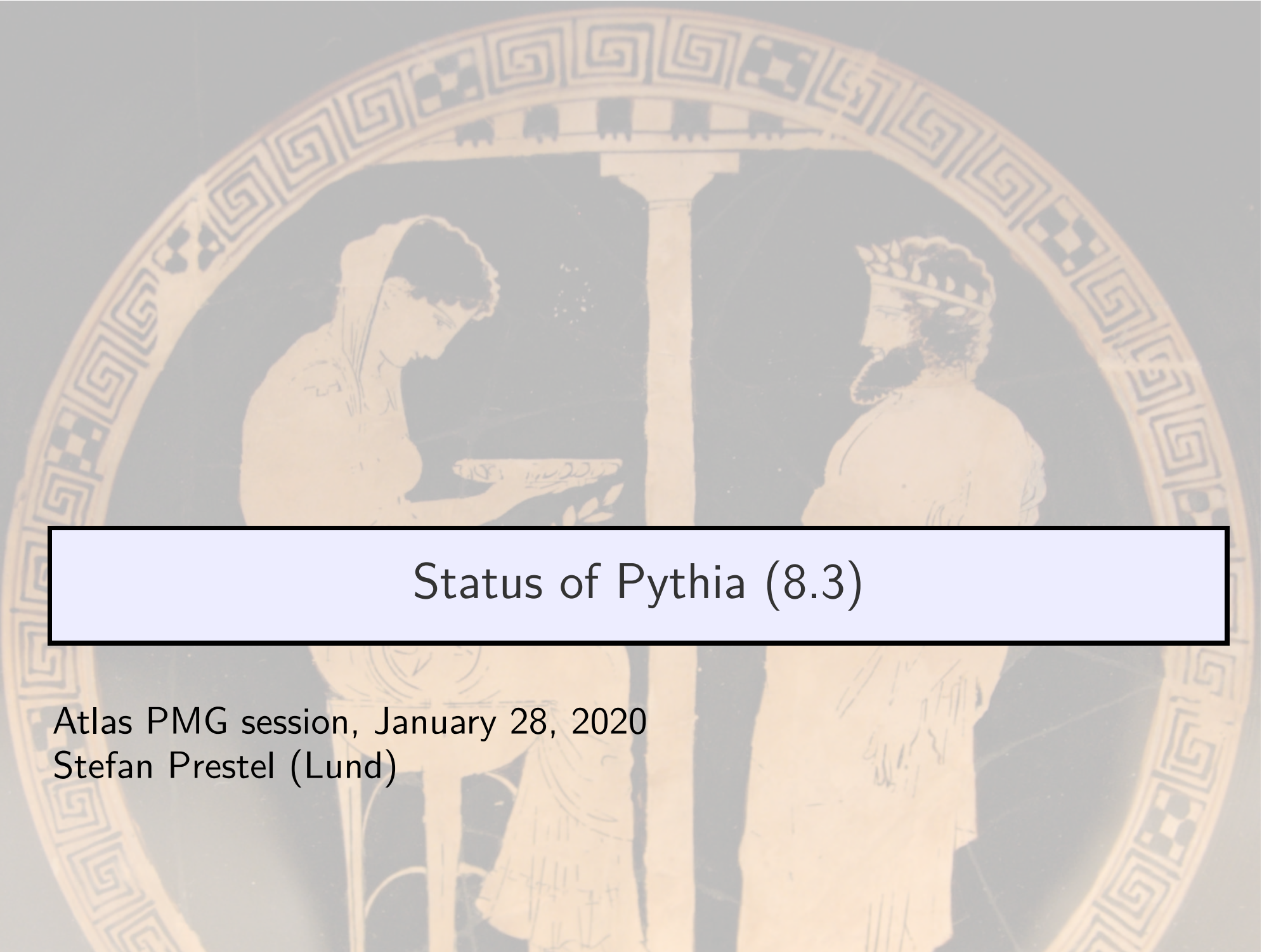


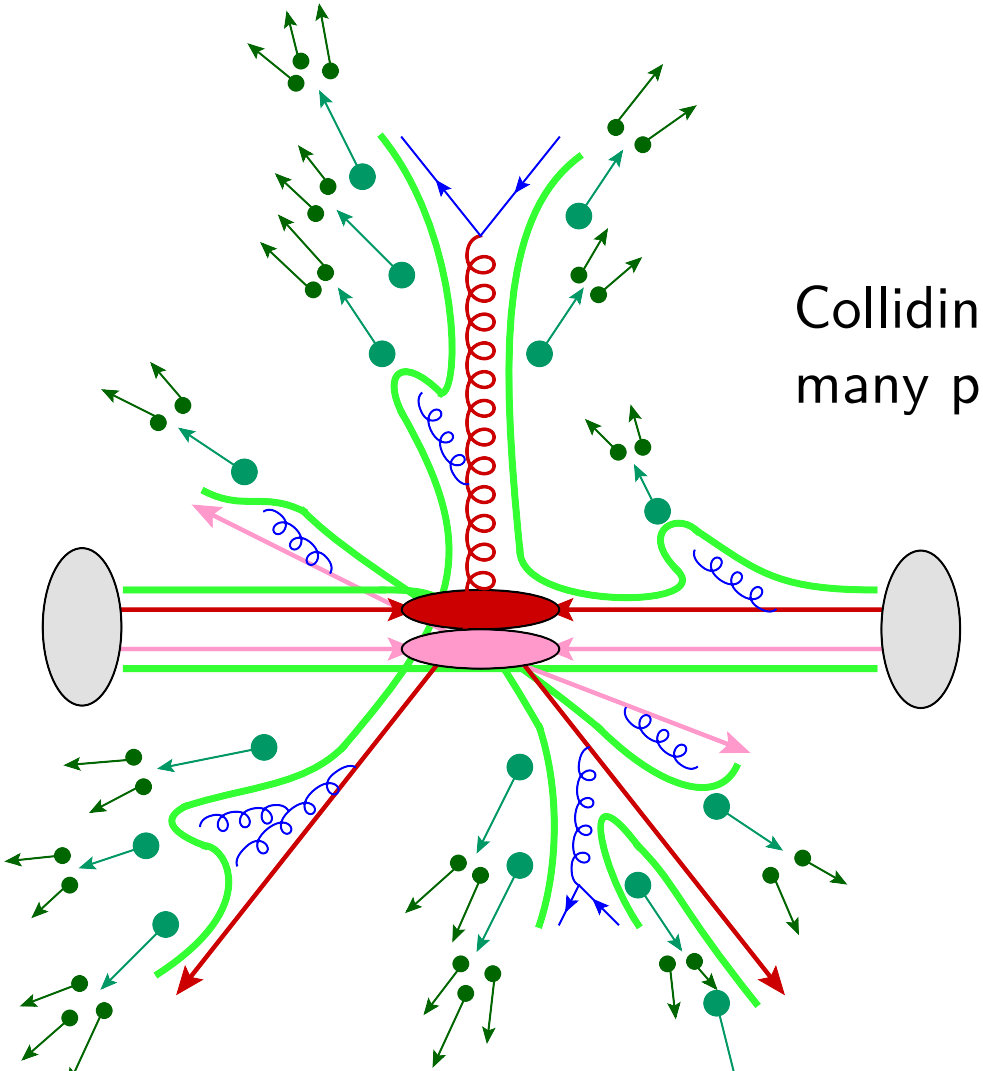
Status of Pythia (8.3)

Atlas PMG session, January 28, 2020

Stefan Prestel (Lund)

Detailed Monte Carlo predictions

