

# HighTEA: High energy Theory Event Analyser

---

Michał Czakon,<sup>a</sup> Zahari Kassabov,<sup>b</sup> Alexander Mitov,<sup>c</sup> Rene Poncelet,<sup>c</sup> Andrei Popescu<sup>c</sup>

<sup>a</sup>*Institut für Theoretische Teilchenphysik und Kosmologie, RWTH Aachen University, D-52056 Aachen, Germany*

<sup>b</sup>*DAMTP, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WA, United Kingdom*

<sup>c</sup>*Cavendish Laboratory, University of Cambridge, Cambridge CB3 0HE, United Kingdom*

*E-mail:* [mczakon@physik.rwth-aachen.de](mailto:mczakon@physik.rwth-aachen.de), [zk261@cam.ac.uk](mailto:zk261@cam.ac.uk), [adm74@cam.ac.uk](mailto:adm74@cam.ac.uk),  
[poncelet@hep.phy.cam.ac.uk](mailto:poncelet@hep.phy.cam.ac.uk), [andrei.popescu@cantab.net](mailto:andrei.popescu@cantab.net)

ABSTRACT: We introduce HighTEA, a new paradigm for deploying fully-differential next-to-next-to leading order (NNLO) calculations for collider observables. In principle, any infrared safe observable can be computed and, with very few restrictions, the user has complete freedom in defining their calculation's setup. For example, one can compute generic  $n$ -dimensional distributions, can define kinematic variables and factorization/renormalization scales, and can modify the strong coupling and parton distributions. HighTEA operates on the principle of analyzing precomputed events. It has all the required hardware and software infrastructure such that users only need to request their calculation via the internet before receiving the results, typically within minutes, in the form of a histogram. No specialized knowledge or computing infrastructure is required to fully utilize HighTEA, which could be used by both experts in particle physics and the general public. The current focus is on all classes of LHC processes. Extensions beyond NNLO, or to  $e^+e^-$  colliders, are natural next steps.

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>HighTEA in a nutshell</b>	<b>2</b>
2.1	Utility of HighTEA	2
2.2	HighTEA is designed to be used by <i>everyone</i>	3
2.3	Limitations of HighTEA	3
2.4	Under the hood: a gentle introduction to the inner workings of HighTEA	4
<b>3</b>	<b>High-level overview of HighTEA's structure and usage</b>	<b>5</b>
3.1	Server	6
3.2	Front end	7
3.2.1	API	7
3.2.2	CLI	7
3.2.3	Web form	7
3.2.4	Python library	8
<b>4</b>	<b>Examples of uses and test cases</b>	<b>8</b>
<b>5</b>	<b>Conclusions</b>	<b>10</b>

---

## 1 Introduction

The so-called *NLO revolution*, which peaked about 10 years ago, resulted in two important developments: first, the technology for performing any next-to-leading (NLO) calculation was established and refined and, second, a number of flexible, publicly available computer codes [1–6] and general purpose event generators [7–11] appeared. The constant flow of ever-increasing-in-precision LHC measurements, however, quickly made it clear that the demand was shifting towards calculations with precision higher than NLO.

Calculations with next-to-next-to-leading (NNLO) accuracy satisfy this requirement for almost all LHC processes, while for selected few cases N<sup>3</sup>LO accuracy is required and has been achieved [12–20]. The technology for performing NNLO calculations for a generic LHC process has already been developed and is fairly well established [21–30]. It has resulted in the calculation of almost all  $2 \rightarrow 2$  and many  $2 \rightarrow 3$  processes. What is lacking at present is the possibility for easily obtaining such results. To be specific, NNLO calculations at present are being performed either with non-public codes (which are not directly accessible by general users), or with a small number of public codes [31, 32] which can compute an important but

nonetheless limited range of processes. Running such programs already requires a relatively high level of sophistication on the side of the user. Additionally, NNLO calculations are computationally very expensive and the vast majority of users do not have such computational resources at their disposal. For example, a reasonably straightforward NNLO calculation takes about 50k CPU hours which already requires significant, cluster-sized computing infrastructure. On the other hand, some of the most demanding NNLO calculations performed to date [33, 34] have required up to tens of millions of CPU hours, even with the relative maturity and multi-year optimization efforts they have benefited from. For these reasons it is unlikely that fast, general, publicly available NNLO codes will appear in the near future.

With the clear demand for NNLO predictions, the question then arises how NNLO calculations can easily be made available to the general user. In this work we propose one possible solution, called `HighTEA`.

What is `HighTEA`? This is a web service which stores Monte Carlo events produced in the course of a NNLO calculation, such that they can be analyzed in new calculations with minimal effort. While the basic idea is not new [35–38], the utility of `HighTEA` stems from the fact that it offers extensive functionality for efficiently storing, processing, analyzing and visualizing the results of such a calculation.

Who is `HighTEA` for? It is for *everyone*. `HighTEA` is structured in such a way that it requires neither technical knowledge of higher-order perturbative calculations nor computing infrastructure or technical resources.

