

Tools for perturbative high-precision calculations

Tobias Neumann^{a,*}

^a*Department of Physics, Brookhaven National Laboratory, Upton, New York 11973, USA*

E-mail: tneumann@bnl.gov

Current higher-order calculations heavily rely on publicly available tools. These are not necessarily computer programs, but it is anything that aids in different steps of calculations, e.g. also includes published loop-amplitudes, integrals and algorithms that eventually lead to important automatization. In these proceedings we give an overview of tools used for perturbative precision calculations. The availability, composability and hackability of such tools are what lets our field flourish. And while our field is already positively standing out in the development of such tools, we would like to stress their importance for the success of future developments.

The Ninth Annual Conference on Large Hadron Collider Physics - LHCP2021
7-12 June 2021
Online

*Speaker

Some tools build the foundation of every calculation; among these are computer algebra packages like Wolfram Mathematica, Maple, FORM [1] and GiNaC [2]. In these proceedings we highlight a few specialized tools that are important for LHC precision calculations and phenomenology. The list of tools presented here is not exhaustive but merely gives a starting point for further exploration. We furthermore only cover publicly available tools, since these are the ones that allow for a benefit for the whole community and therefore are key in driving most progress.

Typical calculations of higher-order cross-sections involve multiple stages. First, the calculation of amplitude expressions in terms of loop integrals and the evaluation of the loop integrals. Then the development and use of subtraction methods to produce infrared finite results, and finally the convolution with parton distribution functions (PDFs). At the last stage event generators add parton shower, hadronization and matching and merging of multiple fixed-order calculations.

Amplitudes. The calculation of amplitudes can be a daunting task for processes with increasing number of scales, i.e. loops and legs, even without calculating the loop integrals. Even NLO calculations reach their practical limitations for a sufficient number of legs and internal masses. Traditionally, amplitudes are calculated using Feynman diagrams in a brute-force approach while using clever hand-tuned simplification rules in FORM or Mathematica. But the growth of algebraic complexity of modern problems is significant, and with growing expressions one runs into several issues. For example numerical cancellations can cause loss of precision in the numerical evaluation. Sufficiently large expressions are bound to run into issues with a fixed machine precision and also the growing evaluation time becomes problematic.

On the other hand we know that NLO (one-loop) calculations are meanwhile heavily automatized and expressions can be compact and fast. This is because of the progress in (numerical) unitarity techniques [3, 4] that can be efficiently automatized. A lot of progress has been made in applying numerical unitarity methods to two-loop amplitudes [5–7]. Another turning point is the widespread use of rational function reconstruction algorithms. These can be combined with numerical evaluations of amplitudes to obtain compact analytical expressions. A code that combines numerical unitarity with rational reconstruction for one- and two-loop processes is Caravel [8]. So far it is limited to QCD at leading color, but it is written in a modular way which can be extended to more complicated situations. An extensive and general framework for rational reconstruction is FiniteFlow [9], which can also be useful in other situations.

Reduction of loop integrals. Once the amplitude expressions have been obtained in terms of the loop integrals, the latter have to be calculated, analytically or numerically. A first step is the reduction to a set of master integrals. Tools like Reduze [10] and Fire [11–14] are established players in this field. Kira [15, 16] + FireFly [17, 18] offers strong competition since it automatically uses rational function reconstruction to handle intermediate expression growth. Note that FireFly is a dedicated program for rational function reconstruction and can also be used separately.

Even with such improvements the reduction of integral families at the current level of interest can take terabytes of computer memory. Therefore, further developments are necessary in the future. For example, it has been suggested to use the Baikov Feynman integral representation to keep the reduction system small, and additionally use multivariate partial fractioning to keep partial expressions small. Such an algorithm has been implemented in the Singular computer algebra

system [19]. A dedicated package for multivariate partial fractioning is meanwhile also available as `MultivariateApart` [20].