Detailed pseudodata from theory tools $\rightarrow$ better analyses of backgrounds, better analyses of signals



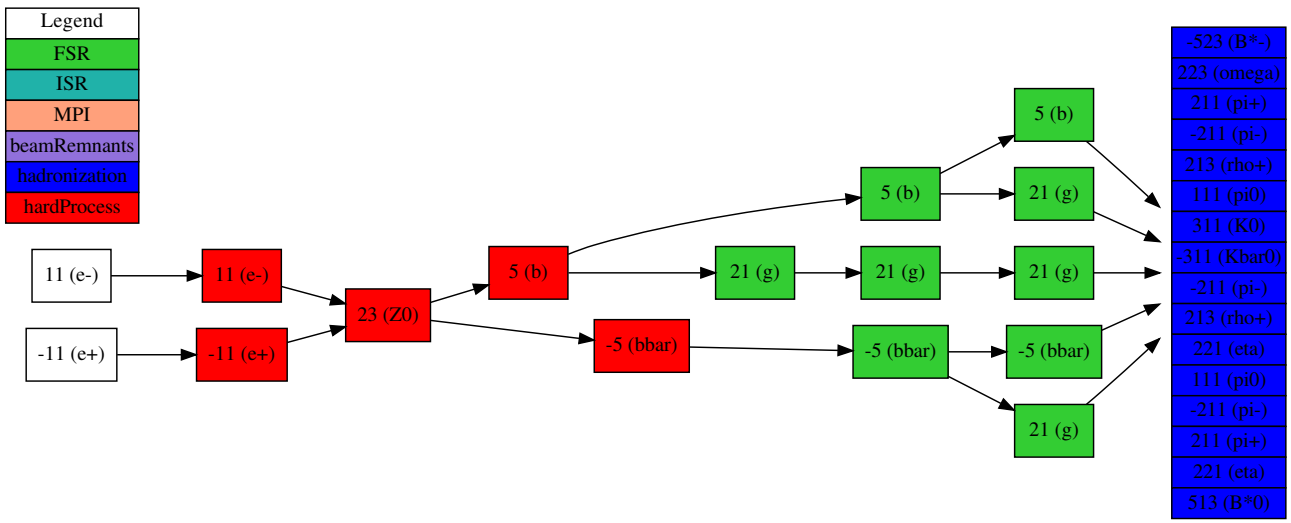
Colliding composite objects kick-starts many processes:

- hard scattering
- radiation cascade
- multiparton interactions
- hadronization and decay

For new users: Some fun visualization

New users often wonder what goes on in PYTHIA. We included a bit of silly visualization for that, see `include/Pythia8Plugins/Visualization.h`

LEP looks pretty simple...

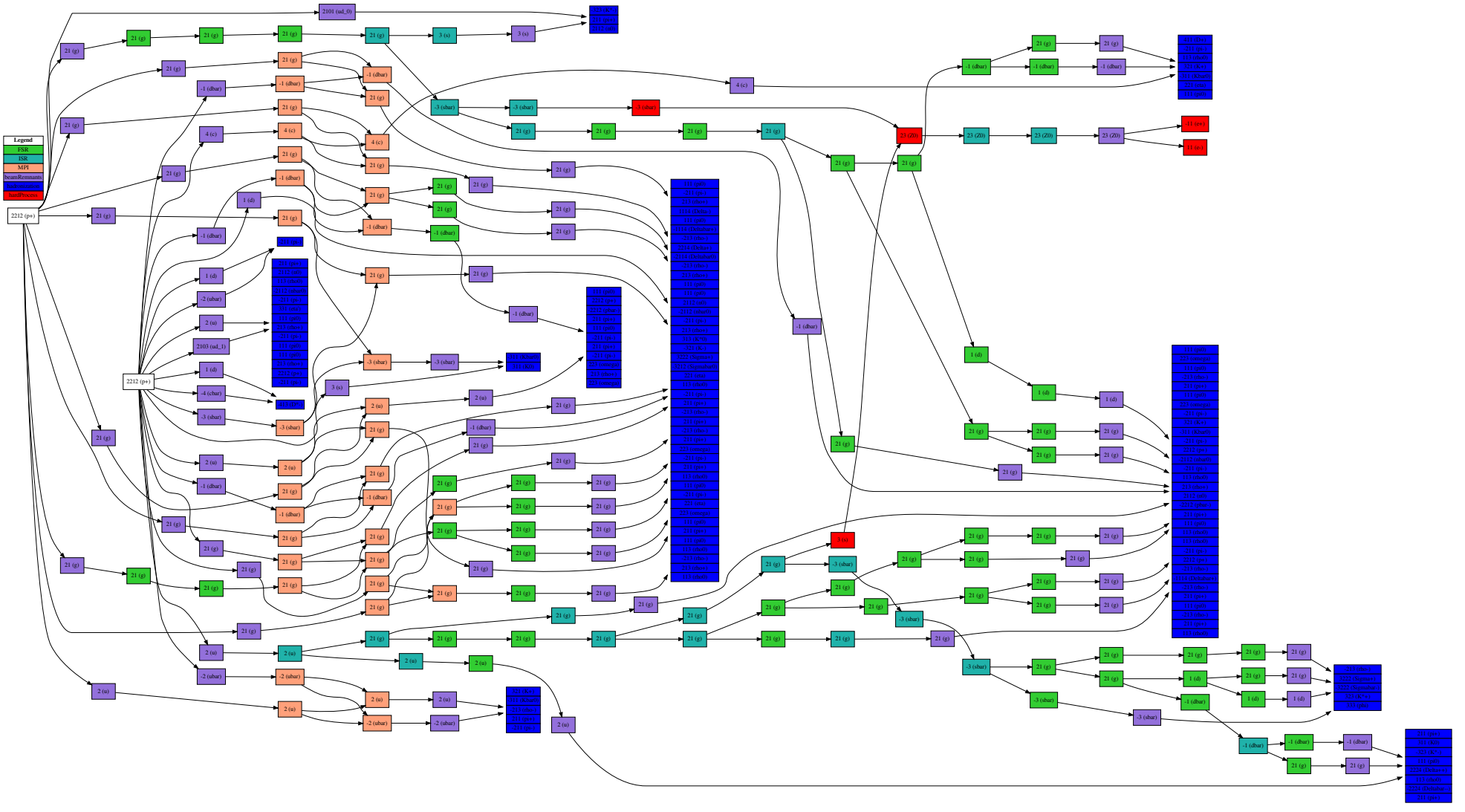


Generated with...

```
./main300 --input main300.cmnd --visualize_event
```

For new users: Some fun visualization

...and LHC is a bit more complicated.



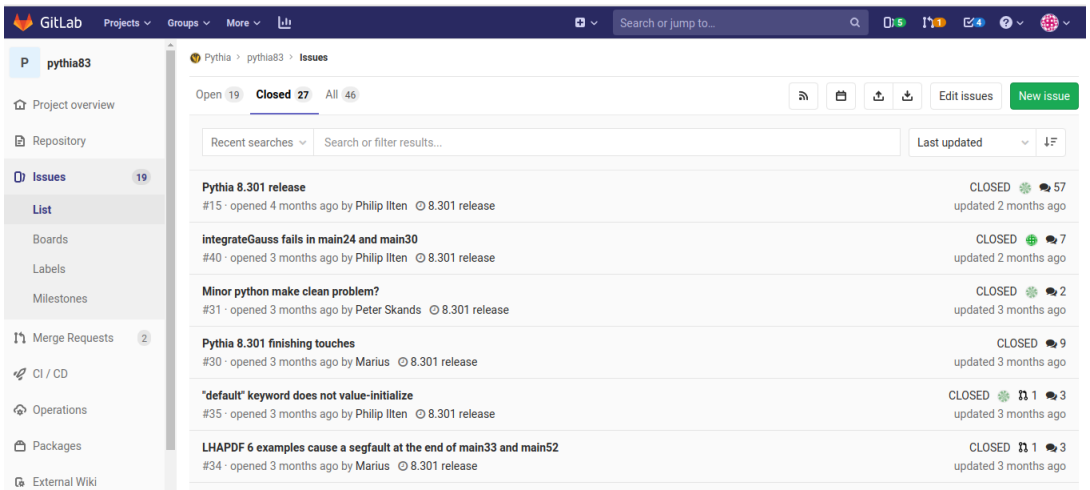
PYTHIA 8.301	is the latest release
BEAMS	ee, ep, pp, γx, pA, AA, DM
HARD SCATTERING	Core lib. of internal processes, otherwise from external tools. NLO+PS matching/merging with both aMC@NLO and POWHEG-BOX processes.
PARTON SHOWER	Three models: Default, Vincia and Dire.
MULTIPARTON INTERACTIONS	Regularised secondary $2 \to 2$ SM scatterings, interleaved with shower evolution.
SOFT PHYSICS	Regge-based diffraction and x-sections
FRAGMENTATION	String hadronization with Schwinger-based or thermal transition probabilities.

News: Code revamp under the hood (but should feel the same to users), **Vincia & Dire now core components**. **New manual** for 8.3:
http://home.thep.lu.se/~torbjorn/pythia83html/Welcome.html

New technical features I

Pythia 8.3 mainly hosted at https://gitlab.com/Pythia8/releases and is relying on C++11

We internally use the gitlab issue tracking
...and built up a unit test suite & continuous integration w/ help of docker.



Would e.g. a bug tracker also be useful for you? Should we expose more of our inner workings to the world?

For our release testing, we start relying on CI and docker, see e.g. https://hub.docker.com/r/pythia8/dev/tags or the tutorial http://home.thep.lu.se/~prestel/Tutorials.html and http://home.thep.lu.se/~leifg/tutorials/ (still for 8.2, though)

We currently use a
◇ lightweight container for only Pythia
◇ heavyweight container to test all our dependencies
◇ a container to generate `Python` interfaces.

Should we make "blessed containers" available to the world?