In the following sections we explain in detail the features and workings of `HighTEA`.

## 2 HighTEA in a nutshell

### 2.1 Utility of HighTEA

`HighTEA`'s utility can be summarized in the following way:

1. It allows the user to compute *any* infrared safe  $n$ -dimensional differential distribution in *any* process which has already been added to the library of available processes. The up-to-date list of all available processes can be obtained from `HighTEA` itself (see sec. 4).
2. The output of a `HighTEA` computation is a histogram, and the input is the histogram's specification. In addition, `HighTEA` automatically produces nice plots of the computed histograms. If experimental data is available, the user can directly be provided with a theory/data comparison plot, including all relevant theory uncertainties: MC, scale and pdf. This is in addition to the data uncertainty, if known. See sec. 4 for an example.
3. Predictions at present are in fixed-order perturbation theory, but can potentially be extended to include resummed/showered events.
4. `HighTEA` is not tied to a specific perturbative order. Any perturbative order that can be computed with Monte Carlo methods can be included. At present, our main goal is to offer NNLO QCD accurate corrections for LHC processes.

5. `HighTEA` “knows” most standard kinematic variables relevant for a given process. For added flexibility, users can additionally define their own kinematic variables in terms of commonly available mathematical functions [39].
6. `HighTEA` offers the user *complete* control over the functional forms of the factorization and renormalization scales. This is achieved similarly to the user-defined kinematic variables discussed above. Scale variation is flexible, too; one can choose among the standard 3-point or 7-point variations or can define a custom variation.
7. Any pdf set available in the LHAPDF library [40] can be used. `HighTEA` produces pdf variation based on either standard or specialized (recommended - see sec. 4) pdf sets.
8. The jet size in processes involving jets can also be varied.

## 2.2 `HighTEA` is designed to be used by *everyone*

Here are the main reasons why one would want to use `HighTEA` :

1. With the help of `HighTEA` one can perform their own state-of-the-art calculation without the need for complex computing codes or the knowledge of a technical subject like higher-order perturbative calculations.
2. It can be used to compute processes that may not be available in public codes or to cross-check and/or supplement existing codes and calculations.
3. No need for major computing infrastructures (typically, a large cluster). Any user can obtain their own state-of-the-art predictions that are equivalent to a full-fledged NNLO calculation using only their own personal computer or smart phone.
4. Predictions derived from `HighTEA` are very fast. A typical calculation takes between seconds and an hour, depending on the setup.

## 2.3 Limitations of `HighTEA`

Like any other tool, `HighTEA` has its limitations:

1. The values of parameters intrinsic to the collider or the process cannot be changed. Examples are the collider c.o.m. energy and the value of the top quark mass in processes involving top quarks.
2. Predictions for processes with no intrinsic hard transverse scale (jet production, for example) are subject to a set of pre-defined generation-level cuts. These are needed for the definition of the process. The user does not need to be too concerned with those beyond ensuring that every event that is accepted by the user’s cuts is also accepted by the generation cuts.

3. Fixed Monte Carlo statistics. The statistics of theoretical events is fixed and the user cannot increase it. This means that if one requests, for example, very small bins and/or is interested in the tails of kinematic distributions, one may get predictions with unsuitably large MC uncertainty. `HighTEA` always returns an estimate of the MC uncertainty, i.e. the user is always provided with a quantitative measure of how reliable the prediction is in terms of statistics. If needed, MC statistics can be increased by the authors of `HighTEA` upon request. One should keep in mind that separate evaluations using the same dataset are not statistically independent and, for example, one cannot reduce the MC uncertainty by averaging multiple calculations.
4. Only processes that have been added to the library can be analyzed. If the process of interest is not available please request it from the `HighTEA` authors.

The above limitations are not unusual and are, in fact, inherent to any Monte Carlo program. Here are some hints about how one can deal with some of these limitations.

Some parameters like the value of the strong coupling constant, pdf and scales are at the complete disposal of the user. Other parameters, like the top quark mass, cannot be changed, yet predictions for a set of values are often required. A practical solution is to make available with `HighTEA` three datasets that have different values of  $m_t$ : one around the world average and two more with  $m_t$  values that are about  $\pm 1$  GeV away from it. The user can then interpolate on a per-bin basis between these 3 values to derive approximate prediction for any  $m_t$  value within this range. An extensive application of this interpolation technique can be found in ref. [41].

Whenever generation-level cuts are applied to a process, this is specified in the info file for that process (see sec. 4 for information about how to access it). As a rule, such cuts are applied to typical for that process variables like a suitable invariant mass or  $H_T$ . It is easy to ensure consistency when the user applies cuts to the same variable, because in such a case all one needs is a user-level cut that is more exclusive than the generation-level one. In case user-level cuts are applied to a different variable, the user must ensure consistency between the cuts. This can be a non-trivial task, however it is an important one for maintaining the correctness of the results.