Evaluation of loop integrals. The first step when the solution of a loop integral is needed, is to check whether it has already been done. For that one can use the `Loopedia` online database [21]. In a narrower sense also solved integrals and explicit expressions of amplitudes count as tools, as long as they are presented in a machine readable way. Very successful examples of this are the diboson two-loop amplitudes published in the package `VVAmp` [10, 22]. The package is merely an algebraic amplitude expression, but fueled all the diboson NNLO calculations that we now have. There are several more examples that follow this approach at a more recent level of three-loops $2 \rightarrow 2$, three-loops $2 \rightarrow 1$ with masses, or two-loop $2 \rightarrow 3$.

The current standard technique for the evaluation of loop integrals is to use differential equations and to ideally find a canonical basis where the dimensional ϵ -dependence factors out [23], making it amenable for an analytical integration. Under certain conditions the integrals can be solved iteratively in terms of multiple polylogarithms. For the evaluation of such functions there are codes like `GiNaC` [2, 24] and computer algebra libraries like `PolyLogTools` [25] for algebraic manipulations. Libraries for the evaluation of simpler harmonic polylogs are available, e.g. `TDHPL` [26], `CHAPLIN` [27], `HPL` [28], `HPL4` [29] and `HPOLY` [30]. Most current NNLO calculation make use of these libraries.

Whether one has a canonical basis or not, one can solve the system of differential equations numerically. This can be done for example using the `Mathematica` package `DiffExp` [31]. But the canonical basis makes the evaluation significantly more efficient, both numerically and analytically.

Tools like `epsilon` [32], `Fuchsia` [33] and `Libra` [34] implement an algorithm to find a canonical basis for single scale problems. For multi-scale problems there is no general algorithm known, but the `CANONICA` package [35] is able to find a canonical basis in certain cases and has been used in several multi-loop calculations.

Apart from differential equations, other tools and methods exist, of course. For example in some cases `HyperInt` [36] can give a solution by direct integration of the Schwinger integral representation. In other situations deriving nested sum representations turns out to be successful, which can be handled using the `Sigma` package [37]. Mellin Barnes representations are also used in some cases and there are several `Mathematica` packages available for that purpose like `AMBRE` [38, 39] or `MB.m` [40, 41].

As a last resort and for cross-checks one can use fully numerical sector decomposition with codes like `FIESTA` [42–45] and `SecDec` [46–49]. These are intrinsically slow and “inefficient” compared to the previously discussed techniques, but they always work and complete multi-loop calculations have been set up with them.

Cross-sections. Obtaining a cross-section requires the combination of purely virtual corrections with real emission amplitudes for an infrared finite result. This is done using subtraction methods, which must also count as tools. There are meanwhile a dozen available at the level of NNLO. Unfortunately none reach the generality or flexibility of NLO methods yet [50, 51]. For slicing subtractions one typically starts with a factorization theorem, and for the necessary ingredients some automatization is being worked on. For example the library `SoftServe` [52, 53] allows for the

calculation of a class of soft functions in SCET, but with its current limitations it is currently only used in the context of resummation.

Real emission contributions to NNLO calculations involve NLO calculations with an additional leg that have to be calculated efficiently and numerically stable in infrared singular regions. The gold standard are optimized analytical expressions, as for example available in MCFM [54]. In general, one wants to make use of the great one-loop automatization advancements. Some libraries can give one-loop amplitudes with the press of a button; either for any process like in Recola [55–58] or for large pregenerated lists as in OpenLoops [59–61]. Further available codes are GoSam [62, 63] and Samurai [64]. The packages Recola and OpenLoops can also compute NLO electroweak corrections in their most recent versions. Last, libraries like QCDLoop [65, 66], Collier [67] and OneLOop [68] are available for the evaluation of one-loop integrals when used with analytical amplitude expressions, for example.

