ATLAS Open Event Generation

LHC BSM Working Group general meeting

13 November 2025

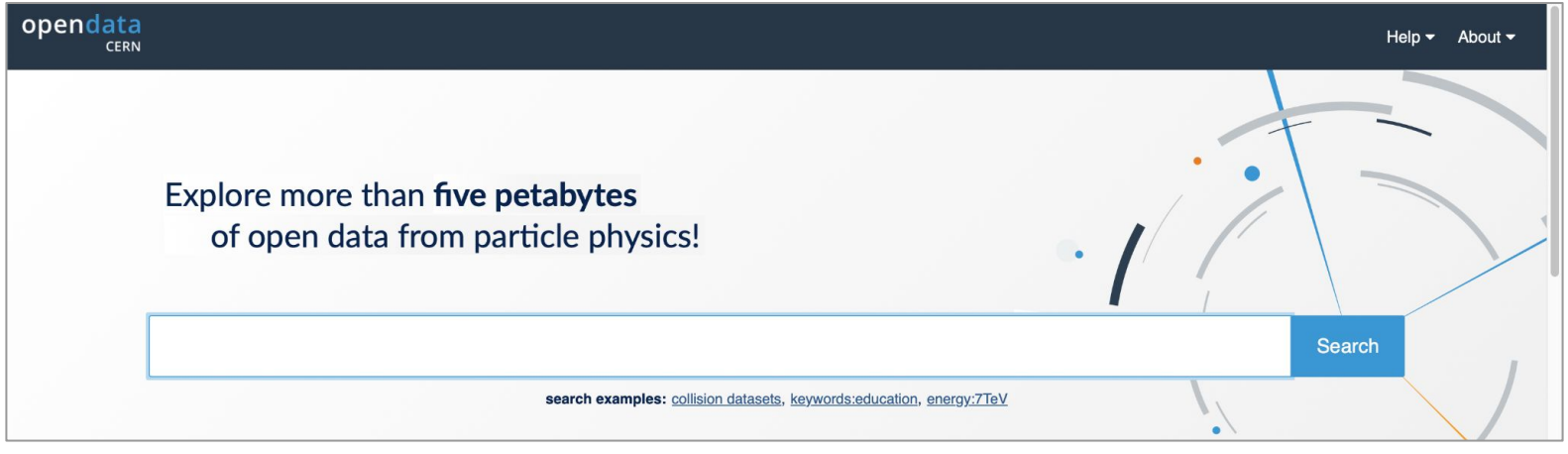
Mariana Vivas Albornoz (DESY) on behalf of the ATLAS Collaboration

What Have We Released?

ATLAS has released substantial event generation.

Initial release of 24M (~1TB) events for specialised SM Z-boson as a test sample.

- Available in the CERN Open Data portal.



⚠️ If using the data, please cite these records using their DOI.

What Have We Released?

ATLAS has released substantial event generation.

Initial release of 24M (~1TB) events for specialised SM Z-boson as a test sample.

Currently deploying 12.77B events in 6213 samples.

- Available in `atlasopenmagic` (more in a moment)

ATLAS Open Magic 🪄📊

Tests passing | pypi 1.6.1 | codecov 100%

`atlasopenmagic` is a Python package made to simplify working with ATLAS Open Data by providing utilities to manage metadata and URLs for streaming the data.

⚠️ The data are fine to be used but there is just a risk of breaking changes in some APIs.

What Have We Released?

ATLAS has released substantial event generation.

Initial release of 24M (~1TB) events for specialised SM Z-boson as a test sample.

Currently deploying 12.77B events in 6213 samples.

Ultimate plan for this year: 13.4B events in ~6541 samples (~900 TB)

SM and BSM

Different generators

HEPMC format (txt.tgz)

13 TeV and 13.6 TeV

Enough events for double the effective luminosity

‼ Have **not committed to any updates** of these samples. It will depend on community interest and engagement

What Have We Released?

ATLAS has released substantial event generation.

