+e^- → 2 *jets at NNLO*, Nucl. Phys. B **691**, 195 (2004), doi:10.1016/j.nuclphysb.2004.05.017.

[124] A. Gehrmann-De Ridder, T. Gehrmann and E. W. N. Glover, *Quark-gluon antenna functions from neutralino decay*, Phys. Lett. B **612**, 36 (2005), doi:10.1016/j.physletb.2005.02.039.

[125] A. Gehrmann-De Ridder, T. Gehrmann and E. W. N. Glover, *Gluon-gluon antenna functions from Higgs boson decay*, Phys. Lett. B **612**, 49 (2005), doi:10.1016/j.physletb.2005.03.003.

[126] A. Gehrmann-De Ridder, T. Gehrmann and E. W. N. Glover, *Antenna subtraction at NNLO*, J. High Energy Phys. **09**, 056 (2005), doi:10.1088/1126-6708/2005/09/056.

[127] H. W. Kuhn, *The Hungarian method for the assignment problem*, Springer, Berlin, Heidelberg, ISBN 9783540682790 (2009), doi:10.1007/978-3-540-68279-0_2.

[128] J. Munkres, *Algorithms for the assignment and transportation problems*, J. Soc. Ind. Appl. Math. **5**, 32 (1957).

[129] R. Jonker and A. Volgenant, *A shortest augmenting path algorithm for dense and sparse linear assignment problems*, Computing **38**, 325 (1987), doi:10.1007/BF02278710.

[130] J. Chen, T. Han and B. Tweedie, *Electroweak splitting functions and high energy showering*, J. High Energy Phys. **11**, 093 (2017), doi:10.1007/JHEP11(2017)093.

[131] S. Catani, Yu. L. Dokshitzer, M. H. Seymour and B. R. Webber, *Longitudinally-invariant K_t-clustering algorithms for hadron-hadron collisions*, Nucl. Phys. B **406**, 187 (1993), doi:10.1016/0550-3213(93)90166-M.

[132] S. Höche and S. Prestel, *Triple collinear emissions in parton showers*, Phys. Rev. D **96**, 074017 (2017), doi:10.1103/PhysRevD.96.074017.

[133] S. Höche, F. Krauss and S. Prestel, *Implementing NLO DGLAP evolution in parton showers*, J. High Energy Phys. **10**, 093 (2017), doi:10.1007/JHEP10(2017)093.

[134] F. Dulat, S. Höche and S. Prestel, *Leading-color fully differential two-loop soft corrections to QCD dipole showers*, Phys. Rev. D **98**, 074013 (2018), doi:10.1103/PhysRevD.98.074013.

[135] S. Prestel and M. Spannowsky, *HYTREES: Combining matrix elements and parton shower for hypothesis testing*, Eur. Phys. J. C **79**, 546 (2019), doi:10.1140/epjc/s10052-019-7030-y.

[136] J. R. Andersen, C. Gütschow, A. Maier and S. Prestel, *A positive resampler for Monte Carlo events with negative weights*, Eur. Phys. J. C **80**, 1007 (2020), doi:10.1140/epjc/s10052-020-08548-w.

[137] L. Gellersen, S. Prestel and M. Spannowsky, *Coloring mixed QCD/QED evolution*, SciPost Phys. **13**, 034 (2022), doi:10.21468/SciPostPhys.13.2.034.

[138] S. Dittmaier, *A general approach to photon radiation off fermions*, Nucl. Phys. B **565**, 69 (2000), doi:10.1016/S0550-3213(99)00563-5.

[139] L. Gellersen, S. Höche and S. Prestel, *Disentangling soft and collinear effects in QCD parton showers*, Phys. Rev. D **105**, 114012 (2022), doi:10.1103/PhysRevD.105.114012.

[140] M. Schönherr, *An automated subtraction of NLO EW infrared divergences*, Eur. Phys. J. C **78**, 119 (2018), doi:10.1140/epjc/s10052-018-5600-z.

[141] F. Krauss, P. Petrov, M. Schönherr and M. Spannowsky, *Measuring collinear W emissions inside jets*, Phys. Rev. D **89**, 114006 (2014), doi:10.1103/PhysRevD.89.114006.

[142] M. Rubin, G. P. Salam and S. Sapeta, *Giant QCD K-factors beyond NLO*, J. High Energy Phys. **09**, 084 (2010), doi:10.1007/JHEP09(2010)084.

[143] A. Schälicke and F. Krauss, *Implementing the ME+PS merging algorithm*, J. High Energy Phys. **07**, 018 (2005), doi:10.1088/1126-6708/2005/07/018.

[144] J. R. Christiansen and S. Prestel, *Merging weak and QCD showers with matrix elements*, Eur. Phys. J. C **76**, 39 (2016), doi:10.1140/epjc/s10052-015-3871-1.

[145] M. H. Seymour, *Matrix-element corrections to parton shower algorithms*, Comput. Phys. Commun. **90**, 95 (1995), doi:10.1016/0010-4655(95)00064-M.

[146] M. H. Seymour, *A simple prescription for first-order corrections to quark scattering and annihilation processes*, Nucl. Phys. B **436**, 443 (1995), doi:10.1016/0550-3213(94)00554-R.

[147] J. André and T. Sjöstrand, *Matching of matrix elements and parton showers*, Phys. Rev. D **57**, 5767 (1998), doi:10.1103/PhysRevD.57.5767.

[148] N. Fischer and S. Prestel, *Combining states without scale hierarchies with ordered parton showers*, Eur. Phys. J. C **77**, 601 (2017), doi:10.1140/epjc/s10052-017-5160-7.

[149] S. Frixione and B. R. Webber, *Matching NLO QCD computations and parton shower simulations*, J. High Energy Phys. **06**, 029 (2002), doi:10.1088/1126-6708/2002/06/029.

[150] P. Nason, *A new method for combining NLO QCD with shower Monte Carlo algorithms*, J. High Energy Phys. **11**, 040 (2004), doi:10.1088/1126-6708/2004/11/040.

[151] S. Frixione, P. Nason and C. Oleari, *Matching NLO QCD computations with parton shower simulations: The POWHEG method*, J. High Energy Phys. **11**, 070 (2007), doi:10.1088/1126-6708/2007/11/070.

[152] S. Höche, F. Krauss, M. Schönherr and F. Siegert, *Automating the Powheg method in Sherpa*, J. High Energy Phys. **04**, 024 (2011), doi:10.1007/JHEP04(2011)024.

[153] J. Alwall et al., *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*, J. High Energy Phys. **07**, 079 (2014), doi:10.1007/JHEP07(2014)079.

[154] S. Höche, F. Krauss, M. Schönherr and F. Siegert, *A critical appraisal of NLO+PS matching methods*, J. High Energy Phys. **09**, 049 (2012), doi:10.1007/JHEP09(2012)049.

[155] S. Alioli, P. Nason, C. Oleari and E. Re, *A general framework for implementing NLO calculations in shower Monte Carlo programs: The POWHEG BOX*, J. High Energy Phys. **06**, 043 (2010), doi:10.1007/JHEP06(2010)043.

[156] M. L. Mangano, M. Moretti and R. Pittau, *Multijet matrix elements and shower evolution in hadronic collisions: $Wb\bar{b} + n$ jets as a case study*, Nucl. Phys. B **632**, 343 (2002), doi:10.1016/S0550-3213(02)00249-3.

[157] M. L. Mangano, M. Moretti, F. Piccinini and M. Treccani, *Matching matrix elements and shower evolution for top-pair production in hadronic collisions*, J. High Energy Phys. **01**, 013 (2007), doi:10.1088/1126-6708/2007/01/013.

[158] S. Catani, F. Krauss, B. R. Webber and R. Kuhn, *QCD matrix elements + parton showers*, J. High Energy Phys. **11**, 063 (2001), doi:10.1088/1126-6708/2001/11/063.

[159] K. Hamilton, P. Richardson and J. Tully, *A modified CKKW matrix element merging approach to angular-ordered parton showers*, J. High Energy Phys. **11**, 038 (2009), doi:10.1088/1126-6708/2009/11/038.

[160] L. Lönnblad, *Correcting the colour-dipole cascade model with fixed order matrix elements*, J. High Energy Phys. **05**, 046 (2002), doi:10.1088/1126-6708/2002/05/046.

[161] L. Lönnblad and S. Prestel, *Matching tree-level matrix elements with interleaved showers*, J. High Energy Phys. **03**, 019 (2012), doi:10.1007/JHEP03(2012)019.

[162] H. Brooks and C. T. Preuss, *Efficient multi-jet merging with the Vincia sector shower*, Comput. Phys. Commun. **264**, 107985 (2021), doi:10.1016/j.cpc.2021.107985.

[163] S. Höche, F. Krauss, S. Schumann and F. Siegert, *QCD matrix elements and truncated showers*, J. High Energy Phys. **05**, 053 (2009), doi:10.1088/1126-6708/2009/05/053.

[164] L. Lönnblad and S. Prestel, *Unitarising matrix element + parton shower merging*, J. High Energy Phys. **02**, 094 (2013), doi:10.1007/JHEP02(2013)094.