Parton distribution functions. The last part that enters collider cross-sections are PDFs. They meanwhile constitute an equal or larger uncertainty than the hard multi-loop piece for a lot of processes. This means we need the tools to push for their better understanding and higher precision.

The standard to interface PDF grids is LHAPDF [69], which offers a unified interface to dozens of fits. To make actual PDF fits, one has to evaluate cross-sections many times in the fitting process. This means that very fast evaluations with modified input parameters and PDFs are needed. For that purpose grid interpolation tools like fastNLO [70–72], ApplGrid [73] and PineAppl [74] have been developed. Toolkits like Hoppet [75] and Apfel [76] are used for the perturbative PDF evolution. Very recently the NNPDF collaboration published their complete fitting infrastructure with version 4.0 [77], which opens the way for everyone to study data tensions and fitting methodology systematics in more detail.

GPUs and Machine Learning. Emerging topics of significant interest in high-energy physics are machine learning and the use of GPUs. While these are clearly useful tools for data analysis, i.e. especially on the experimental side, for the frameworks covered in these proceedings their use is still under exploration. For example machine learning has been used for phase-space sampling [78–80] and amplitudes [81]. But the use of these techniques in the context of higher-order calculations where infrared singular regions are probed deeply and numerical precision is important, is likely difficult. GPUs also still have to prove to be useful for general multi-loop applications, because these typically consist of large amplitude expressions and complex algorithms. On the other hand, linear algebra problems and easily vectorizable problems like PDF evaluation can clearly benefit from GPU evaluation [82].

Conclusions. In these proceedings we have briefly surveyed public tools for higher-order perturbative collider calculations. We advocate for tools that are public, reusable, composable and hackable. We also advocate for tools that perform one thing and one thing well. Tools that try to achieve too many things at once are less suitable for the most complicated calculations on our higher-order wish list that need sufficient flexibility. These are core principles that let our community prosper, and we hereby encourage journals to include these criteria in evaluations. We also emphasize that machine-readable loop integrals and amplitudes and the development of algorithms also comprise such tools, as they are an integral part of higher-order calculations.

References

- [1] B. Ruijl, T. Ueda and J. Vermaseren, *FORM version 4.2*, [1707.06453](#).
- [2] C. W. Bauer, A. Frink and R. Kreckel, *Introduction to the GiNaC framework for symbolic computation within the C++ programming language*, *J. Symb. Comput.* **33** (2002) 1 [[cs/0004015](#)].
- [3] G. Ossola, C. G. Papadopoulos and R. Pittau, *Reducing full one-loop amplitudes to scalar integrals at the integrand level*, *Nucl. Phys. B* **763** (2007) 147 [[hep-ph/0609007](#)].
- [4] R. K. Ellis, W. T. Giele and Z. Kunszt, *A Numerical Unitarity Formalism for Evaluating One-Loop Amplitudes*, *JHEP* **03** (2008) 003 [[0708.2398](#)].
- [5] H. Ita, *Two-loop Integrand Decomposition into Master Integrals and Surface Terms*, *Phys. Rev. D* **94** (2016) 116015 [[1510.05626](#)].
- [6] S. Abreu, F. Febres Cordero, H. Ita, M. Jaquier and B. Page, *Subleading Poles in the Numerical Unitarity Method at Two Loops*, *Phys. Rev. D* **95** (2017) 096011 [[1703.05255](#)].
- [7] S. Abreu, F. Febres Cordero, H. Ita, M. Jaquier, B. Page and M. Zeng, *Two-Loop Four-Gluon Amplitudes from Numerical Unitarity*, *Phys. Rev. Lett.* **119** (2017) 142001 [[1703.05273](#)].
- [8] S. Abreu, J. Dormans, F. Febres Cordero, H. Ita, M. Kraus, B. Page et al., *Caravel: A C++ framework for the computation of multi-loop amplitudes with numerical unitarity*, *Comput. Phys. Commun.* **267** (2021) 108069 [[2009.11957](#)].
- [9] T. Peraro, *FiniteFlow: multivariate functional reconstruction using finite fields and dataflow graphs*, *JHEP* **07** (2019) 031 [[1905.08019](#)].
- [10] A. von Manteuffel and C. Studerus, *Reduze 2 - Distributed Feynman Integral Reduction*, [1201.4330](#).
- [11] A. V. Smirnov, *Algorithm FIRE – Feynman Integral REDuction*, *JHEP* **10** (2008) 107 [[0807.3243](#)].
- [12] A. V. Smirnov and V. A. Smirnov, *FIRE4, LiteRed and accompanying tools to solve integration by parts relations*, *Comput. Phys. Commun.* **184** (2013) 2820 [[1302.5885](#)].
- [13] A. V. Smirnov, *FIRE5: a C++ implementation of Feynman Integral REDuction*, *Comput. Phys. Commun.* **189** (2015) 182 [[1408.2372](#)].
- [14] A. V. Smirnov and F. S. Chuharev, *FIRE6: Feynman Integral REDuction with Modular Arithmetic*, *Comput. Phys. Commun.* **247** (2020) 106877 [[1901.07808](#)].
- [15] P. Maierhöfer, J. Usovitsch and P. Uwer, *Kira—A Feynman integral reduction program*, *Comput. Phys. Commun.* **230** (2018) 99 [[1705.05610](#)].
- [16] J. Klappert, F. Lange, P. Maierhöfer and J. Usovitsch, *Integral reduction with Kira 2.0 and finite field methods*, *Comput. Phys. Commun.* **266** (2021) 108024 [[2008.06494](#)].