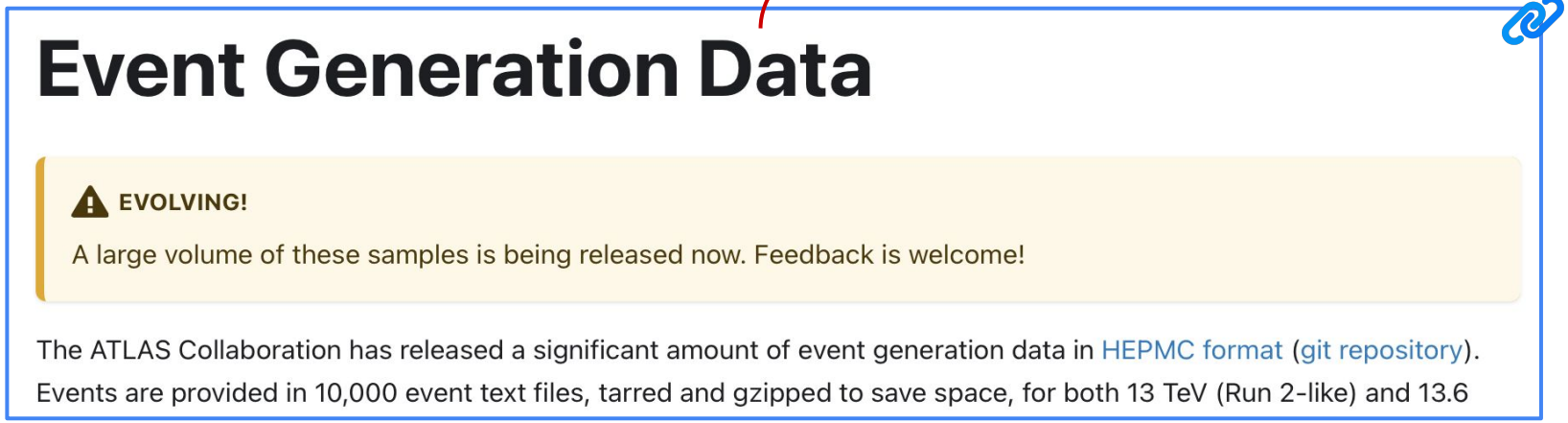
Initial release of 24M (~1TB) events for specialised SM Z-boson as a test sample.

Currently deploying 12.77B events in 6213 samples.

Ultimate plan for this year: 13.4B events in ~6541 samples (~900 TB)

And check out our broader Open Data program ❗

[Check the website]



What Have We Documented?

Open data are very hard to use without good documentation:

The Data › Data for research › Event Generation Data

Event Generation Data

EVOLVING!

A large volume of these samples is being released now. Feedback is welcome!

The ATLAS Collaboration has released a significant amount of event generation data in HEPMC format (git repository). Events are provided in 10,000 event text files, tarred and gzipped to save space, for both 13 TeV (Run 2-like) and 13.6 TeV (Run 3-like) configurations. These files can be passed into common phenomenological tools like Delphes and Rivet. Please don't forget to cite these samples if you use them!

Search for Data

The data are gathered into records on the CERN Open Data Portal. A single record serves as an entry point, with pointers to all the other records. They are grouped into logically consistent sets according to keyword rules, in order to make the identification of sample sets (e.g. phase-space slices of a single physics sample, or beyond the Standard Model (BSM) signal samples intended for use with a single analysis) straightforward. The most convenient way to find and access these samples is by using atlasopenmagic. You can follow the metadata tutorial to learn how to use atlasopenmagic.

Sample Availability

In order to minimize the storage required for these samples, we have released up to 10M events per individual sample, up to double the luminosity of the collected dataset at the corresponding center of mass energy, or all the events that we have generated, which ever is least. In some cases, more events are available. If you require more events for a specific

The Data › Data for research › Event Generation Data › Combining Samples

Combining Samples

Many of the common Standard Model processes are not generated as a single inclusive process, but as many individual samples that have to be combined in order to be used in a physics analysis. In many cases, it's not obvious which samples are "safe" (or necessary) to combine. This is tricky for a variety of reasons, and this page is intended to help guide users through some of the common issues.

