



CASCADE3 A Monte Carlo event generator based on TMDs

S. Baranov¹, A. Bermudez Martinez², L. I. Estevez Banos², F. Guzman³, F. Hautmann^{4,5}, H. Jung^{2,a}, A. Lelek⁴, J. Lidrych², A. Lipatov⁶, M. Malyshev⁶, M. Mendizabal², S. Taheri Monfared², A. M. van Kampen⁴, Q. Wang^{2,7}, H. Yang^{2,7}

¹ Lebedev Physics Institute, Moscow, Russia

² DESY, Hamburg, Germany

³ InSTEC, Universidad de La Habana, Havana, Cuba

⁴ Elementary Particle Physics, University of Antwerp, Antwerp, Belgium

⁵ RAL and University of Oxford, Oxford, UK

⁶ SINP, Moscow State University, Moscow, Russia

⁷ School of Physics, Peking University, Beijing, China

Received: 1 February 2021 / Accepted: 2 May 2021

© The Author(s) 2021

Abstract The CASCADE3 Monte Carlo event generator based on Transverse Momentum Dependent (TMD) parton densities is described. Hard processes which are generated in collinear factorization with LO multileg or NLO parton level generators are extended by adding transverse momenta to the initial partons according to TMD densities and applying dedicated TMD parton showers and hadronization. Processes with off-shell kinematics within k_t -factorization, either internally implemented or from external packages via LHE files, can be processed for parton showering and hadronization. The initial state parton shower is tied to the TMD parton distribution, with all parameters fixed by the TMD distribution.

1 Introduction

The simulation of processes for high energy hadron colliders has been improved significantly in the past years by automation of next-to-leading order (NLO) calculations and matching of the hard processes to parton shower Monte Carlo event generators which also include a simulation of hadronization. Among those automated tools are the MADGRAPH5_aMC@NLO [1] generator based on the MC@NLO [2–5] method or the POWHEG [6, 7] generator for the calculation of the hard process. The results from these packages are then combined with either the HERWIG [8] or PYTHIA [9] packages for parton showering and hadronization. Different jet multiplicities can be combined at the matrix element level and then merged with special procedures, like the MLM [10] or CKKW [11] merging for LO processes, the FxFx [12]

or MiNLO method [13] for merging at NLO, among others. While the approaches of matching and merging matrix element calculations and parton showers are very successful, two ingredients important for high energy collisions are not (fully) treated: the matrix elements are calculated with collinear dynamics and the inclusion of initial state parton showers results in a net transverse momentum of the hard process; the special treatment of high energy effects (small x) is not included.

The CASCADE Monte Carlo event generator, developed originally for small x processes based on high-energy factorization [14] and the CCFM [15–18] evolution equation, has been extended to cover the full kinematic range (not only small x) by applying the Parton Branching (PB) method and the corresponding PB Transverse Momentum Dependent (TMD) parton densities [19, 20]. The initial state evolution is fully described and determined by the TMD density, as it was in the case of the CCFM gluon density, but now available for all flavor species, including quarks, gluons and photons at small and large x and any scale μ . For a general overview of TMD parton densities, see Ref. [21].

With the advances in determination of PB TMDs [19, 20], it is natural to develop a scheme, where the initial parton shower follows as close as possible the TMD parton density and where either collinear (on-shell) or k_t -dependent (off-shell) hard process calculations can be included at LO or NLO. In order to be flexible and to use the latest developments in automated matrix element calculations of hard process at higher order in the strong coupling α_s , events available in the Les Houches Event (LHE) file format [22], which contains all the information of the hard process including the color

^a e-mail: jung@mail.desy.de (corresponding author)

structure, can be further processed for parton shower and hadronization in CASCADE3.

In this report we describe the new developments in CASCADE3 for a full PB-TMD parton shower and the matching of TMD parton densities to collinear hard process calculations. We also mention features of the small- x mode of CASCADE3.

2 The hard process

The cross section for the scattering process of two hadrons A and B can be written in collinear factorization as a convolution of the partonic cross section of partons a and b , $a + b \rightarrow X$, and the densities $f_{a(b)}(x, \mu)$ of partons a (b) inside the hadrons A (B),

$$\begin{aligned} \sigma(A + B \rightarrow Y) \\ = \int dx_a \int dx_b f_a(x_a, \mu) f_b(x_b, \mu) \sigma(a + b \rightarrow X), \end{aligned} \quad (1)$$

where $x_a(x_b)$ are the fractions of the longitudinal momenta of hadrons A , B carried by the partons $a(b)$, $\sigma(a + b \rightarrow X)$ is the partonic cross section, and μ is the factorization scale of the process. The final state Y contains the partonic final state X and the recoils from the parton evolution and hadron remnants.

In CASCADE3 we extend collinear factorization to include transverse momenta in the initial state, either by adding a transverse momentum to an on-shell process or by using off-shell processes directly, as described in detail in Sects. 2.1 and 2.2. TMD factorization is proven for semi-inclusive deep-inelastic scattering, Drell–Yan production in hadron-hadron collisions and e^+e^- annihilation [23–35]. In the high-energy limit (small- x) k_T -factorization has been formulated also in hadronic collisions for processes like heavy flavor or heavy boson (including Higgs) production [14, 36–38], with so-called *unintegrated* parton distribution functions (uPDFs), see e.g. Refs. [39–49].

2.1 On-shell processes

The hard processes in collinear factorization (with on-shell initial partons, without transverse momenta) can be calculated by standard automated methods like MADGRAPH5_AMC@NLO [1] for multileg processes at LO or NLO accuracy. The matrix element processes are calculated with collinear parton densities (PDF), as provided by LHAPDF [50].

We extend the factorization formula given in Eq. (1) by replacing the collinear parton densities $f(x, \mu)$ by TMD densities $\mathcal{A}(x, k_t, \mu)$ with k_t being the transverse momentum of

the interacting parton, and integrating over the transverse momenta.

However, when the hard process is to be combined with a TMD parton density, as described later, the integral over k_t of the TMD density must agree with the collinear (k_t -integrated) density; this feature is guaranteed by construction for the PB-TMDs (also available as integrated PDFs in LHAPDF format).

In a LO partonic calculation the TMD or the parton shower can be included respecting energy momentum conservation, as described below. In an NLO calculation based on the MC@NLO method [2–5] the contribution from collinear and soft partons is subtracted, as this is added later with the parton shower. For the use with PB TMDs, the HERWIG6 subtraction terms are best suited as the angular ordering conditions coincide with those applied in the PB-method. The PB TMDs play the same role as a parton shower does, in the sense that a finite transverse momentum is created as a result of the parton evolution [51, 52].

When transverse momenta of the initial partons from TMDs are to be included to the hard scattering process, which was originally calculated under the assumption of collinear initial partons, care has to be taken that energy and momentum are still conserved. When the initial state partons have transverse momenta, they also acquire virtual masses. The procedure adopted in CASCADE3 is the following: for each initial parton, a transverse momentum is assigned according to the TMD density, and the parton-parton system is boosted to its center-of-mass frame and rotated such that only the longitudinal and energy components are non-zero. The energy and longitudinal component of the initial momenta $p_{a,b}$ are recalculated taking into account the virtual masses $Q_a^2 = k_{t,a}^2$ and $Q_b^2 = k_{t,b}^2$ [53],

$$E_{a,b} = \frac{1}{2\sqrt{\hat{s}}} (\hat{s} \pm (Q_b^2 - Q_a^2)) \quad (2)$$

$$p_{z\,a,b} = \pm \frac{1}{2\sqrt{\hat{s}}} \sqrt{(\hat{s} + Q_a^2 + Q_b^2)^2 - 4Q_a^2 Q_b^2} \quad (3)$$

with $\hat{s} = (p_a + p_b)^2$ with $p_a(p_b)$ being the four-momenta of the interacting partons a and b . The partonic system is then rotated and boosted back to the overall center-of-mass system of the colliding particles. By this procedure, the parton-parton mass $\sqrt{\hat{s}}$ is exactly conserved, while the rapidity of the partonic system is approximately restored, depending on the transverse momenta.