New technical features III: Lightweight python interface

Pythia 8.3 comes with a new light-weight python interface via PyBind11, see
http://home.thep.lu.se/~torbjorn/pythia83html/PythonInterface.html

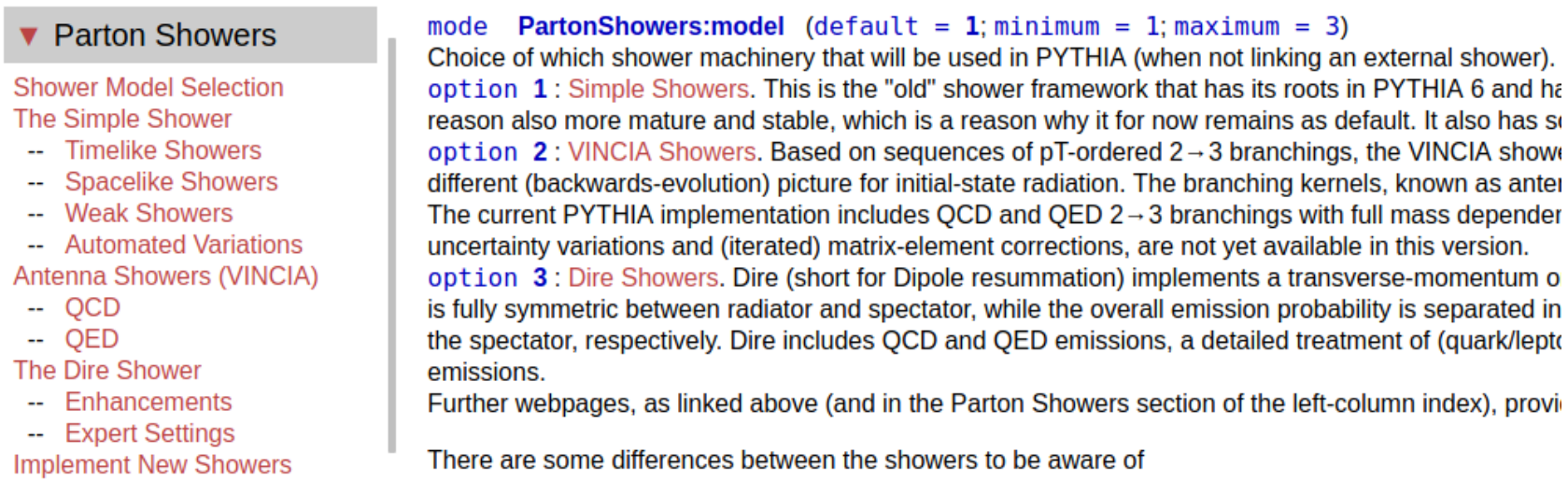
```python
# Import the Pythia module.
import pythia8
pythia = pythia8.Pythia()
pythia.readString("HardQCD:all = on")
pythia.readString("PhaseSpace:pTHatMin = 20.")
pythia.init()
mult = pythia8.Hist("charged multiplicity", 100, -0.5, 799.5)

# Begin event loop. Generate event. Skip if error. List first one.
for iEvent in range(0, 100):
    if not pythia.next(): continue
    # Find number of all final charged particles and fill histogram.
    nCharged = 0
    for prt in pythia.event:
        if prt.isFinal() and prt.isCharged(): nCharged += 1
    mult.fill(nCharged)
# End of event loop. Statistics. Histogram. Done.
pythia.stat();
print(mult)
```

...wich allows inheritance. See `main01.py`, `main10.py` for `UserHooks` written in python
...you can also regenerate the interface, if you e.g. change/introduce C++ headers (UserHooks...)

Main new physics feature: Native Vincia/Dire

VINCIA and **DIRE** are now part of PYTHIA core code. You can just switch them on:



▼ Parton Showers

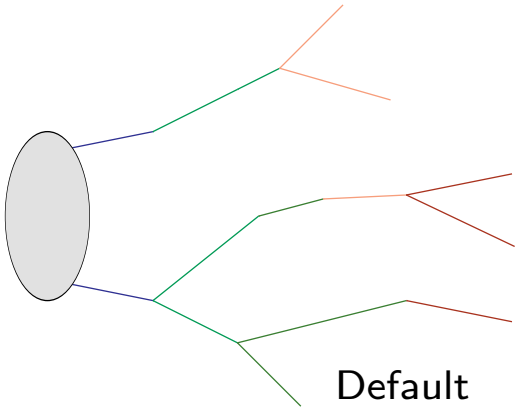
Shower Model Selection
The Simple Shower
-- Timelike Showers
-- Spacelike Showers
-- Weak Showers
-- Automated Variations
Antenna Showers (VINCIA)
-- QCD
-- QED
The Dire Shower
-- Enhancements
-- Expert Settings
Implement New Showers