[165] S. Plätzer, *Controlling inclusive cross sections in parton shower + matrix element merging*, J. High Energy Phys. **08**, 114 (2013), doi:10.1007/JHEP08(2013)114.

[166] N. Lavesson and L. Lönnblad, *Extending CKKW-merging to one-loop matrix elements*, J. High Energy Phys. **12**, 070 (2008), doi:10.1088/1126-6708/2008/12/070.

[167] L. Lönnblad and S. Prestel, *Merging multi-leg NLO matrix elements with parton showers*, J. High Energy Phys. **03**, 166 (2013), doi:10.1007/JHEP03(2013)166.

[168] J. Bellm, S. Gieseke and S. Plätzer, *Merging NLO multi-jet calculations with improved unitarization*, Eur. Phys. J. C **78**, 244 (2018), doi:10.1140/epjc/s10052-018-5723-2.

[169] K. Hamilton and P. Nason, *Improving NLO-parton shower matched simulations with higher order matrix elements*, J. High Energy Phys. **06**, 039 (2010), doi:10.1007/JHEP06(2010)039.

[170] S. Höche, F. Krauss, M. Schönherr and F. Siegert, *NLO matrix elements and truncated showers*, J. High Energy Phys. **08**, 123 (2011), doi:10.1007/JHEP08(2011)123.

[171] T. Gehrmann, S. Höche, F. Krauss, M. Schönherr and F. Siegert, *NLO QCD matrix elements + parton showers in $e^+e^- \to$ hadrons*, J. High Energy Phys. **01**, 144 (2013), doi:10.1007/JHEP01(2013)144.

[172] S. Höche, F. Krauss, M. Schönherr and F. Siegert, *QCD matrix elements + parton showers. The NLO case*, J. High Energy Phys. **04**, 027 (2013), doi:10.1007/JHEP04(2013)027.

[173] R. Frederix and S. Frixione, *Merging meets matching in MC@NLO*, J. High Energy Phys. **12**, 061 (2012), doi:10.1007/JHEP12(2012)061.

[174] K. Hamilton, P. Nason and G. Zanderighi, *MINLO: Multi-scale improved NLO*, J. High Energy Phys. **10**, 155 (2012), doi:10.1007/JHEP10(2012)155.

[175] R. Frederix and K. Hamilton, *Extending the Minlo method*, J. High Energy Phys. **05**, 042 (2016), doi:10.1007/JHEP05(2016)042.

[176] M. Mangano, *Exploring theoretical systematics in the ME-to-shower MC merging for multijet process*, in *E-proceedings of matrix element/Monte Carlo tuning working group*, Fermilab (2002).

[177] R. Frederix, S. Frixione, S. Prestel and P. Torrielli, *On the reduction of negative weights in MC@NLO-type matching procedures*, J. High Energy Phys. **07**, 238 (2020), doi:10.1007/JHEP07(2020)238.

[178] T. Ježo and P. Nason, *On the treatment of resonances in next-to-leading order calculations matched to a parton shower*, J. High Energy Phys. **12**, 001 (2015), doi:10.1007/JHEP12(2015)065.

[179] S. F. Ravasio, T. Ježo, P. Nason and C. Oleari, *Addendum to: A theoretical study of top-mass measurements at the LHC using NLO+PS generators of increasing accuracy*, Eur. Phys. J. C **79**, 859 (2019), doi:10.1140/epjc/s10052-019-7336-9.

[180] S. Alioli, C. W. Bauer, C. Berggren, A. Hornig, F. J. Tackmann, C. K. Vermilion, J. R. Walsh and S. Zuberi, *Combining higher-order resummation with multiple NLO calculations and parton showers in GENEVA*, J. High Energy Phys. **09**, 120 (2013), doi:10.1007/JHEP09(2013)120.

[181] S. Alioli, C. W. Bauer, C. Berggren, F. J. Tackmann, J. R. Walsh and S. Zuberi, *Matching fully differential NNLO calculations and parton showers*, J. High Energy Phys. **06**, 089 (2014), doi:10.1007/JHEP06(2014)089.

[182] S. Alioli, C. W. Bauer, C. Berggren, F. J. Tackmann and J. R. Walsh, *Drell-Yan production at NNLL'+NNLO matched to parton showers*, Phys. Rev. D **92**, 094020 (2015), doi:10.1103/PhysRevD.92.094020.

[183] S. Alioli, C. W. Bauer, S. Guns and F. J. Tackmann, *Underlying-event sensitive observables in Drell-Yan production using GENEVA*, Eur. Phys. J. C **76**, 614 (2016), doi:10.1140/epjc/s10052-016-4458-1.

[184] S. Höche, S. Kuttimalai and Y. Li, *Hadronic final states in DIS at NNLO QCD with parton showers*, Phys. Rev. D **98**, 114013 (2018), doi:10.1103/PhysRevD.98.114013.

[185] L. Gellersen and S. Prestel, *Scale and scheme variations in unitarized NLO merging*, Phys. Rev. D **101**, 114007 (2020), doi:10.1103/PhysRevD.101.114007.

[186] H. T. Li and P. Skands, *A framework for second-order parton showers*, Phys. Lett. B **771**, 59 (2017), doi:10.1016/j.physletb.2017.05.011.

[187] J. M. Campbell, S. Höche, H. T. Li, C. T. Preuss and P. Skands, *Towards NNLO+PS matching with sector showers*, arXiv:2108.07133.

[188] J. J. Lopez-Villarejo and P. Skands, *Efficient matrix-element matching with sector showers*, J. High Energy Phys. **11**, 150 (2011), doi:10.1007/JHEP11(2011)150.

[189] S. Höche, S. Mrenna, S. Payne, C. T. Preuss and P. Skands, *A study of QCD radiation in VBF Higgs production with Vincia and Pythia*, SciPost Phys. **12**, 010 (2022), doi:10.21468/SciPostPhys.12.1.010.

[190] S. Prestel, *Matching N3LO QCD calculations to parton showers*, J. High Energy Phys. **11**, 041 (2021), doi:10.1007/JHEP11(2021)041.

[191] V. N. Gribov, *A Reggeon diagram technique*, Zh. Eksp. Teor. Fiz. **53**, 654 (1967).

[192] P. D. B. Collins, *An introduction to Regge theory and High energy physics*, Cambridge University Press, Cambridge, UK, ISBN 9780521212458 (1977), doi:10.1017/CBO9780511897603.

[193] J. R. Forshaw and D. A. Ross, *Quantum chromodynamics and the pomeron*, Cambridge University Press, Cambridge, UK, ISBN 9780511524387 (2011), doi:10.1017/CBO9780511524387.

[194] S. Donnachie, H. G. Dosch, O. Nachtmann and P. Landshoff, *Pomeron physics and QCD*, Cambridge University Press, Cambridge, UK, ISBN 9780511534935 (2002), doi:10.1017/CBO9780511534935.

[195] V. Barone and E. Predazzi, *High-energy particle diffraction*, Springer, Berlin, Heidelberg, ISBN 9783662047248 (2002), doi:10.1007/978-3-662-04724-8.

[196] G. Antchev et al., *First determination of the ρ parameter at $\sqrt{s} = 13$ TeV: Probing the existence of a colourless C-odd three-gluon compound state*, Eur. Phys. J. C **79**, 785 (2019), doi:10.1140/epjc/s10052-019-7223-4.

[197] M. Froissart, *Asymptotic behavior and subtractions in the Mandelstam representation*, Phys. Rev. **123**, 1053 (1961), doi:10.1103/PhysRev.123.1053.

[198] G. A. Schuler and T. Sjöstrand, *Hadronic diffractive cross sections and the rise of the total cross section*, Phys. Rev. D **49**, 2257 (1994), doi:10.1103/PhysRevD.49.2257.

[199] C. O. Rasmussen and T. Sjöstrand, *Models for total, elastic and diffractive cross sections*, Eur. Phys. J. C **78**, 461 (2018), doi:10.1140/epjc/s10052-018-5940-8.

[200] A. Donnachie and P. V. Landshoff, *Total cross sections*, Phys. Lett. B **296**, 227 (1992), doi:10.1016/0370-2693(92)90832-O.

[201] R. Ciesielski, *MBR Monte Carlo simulation in PYTHIA8*, Proc. Sci. **ICHEP2012**, 301 (2013), doi:10.22323/1.174.0301.

[202] R. B. Appleby, R. J. Barlow, J. G. Molson, M. Serluca and A. Toader, *The practical Pomeron for high energy proton collimation*, Eur. Phys. J. C **76**, 520 (2016), doi:10.1140/epjc/s10052-016-4363-7.

[203] K. A. Olive, *Review of particle physics*, Chin. Phys. C **40**, 100001 (2016), doi:10.1088/1674-1137/40/10/100001.

[204] A. Donnachie and P. V. Landshoff, *Dynamics of elastic scattering*, Nucl. Phys. B **267**, 690 (1986), doi:10.1016/0550-3213(86)90137-9.