In Fig. 1 a comparison of the Drell–Yan (DY) mass, transverse momentum and rapidity is shown for an NLO calculation of DY production in pp collisions at $\sqrt{s} = 13$ TeV in the mass range $30 < m_{DY} < 2000$ GeV. The curve labelled NLO(LHE) is the calculation of MADGRAPH5_AMC@NLO with the subtraction terms, the curve NLO(LHE+TMD) is the prediction after the transverse momentum is included

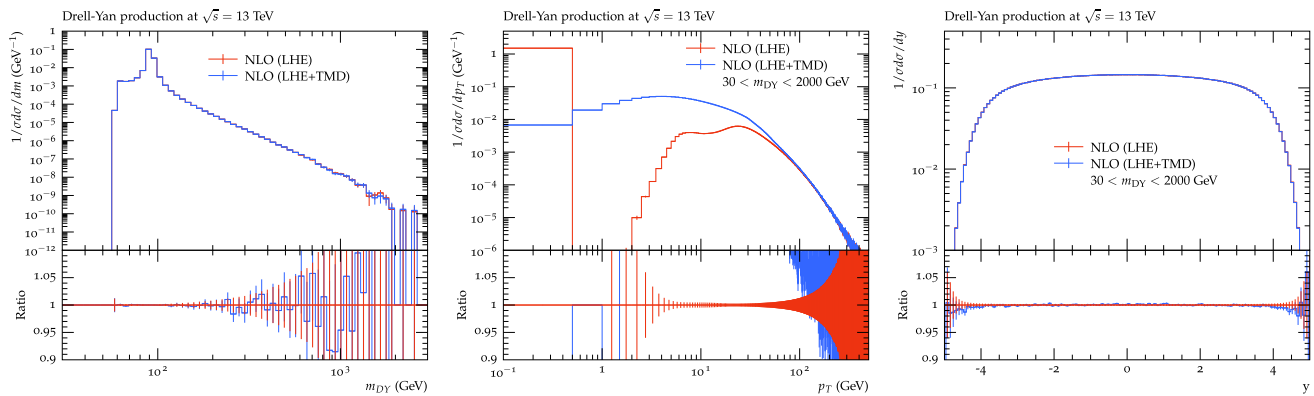


Fig. 1 Distributions of Drell-Yan mass, transverse momentum and rapidity for $pp \rightarrow DY + X$ at $\sqrt{s} = 13$ TeV. The hard process is calculated with MADGRAPH5_aMC@NLO. NLO(LHE) is the predic-

tion including subtraction terms, NLO(LHE+TMD) includes transverse momenta of the interacting partons according to the description in the text

according to the procedure described above. In the p_T spectrum one can clearly see the effect of including transverse momenta from the TMD distribution. The DY mass distribution is not changed, and the rapidity distribution is almost exactly reproduced, only at large rapidities small differences are observed.

The transverse momenta k_t are generated according to the TMD density $\mathcal{A}(x, k_t, \mu)$, at the original longitudinal momentum fraction x and the hard process scale μ . In a LO calculation, the full range of k_t is available, but in an NLO calculation via the MC@NLO method a *shower scale* defines the boundary between parton shower and real emissions from the matrix element, limiting the transverse momentum k_t . Technically the factorization scale μ is calculated within CASCADE3 (see parameter `lhescale`) as it is not directly accessible from the LHE file, while the *shower scale* is given by SCALUP. The limitation of the transverse momenta coming from the TMD distribution and TMD shower to be smaller than the *shower scale* SCALUP guarantees that the overlap with real emissions from the matrix element is minimized according to the subtraction of counterterms in the MC@NLO method.

The advantage of using TMDs for the complete process is that the kinematics are fixed, independent of simulating explicitly the radiation history from the parton shower. For inclusive processes, for example inclusive Drell-Yan processes, the details of the hadronic final state generated by a parton shower do not matter, and only the net effect of the transverse momentum distribution is essential. However, for processes which involve jets, the details of the parton shower become also important. The parton shower, as described below, follows very closely the transverse momentum distribution of the TMD and thus does not change any kinematic distribution after the transverse momentum of the initial partons are included.

All hard processes, which are available in MADGRAPH5_aMC@NLO can be used within CASCADE3. The treatment of multijet merging is described in Sect. 8.

2.2 Off-shell processes

In a region of phase space, where the longitudinal momentum fractions x become very small, the transverse momentum of the partons cannot be neglected and has to be included already at the matrix element level, leading to so-called *off-shell* processes.

In off-shell processes a natural suppression at large k_t [54] (with $k_t > \mu$) is obtained, shown explicitly in Fig. 2, where the matrix element for $g^*g^* \rightarrow Q\bar{Q}$, with Q being a heavy quark, is considered. The process is integrated over the final state phase space [55],

$$\tilde{\sigma}(k_t) = \int \frac{dx_2}{x_2} d\phi_{1,2} d\text{Lips} |ME|^2 (1-x_2)^5, \quad (4)$$

where dLips is the Lorentz-invariant phase space of the final state, ME is the matrix-element for the process, $\phi_{1,2}$ is the azimuthal angle between the two initial partons, and a simple scale-independent and k_t -independent gluon density $xG(x) = (1-x)^5$ is included which suppresses large- x contributions. In Fig. 2 we show $\tilde{\sigma}(k_t)$ normalized to its on-shell value $\tilde{\sigma}(0)$ at $\sqrt{s} = 13000$ GeV as a function of the transverse momentum of the incoming gluon $k_{t,2}$ for different values of x_1 , which are chosen such that the ratio $k_{t,1}^2/(x_1s)$ is kept constant.

In Fig. 2 (left) predictions are shown for bottom quarks with mass $m = 5$ GeV and different $k_{t,1}$, in Fig. 2 (right) a comparison is made for different heavy quark masses. Using off-shell matrix elements a suppression at large transverse momenta of the initial partons is obtained, depending on the heavy flavor mass and the transverse momentum. In a collinear approach, with implicit integration over transverse

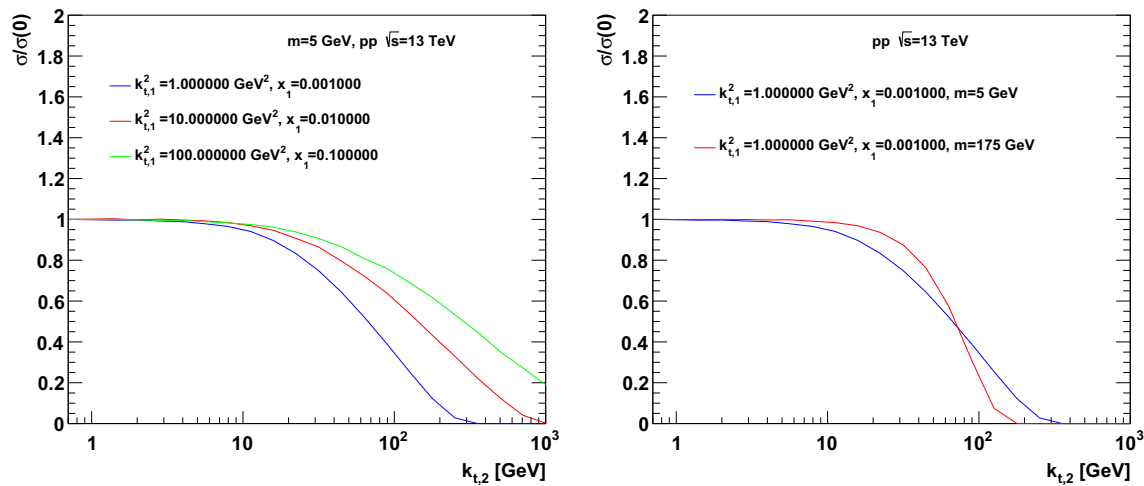


Fig. 2 The reduced cross section $\tilde{\sigma}(k_t)/\tilde{\sigma}(0)$ as a function of the transverse momentum $k_{t,2}$ of the incoming gluon at $\sqrt{s} = 13000$ GeV. (Left) for different values of $k_{t,1}$ and x_1 , (right) for different heavy flavor masses and fixed values of $k_{t,1}$ and x_1

momenta of the initial state partons, the transverse momenta are limited by a theta function at the factorization scale, while off-shell matrix elements give a smooth transition to a high k_t tail.