## 2.4 Under the hood: a gentle introduction to the inner workings of `HighTEA`

To be able to make state of the art calculations and take full advantage of `HighTEA` one does not need to have any knowledge about how it works. The purpose of this section is to satisfy the curiosity of the reader about the inner workings of the program.

`HighTEA` analyzes precomputed and stored weighted “theoretical events”. An event is a set of parton id’s, parton momenta and a weight. The weight represents the value of a probability density distribution in one point (specified by the set of momenta and partonic fractions  $x_{1,2}$  and/or  $z$ ). Beyond the leading order, an observable receives contributions from additional partonic channels and from final states with different multiplicities. Furthermore, the cancellation of infrared divergences necessitates the inclusion of so-called “counterevents”

(these are also weighted events but are derived from reduced matrix elements). For these reasons, in the context of MC event generation, a complete prediction is expressed not through a single event but by a set of events that typically have different dimensionalities. `HighTEA` automatically accounts for all such contributions.

When a user submits a request for an analysis, `HighTEA` analyzes all saved events and bins them accordingly. The user can modify parameters like the value of the strong coupling constant, the pdf set and the renormalization and factorization scales. This is achieved by re-weighting each event during the course of the analysis. In order to gain access to these scales, to  $\alpha_S$  and to pdfs, an event's weight is saved as an array which contains the coefficients of all terms of the type  $\log^n(\mu_F) \log^m(\mu_R)$ , and factorizes the dependence on  $\alpha_S$  and pdfs. It is then easy to reconstruct the complete weight for any value or functional form of these two scales, as well as for any value of the strong coupling constant and pdf set available via the `LHAPDF` library.

We partially unweight events to reduce the size of the tables of events. We introduce a maximum weight to which we unweight using a hit-and-miss Monte-Carlo. Events with larger event weight than this maximum are kept using their weight. The value of the maximum weight is tuned to balance a high acceptance rate and a minimal number of weighted events. A higher acceptance rate translates directly to shorter computation times for a given sample size. At the same time, a smaller number of weighted events minimizes the resulting Monte Carlo uncertainty for a sample of a given size. The working point depends on the process and the optimization of the phase space sampling. Related ideas have been discussed in refs. [42–44].

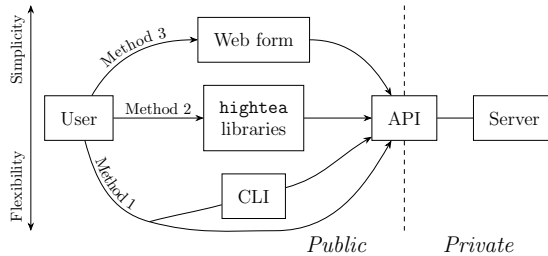
Processes with jets deserve special attention. For such processes, `HighTEA` retains the full parton-level information for all events/counterevents. Nonetheless, there is an implicit dependence on a jet algorithm and jet radius  $R$  since, during the generation of the events, generation-level selection cuts are applied at the jet level. Once an event passes the jet-level selection requirement, all information about the event is saved at the parton-level. When the user requests a calculation with jets, the jet algorithm specified by the user is used to cluster the partons into jets. Correct results can only be guaranteed if the user uses the same jet algorithm as the algorithm used during the generation of the dataset. This information is available in the process info file.

### 3 High-level overview of `HighTEA`'s structure and usage

This section is the central part of this paper and should be read by all users. It gives a high-level overview of the structure and usage of this project and explains everything in non-technical, intuitive terms. The depth is sufficient so a user with basic knowledge can start using the project right away.

From the point of view of the user, `HighTEA` is naturally divided in two parts, see fig. 1:

- **Server:** it is completely hidden from the user; the user has no direct access to it and in effect it should be thought of by the users as a black box,



**Figure 1.** High-level structure of HighTEA. Shown are library’s main components together with the three access methods. The vertical dashed line delineates the separation of public and private components.

- **Front end:** directly accessible by the user. It allows the user to submit requests and to receive and visualize results.

### 3.1 Server

We expect this section to be of limited interest and will provide a brief overview only. The server side consists of the databases for all precomputed processes as well as various software facilities that enable the analysis of these databases and the transfer of results and/or requests to/from the user interface. The sizes of the various databases vary greatly from process to process and can be between hundreds of GBytes to many TBytes. For ease of storage, analysis and server memory management, each dataset is split into a number of compressed files of approximately equal size. The database files are stored in the parquet format [45], which is designed for efficient data storage and retrieval.

A typical analysis can take between seconds and an hour, depending on the size of the database and how extensive the requested analysis is. The bulk of the time is spent not on reading the database but on the various mathematical operations needed to recompute scales and especially pdf sets. In fact, re-computations with different pdf sets tend to be the slowest part of any computation. The time is proportional to the number of pdf members used. For this reason we recommend the use of Specialized Minimal pdf (SMPDF) sets [46] whenever pdf variation is required and when such a set has been made available by the authors.

The CPU and memory consumption for such an analysis can also be significant but are within the capabilities of a standard <sup>1</sup> server.