[205] P. Aurenche, F. W. Bopp, A. Capella, J. Kwiecinski, M. Maire, J. Ranft and J. T. T. Van, *Multiparticle production in a two-component dual parton model*, Phys. Rev. D **45**, 92 (1992), doi:10.1103/PhysRevD.45.92.

[206] M. L. Good and W. D. Walker, *Diffraction dissociation of beam particles*, Phys. Rev. **120**, 1857 (1960), doi:10.1103/PhysRev.120.1857.

[207] G. A. Schuler and T. Sjöstrand, *A scenario for high-energy γγ interactions*, Z. Phys. C **73**, 677 (1997), doi:10.1007/s002880050359.

[208] E. M. Levin and L. L. Frankfurt, *The quark hypothesis and relations between cross-sections at high-energies*, J. Exp. Theor. Phys. Lett. **2**, 65 (1965).

[209] H. J. Lipkin, *Quarks for pedestrians*, Phys. Rep. **8**, 173 (1973), doi:10.1016/0370-1573(73)90002-1.

[210] S. Okubo, *φ-meson and unitary symmetry model*, Phys. Lett. **5**, 165 (1963), doi:10.1016/S0375-9601(63)92548-9.

[211] G. Zweig, *An SU_3 model for strong interaction symmetry and its breaking. Version 1*, CERN-TH-401 (1964).

[212] J. Iizuka, *A systematics and phenomenology of meson family*, Prog. Theor. Phys. Suppl. **37**, 21 (1966), doi:10.1143/PTPS.37.21.

[213] T. Sjöstrand and M. Utheim, *A framework for hadronic rescattering in pp collisions*, Eur. Phys. J. C **80**, 907 (2020), doi:10.1140/epjc/s10052-020-8399-3.

[214] M. Tanabashi et al., *Review of particle physics*, Phys. Rev. D **98**, 030001 (2018), doi:10.1103/PhysRevD.98.030001.

[215] S. Bass, *Microscopic models for ultrarelativistic heavy ion collisions*, Prog. Part. Nucl. Phys. **41**, 255 (1998), doi:10.1016/S0146-6410(98)00058-1.

[216] R. García-Martín, R. Kamiński, J. R. Peláez, J. R. de Elvira and F. J. Ynduráin, *Pion-pion scattering amplitude. IV. Improved analysis with once subtracted Roy-like equations up to 1100 MeV*, Phys. Rev. D **83**, 074004 (2011), doi:10.1103/PhysRevD.83.074004.

[217] J. R. Pelaez, A. Rodas and J. R. de Elvira, *Global parameterization of ππ scattering up to 2 GeV*, Eur. Phys. J. C **79**, 1008 (2019), doi:10.1140/epjc/s10052-019-7509-6.

[218] J. R. Pelaez and A. Rodas, *Pion-kaon scattering amplitude constrained with forward dispersion relations up to 1.6 GeV*, Phys. Rev. D **93**, 074025 (2016), doi:10.1103/PhysRevD.93.074025.

[219] L. Montanet et al., *Review of particle properties*, Phys. Rev. D **50**, 1173 (1994), doi:10.1103/PhysRevD.50.1173.

[220] P. Koch and C. B. Dover, $K^{\pm}$, $\bar{p}$ and Ω^{-} production in relativistic heavy ion collisions, Phys. Rev. C **40**, 145 (1989), doi:10.1103/PhysRevC.40.145.

[221] P. V. Landshoff and J. C. Polkinghorne, *Calorimeter triggers for hard collisions*, Phys. Rev. D **18**, 3344 (1978), doi:10.1103/PhysRevD.18.3344.

[222] C. Goebel, D. M. Scott and F. Halzen, *Double Drell-Yan annihilations in hadron collisions: Novel tests of the constituent picture*, Phys. Rev. D **22**, 2789 (1980), doi:10.1103/PhysRevD.22.2789.

[223] V. A. Abramovsky, V. N. Gribov and O. V. Kancheli, *Character of inclusive spectra and fluctuations produced in inelastic processes by multi-pomeron exchange*, Yad. Fiz. **18**, 595 (1973).

[224] T. Sjöstrand, *Multiple parton-parton interactions in hadronic events*, FERMILAB-Pub-85/119-T (1985).

[225] P. Bartalini and J. R. Gaunt, *Multiple parton interactions at the LHC*, World Scientific, Singapore, ISBN 9789813227750 (2017), doi:10.1142/10646.

[226] T. Sjöstrand, *The development of MPI modeling in Pythia*, World Scientific, Singapore, ISBN 9789813227767 (2018), doi:10.1142/9789813227767_0010.

[227] L. McLerran and R. Venugopalan, *Computing quark and gluon distribution functions for very large nuclei*, Phys. Rev. D **49**, 2233 (1994), doi:10.1103/PhysRevD.49.2233.

[228] F. Gelis, E. Iancu, J. Jalilian-Marian and R. Venugopalan, *The color glass condensate*, Annu. Rev. Nucl. Part. Sci. **60**, 463 (2010), doi:10.1146/annurev.nucl.010909.083629.

[229] R. D. Ball, V. Bertone, S. Carrazza, L. Del Debbio, S. Forte, A. Guffanti, N. P. Hartland and J. Rojo, *Parton distributions with QED corrections*, Nucl. Phys. B **877**, 290 (2013), doi:10.1016/j.nuclphysb.2013.10.010.

[230] L. Gribov, *Semihard processes in QCD*, Phys. Rep. **100**, 1 (1983), doi:10.1016/0370-1573(83)90022-4.

[231] A. H. Mueller and J. Qiu, *Gluon recombination and shadowing at small values of x*, Nucl. Phys. B **268**, 427 (1986), doi:10.1016/0550-3213(86)90164-1.

[232] R. J. Glauber, *High-energy collision theory*, in *Lectures in theoretical physics* **I**, Interscience, New York, US, 315 (1959).

[233] T. T. Chou and C. N. Yang, *Model of elastic high-energy scattering*, Phys. Rev. **170**, 1591 (1968), doi:10.1103/PhysRev.170.1591.

[234] C. Bourrely, J. Soffer and T. T. Wu, *Impact-picture expectations for very high-energy elastic pp and p$\bar{p}$ scattering*, Nucl. Phys. B **247**, 15 (1984), doi:10.1016/0550-3213(84)90369-9.

[235] P. L'Heureux, B. Margolis and P. Valin, *Quark-gluon model for diffraction at high energies*, Phys. Rev. D **32**, 1681 (1985), doi:10.1103/PhysRevD.32.1681.

[236] L. Frankfurt, M. Strikman and C. Weiss, *Small-χ physics: From HERA to LHC and beyond*, Annu. Rev. Nucl. Part. Sci. **55**, 403 (2005), doi:10.1146/annurev.nucl.53.041002.110615.

[237] R. Corke and T. Sjöstrand, *Multiparton interactions with an x-dependent proton size*, J. High Energy Phys. **05**, 009 (2011), doi:10.1007/JHEP05(2011)009.

[238] J. R. Gaunt and W. J. Stirling, *Double parton distributions incorporating perturbative QCD evolution and momentum and quark number sum rules*, J. High Energy Phys. **03**, 005 (2010), doi:10.1007/JHEP03(2010)005.

[239] K. Konishi, A. Ukawa and G. Veneziano, *Jet calculus: A simple algorithm for resolving QCD jets*, Nucl. Phys. B **157**, 45 (1979), doi:10.1016/0550-3213(79)90053-1.

[240] R. Kirschner, *Generalized Lipatov-Altarelli-Parisi equations and jet calculus rules*, Phys. Lett. B **84**, 266 (1979), doi:10.1016/0370-2693(79)90300-9.

[241] N. Paver and D. Treleani, *Multiquark scattering and large-p_T jet production in hadronic collisions*, Nuov. Cim. A **70**, 215 (1982), doi:10.1007/BF02814035.

[242] N. Paver and D. Treleani, *Multiple parton interactions and multijet events at collider and tevatron energies*, Phys. Lett. B **146**, 252 (1984), doi:10.1016/0370-2693(84)91029-3.

[243] N. Paver and D. Treleani, *Multiple parton processes in the TeV region*, Z. Phys. C **28**, 187 (1985), doi:10.1007/BF01575722.

[244] R. Corke and T. Sjöstrand, *Multiparton interactions and rescattering*, J. High Energy Phys. **01**, 035 (2010), doi:10.1007/JHEP01(2010)035.

[245] E. Avsar, G. Gustafson and L. Lönnblad, *Small-x dipole evolution beyond the large-N(c) limit*, J. High Energy Phys. **01**, 012 (2007), doi:10.1088/1126-6708/2007/01/012.

[246] C. Bierlich, G. Gustafson, L. Lönnblad and A. Tarasov, *Effects of overlapping strings in pp collisions*, J. High Energy Phys. **03**, 148 (2015), doi:10.1007/JHEP03(2015)148.

[247] C. Bierlich and C. O. Rasmussen, *Dipole evolution: Perspectives for collectivity and γ^*A collisions*, J. High Energy Phys. **10**, 026 (2019), doi:10.1007/JHEP10(2019)026.