The metadata page is an extremely useful reference to identify samples that are meant for specific use-cases (e.g. as the baseline, starting point estimate for a physics process, or for calculating a systematic uncertainty). With a bit of practice, reading the "Physics short" (short description of a sample) becomes easier and can reveal most of the key features of a sample.

The QCD jet samples provide a nice example of a combination. They are named "JZ0", "JZ1", "JZ2", and so on, with each sample corresponding to a specific range of leading jet transverse momenta. In order to create a complete, smooth jet spectrum, one must combine all of these samples according to their cross sections, filter efficiencies, and k-factors. If an analysis is being done that requires a 500 GeV jet, then there is no need to consider the first few samples. "Slicing" the transverse momentum spectrum provides an analysis with roughly constant statistical uncertainties with increasing jet transverse momentum without the need to generate (and analyze) billions of events.

The divisions between samples might not be consistent between event generators, which can lead to further confusion. Some event generators, like Sherpa, can generate final states directly (e.g. "four leptons") without regard to the internal configuration of the physics process. Another generator might only be able to generate "two Z boson" events that then decay into four leptons. With Z bosons and four leptons, there is likely little ambiguity. Two charged leptons and two neutrinos, however, can be created by either two Z bosons or two W bosons. Some generators will provide these in two separate samples (ZZ and WW), and others will provide these as a single sample (llvv). When comparing these two configurations, it is important to compare apples to apples, and therefore to carefully consider these sorts of issues.

Explaining the origins and combinations of all the samples available is a significant undertaking. On this page, we aim to document the most common setups and questions. As users request help with other samples, we will continue to add to these pages to document features and clarify points of confusion. In order to make the discovery process as painless as possible, related samples in the Open Data portal have been gathered into collections.

What Have We Documented?

Open data are very hard to use without good documentation:

The Data › Data for research › Event Generation Data › Limitations

Limitations

Although these data are all intended for use in scientific research, they are not without limitations. It is important to keep in mind the limitations of these datasets when using them in order to avoid false-positive results or dead ends. Although some of the most important limitations are documented here, if you have any questions about whether a particular analysis is possible please feel free to contact us. In case the analysis is not possible with the available public data, you are also welcome to consider a short term association with ATLAS in order to perform the analysis like a member of the collaboration would.

Data format limitations

The HEPMC2 event-generation data format released for research use contains all particles in the event record, slightly modified from what the generator would normally provide. In particular:

- Loops in the event record have been broken apart.
- The events have been checked for unexpected features (like stable gluons), and any such events have been removed. These can result from rare issues in specific event generators.

The data format is without any detector simulation applied. It is necessary, therefore, to apply some smearing and inefficiencies to the particle records before comparing them to reconstructed data. Doing so is rather delicate, but standard programs like DELPHES and PGS, as well as Rivet functionality, should suffice for many cases.

Data sample limitations

The samples that have been released are sufficient to establish standard backgrounds and generically-applicable systematic uncertainties. In some cases, establishing key systematic uncertainties requires additional samples that have

The Data › Data for research › Event Generation Data › Metadata

Metadata

Below, you will find the metadata for all the samples, which includes the following:

- **Dataset ID**: Unique identifier assigned to each dataset.
- **Physics short**: Short name with information regarding the content of the dataset.
- **CoM Energy (GeV)**: Center of mass energy.
- **Cross section (pb)**: A cross section represents the probability of a particular interaction occurring, measured in picobarns (pb). It is a fundamental parameter that helps understanding the likelihood of specific particle interactions under given conditions. The cross section is usually what is returned by the generator - a user needs to multiply by the filter efficiency and k-factor (see below) to get a complete sample weight.
- **Filter efficiency**: Measure of the effectiveness of the selection criteria applied to the data. It indicates the fraction of events that pass the filters applied during the data processing stages.
- **K-factor**: Multiplicative correction factor used to account for higher-order effects in theoretical calculations. It adjusts the leading-order theoretical predictions to better match the observed data by incorporating next-to-leading order (NLO) or next-to-next-to-leading order (NNLO) corrections.
- **Number of Events**: Total count of events in the released HEPMC dataset.
- **Generated events**: Total count of generated events available internally. If you require more events and we have them available, please get in touch.
- **Generator used**: Specifies the simulation software used to generate the data. Information about the generators can be found in the Simulation Tools section.
- **Generator tune**: The parameter settings used in the event generation, in particular for fragmentation and hadronization.
- **PDF**: The parton distribution function used for the event generation.
- **Keywords**: Terms or phrases associated with the dataset that help to find specific datasets.
- **Process Description**: Brief description of the physics process being studied. For instance, "H->γγ" denotes the Higgs boson decaying into two photons.
- **Release**: Version of the ATLAS software (usually the AthGeneration release version) used to generate the events.
- **Filters**: A list of any filters applied during event generation (e.g. requiring at least two leptons.)
- **Job options**: Link to the specific code or configuration files used to generate the sample.

What're the Metadata Like?

Metadata is important to be able to use the data. We have a notebook tutorial showing how to access it. Runs on google Colab, myBinder, and SWAN.

```
[ ]: # First we install atlasopenmagic into our environment
     %pip install atlasopenmagic
```

```
[2]: # Now we can safely import atlasopenmagic
     import atlasopenmagic as atom
```

```
[3]: # Now let's see what releases are available to us
     atom.available_releases()
```

```
Available releases:
=======================================
2016e-8tev          2016 Open Data for education release of 8 TeV proton-proton collisions (https://opendata.cern.ch/record/3860).
2020e-13tev         2020 Open Data for education release of 13 TeV proton-proton collisions (https://cern.ch/2r7xt).
2024r-pp            2024 Open Data for research release for proton-proton collisions (https://opendata.cern.record/80020).
2024r-hi            2024 Open Data for research release for heavy-ion collisions (https://opendata.cern.ch/record/80035).
2025e-13tev-beta    2025 Open Data for education and outreach beta release for 13 TeV proton-proton collisions (https://opendata.cern.ch/record/93910).
2025r-evgen-13tev   2025 Open Data for research release for event generation at 13 TeV (https://opendata.cern.ch/record/160000).
2025r-evgen-13p6tev 2025 Open Data for research release for event generation at 13.6 TeV (https://opendata.cern.ch/record/160000).
```

ATLAS Open Magic

Tests passing | pypi 1.6.0 | codecov 100%

`atlasopenmagic` is a Python package made to simplify working with ATLAS Open Data by providing utilities to manage metadata and URLs for streaming the data.

```
[12]: # And let's use the latest release of Event Generation Open Data
      atom.set_release('2025r-evgen-13p6tev')
```

```
Fetching and caching all metadata for release: 2025r-evgen-13p6tev...
Fetched 1509 datasets so far...
Successfully cached 1509 datasets.
Active release: 2025r-evgen-13p6tev. (Datasets path: REMOTE)
```

Loading the 13 TeV data takes ~20-30s (it's a lot of datasets!)

(It might wipe out the available memory in binder)

What're the Metadata Like?

Metadata is important to be able to use the data. We have a [notebook tutorial] showing how to access it. Runs on google Colab, myBinder, and SWAN.

```
[14]: # Now we can look at the metadata for a specific sample
      atom.get_metadata(510203)
      # Notice that the function here will accept either the dataset identifier or the
      # "physics short", a short unique descriptor for the sample %% [markdown] That's
      # a lot of metadata! Let's go through the fields a bit:
```

```
[14]: {'dataset_number': '510203',
       'physics_short': 'MGPy8EG_A14NNPDF30_SM4topsLOInclusive_run3',
       'e_tag': None,
       'cross_section_pb': 0.0092591,
       'genFiltEff': 1.0,
       'kFactor': 1.0,
       'nEvents': 10000,
       'sumOfWeights': None,
       'sumOfWeightsSquared': None,
       'process': None,
       'generator': 'MadGraph(v.3.5.3.atlas4)+Pythia8(v.310)+EvtGen(v.2.2.1)',
       'keywords': ['4top', 'Systematic', 'lo', 'sm', 'top'],
       'description': 'Standard-Model 4tops production at LO with MadGraph5 and Pythia8',
       'job_path': 'https://gitlab.cern.ch/atlas-physics/pmg/mcjoboptions/-/blob/master/510xxx/510203/mc.MGPy8EG_A14NNPDF30_SM4topsLOInclusive_run3.py',
       'CoMEnergy': 13600.0,
       'GenEvents': 41240000,
       'GenTune': 'A14 NNPDF23LO',
       'PDF': 'NULL',
       'Release': 'AthGeneration_23.6.24',
       'Filters': '',
       'cross_section_uncertainty': 0.0,
       'hepmc_version': 2,
       'release': {'name': '2025r-evgen-13p6tev'}}
```