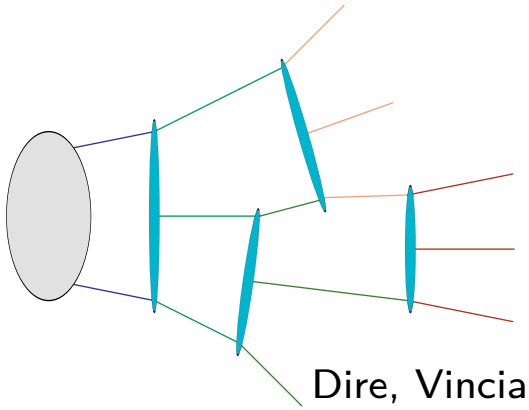
mode **PartonShowers:model** (default = **1**; minimum = 1; maximum = 3)
Choice of which shower machinery that will be used in PYTHIA (when not linking an external shower).
option **1** : Simple Showers. This is the "old" shower framework that has its roots in PYTHIA 6 and ha
reason also more mature and stable, which is a reason why it for now remains as default. It also has s
option **2** : VINCIA Showers. Based on sequences of pT-ordered 2→3 branchings, the VINCIA showe
different (backwards-evolution) picture for initial-state radiation. The branching kernels, known as anter
The current PYTHIA implementation includes QCD and QED 2→3 branchings with full mass depender
uncertainty variations and (iterated) matrix-element corrections, are not yet available in this version.
option **3** : Dire Showers. Dire (short for Dipole resummation) implements a transverse-momentum o
is fully symmetric between radiator and spectator, while the overall emission probability is separated in
the spectator, respectively. Dire includes QCD and QED emissions, a detailed treatment of (quark/lepto
emissions.
Further webpages, as linked above (and in the Parton Showers section of the left-column index), provi

There are some differences between the showers to be aware of

Development of the plugins will be phased out over the next year or so.

Main new physics feature: Native Vincia/Dire


Default


Dire, Vincia

Default

◇ Improved DGLAP evolution in $p_\perp$ ◇ ME corrections for 1st splitting.
◇ QCD, QED, EW, hidden valley ◇ Extensive tuning expertise.

Vincia

◇ Coherent evolution in 1/eikonal, antenna pattern
◇ Implements iterated LO matrix element corrections.
◇ QCD, QED, coherence in res. decays