[248] T. Sjöstrand and P. Skands, *Multiple interactions and the structure of beam remnants*, J. High Energy Phys. **03**, 053 (2004), doi:10.1088/1126-6708/2004/03/053.

[249] J. R. Christiansen and P. Z. Skands, *String formation beyond leading colour*, J. High Energy Phys. **08**, 003 (2015), doi:10.1007/JHEP08(2015)003.

[250] G. Ingelman and P. E. Schlein, *Jet structure in high mass diffractive scattering*, Phys. Lett. B **152**, 256 (1985), doi:10.1016/0370-2693(85)91181-5.

[251] S. Navin, *Diffraction in Pythia*, arXiv:1005.3894.

[252] T. Affolder et al., *Diffractive dijets with a leading antiproton in $\bar{p}p$ collisions at $\sqrt{s} = 1800$ GeV*, Phys. Rev. Lett. **84**, 5043 (2000), doi:10.1103/PhysRevLett.84.5043.

[253] G. Aad et al., *Dijet production in $\sqrt{s} = 7$ TeV pp collisions with large rapidity gaps at the ATLAS experiment*, Phys. Lett. B **754**, 214 (2016), doi:10.1016/j.physletb.2016.01.028.

[254] C. O. Rasmussen and T. Sjöstrand, *Hard diffraction with dynamic gap survival*, J. High Energy Phys. **02**, 142 (2016), doi:10.1007/JHEP02(2016)142.

[255] CMS Collaboration, *Measurement of dijet production with a leading proton in proton-proton collisions at $\sqrt{s} = 8$ TeV*, (2018).

[256] R. Kleiss et al., *Monte Carlos for electroweak physics*, in *LEP Physics Workshop* (1989).

[257] D. Schulte, *Beam-beam simulations with GUINEA-PIG* (1999).

[258] T. Ohl, *CIRCE Version 1.02: Beam spectra for simulating linear collider physics*, Comput. Phys. Commun. **101**, 269 (1997), doi:10.1016/S0010-4655(96)00167-1.

[259] R. P. Feynman, *Very high-energy collisions of hadrons*, Phys. Rev. Lett. **23**, 1415 (1969), doi:10.1103/PhysRevLett.23.1415.

[260] J. D. Bjorken and E. A. Paschos, *Inelastic electron proton and gamma proton scattering, and the structure of the nucleon*, Phys. Rev. **185**, 1975 (1969), doi:10.1103/PhysRev.185.1975.

[261] J. C. Collins, D. E. Soper and G. Sterman, *Transverse momentum distribution in Drell-Yan pair and W and Z boson production*, Nucl. Phys. B **250**, 199 (1985), doi:10.1016/0550-3213(85)90479-1.

[262] R. Angeles-Martinez et al., *Transverse momentum dependent (TMD) parton distribution functions: Status and prospects*, Acta Phys. Pol. B **46**, 2501 (2015), doi:10.5506/APhysPolB.46.2501.

[263] A. Kerbizi and L. Lönnblad, *StringSpinner - Adding spin to the PYTHIA string fragmentation*, Comput. Phys. Commun. **272**, 108234 (2022), doi:10.1016/j.cpc.2021.108234.

[264] F. Cornet, P. Jankowski, M. Krawczyk and A. Lorca, *New 5-flavor LO analysis and parametrization of parton distributions in the real photon*, Phys. Rev. D **68**, 014010 (2003), doi:10.1103/PhysRevD.68.014010.

[265] C. F. v. Weizsäcker, *Ausstrahlung bei Stößen sehr schneller Elektronen*, Z. Phys. **88**, 612 (1934), doi:10.1007/BF01333110.

[266] E. J. Williams, *Nature of the high energy particles of penetrating radiation and status of ionization and radiation formulae*, Phys. Rev. **45**, 729 (1934), doi:10.1103/PhysRev.45.729.

[267] G. A. Schuler and T. Sjöstrand, *Parton distributions of the virtual photon*, Phys. Lett. B **376**, 193 (1996), doi:10.1016/0370-2693(96)00265-1.

[268] I. Helenius, *Simulations of photo-nuclear dijets with PYTHIA 8 and their sensitivity to nuclear PDFs*, Proc. Sci. **DIS2018**, 113 (2018), doi:10.22323/1.316.0113.

[269] S. Chekanov et al., *Diffractive photoproduction of dijets in ep collisions at HERA*, Eur. Phys. J. C **55**, 177 (2008), doi:10.1140/epjc/s10052-008-0598-2.

[270] V. Andreev et al., *Diffractive dijet production with a leading proton in ep collisions at HERA*, J. High Energy Phys. **05**, 056 (2015), doi:10.1007/JHEP05(2015)056.

[271] I. Helenius and C. O. Rasmussen, *Hard diffraction in photoproduction with Pythia 8*, Eur. Phys. J. C **79**, 413 (2019), doi:10.1140/epjc/s10052-019-6914-1.

[272] T. H. Bauer, R. D. Spital, D. R. Yennie and F. M. Pipkin, *The hadronic properties of the photon in high-energy interactions*, Rev. Mod. Phys. **50**, 261 (1978), doi:10.1103/RevModPhys.50.261.

[273] M. Derrick et al., *Measurement of elastic ρ^0 photoproduction at HERA*, Z. Phys. C **69**, 39 (1995), doi:10.1007/s002880050004.

[274] M. Derrick et al., *Measurement of elastic ω photoproduction at HERA*, Z. Phys. C **73**, 73 (1996), doi:10.1007/s002880050297.

[275] M. Derrick et al., *Measurement of elastic φ photoproduction at HERA*, Phys. Lett. B **377**, 259 (1996), doi:10.1016/0370-2693(96)00172-4.

[276] J. Breitweg et al., *Measurement of elastic J/ψ photoproduction at HERA*, Z. Phys. C **75**, 215 (1997), doi:10.1007/s002880050464.

[277] S. Chekanov et al., *Exclusive photoproduction of J/ψ mesons at HERA*, Eur. Phys. J. C **24**, 345 (2002), doi:10.1007/s10052-002-0953-7.

[278] A. Aktas et al., *Elastic J/ψ production at HERA*, Eur. Phys. J. C **46**, 585 (2006), doi:10.1140/epjc/s2006-02519-5.

[279] C. Alexa et al., *Elastic and proton-dissociative photoproduction of J/ψ mesons at HERA*, Eur. Phys. J. C **73**, 2466 (2013), doi:10.1140/epjc/s10052-013-2466-y.

[280] M. Drees and D. Zeppenfeld, *Production of supersymmetric particles in elastic* ep *collisions*, Phys. Rev. D **39**, 2536 (1989), doi:10.1103/PhysRevD.39.2536.

[281] V. M. Budnev, I. F. Ginzburg, G. V. Meledin and V. G. Serbo, *The two-photon particle production mechanism. Physical problems. Applications. Equivalent photon approximation*, Phys. Rep. **15**, 181 (1975), doi:10.1016/0370-1573(75)90009-5.

[282] J. D. Jackson, *Classical electrodynamics*, Wiley, Hoboken, New Jersey, US, ISBN 9780471309321 (1998).

[283] C. Bierlich, G. Gustafson, L. Lönnblad and H. Shah, *The Angantyr model for heavy-ion collisions in Pythia8*, J. High Energy Phys. **10**, 134 (2018), doi:10.1007/JHEP10(2018)134.

[284] B. Andersson, G. Gustafson and B. Nilsson-Almqvist, *A model for low-p_T hadronic reactions with generalizations to hadron-nucleus and nucleus-nucleus collisions*, Nucl. Phys. B **281**, 289 (1987), doi:10.1016/0550-3213(87)90257-4.

[285] A. Białłas, M. Bleszyński and W. Czyż, *Multiplicity distributions in nucleus-nucleus collisions at high energies*, Nucl. Phys. B **111**, 461 (1976), doi:10.1016/0550-3213(76)90329-1.

[286] C. Bierlich, G. Gustafson and L. Lönnblad, *Diffractive and non-diffractive wounded nucleons and final states in pA collisions*, J. High Energy Phys. **10**, 139 (2016), doi:10.1007/JHEP10(2016)139.

[287] W. Broniowski, M. Rybczyński and P. Bożek, *GLISSANDO: GLauber Initial-State Simulation AND mOre...*, Comput. Phys. Commun. **180**, 69 (2009), doi:10.1016/j.cpc.2008.07.016.

[288] M. Rybczyński, G. Stefanek, W. Broniowski and P. Bożek, *GLISSANDO 2: GLauber Initial-State Simulation AND mOre..., ver. 2*, Comput. Phys. Commun. **185**, 1759 (2014), doi:10.1016/j.cpc.2014.02.016.

[289] B. Andersson, G. Gustafson and B. Söderberg, *A general model for jet fragmentation*, Z. Phys. C **20**, 317 (1983), doi:10.1007/BF01407824.

[290] T. Sjöstrand, *Jet fragmentation of multiparton configurations in a string framework*, Nucl. Phys. B **248**, 469 (1984), doi:10.1016/0550-3213(84)90607-2.