When using off-shell processes, BFKL or CCFM type parton densities should be used to cover the full available phase space in transverse momentum, which can lead to k_t 's larger than the transverse momentum of any of the partons of the hard process [56]. Until now, only gluon densities obtained from CCFM [15–18] or BFKL [57–59] are available, thus limiting the advantages of using off-shell matrix elements to gluon induced processes.

Several processes with off-shell matrix elements are implemented in CASCADE3 as listed in Table 1, and described in detail in [60]. However, many more processes are accessible via the automated matrix element calculators for off-shell processes, KATIE [61] and PEGASUS [62]. The events from the hard process are then read with the CASCADE3 package via LHE files. For processes generated with KATIE or PEGASUS no further corrections need to be performed and the event can be directly passed to the showering procedure, described in the next section.

3 Initial state parton shower based on TMDs

The parton shower, which is described here, follows consistently the parton evolution of the TMDs. By this we mean that the splitting functions P_{ab} , the order and the scale in α_s as well as kinematic restrictions are identical to both the parton shower and the evolution of the parton densities (for NLO PB TMD densities, the NLO DGLAP splitting functions [73, 74] together with NLO α_s is applied, while for the

LO TMD densities the corresponding LO splitting functions [75–77] and LO α_s is used).

3.1 From PB TMD evolution to TMD parton shower

The PB method describes the TMD parton density as (cf Eq. (2.43) in Ref. [19])

$$\begin{aligned}
 x\mathcal{A}_a(x, k_t, \mu) &= \Delta_a(\mu) x\mathcal{A}_a(x, k_t, \mu_0) \\
 &+ \sum_b \int \frac{dq^2}{q^2} \frac{d\phi}{2\pi} \frac{\Delta_a(\mu)}{\Delta_a(q)} \Theta(\mu - q) \Theta(q - \mu_0) \\
 &\times \int_x^{z_M} dz P_{ab}^{(R)}(\alpha_s(f(z, q)), z) \frac{x}{z} \mathcal{A}_b\left(\frac{x}{z}, k'_t, q\right), \quad (5)
 \end{aligned}$$

with $z_M < 1$ defining resolvable branchings, \mathbf{k} (\mathbf{q}_c) being the transverse momentum vector of the propagating (emitted) parton, respectively. The transverse momentum of the parton before branching is defined as $k'_t = |\mathbf{k} + (1 - z)\mathbf{q}|$ with $\mathbf{q} = \mathbf{q}_c/(1 - z)$ being the rescaled transverse momentum vector of the emitted parton (see Fig. 3, with the notation $k_t = |\mathbf{k}|$ and $q = |\mathbf{q}|$) and ϕ being the azimuthal angle between \mathbf{q} and \mathbf{k} . The argument in α_s is in general a function of the evolution scale q . Higher order calculations indicate the transverse momentum of the emitted parton as the preferred scale. The real emission branching probability is denoted by $P_{ab}^{(R)}(\alpha_s(f(z, q)), z)$ including α_s as described in Ref. [19] (in the following we omit α_s in the argument of $P_{ab}^{(R)}$ for easier reading). The Sudakov form factor is given by:

$$\Delta_a(z_M, \mu, \mu_0) = \exp\left(-\sum_b \int_{\mu_0^2}^{\mu^2} \frac{dq^2}{q^2} \int_0^{z_M} dz z P_{ba}^{(R)}\right). \quad (6)$$

Table 1 Processes included in CASCADE3. Q stands for heavy quarks, q for light quarks

Lepto(photo)production	Process	IPRO	References
	$\gamma^* g^* \rightarrow q \bar{q}$	10	[63]
	$\gamma^* g^* \rightarrow Q \bar{Q}$	11	[63]
	$\gamma^* g^* \rightarrow J/\psi g$	2	[64–67]
Hadroproduction			
	$g^* g^* \rightarrow q \bar{q}$	10	[63]
	$g^* g^* \rightarrow Q \bar{Q}$	11	[63]
	$g^* g^* \rightarrow J/\psi g$	2	[67]
	$g^* g^* \rightarrow \Upsilon g$	2	[67]
	$g^* g^* \rightarrow \chi_c$	3	[67]
	$g^* g^* \rightarrow \chi_b$	3	[67]
	$g^* g^* \rightarrow J/\psi J/\psi$	21	[68]
	$g^* g^* \rightarrow h^0$	102	[38]
	$g^* g^* \rightarrow Z Q \bar{Q}$	504	[69, 70]
	$g^* g^* \rightarrow Z q \bar{q}$	503	[69, 70]
	$g^* g^* \rightarrow W q_i Q_j$	514	[69, 70]
	$g^* g^* \rightarrow W q_i q_j$	513	[69, 70]
	$q g^* \rightarrow Z q$	501	[71]
	$q g^* \rightarrow W q$	511	[71]
	$q g^* \rightarrow q g$	10	[72]
	$g g^* \rightarrow g g$	10	[72]

Dividing Eq. (5) by $\Delta_a(\mu^2)$ and differentiating with respect to μ^2 gives the differential form of the evolution equation describing the probability for resolving a parton with transverse momentum \mathbf{k}' and momentum fraction x/z into a parton with momentum fraction x and emitting another parton during a small decrease of μ ,

$$\mu^2 \frac{d}{d\mu^2} \left(\frac{x \mathcal{A}_a(x, k_t, \mu)}{\Delta_a(\mu)} \right) = \sum_b \int_x^{z_M} dz \frac{d\phi}{2\pi} P_{ab}^{(R)} \frac{x}{z} \frac{\mathcal{A}_b\left(\frac{x}{z}, k'_t, \mu\right)}{\Delta_a(\mu)}. \quad (7)$$

The normalized probability is then given by

$$\frac{\Delta_a(\mu)}{x \mathcal{A}_a(x, k_t, \mu)} d \left(\frac{x \mathcal{A}_a(x, k_t, \mu)}{\Delta_a(\mu)} \right) = \sum_b \frac{d\mu^2}{\mu^2} \int_x^{z_M} dz \frac{d\phi}{2\pi} P_{ab}^{(R)} \frac{\frac{x}{z} \mathcal{A}_b\left(\frac{x}{z}, k'_t, \mu\right)}{x \mathcal{A}_a(x, k_t, \mu)} \quad (8)$$

This equation can be integrated between μ_{i-1}^2 and μ^2 to give the no-branching probability (Sudakov form factor) for the backward evolution Δ_{bw} ,¹

$$\log \Delta_{bw}(x, k_t, \mu, \mu_{i-1})$$

¹ In Eq. (9) ordering in μ is assumed. However, if angular ordering as in CCFM [15–18] is applied then the ratio of parton densities would change to $[x' \mathcal{A}_b(x', k'_t, q'/z)]/[x \mathcal{A}_a(x, k_t, q')]$ as discussed in [60].

$$\begin{aligned} &= \log \left(\frac{\Delta_a(\mu)}{\Delta_a(\mu_{i-1})} \frac{x \mathcal{A}_a(x, k_t, \mu_{i-1})}{x \mathcal{A}_a(x, k_t, \mu)} \right) \\ &= - \sum_b \int_{\mu_{i-1}^2}^{\mu^2} \frac{dq'^2}{q'^2} \frac{d\phi}{2\pi} \int_x^{z_M} dz P_{ab}^{(R)} \frac{x' \mathcal{A}_b(x', k'_t, q')}{x \mathcal{A}_a(x, k_t, q')}, \end{aligned} \quad (9)$$

with $x' = x/z$. This Sudakov form factor is very similar to the Sudakov form factor in ordinary parton shower approaches, with the difference that for the PB TMD shower the ratio of PB TMD densities $[x' \mathcal{A}_b(x', k'_t, q')]/[x \mathcal{A}_a(x, k_t, q')]$ is applied, which includes a dependence on k_t .