The software on the server side also deals with the queuing of requests and the storing of the final results. We have implemented a solution where all requests and results are stored long-term and can be retrieved by the user with the help of a token they are provided with when their request is submitted. For this reason, the user may disconnect once a job request is submitted and can retrieve the job at a later time.

<sup>1</sup>Currently, we are using a DELL PowerEdge R620 (2 x 8-core Xeon E5-2650v2) server with 64GB memory.

## 3.2 Front end

The user interface offers three possible methods of accessing the database of HighTEA, see fig. 1. These methods are designed with different types of users in mind:

1. Directly accessing the project's API.
2. Accessing the project through a Python library. Suitable for the vast majority of users. It is expected to be the most utilized way for accessing the project.
3. Accessing the project via a web form. Suitable for every user. It is trivial to use however it offers less flexibility and sophistication and not all features can be utilized. This method is suitable for quick checks and demonstrations but less so for serious work.

### 3.2.1 API

The API is the only point through which the server can be accessed. It utilizes standard web communication protocols and can be accessed from anywhere on the web via JSON format queries. The complete description of the API can be found in the automatically generated online document [47]. Further details about the API can be found here [48].

The API allows the user to utilize every feature available in HighTEA. API's direct use, however, requires significant technical sophistication since the user needs to prepare their own query in a JSON format, submit it to the corresponding web address, receive and store the token which uniquely identifies the request and, once the request has been processed, retrieve the results by querying the web access point and retrieving the results which are also in a JSON format.

### 3.2.2 CLI

We have provided a Command Line Interface utility, described here [48], which simplifies the task of submitting JSON requests to the API.

### 3.2.3 Web form

For the vast majority of users we have designed two more access methods: via our web form and via our Python library.

The web form access [49] is self explanatory. It represents a standard web form where one can fill in the request, press a button, and then wait for the results to be displayed. Once the process is selected, on the right hand side of the page a detailed information about the process is displayed. In essence, this is the content of the info file mentioned above. Please note that the web form is under constant improvement and so it may evolve with time. Its current functionality is limited by web design constraints but we could imagine future improvements that will make it potentially as flexible and as useable as the Python library.



### 3.2.4 Python library

The most versatile and recommended way of accessing the project is via the Python library. A detailed description of the library can be found here [50]. The fact that the library is written in Python is not essential since, in order to use it, no knowledge of Python is required. We have created higher level wrapper functions which hide the Python language specifics and allow the user to interact with it using high-level intuitive commands. Of course, the user also has direct access to all low-level commands in the Python language. The user-friendliness and visual appeal is ensured by our embedding of the functionalities in a JupyterLab Notebook Interface [51]. The way a user interacts with a Jupyter Notebook is very similar to the Notebooks of popular computer algebra systems like Mathematica [52] and Maple [53].

We have created two different ways of using this library. The first one involves its installation on a personal computer. This is a fairly standard process but it may require the installation of various standard Python libraries. This step should be fairly straightforward for people who are familiar with Python or the basics of installing software on Unix-like platforms. The second way is much easier since it involves no software installation at all. For this reason it is our default recommendation for using HighTEA. In this second approach Jupyter Notebooks are run directly in the cloud. There are many websites and web-based services which allow users to run and collaborate on their own Jupyter Notebooks. One such option is the Google Research Colaboratory (a.k.a. Colab) [54]. It offers the possibility to install, share and collaborate on Jupyter Notebooks. As of the writing of this paper this service is free and only requires a free Google account. On this page [55] we have provided a number of HighTEA examples and a tutorial in the form of ready-to-use Jupyter Notebooks which the user can immediately run directly on Colab. The Notebooks and their results can be saved for later use on the user's Google Drive.

## 4 Examples of uses and test cases

In this section we describe a typical workflow for using HighTEA. It assumes the default Jupyter notebook approach for accessing HighTEA described in sec. 3.2.4. The following discussion can be translated to the other ways of accessing HighTEA in a rather straightforward way.

Before we delve into the specifics of how jobs are submitted and results retrieved, there is one general aspect of HighTEA related to authentication that needs to be clarified. HighTEA is meant to be used concurrently by a large number of users who may submit a large number of jobs nearly simultaneously and these jobs may take rather different times to execute. This aspect requires a system of labeling jobs and users over a potentially extended period of time. In HighTEA we have implemented the following process.

**Job identification.** When submitted, a job is assigned a unique ID in the form of a token. The token is returned to the user who submits the job and is displayed in the Jupyter notebook. They are stored in the `.job` file associated with the given job. The token is needed since it allows users to exit their Notebook right after the job is submitted and then restart the notebook at a later time and retrieve the result. To achieve this the user simply executes

all lines of the Notebook. The Notebook will automatically submit the appropriate token to the HighTEA server and retrieve the result. Results stored on the HighTEA server currently do not expire.