[291] B. Andersson, G. Gustafson and B. Söderberg, *A probability measure on parton and string states*, Nucl. Phys. B **264**, 29 (1986), doi:10.1016/0550-3213(86)90471-2.

[292] M. G. Bowler, e⁺e⁻ *production of heavy quarks in the string model*, Z. Phys. C **11**, 169 (1981), doi:10.1007/BF01574001.

[293] X. Artru and G. Mennessier, *String model and multiproduction*, Nucl. Phys. B **70**, 93 (1974), doi:10.1016/0550-3213(74)90360-5.

[294] D. A. Morris, *Heavy-quark fragmentation functions in a simple string model*, Nucl. Phys. B **313**, 634 (1989), doi:10.1016/0550-3213(89)90399-4.

[295] B. Andersson, G. Gustafson, G. Ingelman and T. Sjöstrand, *Parton fragmentation and string dynamics*, Phys. Rep. **97**, 31 (1983), doi:10.1016/0370-1573(83)90080-7.

[296] C. Bierlich, S. Chakraborty, G. Gustafson and L. Lönnblad, *Hyperfine splitting effects in string hadronization*, Eur. Phys. J. C **82**, 228 (2022), doi:10.1140/epjc/s10052-022-10172-9.

[297] B. Andersson, G. Gustafson and T. Sjöstrand, *A model for baryon production in quark and gluon jets*, Nucl. Phys. B **197**, 45 (1982), doi:10.1016/0550-3213(82)90153-5.

[298] A. Casher, H. Neuberger and S. Nussinov, *Chromoelectric-flux-tube model of particle production*, Phys. Rev. D **20**, 179 (1979), doi:10.1103/PhysRevD.20.179.

[299] B. Andersson, G. Gustafson and T. Sjöstrand, *Baryon production in jet fragmentation and Υ-decay*, Phys. Scr. **32**, 574 (1985), doi:10.1088/0031-8949/32/6/003.

[300] T. Sjöstrand, *The merging of jets*, Phys. Lett. B **142**, 420 (1984), doi:10.1016/0370-2693(84)91354-6.

[301] S. Ferreres-Solé and T. Sjöstrand, *The space-time structure of hadronization in the Lund model*, Eur. Phys. J. C **78**, 983 (2018), doi:10.1140/epjc/s10052-018-6459-8.

[302] T. Sjöstrand and P. Z. Skands, *Baryon number violation and string topologies*, Nucl. Phys. B **659**, 243 (2003), doi:10.1016/S0550-3213(03)00193-7.

[303] E. Norrbin and T. Sjöstrand, *Production mechanisms of charm hadrons in the string model*, Phys. Lett. B **442**, 407 (1998), doi:10.1016/S0370-2693(98)01244-1.

[304] E. Boos et al., *Generic user process interface for event generators*, arXiv:hep-ph/0109068.

[305] J. Alwall et al., *A standard format for Les Houches event files*, Comput. Phys. Commun. **176**, 300 (2007), doi:10.1016/j.cpc.2006.11.010.

[306] H. Fritzsch, *Producing heavy quark flavors in hadronic collisions: A test of quantum chromodynamics*, Phys. Lett. B **67**, 217 (1977), doi:10.1016/0370-2693(77)90108-3.

[307] A. Ali, J. G. Körner, G. Kramer and J. Willrodt, *Nonleptonic weak decays of bottom mesons*, Z. Phys. C **1**, 269 (1979), doi:10.1007/BF01440227.

[308] H. Fritzsch, *How to discover the B mesons*, Phys. Lett. B **86**, 343 (1979), doi:10.1016/0370-2693(79)90853-0.

[309] C. Albajar et al., *A study of the general characteristics of $p\bar{p}$ collisions at $\sqrt{s} = 0.2$ TeV to 0.9 TeV*, Nucl. Phys. B **335**, 261 (1990), doi:10.1016/0550-3213(90)90493-W.

[310] T. Sjöstrand and V. A. Khoze, *On colour rearrangement in hadronic* W$^+$W$^-$ *evemts*, Z. Phys. C **62**, 281 (1994), doi:10.1007/BF01560244.

[311] S. Schael et al., *Electroweak measurements in electron-positron collisions at W-boson-pair energies at LEP*, Phys. Rep. **532**, 119 (2013), doi:10.1016/j.physrep.2013.07.004.

[312] Tevatron Electroweak Working Group, *Combination of CDF and D0 results on the mass of the top quark using up* 9.7 fb^{-1} *at the Tevatron*, arXiv:1608.01881.

[313] M. Aaboud et al., *Measurement of the top quark mass in the* $t\bar{t} \rightarrow$ *lepton+jets channel from* $\sqrt{s} = 8$ *TeV ATLAS data and combination with previous results*, Eur. Phys. J. C **79**, 290 (2019), doi:10.1140/epjc/s10052-019-6757-9.

[314] A. M. Sirunyan et al., *Measurement of the top quark mass in the all-jets final state at* $\sqrt{s} = 13$ *TeV and combination with the lepton+jets channel*, Eur. Phys. J. C **79**, 313 (2019), doi:10.1140/epjc/s10052-019-6788-2.

[315] M. Sandhoff and P. Z. Skands, *Colour annealing - A toy model of colour reconnections*, in *4th Les Houches Workshop on Physics at TeV Colliders*, FERMILAB-CONF-05-518-T (2005).

[316] P. Skands and D. Wicke, *Non-perturbative QCD effects and the top mass at the Tevatron*, Eur. Phys. J. C **52**, 133 (2007), doi:10.1140/epjc/s10052-007-0352-1.

[317] P. Z. Skands, *Tuning Monte Carlo generators: The Perugia tunes*, Phys. Rev. D **82**, 074018 (2010), doi:10.1103/PhysRevD.82.074018.

[318] S. Argyropoulos and T. Sjöstrand, *Effects of color reconnection on* $t\bar{t}$ *final states at the LHC*, J. High Energy Phys. **11**, 043 (2014), doi:10.1007/JHEP11(2014)043.

[319] L. Lönnblad, *Reconnecting coloured dipoles*, Z. Phys. C **70**, 107 (1996), doi:10.1007/s002880050087.

[320] T. Sjöstrand and V. A. Khoze, *Does the* W *reconstruction survive QCD effects?*, Phys. Rev. Lett. **72**, 28 (1994), doi:10.1103/PhysRevLett.72.28.

[321] J. R. Christiansen and T. Sjöstrand, *Color reconnection at future* $e^+ e^-$ *colliders*, Eur. Phys. J. C **75**, 441 (2015), doi:10.1140/epjc/s10052-015-3674-4.

[322] A. Chodos, R. L. Jaffe, K. Johnson, C. B. Thorn and V. F. Weisskopf, *New extended model of hadrons*, Phys. Rev. D **9**, 3471 (1974), doi:10.1103/PhysRevD.9.3471.

[323] D. Wicke and P. Z. Skands, *Non-perturbative QCD effects and the top mass at the Tevatron*, Nuov. Cim. B **123**, 1 (2008), doi:10.1393/ncb/i2009-10749-y.

[324] G. Gustafson and J. Häkkinen, *Colour interference and confinement effects in W-pair production*, Z. Phys. C **64**, 659 (1994), doi:10.1007/BF01957774.

[325] W. Buchmüller and A. Hebecker, *A parton model for diffractive processes in deep inelastic scattering*, Phys. Lett. B **355**, 573 (1995), doi:10.1016/0370-2693(95)00721-V.

[326] A. Edin, G. Ingelman and J. Rathsman, *Soft colour interactions as the origin of rapidity gaps in DIS*, Phys. Lett. B **366**, 371 (1996), doi:10.1016/0370-2693(95)01391-1.

[327] A. Edin, G. Ingelman and J. Rathsman, *Unified description of rapidity gaps and energy flows in DIS final states*, Z. Phys. C **75**, 57 (1997), doi:10.1007/s002880050447.

[328] R. Pasechnik, R. Enberg and G. Ingelman, *Diffractive deep inelastic scattering from multiple soft gluon exchange in QCD*, Phys. Lett. B **695**, 189 (2011), doi:10.1016/j.physletb.2010.11.010.

[329] R. Enberg, G. Ingelman and N. Tîmneanu, *Soft color interactions and diffractive hard scattering at the Fermilab Tevatron*, Phys. Rev. D **64**, 114015 (2001), doi:10.1103/PhysRevD.64.114015.

[330] A. Edin, G. Ingelman and J. Rathsman, *Quarkonium production at the Fermilab Tevatron through soft color interactions*, Phys. Rev. D **56**, 7317 (1997), doi:10.1103/PhysRevD.56.7317.

[331] J. Rathsman, *A generalised area law for hadronic string reinteractions*, Phys. Lett. B **452**, 364 (1999), doi:10.1016/S0370-2693(99)00291-9.

[332] S. Gieseke, C. Röhr and A. Siódmok, *Colour reconnections in Herwig++*, Eur. Phys. J. C **72**, 2225 (2012), doi:10.1140/epjc/s10052-012-2225-5.