In Eq. (9) a relation between the Sudakov form factor Δ_a used in the evolution equation and the Sudakov form factor Δ_{bw} used for the backward evolution of the parton shower is made explicit. A similar relation was also studied in Refs. [78, 79]. In Ref. [78] the z_M limit was identified as a source of systematic uncertainty when using conventional showers with standard collinear pdfs; in the PB approach, the same z_M limit is present in the parton evolution as well as in the PB-shower. The PB approach allows a consistent formulation of the parton shower with the PB TMDs, as in both Sudakov form factors Δ_a and Δ_{bw} the same value of z_M is used.

The splitting functions $P_{ab}^{(R)}$ contain the coupling,

$$P_{ab}(\alpha_s, z) = \sum_{n=1}^{\infty} \left(\frac{\alpha_s(f(z, q))}{2\pi} \right)^n P_{ab}^{(n-1)}(z), \quad (10)$$

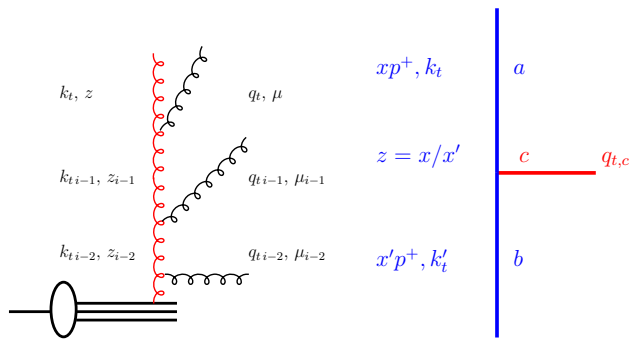


Fig. 3 Left: schematic view of a parton branching process. Right: branching process $b \rightarrow a + c$

where the scale $f(z, q)$ in the coupling depends on the ordering condition as discussed later (see Eq. (11)).

The advantage of using a PB TMD shower is that as long as the parameters of the parton shower are set through TMD distributions the parton shower uncertainties can be recast as uncertainties of the TMDs, which in turn can be fitted to experimental data in a systematic global manner.

3.2 Backward evolution for initial state TMD parton shower

A backward evolution method, as now common in Monte Carlo event generators, is applied for the initial state parton shower, evolving from the large scale of the matrix-element process backwards down to the scale of the incoming hadron. However, in contrast to the conventional parton shower, which generates transverse momenta of the initial state partons during the backward evolution, the transverse momenta of the initial partons of the hard scattering process is fixed by the TMD and the parton shower does not change the kinematics. The transverse momenta during the backward cascade follow the behavior of the TMD. The hard scattering process is obtained as described in Sect. 2. The backward evolution of the initial state parton shower follows very closely the description in [60,80,81], which is based on Ref. [53].

The starting value of the evolution scale μ is calculated from the hard scattering process, as described in Sect. 2. In case of on-shell matrix elements at NLO, the transverse momentum of the hardest parton in the parton shower evolution is limited by the *shower-scale*, as described in Sect. 2.1.

Starting at the hard scale $\mu = \mu_i$, the parton shower algorithm searches for the next scale μ_{i-1} at which a resolvable branching occurs (see Fig. 3 left). This scale μ_{i-1} is selected from the Sudakov form factor Δ_{bw} as given in Eq. (9) (see also [60]). In the parton shower language, the selection of the next branching comes from solving $R = \Delta_{bw}(x, k_t, \mu_i, \mu_{i-1})$ for μ_{i-1} using uniformly distributed random numbers R for given x and μ_i . However, to

solve the integrals in Eq. (9) numerically for every branching would be too time consuming, instead the veto-algorithm [53,82] is applied.

The splitting function P_{ab} as well as the argument $f(z, q)$ in the calculation of α_s is chosen exactly as used in the evolution of the parton density. In a parton shower one treats “resolvable” branchings, defined via a cut in $z < z_M$ in the splitting function to avoid the singular behavior of the terms $1/(1-z)$, and branchings with $z > z_M$ are regarded as “non-resolvable” and are treated similarly as virtual corrections: they are included in the Sudakov form factor Δ_{bw} . The splitting variable z_{i-1} is obtained from the splitting functions following the standard methods (see Eq.(2.37) in [19]).

The calculation of the transverse momentum k_t is sketched in Fig. 3 (right). The transverse momentum $q_{t,c}$ can be calculated in case of angular ordering (where the scale q of each branching is associated with the angle of the emission) in terms of the angle Θ of the emitted parton with respect to the beam directions $q_{t,c} = (1-z)E_b \sin \Theta$,

$$\mathbf{q}_c^2 = (1-z)^2 q^2. \quad (11)$$

Once the transverse momentum of the emitted parton \mathbf{q}_c is known, the transverse momentum of the propagating parton can be calculated from

$$\mathbf{k}' = \mathbf{k} + \mathbf{q}_c \quad (12)$$

with a uniformly distributed azimuthal angle ϕ assumed for the vector components of \mathbf{k} and \mathbf{q}_c . The generation of the parton momenta is performed in the center-of-mass frame of the collision (in contrast to conventional parton showers, which are generated in different partonic frames).

The whole procedure is iterated until one reaches a scale $\mu_{i-1} < q_0$ with q_0 being a cut-off parameter, which can be chosen to be the starting evolution scale of the TMD. It is of advantage to continue the parton shower evolution to lower scales $q_0 \sim \Lambda_{qcd} \sim 0.3$ GeV.

The final transverse momentum of the propagating parton \mathbf{k} is the sum of all transverse momenta \mathbf{q}_c (see Fig. 3 right):

$$\mathbf{k} = \mathbf{k}_0 - \sum_c \mathbf{q}_c. \quad (13)$$

with \mathbf{k}_0 being the intrinsic transverse momentum.

The PB TMD parton shower is selected with `Parton Evolution=2` (or `ICCF=2`).

3.3 CCFM parton evolution and parton shower

The CCFM parton evolution and corresponding parton shower follows a similar approach as described in the previous section and in detail also in Refs. [60,80,81,83]. The main difference to the PB-TMD shower are the splitting functions with the non-Sudakov form factor Δ_{ns} and the allowed

phase space for emission. The original CCFM splitting function $\tilde{P}_g(z, q, k_t)$ for branching $g \rightarrow gg$ is given by²

$$\tilde{P}_g(z, q, k_t) = \frac{\bar{\alpha}_s(q(1-z))}{1-z} + \frac{\bar{\alpha}_s(k_t)}{z} \Delta_{ns}(z, q, k_t), \quad (14)$$

where the non-Sudakov form factor Δ_{ns} is defined as

$$\log \Delta_{ns} = -\bar{\alpha}_s(k_t) \int_0^1 \frac{dz'}{z'} \times \int \frac{dq^2}{q^2} \Theta(k_t - q) \Theta(q - z'q_t), \quad (15)$$

with $q_t = \sqrt{\mathbf{q}_t^2}$ being the magnitude of the transverse vector defined in Eq. (11) and k_t the magnitude of the transverse vector in Eq. (12).