**User authentication.** Internally, HighTEA keeps record of all submitted requests and outputs. To allow users to retrieve their past jobs even without having to keep track of all individual job tokens, we have introduced a user authentication system. If desired, a user can obtain a personalized token which needs to be used with all job submissions. This allows users to retrieve all their past jobs. An example can be found in the Notebook `Example-authentication.ipynb` in [55]. At present users are not required to authenticate and can submit jobs as anonymous users.

**Notebook setup.** The example Notebooks available at [55] are ready to be run. Beforehand, the user needs to clone the desired Notebook to their own Google Drive, see [50] for details. A Notebook saved on the local Drive can directly be run, one does not need to go back to the page [55]. Users can create new Notebooks out of the several examples provided at [55] and use them in a way that suits their needs.

**The actual calculation.** In the rest of this section we introduce the HighTEA functionality, using as an example the Notebook `Example-ttbar.ipynb`. This Notebook consists of the following steps:

- Setup of the notebook,
- Download experimental data from `www.hepdata.net` for each one of the available differential distributions,
- Convert the downloaded data to a format convenient for further processing,
- Automatically extract from the data the binnings of each variable,
- Specify the mathematical expressions for all measured observables,
- For each order of interest (LO, NLO, NNLO, etc) prepare a job (called `job` in this example) by giving it the following attributes, and then submit the job:
  - Initialize `job` by giving it a name,
  - Specify the process ( $t\bar{t}$  production at the LHC at 13 TeV in this case),
  - Specify all one-dimensional distributions that will be computed,
  - Specify if scale uncertainty is required (by saying what scale variation to use: 3-point, 7-point or custom),
  - Submit `job` to HighTEA using the command `job.request()`,
  - Place the results, when ready, in a new object called `jobs`.
- Pre-format data and theory predictions for plotting (using a special Python function `Run`; more info about it can be obtained by typing `?Run`),

- Plot all observables.

In the plots, one can see data (as bars) together with LO, NLO and NNLO predictions. Both the absolute distributions and their ratio w/r to LO are plotted. Shown is the scale variation band as well as the MC uncertainty as a vertical bar for each curve and bin.

The above example does not exhaust all options and possibilities offered by `HighTEA`. For example, the user can print the computed differential distributions in the form of tables using the command `job.show_result()` and then use them in their own plotting routines. Some of the other example Notebooks show its usage.

Here are some more functionalities:

- List all available processes: see the Notebook `Example-lower-level-interface.ipynb`,
- Output the information about a process: `job.process('pp_tt_13000_172.5')` (for the specific process in the above example),
- Define new variables: `job.define_new_variable('name', 'definition')`,
- The renormalization and factorization scales to be used: `job.scales('μR', 'μF')`,
- Specify the pdf set to be used. For example: `job.pdf('CT14nnlo')`.
- To introduce phase space cuts one uses, for example, `job.cuts('pt_t > 30')`, see the Tutorial Notebook for details.
- Generic  $n$ -dimensional differential cross sections can be computed, see the example Notebook `Example-ttbar-simple-2D.ipynb`.
- For calculations of processes with jets see the Notebook `Example-inclusive-jets.ipynb`.

One can also request pdf variation by typing `job.pdf_variation()`. The pdf prescription relevant for the requested pdf set is automatically used. A note about pdf variation and timings: LO and NLO calculations are very fast, and tend to be of the order of seconds. NNLO calculations are much slower and as a rule take about 10 to 100 times longer. As mentioned above, the speed of a calculation is directly related to the number of scales and pdf members used. For this reason when pdf variation is requested a calculation can be slowed down by a factor of about 100. We have implemented by default SMPDF sets, for the processes for which such sets are available, since such sets reduce the time for computing pdf variation by a factor of about 10. If full pdf variation is required one can specify this by typing `job.pdf_variation('full')`.

## 5 Conclusions

In this work we introduce `HighTEA`: a novel approach for distributing and performing fast NNLO calculations for collider processes. The main idea is simple: the `HighTEA` authors pre-compute “theoretical” events using Monte Carlo methods, and store the events for later analysis. To avoid burdening users with technical aspects like transferring, storing and analyzing

vast amounts of data, we have developed the complete software and hardware infrastructure required for dealing with these tasks. The only thing the user needs to do is to request a calculation via the web and then receive the result in the form of a binned distribution, accompanied by a nice plot. This way, the user is completely shielded from the intricacies of doing NNLO calculations, the need for securing huge computing resources and from having to have even basic programming experience. In fact, as of the writing of this article, a user only needs a free Google account and a device for accessing the Internet (like a personal computer or a smart phone) in order to request and receive, *typically within minutes*, a most sophisticated calculation. For further details, and to access ready-to-run examples, please visit HighTEA’s homepage:

<https://www.precision.hep.phy.cam.ac.uk/hightea>

The main restrictions when using HighTEA are that one can only compute processes that have been already been implemented by HighTEA’s authors, and that the statistics of events is fixed, i.e. the user cannot increase it. Our hope is to keep adding new processes on a regular basis. The number of events included in the various datasets is chosen in such a way that at NNLO the MC error is subdominant to the scale one for a typical experimental LHC setup. The number of events can be increased upon request.