[333] S. Gieseke, P. Kirchgaeßer, S. Plätzer and A. Siodmok, *Colour reconnection from soft gluon evolution*, J. High Energy Phys. **11**, 149 (2018), doi:10.1007/JHEP11(2018)149.

[334] J. Bellm, C. B. Duncan, S. Gieseke, M. Myska and A. Siódmok, *Spacetime colour reconnection in Herwig 7*, Eur. Phys. J. C **79**, 1003 (2019), doi:10.1140/epjc/s10052-019-7533-6.

[335] V. A. Khoze, F. Krauss, A. D. Martin, M. G. Ryskin and K. C. Zapp, *Diffraction and correlations at the LHC: Definitions and observables*, Eur. Phys. J. C **69**, 85 (2010), doi:10.1140/epjc/s10052-010-1392-5.

[336] K. Zapp, *JEWEL 2.0.0: Directions for use*, Eur. Phys. J. C **74**, 2762 (2014), doi:10.1140/epjc/s10052-014-2762-1.

[337] S. Cao et al., *Multistage Monte Carlo simulation of jet modification in a static medium*, Phys. Rev. C **96**, 024909 (2017), doi:10.1103/PhysRevC.96.024909.

[338] K. Werner, *Core-corona separation in ultrarelativistic heavy ion collisions*, Phys. Rev. Lett. **98**, 152301 (2007), doi:10.1103/PhysRevLett.98.152301.

[339] T. Pierog, Iu. Karpenko, J. M. Katzy, E. Yatsenko and K. Werner, *EPOS LHC: Test of collective hadronization with data measured at the CERN Large Hadron Collider*, Phys. Rev. C **92**, 034906 (2015), doi:10.1103/PhysRevC.92.034906.

[340] Y. Kanakubo, Y. Tachibana and T. Hirano, *Unified description of hadron yield ratios from dynamical core-corona initialization*, Phys. Rev. C **101**, 024912 (2020), doi:10.1103/PhysRevC.101.024912.

[341] V. Khachatryan et al., *Observation of long-range, near-side angular correlations in proton-proton collisions at the LHC*, J. High Energy Phys. **09**, 091 (2010), doi:10.1007/JHEP09(2010)091.

[342] V. Khachatryan et al., *Strange particle production in pp collisions at $\sqrt{s} = 0.9$ and 7 TeV*, J. High Energy Phys. **05**, 064 (2011), doi:10.1007/JHEP05(2011)064.

[343] R. Aaij et al., *Measurement of prompt hadron production ratios in pp collisions at $\sqrt{s} = 0.9$ and 7 TeV*, Eur. Phys. J. C **72**, 2168 (2012), doi:10.1140/epjc/s10052-012-2168-x.

[344] K. Aamodt et al., *Strange particle production in proton-proton collisions at $\sqrt{s} = 0.9$ TeV with ALICE at the LHC*, Eur. Phys. J. C **71**, 1594 (2011), doi:10.1140/epjc/s10052-011-1594-5.

[345] B. Abelev et al., *Multi-strange baryon production in pp collisions at $\sqrt{s} = 7$ TeV with ALICE*, Phys. Lett. B **712**, 309 (2012), doi:10.1016/j.physletb.2012.05.011.

[346] J. Adam et al., *Enhanced production of multi-strange hadrons in high-multiplicity proton-proton collisions*, Nat. Phys. **13**, 535 (2017), doi:10.1038/nphys4111.

[347] A. O. Velasquez, P. Christiansen, E. C. Flores, I. A. M. Cervantes and G. Paić, *Color reconnection and flowlike patterns in pp collisions*, Phys. Rev. Lett. **111**, 042001 (2013), doi:10.1103/PhysRevLett.111.042001.

[348] C. Bierlich and J. R. Christiansen, *Effects of color reconnection on hadron flavor observables*, Phys. Rev. D **92**, 094010 (2015), doi:10.1103/PhysRevD.92.094010.

[349] C. Bierlich, *Microscopic collectivity: The ridge and strangeness enhancement from string-string interactions*, Nucl. Phys. A **982**, 499 (2019), doi:10.1016/j.nuclphysa.2018.07.015.

[350] V. A. Abramovsky, E. V. Gedalin, E. G. Gurvich and O. V. Kancheli, *Long range azimuthal correlations in multiple production processes at high-energies*, J. Exp. Theor. Phys. Lett. **47**, 337 (1988).

[351] C. Bierlich, *Soft modifications to jet fragmentation in high energy proton-proton collisions*, Phys. Lett. B **795**, 194 (2019), doi:10.1016/j.physletb.2019.06.018.

[352] C. Bierlich, S. Chakraborty, G. Gustafson and L. Lönnblad, *Setting the string shoving picture in a new frame*, J. High Energy Phys. **03**, 270 (2021), doi:10.1007/JHEP03(2021)270.

[353] P. Cea, L. Cosmai, F. Cuteri and A. Papa, *Flux tubes in the $SU(3)$ vacuum: London penetration depth and coherence length*, Phys. Rev. D **89**, 094505 (2014), doi:10.1103/PhysRevD.89.094505.

[354] T. S. Biro, H. B. Nielsen and J. Knoll, *Colour rope model for extreme relativistic heavy ion collisions*, Nucl. Phys. B **245**, 449 (1984), doi:10.1016/0550-3213(84)90441-3.

[355] G. S. Bali, *Casimir scaling of $SU(3)$ static potentials*, Phys. Rev. D **62**, 114503 (2000), doi:10.1103/PhysRevD.62.114503.

[356] S. Jeon and R. Venugopalan, *Random walks of partons in $SU(N_c)$ and classical representations of color charges in QCD at small x*, Phys. Rev. D **70**, 105012 (2004), doi:10.1103/PhysRevD.70.105012.

[357] P. Skands, S. Carrazza and J. Rojo, *Tuning PYTHIA 8.1: The Monash 2013 tune*, Eur. Phys. J. C **74**, 3024 (2014), doi:10.1140/epjc/s10052-014-3024-y.

[358] N. Fischer and T. Sjöstrand, *Thermodynamical string fragmentation*, J. High Energy Phys. **01**, 140 (2017), doi:10.1007/JHEP01(2017)140.

[359] A. Bialas, *Fluctuations of the string tension and transverse mass distribution*, Phys. Lett. B **466**, 301 (1999), doi:10.1016/S0370-2693(99)01159-4.

[360] C. Bierlich, T. Sjöstrand and M. Utheim, *Hadronic rescattering in pA and AA collisions*, Eur. Phys. J. A **57**, 227 (2021), doi:10.1140/epja/s10050-021-00543-3.

[361] R. H. Brown and R. Q. Twiss, *A test of a new type of stellar interferometer on Sirius*, Nature **178**, 1046 (1956), doi:10.1038/1781046a0.

[362] N. Neumeister et al., *Higher order Bose-Einstein correlations in $p\overline{p}$ collisions at $\sqrt{s} = 630$ and 900 GeV*, Phys. Lett. B **275**, 186 (1992), doi:10.1016/0370-2693(92)90874-4.

[363] G. Aad et al., *Two-particle Bose-Einstein correlations in pp collisions at $\sqrt{s} = 0.9$ and 7 TeV measured with the ATLAS detector*, Eur. Phys. J. C **75**, 466 (2015), doi:10.1140/epjc/s10052-015-3644-x.

[364] V. Khachatryan et al., *Measurement of Bose-Einstein correlations in pp collisions at $\sqrt{s} = 0.9$ and 7 TeV*, J. High Energy Phys. **05**, 029 (2011), doi:10.1007/JHEP05(2011)029.

[365] R. Aaij et al., *Bose-Einstein correlations of same-sign charged pions in the forward region in pp collisions at $\sqrt{s} = 7$ TeV*, J. High Energy Phys. **12**, 025 (2017), doi:10.1007/JHEP12(2017)025.

[366] P. D. Acton et al., *A study of Bose-Einstein correlations in e^+e^- annihilations at LEP*, Phys. Lett. B **267**, 143 (1991), doi:10.1016/0370-2693(91)90540-7.

[367] P. D. Acton et al., *A study of $K_s^0 K_s^0$ Bose-Einstein correlations in hadronic Z^0 decays*, Phys. Lett. B **298**, 456 (1993), doi:10.1016/0370-2693(93)91851-D.

[368] D. Decamp et al., *A study of Bose-Einstein correlations in e^+e^- annihilation at 91 GeV*, Z. Phys. C **54**, 75 (1992), doi:10.1007/BF01881709.

[369] P. Abreu et al., *Interference of neutral kaons in the hadronic decays of the Z^0*, Phys. Lett. B **323**, 242 (1994), doi:10.1016/0370-2693(94)90298-4.

[370] L. Lönnblad and T. Sjöstrand, *Bose-Einstein effects and W mass determinations*, Phys. Lett. B **351**, 293 (1995), doi:10.1016/0370-2693(95)00393-Y.

[371] M. Gyulassy, S. K. Kauffmann and L. W. Wilson, *Pion interferometry of nuclear collisions. I. Theory*, Phys. Rev. C **20**, 2267 (1979), doi:10.1103/PhysRevC.20.2267.