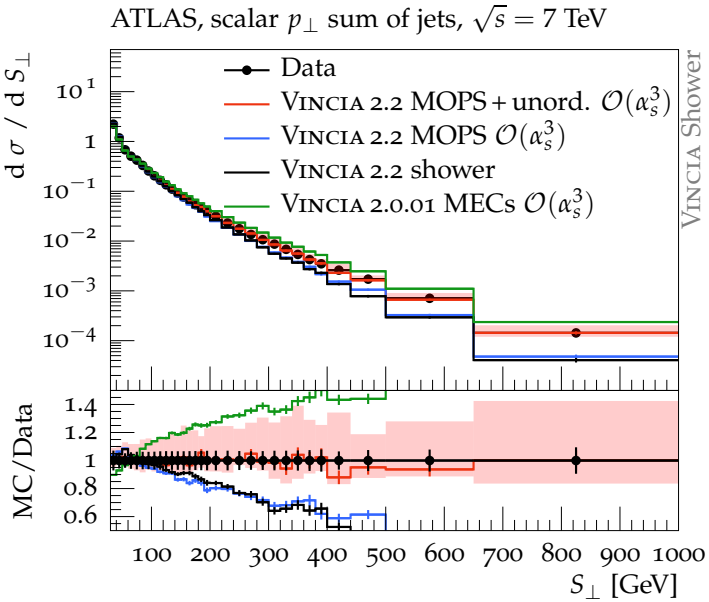
Dire

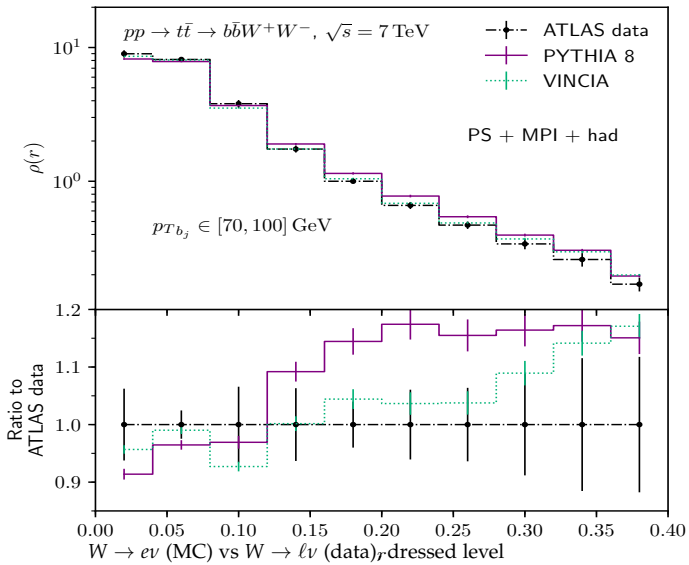
◇ Coherent evolution in 1/eikonal, split into collinear regions
◇ Implements NLO corrections to evolution, matrix element corrections
◇ QCD, QED, iffy EW, dark photons

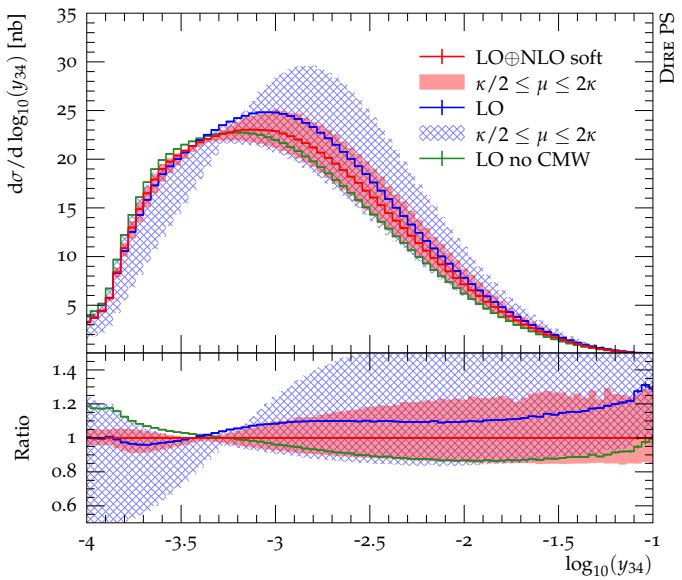
For usage, see `main200-202.cc` and `main300.cc` (which adds `OpenMP`)

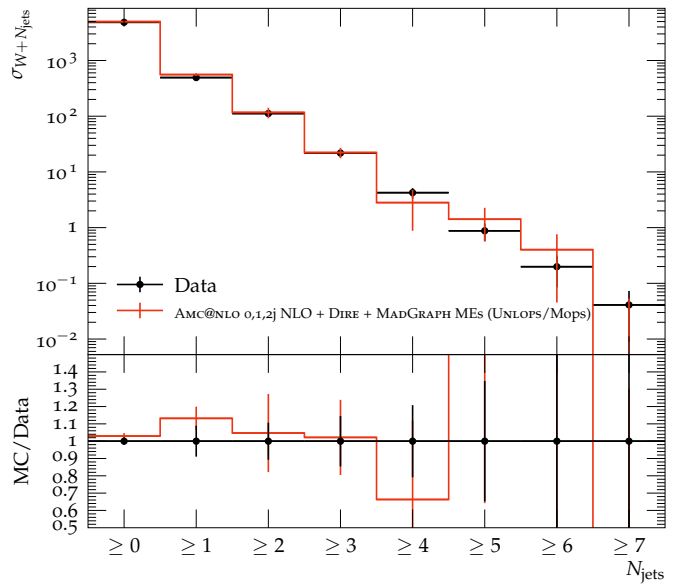
More physics features: Vincia and Dire matching/merging

plots from https://arxiv.org/abs/1907.08980, https://arxiv.org/abs/1706.06218, https://arxiv.org/abs/1805.03757









Vincia and Dire employ C++ matrix-element code to perform MECs & merging

⇒ New, more stable interface to MG5-generated C++ code. (thanks to V. Hirschi!)

Outlook: Weight-handling overhaul needed!