All processes which have been included in HighTEA have been carefully validated versus independent calculations available in the literature. The relevant papers are provided with the info files accompanying each dataset. When using HighTEA please cite the present publication together with the said papers.

HighTEA offers new opportunities in terms of access to cutting precision perturbative calculations for collider processes. For example, the lack until now of accessible NNLO predictions has often precluded the usage of such calculations in various searches and related LHC analyses. HighTEA is meant to effectively close this gap. HighTEA can handle distributions of any dimension and any infrared safe observable, and its predictions are very fast.

HighTEA offers new possibilities for comparing theory to data which is available in alternative and more flexible formats like the unbinned distributions recently proposed in ref. [56], or the CMS Open Data initiative [57]. HighTEA makes it particularly easy to automatically compare theoretical predictions to data which is available in the HEPData repository [58]. A working example is provided.

There are many more potential uses of HighTEA, all of which are reasonably easy to implement and pursue:

- Allow other providers to add their own calculations to the HighTEA. Examples of potential uses are NLO calculations of EW corrections and SMEFT, like refs. [59–62].
- Allow users to access individual pdf channels; incorporate scale variations correlated with pdfs [63].

- Allow users to access contributions from specific powers of couplings (i.e. not just LO, NLO, etc). Such a feature would be relevant for calculations in the full Standard Model or beyond.
- `HighTEA` is very well suited for adding contributions from EFT operators in addition to the usual QCD or SM contributions. This will allow users to directly request an analysis in a complete SMEFT framework.
- In addition to bins and plots, it is relatively easy to modify `HighTEA` by extending our implementation [64] so it can also output `fastNLO` [65, 66] or related [67–69] grids.

Finally, we would like to stress that while `HighTEA` has been developed in the LHC era and, naturally, is currently focused on predictions for LHC physics, it can equally well handle other colliders. Given the increasing interest in future  $e^+e^-$  colliders, `HighTEA` applications to  $e^+e^-$  processes will be particularly useful.

## Acknowledgments

We thank Steve Wotton for his server infrastructure support. The work of M.C. was supported by the Deutsche Forschungsgemeinschaft under grant 396021762 - TRR 257; support by a grant of the Bundesministerium für Bildung und Forschung (BMBF) is additionally acknowledged. The research of Z.K., A.M., R.P. and A.P. has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 Research and Innovation Programme (grant agreement no. 683211). Z.K. is also supported by the European Research Council under the European Union’s Horizon 2020 research and innovation Programme (grant agreement n.950246). A.M. was also supported by the UK STFC grants ST/L002760/1 and ST/K004883/1. R.P. acknowledges support from the Leverhulme Trust and the Isaac Newton Trust. A.M. and R.P. acknowledge the use of the DiRAC Cumulus HPC facility under Grant No. PPSP226.

## References

- [1] S. Frixione, Z. Kunszt and A. Signer, Nucl. Phys. B **467**, 399-442 (1996) [arXiv:hep-ph/9512328 [hep-ph]].
- [2] S. Catani and M. H. Seymour, Nucl. Phys. B **485**, 291-419 (1997) [erratum: Nucl. Phys. B **510**, 503-504 (1998)] [arXiv:hep-ph/9605323 [hep-ph]].
- [3] Z. Nagy, Phys. Rev. D **68**, 094002 (2003) [arXiv:hep-ph/0307268 [hep-ph]].
- [4] J. M. Campbell, R. K. Ellis and C. Williams, JHEP **07**, 018 (2011) [arXiv:1105.0020 [hep-ph]].
- [5] J. M. Campbell, R. K. Ellis and W. T. Giele, Eur. Phys. J. C **75**, no.6, 246 (2015) [arXiv:1503.06182 [physics.comp-ph]].
- [6] J. Campbell and T. Neumann, JHEP **12**, 034 (2019) [arXiv:1909.09117 [hep-ph]].