The CCFM parton shower is selected with `ICCF=1` (`PartonEvolution=1`).³

4 The TMD parton densities

In the previous versions of CASCADE the TMD densities were part of the program. With the development of TMDLIB [84, 85] there is easy access to all available TMDs, including parton densities for photons (as well as Z, W and H densities, if available).

These parton densities can be selected via `PartonDensity` with a value $> 100,000$. For example the TMDs from the parton branching method [19,20] are selected via `PartonDensity=102100` (`102200`) for PB-NLO-HERAI+II-2018-set1 (set2).

Note that the features of the TMD parton shower are only fully available for the PB-TMD sets and the CCFM shower clearly needs CCFM parton densities (like for instance [86]). PB-TMD parton densities are determined in Ref. [87] from fits to HERA DIS F_2 measurements for $Q^2 > 3 \text{ GeV}^2$, giving very good χ^2 values. In Refs. [88,89] the transverse momentum distribution of Drell-Yan pairs at low and high masses, obtained from PB-TMD densities, are compared with experimental measurements in a wide variety of kinematic regions, from low-energy fixed target experiments to high-energy collider experiments. Good agreement is found between predictions and measurements without the need for tuning of non-perturbative parameters, which illustrates the validity of the

approach over a broad kinematic range in energy and mass scales.

5 Final state parton showers

The final state parton shower uses the parton shower routine PYSHOW of PYTHIA. Leptons in the final state, coming for example from Drell-Yan decays, can radiate photons, which are also treated in the final state parton shower. Here the method from PYADSH of PYTHIA is applied, with the scale for the QED shower being fixed at the virtuality of the decaying particle (for example the mass of the Z-boson).

The default scale for the QCD final state shower is $\mu^2 = 2 \cdot (m_{1\perp}^2 + m_{2\perp}^2)$ (`ScaleTimeShower=1`), with $m_{1(2)\perp}$ being the transverse mass of the hard parton 1(2). Other choices are possible: $\mu^2 = \hat{s}$ (`ScaleTimeShower=2`) and $\mu^2 = 2 \cdot (m_1^2 + m_2^2)$ (`ScaleTimeShower=3`). In addition a scale factor can be applied: `ScaleFactorFinalShower` $\times \mu^2$ (default: `ScaleFactorFinalShower=1`).

6 Hadronization

The hadronization (fragmentation of the partons in colorless systems) is done exclusively by PYTHIA. Hadronization (fragmentation) is switched off by `Hadronization = 0` (or `NFRA = 0` for the older steering cards). All parameters of the hadronization model can be changed via the steering cards.

7 Uncertainties

Uncertainties of QCD calculations mainly arise from missing higher order corrections, which are estimated by varying the factorization and renormalization scales up and down by typically a factor of 2. The scale variations are performed when calculating the matrix elements and are stored as additional weights in the LHE file, which are then passed directly via CASCADE3 to the HEPMC [90] output file for further processing.

The uncertainties coming from the PDFs can also be calculated as additional weight factors during the matrix element calculation. However, when using TMDs, additional uncertainties arise from the transverse momentum distribution of the TMD. The PB-TMDs come with uncertainties from the experimental uncertainties as well as from model uncertainties, as discussed in Ref. [87]. These uncertainties can be treated and applied as additional weight factors with the parameter `Uncertainty_TMD=1`.

² Finite terms are neglected as they are not obtained in CCFM at the leading infrared accuracy (cf p. 72 in [17]).

³ A one loop parton shower (DGLAP like) with $\Delta_{ns} = 1$, one loop α_s and strict ordering in q can be selected with `ICCF=0`.

8 Multi-jet merging

Showered multijet LO matrix element calculations can be merged using the prescription discussed in Ref. [91]. The merging performance is controlled by the three parameters R_{clus} , E_{tclus} , E_{tclmax} . Final-state partons with pseudorapidity $\eta < E_{\text{tclmax}}$ present in the event record after the shower step but before hadronization are passed to the merging machinery if $I_{\text{merge}} = 1$. Partons are clustered using the kt-jet algorithm with a cone radius R_{clus} and matched to the PB evolved matrix element partons if the distance between the parton and the jet is $R < 1.5 \times R_{\text{clus}}$. The hardness of the reconstructed jets is controlled by its minimum transverse energy E_{tclus} (merging scale).

The number of light flavor partons is defined by the $N_{\text{qmaxMerge}}$ parameter. Heavy flavor partons and their corresponding radiation are not passed to the merging algorithm. All jet multiplicities are treated in exclusive mode except for the highest multiplicity MaxJetsMerge which is treated in inclusive mode.

9 Program description

In CASCADE3 all variables are declared as `Double Precision`. With CASCADE3 the source of PYTHIA 6.428 is included to avoid difficulties in linking.

9.1 Random numbers

CASCADE3 uses the RANLUX random number generator, with luxury level $\text{LUX} = 4$. The random number seed can be set via the environment variable `CASEED`, the default value is `CASEED=12345`.

9.2 Event output

When HEPMC is included, generated events are written out in HEPMC [90] format for further processing. The environment variable `+HEPMCOUT+` is used to specify the file name, by default this variable is set to `HEPMCOUT=/dev/null`.

The HEPMC events can be further processed, for example with Rivet [92].

9.3 Input parameters

The input parameters are steered via steering files. The new format of steering is discussed in Sect. 9.3.1 and should be used when reading LHE files, while the other format, which is appropriate for the internal off-shell processes, is discussed in Sect. 9.3.2.

9.3.1 Input parameters: new format

Examples for steering files are under `$install_path/share/cascade/LHE`.


```

&CASCADE_input
NrEvents = -1          ! Nr of events to process
Process_Id = -1        ! Read LHE file
Hadronisation = 0      ! Hadronisation (on =1, off = 0)
SpaceShower = 1        ! Space-like Parton Shower
SpaceShowerOrderAlphas=2 ! Order alphas in Space Shower
TimeShower = 1         ! Time-like Parton Shower
ScaleTimeShower = 4    ! Scale choice for Time-like Shower
!                      1: 2(m^2_1t+m^2_2t)
!                      2: shat
!                      3: 2(m^2_1+m^2_2)
!                      4: 2*scalup (from lhe file)
ScaleFactorFinalShower = 1. ! scale factor for Final State Parton Shower
PartonEvolution = 2     ! type of parton evolution in Space-like Shower
!                      1: CCFM
!                      2: full all flavor TMD evolution
! EnergyShareRemnant = 4 ! energy sharing in proton remnant
!                      1: (a+1)(1-z)**a, <z>=1/(a+2)=1/3
!                      2: (a+1)(1-z)**a, <z>=1/(a+2)=mq/(mq+mQ
!                      3: N/(z(1-1/z-c/(1-z))**2), c=(mq/mQ)**2
!                      4: PYZDIS: KFL1=1
! Remnant = 0          ! =0 no remnant treatment
PartonDensity = 102200  ! use TMDlib: PB-TMDNLO-set2
! PartonDensity = 102100 ! use TMDlib: PB-TMDNLO-set1
! TMDDensityPath= './share' ! Path to TMD density for internal files
Uncertainty_TMD = 0     ! calculate and store uncertainty TMD pdfs
lheInput='MCatNLO-example.lhe' ! LHE input file
! HasOnShellPartons = 1 ! = 0 LHE file has off-shell parton configuration
lheReweightTMD = 0      ! Reweight with new TMD given in PartonDensity
lheScale = 2            ! Scale definition for TMD
!                      0: use scalup
!                      1: use shat
!                      2: use 1/2 Sum pt^2 of final parton/particles
!                      3: use shat for Born and 1/2 Sum pt^2 of final parton(particle)
!                      4: use shat for Born and max pt of most forward/backward
!                      parton(particle)
! NNBornpart = 2        ! Nr of hard partons (particles) (Born process)
ScaleFactorMatchingScale = 2. ! Scale factor for matching scale when including TMDs
&End

&PYTHIA6_input
P6_I tune = 370        ! Retune of Perugia 2011 w CTEQ6L1 (Oct 2012)
! P6_MSTJ(41) = 1      ! (D = 2) type of branchings allowed in shower.
!                      1: only QCD
!                      2: QCD and photons off quarks and leptons
P6_MSTJ(45) = 4        ! Nr of flavors in final state shower: g->qqbar
P6_PMAS(4,1)= 1.6      ! charm mass
P6_PMAS(5,1)= 4.75     ! bottom mass
P6_MSTJ(48) = 1        ! (D=0), 0=no max. angle, 1=max angle def. in PARJ(85)
! P6_MSTU(111) = 1     ! = 0 : alpha_s is fixed, =1 first order; =2 2nd order;
! P6_PARU(112) = 0.2   ! lambda_QCD
P6_MSTU(112)= 4        ! nr of flavours wrt lambda_QCD
P6_MSTU(113)=          ! min nr of flavours for alphas
P6_MSTU(114)= 5        ! max nr of flavours for alphas
&End