[372] L. Lönnblad and T. Sjöstrand, *Modeling Bose-Einstein correlations at LEP-2*, Eur. Phys. J. C **2**, 165 (1998), doi:10.1007/s100520050131.

[373] F. Donato, N. Fornengo and P. Salati, *Antideuterons as a signature of supersymmetric dark matter*, Phys. Rev. D **62**, 043003 (2000), doi:10.1103/PhysRevD.62.043003.

[374] A. Andronic, P. Braun-Munzinger, J. Stachel and H. Stöcker, *Production of light nuclei, hypernuclei and their antiparticles in relativistic nuclear collisions*, Phys. Lett. B **697**, 203 (2011), doi:10.1016/j.physletb.2011.01.053.

[375] A. Schwarzschild and Č. Zupančič, *Production of tritons, deuterons, nucleons, and mesons by 30-GeV protons on Al, Be, and Fe targets*, Phys. Rev. **129**, 854 (1963), doi:10.1103/PhysRev.129.854.

[376] J. I. Kapusta, *Mechanisms for deuteron production in relativistic nuclear collisions*, Phys. Rev. C **21**, 1301 (1980), doi:10.1103/PhysRevC.21.1301.

[377] L. A. Dal and A. R. Raklev, *Alternative formation model for antideuterons from dark matter*, Phys. Rev. D **91**, 123536 (2015), doi:10.1103/PhysRevD.91.123536.

[378] J. Beringer et al., *Review of particle physics*, Phys. Rev. D **86**, 010001 (2012), doi:10.1103/PhysRevD.86.010001.

[379] A. De Rújula, H. Georgi and S. L. Glashow, *Hadron masses in a gauge theory*, Phys. Rev. D **12**, 147 (1975), doi:10.1103/PhysRevD.12.147.

[380] D. J. Herndon, P. Söding and R. J. Cashmore, *Generalized isobar model formalism*, Phys. Rev. D **11**, 3165 (1975), doi:10.1103/PhysRevD.11.3165.

[381] M. Ablikim et al., *Dalitz plot analysis of the decay $\omega \to \pi^+\pi^-\pi^0$*, Phys. Rev. D **98**, 112007 (2018), doi:10.1103/PhysRevD.98.112007.

[382] S. Rudaz, *Anomalies, vector mesons and the $\omega \to 3\pi$ contact term*, Phys. Lett. B **145**, 281 (1984), doi:10.1016/0370-2693(84)90355-1.

[383] S. Jadach, Z. Wąs, R. Decker and J. H. Kühn, *The τ decay library TAUOLA, version 2.4*, Comput. Phys. Commun. **76**, 361 (1993), doi:10.1016/0010-4655(93)90061-G.

[384] H. Murayama, I. Watanabe and K. Hagiwara, *HELAS: HELicity Amplitude Subroutines for Feynman diagram evaluations*, KEK Report 91-11 (1992).

[385] J. H. Kühn and A. Santamaria, *τ decays to pions*, Z. Phys. C **48**, 445 (1990), doi:10.1007/BF01572024.

[386] M. Finkemeier and E. Mirkes, *The scalar contribution to $\tau \to K\pi\nu_\tau$*, Z. Phys. C **72**, 619 (1996), doi:10.1007/s002880050284.

[387] D. M. Asner et al., *Hadronic structure in the decay $\tau^- \to \nu_\tau \pi^-\pi^0\pi^0$ and the sign of the tau-neutrino helicity*, Phys. Rev. D **61**, 012002 (1999), doi:10.1103/PhysRevD.61.012002.

[388] M. Finkemeier and E. Mirkes, *Tau decays into kaons*, Z. Phys. C **69**, 243 (1995), doi:10.1007/s002880050024.

[389] R. Decker, E. Mirkes, R. Sauer and Z. Was, *Tau decays into three pseudoscalar mesons*, Z. Phys. C **58**, 445 (1993), doi:10.1007/BF01557702.

[390] A. E. Bondar, S. I. Eidelman, A. I. Milstein, T. Pierzchała, N. I. Root, Z. Wąs and M. Worek, *Novosibirsk hadronic currents for $\tau \to 4\pi$ channels of τ decay library TAUOLA*, Comput. Phys. Commun. **146**, 139 (2002), doi:10.1016/S0010-4655(02)00262-X.

[391] J. H. Kuhn and Z. Was, *τ decays to five mesons in TAUOLA*, Acta Phys. Pol. B **39**, 147 (2008).

[392] P. Ilten, *Electroweak and Higgs measurements using tau final states with the LHCb detector*, arXiv:1401.4902.

[393] P. Golonka, B. Kersevan, T. Pierzchala, E. Richter-Wąs, Z. Wąs and M. Worek, *The tauola-photos-F environment for the TAUOLA and PHOTOS packages, release II*, Comput. Phys. Commun. **174**, 818 (2006), doi:10.1016/j.cpc.2005.12.018.

[394] J. Butterworth et al., *Les Houches 2013: Physics at TeV colliders: Standard model working group report*, arXiv:1405.1067.

[395] A. Buckley, H. Hoeth, H. Lacker, H. Schulz and J. E. von Seggern, *Systematic event generator tuning for the LHC*, Eur. Phys. J. C **65**, 331 (2009), doi:10.1140/epjc/s10052-009-1196-7.

[396] P. Ilten, M. Williams and Y. Yang, *Event generator tuning using Bayesian optimization*, J. Instrum. **12**, P04028 (2017), doi:10.1088/1748-0221/12/04/P04028.

[397] J. Bellm and L. Gellersen, *High dimensional parameter tuning for event generators*, Eur. Phys. J. C **80**, 54 (2020), doi:10.1140/epjc/s10052-019-7579-5.

[398] M. Krishnamoorthy et al., *Apprentice for event generator tuning*, Europhys. J. Web Conf. **251**, 03060 (2021), doi:10.1051/epjconf/202125103060.

[399] J. M. Butterworth et al., *The tools and Monte Carlo working group: Summary report from the Les Houches 2009 workshop on TeV colliders*, arXiv:1003.1643.

[400] P. Skands et al., *SUSY Les Houches accord: Interfacing SUSY spectrum calculators, decay packages, and event generators*, J. High Energy Phys. **07**, 036 (2004), doi:10.1088/1126-6708/2004/07/036.

[401] B. C. Allanach et al., *SUSY Les Houches Accord 2*, Comput. Phys. Commun. **180**, 8 (2009), doi:10.1016/j.cpc.2008.08.004.

[402] J. Alwall et. al, *A Les Houches interface for BSM generators*, arXiv:0712.3311.

[403] E. Bothmann et al., *Event generation with Sherpa 2.2*, SciPost Phys. **7**, 034 (2019), doi:10.21468/SciPostPhys.7.3.034.

[404] S. Höche, S. Prestel and H. Schulz, *Simulation of vector boson plus many jet final states at the high luminosity LHC*, Phys. Rev. D **100**, 014024 (2019), doi:10.1103/PhysRevD.100.014024.

[405] M. R. Whalley, D. Bourilkov and R. C. Group, *The Les Houches Accord PDFs (LHAPDF) and Lhaglue*, arXiv:hep-ph/0508110.

[406] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr and G. Watt, *LHAPDF6: Parton density access in the LHC precision era*, Eur. Phys. J. C **75**, 132 (2015), doi:10.1140/epjc/s10052-015-3318-8.

[407] P. Artoisenet, F. Maltoni and T. Stelzer, *Automatic generation of quarkonium amplitudes in NRQCD*, J. High Energy Phys. **02**, 102 (2008), doi:10.1088/1126-6708/2008/02/102.

[408] H.-S. Shao, *HELAC-Onia 2.0: An upgraded matrix-element and event generator for heavy quarkonium physics*, Comput. Phys. Commun. **198**, 238 (2016), doi:10.1016/j.cpc.2015.09.011.

[409] D. J. Lange, *The EvtGen particle decay simulation package*, Nucl. Instrum. Methods Phys. Res. A **462**, 152 (2001), doi:10.1016/S0168-9002(01)00089-4.

[410] M. Dobbs and J. B. Hansen, *The HepMC C++ Monte Carlo event record for high energy physics*, Comput. Phys. Commun. **134**, 41 (2001), doi:10.1016/S0010-4655(00)00189-2.

[411] A. Buckley, P. Ilten, D. Konstantinov, L. Lönnblad, J. Monk, W. Pokorski, T. Przedzinski and A. Verbytskyi, *The HepMC3 event record library for Monte Carlo event generators*, Comput. Phys. Commun. **260**, 107310 (2021), doi:10.1016/j.cpc.2020.107310.

[412] R. Brun and F. Rademakers, *ROOT: An object oriented data analysis framework*, Nucl. Instrum. Methods Phys. Res. A **389**, 81 (1997), doi:10.1016/S0168-9002(97)00048-X.