- [7] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer, H. S. Shao, T. Stelzer, P. Torrielli and M. Zaro, *JHEP* **07**, 079 (2014) [arXiv:1405.0301 [hep-ph]].
- [8] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert and J. Winter, *JHEP* **02**, 007 (2009) [arXiv:0811.4622 [hep-ph]].
- [9] E. Bothmann *et al.* [Sherpa], *SciPost Phys.* **7**, no.3, 034 (2019) [arXiv:1905.09127 [hep-ph]].
- [10] S. Frixione, P. Nason and C. Oleari, *JHEP* **11**, 070 (2007) [arXiv:0709.2092 [hep-ph]].
- [11] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen and P. Z. Skands, *Comput. Phys. Commun.* **191**, 159-177 (2015) [arXiv:1410.3012 [hep-ph]].
- [12] C. Anastasiou, C. Duhr, F. Dulat, F. Herzog and B. Mistlberger, *Phys. Rev. Lett.* **114**, 212001 (2015) [arXiv:1503.06056 [hep-ph]].
- [13] X. Chen, T. Gehrmann, E. W. N. Glover, A. Huss, B. Mistlberger and A. Pelloni, *Phys. Rev. Lett.* **127**, no.7, 072002 (2021) [arXiv:2102.07607 [hep-ph]].
- [14] S. Camarda, L. Cieri and G. Ferrera, *Phys. Rev. D* **104**, no.11, L111503 (2021) [arXiv:2103.04974 [hep-ph]].
- [15] X. Chen, T. Gehrmann, N. Glover, A. Huss, T. Z. Yang and H. X. Zhu, *Phys. Rev. Lett.* **128**, no.5, 052001 (2022) [arXiv:2107.09085 [hep-ph]].
- [16] X. Chen, T. Gehrmann, E. W. N. Glover, A. Huss, P. F. Monni, E. Re, L. Rottoli and P. Torrielli, *Phys. Rev. Lett.* **128**, no.25, 252001 (2022) [arXiv:2203.01565 [hep-ph]].
- [17] T. Neumann and J. Campbell, *Phys. Rev. D* **107**, no.1, L011506 (2023) [arXiv:2207.07056 [hep-ph]].
- [18] C. Duhr, F. Dulat and B. Mistlberger, *JHEP* **11**, 143 (2020) [arXiv:2007.13313 [hep-ph]].
- [19] J. Baglio, C. Duhr, B. Mistlberger and R. Szafron, *JHEP* **12**, 066 (2022) [arXiv:2209.06138 [hep-ph]].
- [20] S. Camarda, L. Cieri and G. Ferrera, [arXiv:2303.12781 [hep-ph]].
- [21] A. Gehrmann-De Ridder, T. Gehrmann and E. W. N. Glover, *JHEP* **09**, 056 (2005) [arXiv:hep-ph/0505111 [hep-ph]].
- [22] J. Currie, E. W. N. Glover and J. Pires, *Phys. Rev. Lett.* **118**, no.7, 072002 (2017) [arXiv:1611.01460 [hep-ph]].
- [23] J. Gaunt, M. Stahlhofen, F. J. Tackmann and J. R. Walsh, *JHEP* **09**, 058 (2015) [arXiv:1505.04794 [hep-ph]].
- [24] V. Del Duca, C. Duhr, A. Kardos, G. Somogyi and Z. Trócsányi, *Phys. Rev. Lett.* **117**, no.15, 152004 (2016) [arXiv:1603.08927 [hep-ph]].
- [25] S. Catani and M. Grazzini, *Phys. Rev. Lett.* **98**, 222002 (2007) [arXiv:hep-ph/0703012 [hep-ph]].
- [26] R. Boughezal, C. Focke, X. Liu and F. Petriello, *Phys. Rev. Lett.* **115**, no.6, 062002 (2015) [arXiv:1504.02131 [hep-ph]].
- [27] M. Czakon, *Phys. Lett. B* **693**, 259-268 (2010) [arXiv:1005.0274 [hep-ph]].
- [28] M. Czakon and D. Heymes, *Nucl. Phys. B* **890**, 152-227 (2014) [arXiv:1408.2500 [hep-ph]].