- [17] J. Klappert and F. Lange, *Reconstructing rational functions with FireFly*, *Comput. Phys. Commun.* **247** (2020) 106951 [1904.00009].
- [18] J. Klappert, S. Y. Klein and F. Lange, *Interpolation of dense and sparse rational functions and other improvements in FireFly*, *Comput. Phys. Commun.* **264** (2021) 107968 [2004.01463].
- [19] D. Bendle, J. Boehm, M. Heymann, R. Ma, M. Rahn, L. Ristau et al., *Two-loop five-point integration-by-parts relations in a usable form*, **2104.06866**.
- [20] M. Heller and A. von Manteuffel, *MultivariateApart: Generalized Partial Fractions*, **2101.08283**.
- [21] C. Bogner, S. Borowka, T. Hahn, G. Heinrich, S. P. Jones, M. Kerner et al., *Loopedia, a Database for Loop Integrals*, *Comput. Phys. Commun.* **225** (2018) 1 [1709.01266].
- [22] A. von Manteuffel and L. Tancredi, *The two-loop helicity amplitudes for $gg \rightarrow V_1 V_2 \rightarrow 4$ leptons*, *JHEP* **06** (2015) 197 [1503.08835].
- [23] J. M. Henn, *Lectures on differential equations for Feynman integrals*, *J. Phys. A* **48** (2015) 153001 [1412.2296].
- [24] S. Weinzierl, *Symbolic expansion of transcendental functions*, *Comput. Phys. Commun.* **145** (2002) 357 [math-ph/0201011].
- [25] C. Duhr and F. Dulat, *PolyLogTools — polylogs for the masses*, *JHEP* **08** (2019) 135 [1904.07279].
- [26] T. Gehrmann and E. Remiddi, *Numerical evaluation of two-dimensional harmonic polylogarithms*, *Comput. Phys. Commun.* **144** (2002) 200 [hep-ph/0111255].
- [27] S. Buehler and C. Duhr, *CHAPLIN - Complex Harmonic Polylogarithms in Fortran*, *Comput. Phys. Commun.* **185** (2014) 2703 [1106.5739].
- [28] D. Maitre, *HPL, a mathematica implementation of the harmonic polylogarithms*, *Comput. Phys. Commun.* **174** (2006) 222 [hep-ph/0507152].
- [29] M. Czakon, J. Gluza and T. Riemann, *Master integrals for massive two-loop bhabha scattering in QED*, *Phys. Rev. D* **71** (2005) 073009 [hep-ph/0412164].
- [30] J. Ablinger, J. Blümlein, M. Round and C. Schneider, *Numerical Implementation of Harmonic Polylogarithms to Weight $w = 8$* , *Comput. Phys. Commun.* **240** (2019) 189 [1809.07084].
- [31] M. Hidding, *DiffExp, a Mathematica package for computing Feynman integrals in terms of one-dimensional series expansions*, *Comput. Phys. Commun.* **269** (2021) 108125 [2006.05510].
- [32] M. Prausa, *epsilon: A tool to find a canonical basis of master integrals*, *Comput. Phys. Commun.* **219** (2017) 361 [1701.00725].