```

9.3.2 Input parameters: off-shell processes

Examples for steering files are under `$install_path/share/cascade/HERA` and `$install_path/share/cascade/PP`.

```
* OLD STEERING FOR CASCADE
*
* number of events to be generated
*
```

```

NEVENT 100
*
* ++++++ Kinematic parameters ++++++
*
'PBE1' 1 0 -7000. ! Beam energy
'KBE1' 1 0 2212 ! -11: positron, 22: photon 2212: proton
'IRE1' 1 0 1 ! 0: beam 1 has no structure
* ! 1: beam 1 has structure
'PBE2' 1 0 7000. ! Beam energy
'KBE2' 1 0 2212 ! 11: electron, 22: photon 2212: proton
'IRE2' 1 0 1 ! 0: beam 3 has no structure
* ! 1: beam 2 has structure
'NFLA' 1 0 4 ! (D=5) nr of flavours used in str.fct
* ++++++ Hard subprocess selection ++++++
'IPRO' 1 0 2 ! (D=1)
* ! 2: J/psi g
* ! 3: chi_c
'I23S' 1 0 0 ! (D=0) select 2S or 3S state
'IPOL' 1 0 0 ! (D=0) VM->l1 (polarization study)
'IHFL' 1 0 4 ! (D=4) produced flavour for IPRO=11
* ! 4: charm
* ! 5: bottom
'PTCU' 1 0 1. ! (D=0) p_t **2 cut for process
* ++++++ Parton shower and fragmentation ++++++
'NFRA' 1 0 1 ! (D=1) Fragmentation on=1 off=0
'IFPS' 1 0 3 ! (D=3) Parton shower
* ! 0: off
* ! 1: initial state PS
* ! 2: final state PS
* ! 3: initial and final state PS
'IFIN' 1 0 1 ! (D=1) scale switch for FPS
* ! 1: 2(m^2_1t+m^2_2t)
* ! 2: shat
* ! 3: 2(m^2_1+m^2_2)
'SCAF' 1 0 1. ! (D=1) scale factor for FPS
'ITIM' 1 0 0 ! 0: timelike partons may not shower
* ! 1: timelike partons may shower
'ICCF' 1 0 1 ! (D=1) Evolution equation
* ! 0: DGLAP
* ! 1: CCFM
* ! 2: PB TMD evolution
* ++++++ Structure functions and scales ++++++
'IRAM' 1 0 0 ! (D=0) Running of alpha_em(Q2)
* ! 0: fixed
* ! 1: running
'IRAS' 1 0 1 ! (D=1) Running of alpha_s(MU2)
* ! 0: fixed alpha_s=0.3
* ! 1: running
'IQ2S' 1 0 3 ! (D=1) Scale MU2 of alpha_s
* ! 1: MU2= 4*m**2 (only for heavy quarks)
* ! 2: MU2 = shat(only for heavy quarks)
* ! 3: MU2= 4*m**2 + pt**2
* ! 4: MU2 = Q2
* ! 5: MU2 = Q2 + pt**2
* ! 6: MU2 = k_t**2
'SCAL' 1 0 1.0 ! scale factor for renormalisation scale
'SCAF' 1 0 1.0 ! scale factor for factorisation scale*
*'IGLU' 1 0 1201 ! (D=1010)Unintegrated gluon density
* ! > 10000 use TMDlib (i.e. 101201 for JH-2013-set1)
* ! 1201: CCFM set JH-2013-set1 (1201 - 1213)
* ! 1301: CCFM set JH-2013-set2 (1301 - 1313)
* ! 1001: CCFM J2003 set 1
* ! 1002: CCFM J2003 set 2
* ! 1003: CCFM J2003 set 3
* ! 1010: CCFM set A0
* ! 1011: CCFM set A0+

```

```

*          ! 1012: CCFM set A0-
*          ! 1013: CCFM set A1
*          ! 1020: CCFM set B0
*          ! 1021: CCFM set B0+
*          ! 1022: CCFM set B0-
*          ! 1023: CCFM set B1
*          ! 1: CCFM old set JS2001
*          ! 2: derivative of collinear gluon (GRV)
*          ! 3: Bluemlein
*          ! 4: KMS
*          ! 5: GBW (saturation model)
*          ! 6: KMR
*          ! 7: Ryskin,Shabelski
* ++++++ BASES/SPRING Integration procedure ++++++
'NCAL'      1      0      50000 ! (D=20000) Nr of calls per iteration for bases
'ACC1'      1      0      1.0   ! (D=1) relative prec.(
'ACC2'      1      0      0.5   ! (0.5) relative prec.(
* ++++++
*'INTE'      1      0      0     ! Interaction type (D=0)
*          ! = 0 electromagnetic interaction
*'KT1 '      1      0      0.44 ! (D=0.0) intrinsic kt for beam 1
*'KT2 '      1      0      0.44 ! (D=0.0) intrinsic kt for beam 2
*'KTRE'      1      0      0.35 ! (D=0.35) primordial kt when non-trivial
*          ! target remnant is split into two particles
* Les Houches Accord Interface
'ILHA'      1      0      0     ! (D=10) Les Houches Accord
*          ! = 0 use internal CASCADE
*          ! = 1 write event file
*          ! = 10 call PYTHIA for final state PS and remnant frag
* path for updf files
* 'UPDF'      './share'

```

10 Program installation

CASCADE3 now follows the standard AUTOMAKE convention. To install the program, do the following

1) Get the source from <http://www.desy.de/~jung/cascade>

```

tar xvfz cascade-XXXX.tar.gz
cd cascade-XXXX

```

2) Generate the Makefiles (do not use shared libraries)

```

./configure --disable-shared --prefix=install-path --with-lhapdf="lhpdflib_path"
--with-tmdlib="TMDlib_path" --with-hepmc="hepmc_path"

```

with (as example):

```

lhpdflib_path=/Users/jung/MCgenerators/lhapdf/6.2.1/local
TMDlib_path=/Users/jung/jung/cvs/TMDlib/TMDlib2/local
hepmc_path=/Users/jung/MCgenerators/hepmc/HepMC-2.06.09/local

```

3) Compile the binary

```
make
```

4) Install the executable and PDF files

```
make install
```

4) The executable is in bin

run it with:

```

export CASEED=1242425
export HEPMCOUT=outfile.hepmc

```

```
cd $install-path/bin
```

```
./cascade < $install-path/share/cascade/LHE/steering-DY-MCatNLO.txt
```

Acknowledgements FG acknowledges the support and hospitality of DESY, Hamburg, where part of this work started. FH acknowledges the hospitality and support of DESY, Hamburg and of CERN, Theory Division while parts of this work were being done. SB, ALi and MM are grateful the DESY Directorate for the support in the framework of Cooperation Agreement between MSU and DESY on phenomenology of the LHC processes and TMD parton densities. MM was supported by a grant of the foundation for the advancement of theoretical physics and mathematics “Basis” 20-1-3-11-1. STM thanks the Humboldt Foundation for the Georg Forster research fellowship and gratefully acknowledges support from IPM. ALe acknowledges funding by Research Foundation-Flanders (FWO) (application number: 1272421N). QW and HY acknowledge the support by the Ministry of Science and Technology under Grant no. 2018YFA040390 and by the National Natural Science Foundation of China under Grant no. 11661141008.