- [29] F. Caola, K. Melnikov and R. Röntsch, *Eur. Phys. J. C* **77**, no.4, 248 (2017) [arXiv:1702.01352 [hep-ph]].
- [30] L. Magnea, E. Maina, G. Pelliccioli, C. Signorile-Signorile, P. Torrielli and S. Uccirati, *JHEP* **12**, 107 (2018) [erratum: *JHEP* **06**, 013 (2019)] [arXiv:1806.09570 [hep-ph]].
- [31] R. Boughezal, J. M. Campbell, R. K. Ellis, C. Focke, W. Giele, X. Liu, F. Petriello and C. Williams, *Eur. Phys. J. C* **77**, no.1, 7 (2017) [arXiv:1605.08011 [hep-ph]].
- [32] M. Grazzini, S. Kallweit and M. Wiesemann, *Eur. Phys. J. C* **78**, no.7, 537 (2018) [arXiv:1711.06631 [hep-ph]].
- [33] M. Czakon, A. Mitov and R. Poncelet, *Phys. Rev. Lett.* **127**, no.15, 152001 (2021) [erratum: *Phys. Rev. Lett.* **129**, no.11, 119901 (2022); erratum: *Phys. Rev. Lett.* **129**, no.11, 119901 (2022)] [arXiv:2106.05331 [hep-ph]].
- [34] M. Alvarez, J. Cantero, M. Czakon, J. Llorente, A. Mitov and R. Poncelet, *JHEP* **03**, 129 (2023) [arXiv:2301.01086 [hep-ph]].
- [35] Z. Bern, L. J. Dixon, F. Febres Cordero, S. Höche, H. Ita, D. A. Kosower and D. Maitre, *Comput. Phys. Commun.* **185**, 1443-1460 (2014) [arXiv:1310.7439 [hep-ph]].
- [36] J. R. Andersen, S. Bartle, Z. Bern, F. Febres Cordero, S. Höche, D. A. Kosower, H. Ita, N. A. Lo Presti, D. Maitre and K. Ozeren, *PoS LL2014*, 079 (2014) [arXiv:1407.1621 [hep-ph]].
- [37] D. Maitre, *J. Phys. Conf. Ser.* **1085**, no.5, 052017 (2018)
- [38] D. Maitre, *J. Phys. Conf. Ser.* **1525**, no.1, 012014 (2020)
- [39] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg and N. J. Smith, *et al.* *Nature* **585**, no.7825, 357-362 (2020) [arXiv:2006.10256 [cs.MS]].
- [40] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr and G. Watt, *Eur. Phys. J. C* **75**, 132 (2015) [arXiv:1412.7420 [hep-ph]].
- [41] A. M. Cooper-Sarkar, M. Czakon, M. A. Lim, A. Mitov and A. S. Papanastasiou, [arXiv:2010.04171 [hep-ph]].
- [42] J. R. Andersen, A. Maier and D. Maitre, [arXiv:2303.15246 [hep-ph]].
- [43] B. Nachman and J. Thaler, *Phys. Rev. D* **102**, no.7, 076004 (2020) [arXiv:2007.11586 [hep-ph]].
- [44] K. T. Matchev and P. Shyamsundar, *SciPost Phys.* **10**, no.2, 034 (2021) [arXiv:2006.16972 [hep-ph]].
- [45] Apache Parquet data file format <https://parquet.apache.org>
- [46] S. Carrazza, S. Forte, Z. Kassabov and J. Rojo, *Eur. Phys. J. C* **76**, no.4, 205 (2016) [arXiv:1602.00005 [hep-ph]].
- [47] Online documentation of the HighTEA API <https://www.hep.phy.cam.ac.uk/hightea/docs>
- [48] Complete online HighTEA client-side documentation <https://hightea-client.readthedocs.io>
- [49] Access to HighTEA via webform <https://www.hep.phy.cam.ac.uk/hightea/webform/>

- [50] Access to HighTEA via our Python library  
<https://www.precision.hep.phy.cam.ac.uk/hightea/google-colab/>
- [51] The open-source software project Jupyter <https://jupyter.org>
- [52] Wolfram Mathematica <https://www.wolfram.com/mathematica/>
- [53] Maplesoft Maple <https://www.maplesoft.com/products/Maple/>
- [54] Google Research Colaboratory <https://colab.research.google.com>
- [55] Access HighTEA at Google Colab here: <https://colab.research.google.com/github/HighteaCollaboration/hightea-examples/blob/master/Start.ipynb>
- [56] M. Arratia, A. Butter, M. Campanelli, V. Croft, D. Gillberg, A. Ghosh, K. Lohwasser, B. Malaescu, V. Mikuni and B. Nachman, *et al.* JINST **17**, no.01, P01024 (2022) [arXiv:2109.13243 [hep-ph]].
- [57] CMS Open Data initiative <http://opendata.cern.ch>
- [58] HEPData repository of collider data <https://www.hepdata.net>
- [59] R. Cepedello, F. Esser, M. Hirsch and V. Sanz, [arXiv:2302.03485 [hep-ph]].
- [60] T. Giani, G. Magni and J. Rojo, [arXiv:2302.06660 [hep-ph]].
- [61] S. Moretti, L. Panizzi, J. Sjölin and H. Waltari, [arXiv:2302.03401 [hep-ph]].
- [62] Z. Kassabov, M. Madigan, L. Mantani, J. Moore, M. M. Alvarado, J. Rojo and M. Ubiali, [arXiv:2303.06159 [hep-ph]].
- [63] Z. Kassabov, M. Ubiali and C. Voisey, JHEP **03**, 148 (2023) [arXiv:2207.07616 [hep-ph]].
- [64] M. Czakon, D. Heymes and A. Mitov, [arXiv:1704.08551 [hep-ph]].
- [65] D. Britzger *et al.* [fastNLO], [arXiv:1208.3641 [hep-ph]].
- [66] The fastNLO interpolation grids library <https://fastnlo.hepforge.org>
- [67] T. Carli, D. Clements, A. Cooper-Sarkar, C. Gwenlan, G. P. Salam, F. Siegert, P. Starovoitov and M. Sutton, Eur. Phys. J. C **66**, 503-524 (2010) [arXiv:0911.2985 [hep-ph]].
- [68] S. Carrazza, E. R. Nocera, C. Schwan and M. Zaro, JHEP **12**, 108 (2020) [arXiv:2008.12789 [hep-ph]].
- [69] A. Buckley, A. Kvellestad, A. Raklev, P. Scott, J. V. Sparre, J. Van Den Abeele and I. A. Vazquez-Holm, Eur. Phys. J. C **80**, no.12, 1106 (2020) [arXiv:2006.16273 [hep-ph]].