- [33] O. Gituliar and V. Magerya, *Fuchsia: a tool for reducing differential equations for Feynman master integrals to epsilon form*, *Comput. Phys. Commun.* **219** (2017) 329 [1701.04269].
- [34] R. N. Lee, *Libra: A package for transformation of differential systems for multiloop integrals*, *Comput. Phys. Commun.* **267** (2021) 108058 [2012.00279].
- [35] C. Meyer, *Algorithmic transformation of multi-loop master integrals to a canonical basis with CANONICA*, *Comput. Phys. Commun.* **222** (2018) 295 [1705.06252].
- [36] E. Panzer, *Algorithms for the symbolic integration of hyperlogarithms with applications to Feynman integrals*, *Comput. Phys. Commun.* **188** (2015) 148 [1403.3385].
- [37] C. Schneider, *Symbolic summation in difference rings and applications*, in *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation, ISSAC '16*, (New York, NY, USA), p. 9–12, Association for Computing Machinery, 2016, DOI.
- [38] J. Gluza, K. Kajda and T. Riemann, *AMBRE: A Mathematica package for the construction of Mellin-Barnes representations for Feynman integrals*, *Comput. Phys. Commun.* **177** (2007) 879 [0704.2423].
- [39] I. Dubovyk, J. Gluza, T. Riemann and J. Usovitsch, *Numerical integration of massive two-loop Mellin-Barnes integrals in Minkowskian regions*, *PoS LL2016* (2016) 034 [1607.07538].
- [40] M. Czakon, *Automatized analytic continuation of Mellin-Barnes integrals*, *Comput. Phys. Commun.* **175** (2006) 559 [hep-ph/0511200].
- [41] A. V. Smirnov and V. A. Smirnov, *On the Resolution of Singularities of Multiple Mellin-Barnes Integrals*, *Eur. Phys. J. C* **62** (2009) 445 [0901.0386].
- [42] A. V. Smirnov and M. N. Tentyukov, *Feynman Integral Evaluation by a Sector decomposition Approach (FIESTA)*, *Comput. Phys. Commun.* **180** (2009) 735 [0807.4129].
- [43] A. V. Smirnov, V. A. Smirnov and M. Tentyukov, *FIESTA 2: Parallelizeable multiloop numerical calculations*, *Comput. Phys. Commun.* **182** (2011) 790 [0912.0158].
- [44] A. V. Smirnov, *FIESTA 3: cluster-parallelizable multiloop numerical calculations in physical regions*, *Comput. Phys. Commun.* **185** (2014) 2090 [1312.3186].
- [45] A. V. Smirnov, *FIESTA4: Optimized Feynman integral calculations with GPU support*, *Comput. Phys. Commun.* **204** (2016) 189 [1511.03614].
- [46] J. Carter and G. Heinrich, *SecDec: A general program for sector decomposition*, *Comput. Phys. Commun.* **182** (2011) 1566 [1011.5493].
- [47] S. Borowka, J. Carter and G. Heinrich, *Numerical Evaluation of Multi-Loop Integrals for Arbitrary Kinematics with SecDec 2.0*, *Comput. Phys. Commun.* **184** (2013) 396 [1204.4152].
- [48] S. Borowka, G. Heinrich, S. P. Jones, M. Kerner, J. Schlenk and T. Zirke, *SecDec-3.0: numerical evaluation of multi-scale integrals beyond one loop*, *Comput. Phys. Commun.* **196** (2015) 470 [1502.06595].