That's a lot of metadata! Let's go through the fields a bit:

- `dataset_number` : Unique identifier assigned to each dataset.
- `physics_short` : Short name with information regarding the content of the dataset.
- `cross_section_pb` : Represents the probability of a particular interaction occurring, measured in picobarns (pb). It is a fundamental parameter that helps understanding the likelihood of specific particle interactions under given conditions.
- `genFiltEff` : Measure of the effectiveness of the selection criteria applied to the data. It indicates the fraction of events that pass the filters applied during the data processing stages.
- `kFactor` : Multiplicative correction factor used to account for higher-order effects in theoretical calculations. It adjusts the leading-order theoretical predictions to better match the observed data by incorporating next-to-leading order (NLO) or next-to-next-to-leading order (NNLO) corrections.

ATLAS Open Magic

Tests passing | pypi 1.6.0 | codecov 100%

`atlasopenmagic` is a Python package made to simplify working with ATLAS Open Data by providing utilities to manage metadata and URLs for streaming the data.

What're the Metadata Like?

Metadata is important to be able to use the data. We have a notebook tutorial showing how to access it. Runs on google Colab, myBinder, and SWAN.

```
[15]: # Keywords are a great way to find datasets that you're interested in. Let's see
      # what keywords are available.
      atom.available_keywords()

[15]: ['1lepton',
       '2electron',
       '2jet',
       '2lepton',
       '2muon',
       '2photon',
       '2tau',
```

```
[16]: # Now let's find datasets that match one of those keywords
      atom.match_metadata(field='keywords',value='higgs')

[16]: [('525154', 'MGPy8EG_NNPDF30_mh2400_wh25_hbsm4tops_stchan'),
       ('525155', 'MGPy8EG_NNPDF30_mh21000_wh230_hbsm4tops_stchan'),
       ('525156', 'MGPy8EG_NNPDF30_mh22000_wh2100_hbsm4tops_stchan'),
       ('602635', 'PhH7EG_PDF4LHC21_ttH125_dilep'),
       ('602636', 'PhH7EG_PDF4LHC21_ttH125_semilep'),
       ('602637', 'PhPy8EG_PDF4LHC21_ttH125_dilep'),
       ('602638', 'PhPy8EG_PDF4LHC21_ttH125_semilep'),
       ('602852', 'PhPy8EG_PDF4LHC21_ttH125_pThard1_dilep'),
       ('602853', 'PhPy8EG_PDF4LHC21_ttH125_pThard1_semilep'),
       ('603293', 'PhPy8EG_NNPDF30_gg4l_noHiggs_m4l_70_3000'),
       ('604224', 'PhPy8EG_PDF4LHC21_allHad_ttH125_HTop'),
       ('604304', 'PhPy8EG_PDF4LHC21_qqZH125J_Zincl_Hincl_MINLO'),
       ('604305', 'PhPy8EG_PDF4LHC21_WmH125J_Zincl_Hincl_MINLO'),
       ('604306', 'PhPy8EG_PDF4LHC21_WpH125J_Zincl_Hincl_MINLO')]
```

```
[19]: # Now lets look for samples that have `W` somewhere in the description
      # Notice that this also catches things like `Wprime`!
      atom.match_metadata(field='description',value='W')

[19]: [('513093', 'MGPy8EG_Wenu_FxFx3jHT2bias_SW_BFilter'),
       ('513094', 'MGPy8EG_Wenu_FxFx3jHT2bias_SW_CFilterBVeto'),
       ('513095', 'MGPy8EG_Wenu_FxFx3jHT2bias_SW_CVetoBVeto'),
       ('513096', 'MGPy8EG_Wmunu_FxFx3jHT2bias_SW_BFilter'),
       ('513097', 'MGPy8EG_Wmunu_FxFx3jHT2bias_SW_CFilterBVeto'),
       ('513098', 'MGPy8EG_Wmunu_FxFx3jHT2bias_SW_CVetoBVeto'),
       ('513099', 'MGPy8EG_Wtaunu_L_FxFx3jHT2bias_SW_BFilter'),
       ('513100', 'MGPy8EG_Wtaunu_L_FxFx3jHT2bias_SW_CFilterBVeto'),
       ('513101', 'MGPy8EG_Wtaunu_L_FxFx3jHT2bias_SW_CVetoBVeto'),
       ('513102', 'MGPy8EG_Wtaunu_H_FxFx3jHT2bias_SW_BFilter'),
       ('513103', 'MGPy8EG_Wtaunu_H_FxFx3jHT2bias_SW_CFilterBVeto'),
       ('513104', 'MGPy8EG_Wtaunu_H_FxFx3jHT2bias_SW_CVetoBVeto'),
       ('513105', 'MGPy8EG_Zee_FxFx3jHT2bias_SW_BFilter'),
       ('513106', 'MGPy8EG_Zee_FxFx3jHT2bias_SW_CFilterBVeto'),
```