Data Availability Statement This manuscript has no associated data or the data will not be deposited. [Authors’ comment: The source code of the program can be downloaded from <https://www.desy.de/protect/unhbox\voidb\@x\penalty\@M\jung/cascade/>.]

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. Funded by SCOAP³.

Appendix A: Program summary

Title of Program: CASCADE3 3.1.0

Computer for which the program is designed and others on which it is operable: any with standard Fortran 77 (gfortran)

Programming Language used: FORTRAN 77

High-speed storage required: No

Separate documentation available: No

Keywords: QCD, TMD parton distributions.

Method of solution Since measurements involve complex cuts and multi-particle final states, the ideal tool for any theoretical description of the data is a Monte Carlo event generator which generates initial state parton showers according to Transverse Momentum Dependent (TMD) parton densities, in a backward evolution, which follows the evolution equation as used for the determination of the TMD.

Restrictions on the complexity of the problem Any LHE file (with on-shell or off-shell) initial state partons can be processed.

Other Program used PYTHIA (version > 6.4) for final state parton shower and hadronization, BASES/SPRING 5.1 for integration (both supplied with the program package),

TMDLIB as a library for TMD parton densities.

Download of the program <https://www.desy.de/~jung/cascade/>

Unusual features of the program None

References

1. J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni et al., *JHEP* **1407**, 079 (2014). [arXiv:1405.0301](https://arxiv.org/abs/1405.0301)
2. S. Frixione, B.R. Webber (2006) [arXiv:hep-ph/0612272](https://arxiv.org/abs/hep-ph/0612272)
3. S. Frixione, P. Nason, B.R. Webber, *JHEP* **08**, 007 (2003). [arXiv:hep-ph/0305252](https://arxiv.org/abs/hep-ph/0305252)
4. S. Frixione, B.R. Webber (2002) [arXiv:hep-ph/0207182](https://arxiv.org/abs/hep-ph/0207182)
5. S. Frixione, B.R. Webber, *JHEP* **06**, 029 (2002). [arXiv:hep-ph/0204244](https://arxiv.org/abs/hep-ph/0204244)
6. S. Alioli, K. Hamilton, P. Nason, C. Oleari, E. Re, *JHEP* **04**, 081 (2011). [arXiv:1012.3380](https://arxiv.org/abs/1012.3380)
7. S. Frixione, P. Nason, C. Oleari, *JHEP* **0711**, 070 (2007). [arXiv:0709.2092](https://arxiv.org/abs/hep-ph/0709.2092)
8. M. Bahr, S. Gieseke, M. Gigg, D. Grellscheid, K. Hamilton et al., *Eur. Phys. J. C* **58**, 639 (2008). [arXiv:0803.0883](https://arxiv.org/abs/0803.0883)
9. T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C.O. Rasmussen, P.Z. Skands, *Comput. Phys. Commun.* **191**, 159 (2015). [arXiv:1410.3012](https://arxiv.org/abs/1410.3012)
10. J. Alwall, S. Hoche, F. Krauss, N. Lavesson, L. Lonnblad et al., *Eur. Phys. J. C* **53**, 473 (2008). [arXiv:0706.2569](https://arxiv.org/abs/0706.2569)
11. S. Catani, F. Krauss, R. Kuhn, B.R. Webber, *JHEP* **11**, 063 (2001). [arXiv:hep-ph/0109231](https://arxiv.org/abs/hep-ph/0109231)
12. R. Frederix, S. Frixione, *JHEP* **12**, 061 (2012). [arXiv:1209.6215](https://arxiv.org/abs/1209.6215)
13. K. Hamilton, P. Nason, G. Zanderighi, *JHEP* **10**, 155 (2012). [arXiv:1206.3572](https://arxiv.org/abs/1206.3572)
14. S. Catani, M. Ciafaloni, F. Hautmann, *Phys. Lett. B* **242**, 97 (1990)
15. M. Ciafaloni, *Nucl. Phys. B* **296**, 49 (1988)
16. S. Catani, F. Fiorani, G. Marchesini, *Phys. Lett. B* **234**, 339 (1990)
17. S. Catani, F. Fiorani, G. Marchesini, *Nucl. Phys. B* **336**, 18 (1990)
18. G. Marchesini, *Nucl. Phys. B* **445**, 49 (1995). [arXiv:hep-ph/9412327](https://arxiv.org/abs/hep-ph/9412327)
19. F. Hautmann, H. Jung, A. Lelek, V. Radescu, R. Zlebcik, *JHEP* **01**, 070 (2018). [arXiv:1708.03279](https://arxiv.org/abs/1708.03279)
20. F. Hautmann, H. Jung, A. Lelek, V. Radescu, R. Zlebcik, *Phys. Lett. B* **772**, 446 (2017). [arXiv:1704.01757](https://arxiv.org/abs/1704.01757)
21. R. Angeles-Martinez et al., *Acta Phys. Pol. B* **46**, 2501 (2015). [arXiv:1507.05267](https://arxiv.org/abs/1507.05267)
22. J. Alwall et al., *Comput. Phys. Commun.* **176**, 300 (2007). [arXiv:hep-ph/0609017](https://arxiv.org/abs/hep-ph/0609017)
23. J.C. Collins, D.E. Soper, *Nucl. Phys. B* **193**, 381 (1981). (Erratum: *Nucl. Phys. B* **213**, 545 (1983))
24. J.C. Collins, D.E. Soper, *Nucl. Phys. B* **194**, 445 (1982)
25. J.C. Collins, D.E. Soper, G.F. Sterman, *Nucl. Phys. B* **223**, 381 (1983)
26. J.C. Collins, D.E. Soper, G.F. Sterman, *Phys. Lett. B* **109**, 388 (1982)
27. J.C. Collins, D.E. Soper, G.F. Sterman, *Nucl. Phys. B* **250**, 199 (1985)
28. J. Collins, *Foundations of Perturbative QCD. Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology*, vol. 32 (2011)
29. R. Meng, F.I. Olness, D.E. Soper, *Phys. Rev. D* **54**, 1919 (1996). [arXiv:hep-ph/9511311](https://arxiv.org/abs/hep-ph/9511311)
30. P.M. Nadolsky, D.R. Stump, C.P. Yuan, *Phys. Rev. D* **61**, 014003 (2000). (Erratum: *Phys. Rev. D* **64**, 059903 (2001)). [arXiv:hep-ph/9906280](https://arxiv.org/abs/hep-ph/9906280)
31. P.M. Nadolsky, D.R. Stump, C.P. Yuan, *Phys. Rev. D* **64**, 114011 (2001). [arXiv:hep-ph/0012261](https://arxiv.org/abs/hep-ph/0012261)