- [49] S. Borowka, G. Heinrich, S. Jahn, S. P. Jones, M. Kerner, J. Schlenk et al., *pySecDec: a toolbox for the numerical evaluation of multi-scale integrals*, *Comput. Phys. Commun.* **222** (2018) 313 [1703.09692].
- [50] S. Catani and M. H. Seymour, *A General algorithm for calculating jet cross-sections in NLO QCD*, *Nucl. Phys. B* **485** (1997) 291 [hep-ph/9605323].
- [51] S. Frixione, Z. Kunszt and A. Signer, *Three jet cross-sections to next-to-leading order*, *Nucl. Phys. B* **467** (1996) 399 [hep-ph/9512328].
- [52] G. Bell, R. Rahn and J. Talbert, *Two-loop anomalous dimensions of generic dijet soft functions*, *Nucl. Phys. B* **936** (2018) 520 [1805.12414].
- [53] G. Bell, R. Rahn and J. Talbert, *Generic dijet soft functions at two-loop order: correlated emissions*, *JHEP* **07** (2019) 101 [1812.08690].
- [54] J. M. Campbell, S. Höche and C. T. Preuss, *Accelerating LHC phenomenology with analytic one-loop amplitudes: A C++ interface to MCFM*, **2107.04472**.
- [55] S. Actis, A. Denner, L. Hofer, A. Scharf and S. Uccirati, *Recursive generation of one-loop amplitudes in the Standard Model*, *JHEP* **04** (2013) 037 [1211.6316].
- [56] S. Actis, A. Denner, L. Hofer, J.-N. Lang, A. Scharf and S. Uccirati, *RECOLA: REcursive Computation of One-Loop Amplitudes*, *Comput. Phys. Commun.* **214** (2017) 140 [1605.01090].
- [57] A. Denner, J.-N. Lang and S. Uccirati, *NLO electroweak corrections in extended Higgs Sectors with RECOLA2*, *JHEP* **07** (2017) 087 [1705.06053].
- [58] A. Denner, J.-N. Lang and S. Uccirati, *Recola2: REcursive Computation of One-Loop Amplitudes 2*, *Comput. Phys. Commun.* **224** (2018) 346 [1711.07388].
- [59] F. Cascioli, P. Maierhofer and S. Pozzorini, *Scattering Amplitudes with Open Loops*, *Phys. Rev. Lett.* **108** (2012) 111601 [1111.5206].
- [60] F. Buccioni, S. Pozzorini and M. Zoller, *On-the-fly reduction of open loops*, *Eur. Phys. J. C* **78** (2018) 70 [1710.11452].
- [61] F. Buccioni, J.-N. Lang, J. M. Lindert, P. Maierhöfer, S. Pozzorini, H. Zhang et al., *OpenLoops 2*, *Eur. Phys. J. C* **79** (2019) 866 [1907.13071].
- [62] G. Cullen, N. Greiner, G. Heinrich, G. Luisoni, P. Mastrolia, G. Ossola et al., *Automated One-Loop Calculations with GoSam*, *Eur. Phys. J. C* **72** (2012) 1889 [1111.2034].
- [63] G. Cullen et al., *GOSAM-2.0: a tool for automated one-loop calculations within the Standard Model and beyond*, *Eur. Phys. J. C* **74** (2014) 3001 [1404.7096].
- [64] P. Mastrolia, G. Ossola, T. Reiter and F. Tramontano, *Scattering AMplitudes from Unitarity-based Reduction Algorithm at the Integrand-level*, *JHEP* **08** (2010) 080 [1006.0710].