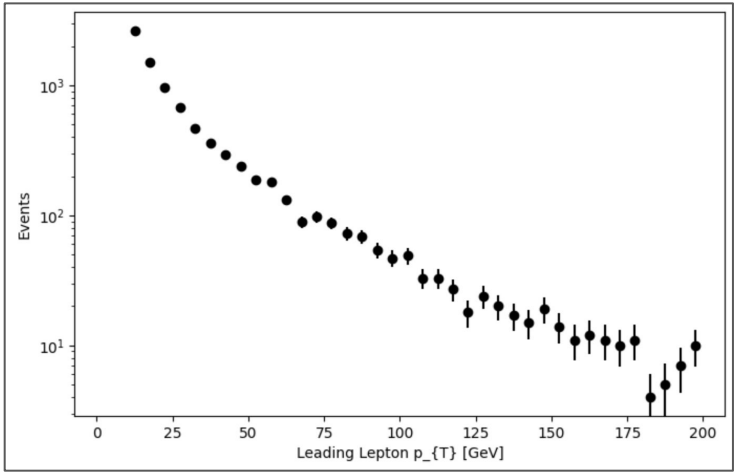
How about an Example?

We provide a fully worked-out example notebook.

- Runs on Binder; some features are not as nice on Colab.
- Set up and sample identification.
- Getting files locally (sometimes a 504 from the gateway).
- Visualizing events with pyhepmc.
- Making basic plots.
- Running Delphes (in the notebook!).
- Examining Delphes output.
- Takes <5 minutes to run.

Combines a few different goals

- Demonstrating access / use of the open data.
- Teaching some basics about HEPMC records.
- Demonstrating parametric simulation.




```
[16]: import uproot # for reading .root files
      # Get the tree with our data directly from the ROOT file
      tree = uproot.open("delphes_output.root:Delphes")
      # Just for an example here, we'll print the transverse momenta of the electrons
      tree["Electron.PT"].arrays()

[16]: [{'Electron.PT': []},
       {'Electron.PT': []},
       {'Electron.PT': []},
       {'Electron.PT': []},
       {'Electron.PT': []},
       {'Electron.PT': [75.4]},
       {'Electron.PT': []},
       {'Electron.PT': [39.1]},
       {'Electron.PT': []},
       {'Electron.PT': []},
       ...,
       {'Electron.PT': []},
       {'Electron.PT': [51.1]},
       {'Electron.PT': []},
       {'Electron.PT': []},
       {'Electron.PT': []},
       {'Electron.PT': [78.5]},
```