32. X.-D. Ji, J.-P. Ma, F. Yuan, Phys. Rev. D **71**, 034005 (2005). [arXiv:hep-ph/0404183](#)
33. X.-D. Ji, J.-P. Ma, F. Yuan, Phys. Lett. B **597**, 299 (2004). [arXiv:hep-ph/0405085](#)
34. M.G. Echevarria, A. Idilbi, I. Scimemi, JHEP **07**, 002 (2012). [arXiv:1111.4996](#)
35. J.-Y. Chiu, A. Jain, D. Neill, I.Z. Rothstein, Phys. Rev. Lett. **108**, 151601 (2012). [arXiv:1104.0881](#)
36. E.M. Levin, M.G. Ryskin, Y.M. Shabelski, A.G. Shuvaev, Sov. J. Nucl. Phys. **53**, 657 (1991)
37. J.C. Collins, R.K. Ellis, Nucl. Phys. B **360**, 3 (1991)
38. F. Hautmann, Phys. Lett. B **535**, 159 (2002). [arXiv:hep-ph/0203140](#)
39. E. Avsar (2012) [arXiv:1203.1916](#)
40. E. Avsar, Int. J. Mod. Phys. Conf. Ser. **04**, 74 (2011). [arXiv:1108.1181](#)
41. S. Jadach, M. Skrzypek, Acta Phys. Pol. B **40**, 2071 (2009). [arXiv:0905.1399](#)
42. F. Dominguez, *Unintegrated Gluon Distributions at Small-x*. Ph.D. Thesis, Columbia U (2011)
43. F. Dominguez, J.-W. Qiu, B.-W. Xiao, F. Yuan, Phys. Rev. D **85**, 045003 (2012). [arXiv:1109.6293](#)
44. F. Dominguez, A. Mueller, S. Munier, B.-W. Xiao, Phys. Lett. B **705**, 106 (2011). [arXiv:1108.1752](#)
45. F. Hautmann, Acta Phys. Pol. B **40**, 2139 (2009)
46. F. Hautmann, M. Hentschinski, H. Jung (2012) [arXiv:1205.6358](#)
47. F. Hautmann, H. Jung, Nucl. Phys. Proc. Suppl. **184**, 64 (2008). [arXiv:0712.0568](#)
48. S. Catani, M. Ciafaloni, F. Hautmann, Phys. Lett. B **307**, 147 (1993)
49. S. Catani, F. Hautmann, Nucl. Phys. B **427**, 475 (1994). [arXiv:hep-ph/9405388](#)
50. A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, G. Watt, Eur. Phys. J. C **75**, 132 (2015). [arXiv:1412.7420](#)
51. S. Dooling, P. Gunnellini, F. Hautmann, H. Jung, Phys. Rev. D **87**, 094009 (2013). [arXiv:1212.6164](#)
52. F. Hautmann, H. Jung, Eur. Phys. J. C **72**, 2254 (2012). [arXiv:1209.6549](#)
53. M. Bengtsson, T. Sjöstrand, M. van Zijl, Z. Phys. C **32**, 67 (1986)
54. S. Catani, M. Ciafaloni, F. Hautmann, Nucl. Phys. B Proc. Suppl. **29**, 182 (1992)
55. G. Marchesini, B.R. Webber, Nucl. Phys. B **386**, 215 (1992)
56. F. Hautmann, H. Jung, S.T. Monfared, Eur. Phys. J. C **74**, 3082 (2014). [arXiv:1407.5935](#)
57. E.A. Kuraev, L.N. Lipatov, V.S. Fadin, Sov. Phys. JETP **44**, 443 (1976)
58. E.A. Kuraev, L.N. Lipatov, V.S. Fadin, Sov. Phys. JETP **45**, 199 (1977)
59. I.I. Balitsky, L.N. Lipatov, Sov. J. Nucl. Phys. **28**, 822 (1978)
60. H. Jung, S. Baranov, M. Deak, A. Grebenyuk, F. Hautmann et al., Eur. Phys. J. C **70**, 1237 (2010). [arXiv:1008.0152](#)
61. A. van Hameren, Comput. Phys. Commun. **224**, 371 (2018). [arXiv:1611.00680](#)
62. A. Lipatov, M. Malyshev, S. Baranov, Eur. Phys. J. C **80**, 330 (2020). [arXiv:1912.04204](#)
63. S. Catani, M. Ciafaloni, F. Hautmann, Nucl. Phys. B **366**, 135 (1991)
64. V. Saleev, N. Zotov, Mod. Phys. Lett. A **9**, 151 (1994). (**Erratum: Mod. Phys. Lett. A 9, 1517–1518 (1994)**)
65. A. Lipatov, N. Zotov, Eur. Phys. J. C **27**, 87 (2003). [arXiv:hep-ph/0210310](#)
66. S. Baranov, N. Zotov, J. Phys. G **29**, 1395 (2003). [arXiv:hep-ph/0302022](#)
67. S.P. Baranov, Phys. Rev. D **66**, 114003 (2002)
68. S. Baranov, Phys. Rev. D **84**, 054012 (2011)
69. S.P. Baranov, A.V. Lipatov, N.P. Zotov, Phys. Rev. D **78**, 014025 (2008). [arXiv:0805.4821](#)
70. M. Deak, F. Schwennsen, JHEP **09**, 035 (2008). [arXiv:0805.3763](#)
71. S. Marzani, R.D. Ball, Nucl. Phys. B **814**, 246 (2009). [arXiv:0812.3602](#)
72. M. Deak, F. Hautmann, H. Jung, K. Kutak, JHEP **09**, 121 (2009). [arXiv:0908.0538](#)
73. W. Furmanski, R. Petronzio, Z. Phys. C **11**, 293 (1982)
74. G. Curci, W. Furmanski, R. Petronzio, Nucl. Phys. B **175**, 27 (1980)
75. Y.L. Dokshitzer, Sov. Phys. JETP **46**, 641 (1977). (**Zh. Eksp. Teor. Fiz.** **73**, 1216 (1977))
76. G. Altarelli, G. Parisi, Nucl. Phys. B **126**, 298 (1977)
77. V.N. Gribov, L.N. Lipatov, Sov. J. Nucl. Phys. **15**, 438 (1972). (**Yad. Fiz.** **15**, 781 (1972))
78. Z. Nagy, D.E. Soper, Phys. Rev. D **102**, 014025 (2020). [arXiv:2002.04125](#)
79. L. Gellersen, D. Napoletano, S. Prestel, Monte Carlo studies, in *11th Les Houches Workshop on Physics at TeV Colliders: Les Houches 2019: Physics at TeV Colliders: Standard Model Working Group Report*, p. 131 (2020). Also in preprint 2003.01700
80. H. Jung, Comput. Phys. Commun. **143**, 100 (2002). [arXiv:hep-ph/0109102](#)
81. H. Jung, G.P. Salam, Eur. Phys. J. C **19**, 351 (2001). [arXiv:hep-ph/0012143](#)
82. S. Platzer, M. Sjödal, Eur. Phys. J. Plus **127**, 26 (2012). [arXiv:1108.6180](#)
83. F. Hautmann, H. Jung, JHEP **10**, 113 (2008). [arXiv:0805.1049](#)
84. F. Hautmann, H. Jung, M. Krämer, P. Mulders, E. Nocera et al., Eur. Phys. J. C **74**, 3220 (2014). [arXiv:1408.3015](#)
85. N.A. Abdulov et al., TMDlib2 and TMDplotter: a platform for 3D hadron structure studies. [arXiv:2103.09741](#)
86. F. Hautmann, H. Jung, Nucl. Phys. B **883**, 1 (2014). [arXiv:1312.7875](#)
87. A.B. Martinez, P. Connor, F. Hautmann, H. Jung, A. Lelek, V. Radescu, R. Zlebcik, Phys. Rev. D **99**, 074008 (2019). [arXiv:1804.11152](#)
88. A.B. Martinez et al., Eur. Phys. J. C **80**, 598 (2020). [arXiv:2001.06488](#)
89. A.B. Martinez et al., Phys. Rev. D **100**, 074027 (2019). [arXiv:1906.00919](#)
90. M. Dobbs, J.B. Hansen, Comput. Phys. Commun. **134**, 41 (2001)
91. A.B. Martinez et al., *Jet merging with TMD parton branching* (To be published) (2021)
92. A. Buckley, J. Butterworth, L. Lonnblad, D. Grellscheid, H. Hoeth, J. Monk, H. Schulz, F. Siegert, Comput. Phys. Commun. **184**, 2803 (2013). [arXiv:1003.0694](#)