- [65] R. K. Ellis and G. Zanderighi, *Scalar one-loop integrals for QCD*, *JHEP* **02** (2008) 002 [0712.1851].
- [66] S. Carrazza, R. K. Ellis and G. Zanderighi, *QCDLoop: a comprehensive framework for one-loop scalar integrals*, *Comput. Phys. Commun.* **209** (2016) 134 [1605.03181].
- [67] A. Denner, S. Dittmaier and L. Hofer, *Collier: a fortran-based Complex One-Loop Library in Extended Regularizations*, *Comput. Phys. Commun.* **212** (2017) 220 [1604.06792].
- [68] A. van Hameren, *OneLoop: For the evaluation of one-loop scalar functions*, *Comput. Phys. Commun.* **182** (2011) 2427 [1007.4716].
- [69] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht et al., *LHAPDF6: parton density access in the LHC precision era*, *Eur. Phys. J. C* **75** (2015) 132 [1412.7420].
- [70] T. Kluge, K. Rabbertz and M. Wobisch, *FastNLO: Fast pQCD calculations for PDF fits*, in *14th International Workshop on Deep Inelastic Scattering*, pp. 483–486, 9, 2006, [hep-ph/0609285](#), DOI.
- [71] FASTNLO collaboration, *New features in version 2 of the fastNLO project*, in *20th International Workshop on Deep-Inelastic Scattering and Related Subjects*, pp. 217–221, 2012, [1208.3641](#), DOI.
- [72] D. Britzger, G. S. Klaus Rabbertz, F. Stober and M. Wobisch, *Recent Developments of the fastNLO Toolkit*, *PoS DIS2015* (2015) 055.
- [73] T. Carli, D. Clements, A. Cooper-Sarkar, C. Gwenlan, G. P. Salam, F. Siegert et al., *A posteriori inclusion of parton density functions in NLO QCD final-state calculations at hadron colliders: The APPLGRID Project*, *Eur. Phys. J. C* **66** (2010) 503 [0911.2985].
- [74] S. Carrazza, E. R. Nocera, C. Schwan and M. Zaro, *PineAPPL: combining EW and QCD corrections for fast evaluation of LHC processes*, *JHEP* **12** (2020) 108 [2008.12789].
- [75] G. P. Salam and J. Rojo, *A Higher Order Perturbative Parton Evolution Toolkit (HOPPET)*, *Comput. Phys. Commun.* **180** (2009) 120 [0804.3755].
- [76] V. Bertone, S. Carrazza and J. Rojo, *APFEL: A PDF Evolution Library with QED corrections*, *Comput. Phys. Commun.* **185** (2014) 1647 [1310.1394].
- [77] R. D. Ball et al., *The Path to Proton Structure at One-Percent Accuracy*, [2109.02653](#).
- [78] C. Gao, S. Höche, J. Isaacson, C. Krause and H. Schulz, *Event Generation with Normalizing Flows*, *Phys. Rev. D* **101** (2020) 076002 [2001.10028].
- [79] C. Gao, J. Isaacson and C. Krause, *i-flow: High-dimensional Integration and Sampling with Normalizing Flows*, *Mach. Learn. Sci. Tech.* **1** (2020) 045023 [2001.05486].
- [80] M. D. Klimek and M. Perelstein, *Neural Network-Based Approach to Phase Space Integration*, *SciPost Phys.* **9** (2020) 053 [1810.11509].

- [81] S. Badger and J. Bullock, *Using neural networks for efficient evaluation of high multiplicity scattering amplitudes*, *JHEP* **06** (2020) 114 [2002.07516].
- [82] S. Carrazza, J. M. Cruz-Martinez and M. Rossi, *PDFFlow: Parton distribution functions on GPU*, *Comput. Phys. Commun.* **264** (2021) 107995 [2009.06635].