Some technicalities

- The files are **stored on a Rucio endpoint** at CERN (on eospublic)

```
atom.get_urls('301057',protocol='eos')
      ['/eos/opendata/atlas/rucio/mc16_13TeV/HEPMC.43646538._000001.tar.gz.1',
       '/eos/opendata/atlas/rucio/mc16_13TeV/HEPMC.43646538._000002.tar.gz.1',
       '/eos/opendata/atlas/rucio/mc16_13TeV/HEPMC.43646538._000003.tar.gz.1',
       '/eos/opendata/atlas/rucio/mc16_13TeV/HEPMC.43646538._000004.tar.gz.1',
       '/eos/opendata/atlas/rucio/mc16_13TeV/HEPMC.43646538._000005.tar.gz.1',
       '/eos/opendata/atlas/rucio/mc16_13TeV/HEPMC.43646538._000006.tar.gz.1',...
```

- **Can be accessed** via
  - xrootd (download/stream)
  - https (download/stream)
  - /eos (mounted volume for those at CERN)
  - rucio directly (with an ATLAS Rucio account)

- EVNT → HEPMC via a **thin transformation layer**.
  - Currently not automated. Long-term would be nice to get away from this, if the user base becomes significant and this takes off.
  - Intentionally minimal internal manipulation: **You really get what we use, for better or worse!**

Experience so far

Mostly positive responses:

"FYI our undergrad looked at the [Drell-Yan] MC and said it was super easy to run on (for our workflow), and included all the info she expected in a HepMC"

MCNet student working with CMSSW: mostly "all good", but reported one issue with beam energy (discussed in following slide.)

"Checkmate - runs with no problems, but requires as input the MC output cross section and uncertainty (checkmate doesn't read this from the hepmc files). Would it be possible to add this info to the metadata? Rivet (with and without smearing) - runs with no problems. Delphes - DelphesHepMC2 runs with no problems, but DelphesHepMC3 crashes with the error message 'invalid vertex format'..."

Both (uncertainty, where available, and HEPMC version) now added to the metadata!

Not clear yet what the broader uptake is or will be (but this is relatively new)

Known issues

We are far from perfect. Some issues in the samples have already been reported

- Sherpa samples seem to report **incorrect energy units** in file → Will be fixed
- Some samples report **all event numbers as '1'** → Probably not going to be fixed unless there is strong demand
- Rivet complains about **beam energies for some events.** → (mcnet/short-term-studentships#4 ; not yet understood)

What More Could We Do?

Release more samples

- Primary limitation right now is **disk space** provided by CERN IT (this much was a fight).

Develop more notebooks

- If we have a big (and loudly supportive) user base, I imagine this will be ‘easier’.

ATLAS and CMS sharing event generation

- No objection to releasing LHE format files as well (we’d need a request.)

Other formats

What More Could We Do?

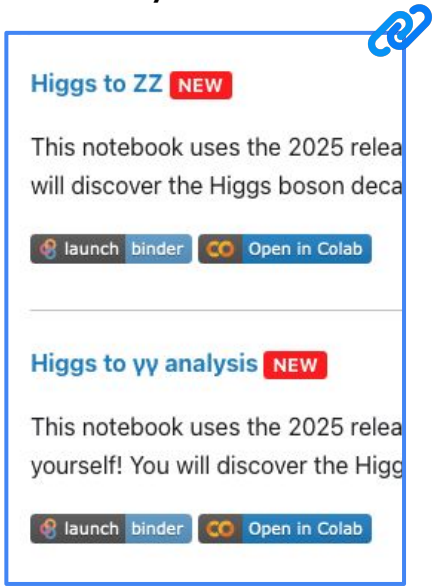
Release more samples

Develop more notebooks

ATLAS and CMS sharing event generation

Other formats

- We have these sorts of things for ATLAS Open Data already



- HighTEA looks similar to some of our open data setups, at least in principle

What More Could We Do?