[413] A. Buckley, J. Butterworth, D. Grellscheid, H. Hoeth, L. Lönnblad, J. Monk, H. Schulz and F. Siegert, *Rivet user manual*, Comput. Phys. Commun. **184**, 2803 (2013), doi:10.1016/j.cpc.2013.05.021.

[414] C. Bierlich et al., *Robust independent validation of experiment and theory: Rivet version 3*, SciPost Phys. **8**, 026 (2020), doi:10.21468/SciPostPhys.8.2.026.

[415] M. R. Whalley and R. G. Roberts, *A user guide to HEPData: The Durham/RAL HEP databases on the RAL CMS system*, RAL–88-003 (1988).

[416] E. Maguire, L. Heinrich and G. Watt, *HEPData: A repository for high energy physics data*, J. Phys.: Conf. Ser. **898**, 102006 (2017), doi:10.1088/1742-6596/898/10/102006.

[417] C. Bierlich et al., *Confronting experimental data with heavy-ion models: Rivet for heavy ions*, Eur. Phys. J. C **80**, 485 (2020), doi:10.1140/epjc/s10052-020-8033-4.

[418] M. Cacciari, G. P. Salam and G. Soyez, *FastJet user manual*, Eur. Phys. J. C **72**, 1896 (2012), doi:10.1140/epjc/s10052-012-1896-2.

[419] J. D. Bjorken, *Particle physics: Where do we go from here?*, SLAC Beam Line **22**, 8 (1992).

[420] B. L. Combridge, J. Kripfganz and J. Ranft, *Hadron production at large transverse momentum and QCD*, Phys. Lett. B **70**, 234 (1977), doi:10.1016/0370-2693(77)90528-7.

[421] R. Cutler and D. Sivers, *Quantum chromodynamic gluon contributions to large p_T reactions*, Phys. Rev. D **17**, 196 (1978), doi:10.1103/PhysRevD.17.196.

[422] H.-U. Bengtsson, *The Lund Monte Carlo for high-p_T physics*, Comput. Phys. Commun. **31**, 323 (1984), doi:10.1016/0010-4655(84)90018-3.

[423] E. Eichten, I. Hinchliffe, K. Lane and C. Quigg, *Supercollider physics*, Rev. Mod. Phys. **56**, 579 (1984), doi:10.1103/RevModPhys.56.579.

[424] B. L. Combridge, *Associated production of heavy flavour states in pp and p̄p interactions: Some QCD estimates*, Nucl. Phys. B **151**, 429 (1979), doi:10.1016/0550-3213(79)90449-8.

[425] F. A. Berends, R. Kleiss, P. De Causmaecker, R. Gastmans and T. T. Wu, *Single bremsstrahlung processes in gauge theories*, Phys. Lett. B **103**, 124 (1981), doi:10.1016/0370-2693(81)90685-7.

[426] F. Halzen and D. M. Scott, *Hadroproduction of photons and leptons*, Phys. Rev. D **18**, 3378 (1978), doi:10.1103/PhysRevD.18.3378.

[427] V. Costantini, B. De Tollis and G. Pistoni, *Nonlinear effects in quantum electrodynamics*, Nuov. Cim. A **2**, 733 (1971), doi:10.1007/BF02736745.

[428] E. L. Berger, E. Braaten and R. D. Field, *Large-p_T production of single and double photons in proton-proton and pion-proton collisions*, Nucl. Phys. B **239**, 52 (1984), doi:10.1016/0550-3213(84)90084-1.

[429] D. A. Dicus and S. S. D. Willenbrock, *Photon pair production and the intermediate-mass Higgs boson*, Phys. Rev. D **37**, 1801 (1988), doi:10.1103/PhysRevD.37.1801.

[430] G. Ingelman et al., *Deep inelastic physics and simulation*, in *DESY Workshop 1987: Physics at HERA* (1987).

[431] D. Bardin, M. Bilenky, D. Lehner, A. Olchevski and T. Riemann, *Semi-analytical approach to four-fermion production in* e^+e^- *annihilation*, Nucl. Phys. B Proc. Suppl. **37**, 148 (1994), doi:10.1016/0920-5632(94)90670-X.

[432] E. Gabrielli, *The production of weak intermediate bosons in* ep *reactions*, Mod. Phys. Lett. A **01**, 465 (1986), doi:10.1142/S0217732386000592.

[433] M. A. Samuel, G. Li, N. Sinha, R. Sinha and M. K. Sundaresan, *Bounds on the magnetic moment of the W boson*, Phys. Rev. Lett. **67**, 9 (1991), doi:10.1103/PhysRevLett.67.9.

[434] T. Barklow, *Particle physics research at a 500 GeV* e^+e^- *linear collider*, Conf. Proc. C **9006252**, 440 (1990).

[435] D. W. Duke and J. F. Owens, *Quantum-chromodynamic corrections to deep-inelastic Compton scattering*, Phys. Rev. D **26**, 1600 (1982), doi:10.1103/PhysRevD.26.1600.

[436] M. Fontannaz, B. Pire and D. Schiff, *Inclusive photoproduction cross sections of charmed mesons and baryons*, Z. Phys. C **11**, 211 (1981), doi:10.1007/BF01545678.

[437] G. Bozzi, B. Fuks, B. Herrmann and M. Klasen, *Squark and gaugino hadroproduction and decays in non-minimal flavour violating supersymmetry*, Nucl. Phys. B **787**, 1 (2007), doi:10.1016/j.nuclphysb.2007.05.031.

[438] B. Fuks, B. Herrmann and M. Klasen, *Phenomenology of anomaly-mediated supersymmetry-breaking scenarios with nonminimal flavor violation*, Phys. Rev. D **86**, 015002 (2012), doi:10.1103/PhysRevD.86.015002.

[439] K. Huitu, J. Maalampi, A. Pietilä and M. Raidal, *Doubly charged Higgs at LHC*, Nucl. Phys. B **487**, 27 (1997), doi:10.1016/S0550-3213(97)87466-4.

[440] G. Barenboim, K. Huitu, J. Maalampi and M. Raidal, *Constraints on doubly charged Higgs interactions at linear collider*, Phys. Lett. B **394**, 132 (1997), doi:10.1016/S0370-2693(96)01670-X.

[441] N. Desai, *Collider signatures for dark matter and long-lived particles with Pythia 8*, arXiv:1807.04240.

[442] C. Ciobanu, T. Junk, G. Veramendi, J. Lee, G. De Lentdecker, K. McFarland and K. Maeshima, *Z' generation with PYTHIA*, FERMILAB-FN-0773-E (2005), doi:10.2172/15020136.

[443] G. Altarelli, B. Mele and M. Ruiz-Altaba, *Searching for new heavy vector bosons in* pp̄ *colliders*, Z. Phys. C **45**, 109 (1989), doi:10.1007/BF01556677.

[444] J. L. Hewett and S. Pakvasa, *Scalar-leptoquark production at hadron colliders*, Phys. Rev. D **37**, 3165 (1988), doi:10.1103/PhysRevD.37.3165.

[445] E. J. Eichten, K. D. Lane and M. E. Peskin, *New tests for quark and lepton substructure*, Phys. Rev. Lett. **50**, 811 (1983), doi:10.1103/PhysRevLett.50.811.

[446] U. Baur, M. Spira and P. M. Zerwas, *Excited-quark and -lepton production at hadron colliders*, Phys. Rev. D **42**, 815 (1990), doi:10.1103/PhysRevD.42.815.

[447] J. Bijnens, P. Eerola, M. Maul, A. Månsson and T. Sjöstrand, *QCD signatures of narrow graviton resonances in hadron colliders*, Phys. Lett. B **503**, 341 (2001), doi:10.1016/S0370-2693(01)00238-6.

[448] G. Bella, E. Etzion, N. Hod, Y. Oz, Y. Silver and M. Sutton, *A search for heavy Kaluza-Klein electroweak gauge bosons at the LHC*, J. High Energy Phys. **09**, 025 (2010), doi:10.1007/JHEP09(2010)025.

[449] S. Ask, J. H. Collins, J. R. Forshaw, K. Joshi and A. D. Pilkington, *Identifying the colour of TeV-scale resonances*, J. High Energy Phys. **01**, 018 (2012), doi:10.1007/JHEP01(2012)018.

[450] R. Franceschini, P. P. Giardino, G. F. Giudice, P. Lodone and A. Strumia, *LHC bounds on large extra dimensions*, J. High Energy Phys. **05**, 092 (2011), doi:10.1007/JHEP05(2011)092.

[451] G. Bella, E. Etzion, N. Hod and M. Sutton, *Introduction to the MCnet Moses project and heavy gauge bosons search at the LHC*, arXiv:1004.1649.

[452] S. Ask, *Simulation of Z plus graviton/unparticle production at the LHC*, Eur. Phys. J. C **60**, 509 (2009), doi:10.1140/epjc/s10052-009-0949-7.

[453] S. Ask, I. V. Akin, L. Benucci, A. De Roeck, M. Goebel and J. Haller, *Real emission and virtual exchange of gravitons and unparticles in Pythia8*, Comput. Phys. Commun. **181**, 1593 (2010), doi:10.1016/j.cpc.2010.05.013.