Release more samples

Develop more notebooks

ATLAS and CMS sharing event generation

Other formats

- Raised in the Dark Showers workshop and years ago in the HSF Event Generation WG
- Sharing does **not** mean we disallow separate samples, or even use the same nominal

What More Could We Do?

Release more samples

Develop more notebooks

ATLAS and CMS sharing event generation

Other formats

- Delphes, PGS, Rivet, others? Need a (complete?) list if we embark on this path
- Could consider a document recommending a simulation (configuration) and pointing out some of the known limitations (working document of improvements?)

Shameless Plug: ATLAS Open Data Tutorial

We will run an Open Data Tutorial this November

- Aimed at people (esp. non-ATLAS) interested in learning about and using our open data
  - Two days on Outreach and Education Open Data
  - Two days on Research Open Data
- This tutorial is **in person**
- We expect to follow up with a **virtual** tutorial some months later (once we've smoothed out the kinks)
- Planning a short introduction to the event generation open data on **Thursday** of the tutorial

Summary

- Please feel free to look at the ATLAS Open Event Generation data!
- We aren't perfect, so please report any issues you find with the samples, documentation, tools, or metadata.
- Please let us know if you have requests.
- In both of the latter cases, your *concrete help with improvements* is very welcome
  - If that happens at a significant rate, then (I believe) this effort will be successful long-term

This is a also **test of the community**:

- Can we work together to support common samples?
- When issues are found, are they reported back? Do people help correct issues?
- If samples are insufficient for some reason, is that reported back? Do we make the samples better together?

Extra

Format Technicalities

- Internally, we have a substantial number of events

Table 3: Number of datasets (with unique configurations) and events (in billions) generated with various generators thus far during the MC simulation campaign of Run 2.

Event Generator	Datasets	Generated Events ($\times 10^9$)	Simulated Events ($\times 10^9$)
SHERPA	3887	89.7	27.6
POWHEG	6747	55.7	15.9
MADGRAPH	251023	52.2	12.5
PYTHIA	6240	13.8	7.5
PYTHIA 8B	422	5.1	2.0
HERWIG	813	4.3	2.4
Others	9851	3.5	0.5
Total	280935	224.4	68.4

Totals in Run 2-like configuration. Some samples are obsolete now (newer configurations exist)

Format Technicalities

- Theorists/Phenomenologists seem to want compressed HEPMC (for now) — right?
  - Primarily because that interfaces well with existing tools
  - Could move to an alternative format if the tools support that format; we could push the community towards something ROOT-based(?) if desired
  - ATLAS and CMS both use ROOT-based representations of HEPMC (v3 for ATLAS)
- Quick size comparison test with an Run 3 (13.6 TeV C.o.M) ttbar file (10k events):
  - EVNT (ATLAS internal format): 58.2 kB/event (~2x variations depending on ROOT settings)
  - Compressed HEPMC: 54.5 kB/event (variations depending on compression settings)
  - Uncompressed HEPMC: 210 kB/event
  - TRUTH0 derivation (easy ROOT-readable EVNT): 35.9 kB/event
  - TRUTH1 derivation (TRUTH0+pre-built simplified collections like ‘jets’): 40.8 kB/event
- Converting our EVNT to HEPMC
  - O(4k) CPU-core-days (not much by modern standards; 0.6% of ATLAS for a day)

Why Open Event Generation?

- Theorists and phenomenologists generate their own MC simulation when needed
- The LHC experiments put enormous effort and CPU into running event generation
- Why should we experimentalists help the theorists out?
  - It would support the community — this would be a great service
  - It would avoid **lots** of duplicate and wasted CPU — good for the environment
  - It would encourage phenomenologists and theorists to look at our MC simulation, which could have a number of ancillary benefits: experts coming to help with configurations, identifying issues in our configurations, improved documentation, maybe validation help?
  - (Of course, this would also be a means of sharing our event generation samples between e.g. ATLAS and CMS, but that is only an interesting-to-explore side-effect here)
- Reminder: our software for event generation is public, builds are available publicly as containers or on cvmfs, and configuration is public