More and more parts of the code come with weights:

- heavy-ion event weight,
- merging weights,
- PS enhancement weights,
- Dire PS weights,
- LHE file multiweights,
- PS variation multiweights...

```
HepMC::GenCrossSection xsec;
xsec.set_cross_section( pyinfo->sigmaGen() * 1e9,
  pyinfo->sigmaErr() * 1e9);
evt->set_cross_section(xsec);
//evt->weights().push_back( pyinfo->weight() );
for (int iweight = 0; iweight < pyinfo->numberOfWeights();
  ++iweight) {
  std::string name  = pyinfo->weightNameByIndex(iweight);
  double value      = pyinfo->weightValueByIndex(iweight);
  evt->weights()[name] = value;
}
```

⇒ Really really need easy interface for everyone – and need to handle consistency internally.

Look for improvements in the next release – feedback will be very valuable!

High-multiplicity (MinBias) pp collisions @ LHC suggest extreme QCD behavior, otherwise only seen in pA or AA $\Rightarrow$ Common model needed!

Use full PYTHIA diffractive, MPI, PS, hadronization machinery to develop a microscopic model heavy-ion collisions $\Rightarrow$ ANGANTYR mode.

...switched on by using heavy ion beams

```
Beams:idA = 2212           // Proton beam
Beams:idB = 1000822080     // Lead beam
Beams:eA = 4000            // Proton energy
Beams:eB = 1570            // Energy per lead nucleon
```

...see e.g. `main112.cc` and `main113.cc` examples.

Note: changes in PYTHIA's pp model (diffraction) feed down to AA description. Help from experiments with systematic tuning effort?

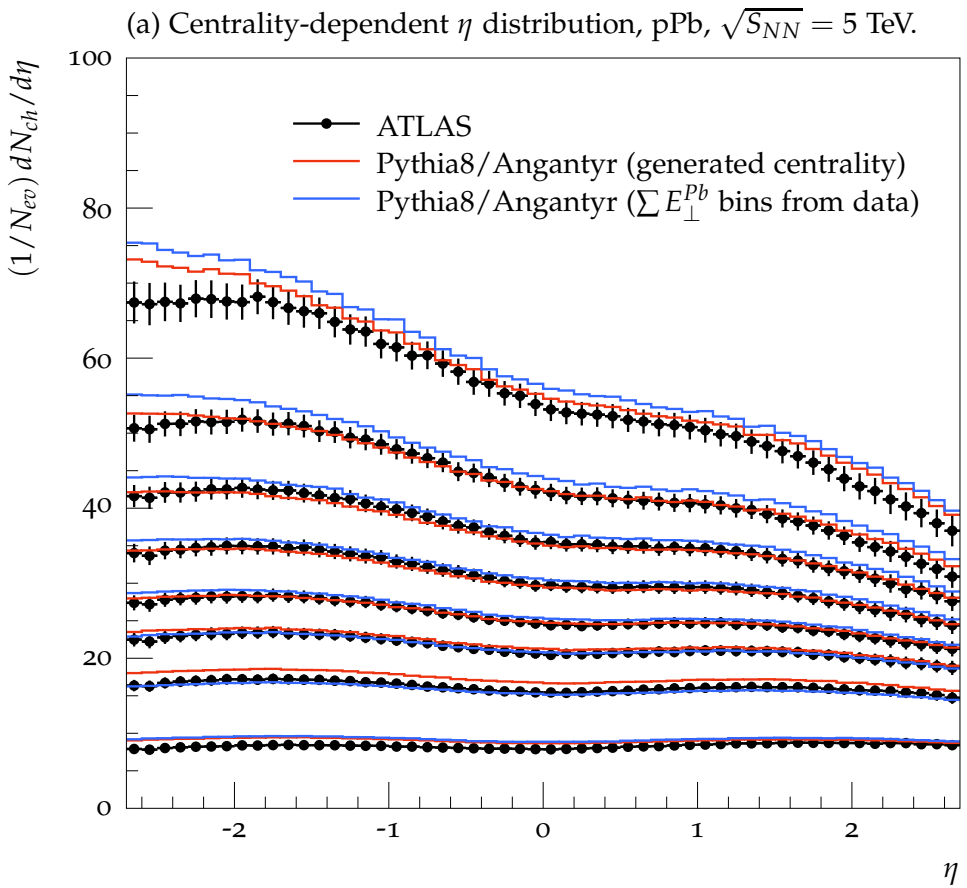
Physics capabilities: Heavy ion cross sections

see https://arxiv.org/abs/1806.10820

The ANGANTYR idea:

- Derive model of nuclear initial state *including event-by-event fluctuations of nucleon wavefunctions*;
- Pick nucleon-nucleon sub-collisions from wounded-nucleon-inspired model;
- Generate & combine full PYTHIA evts for each subcollision, secondary wounded nucleons are diffractive-like events.

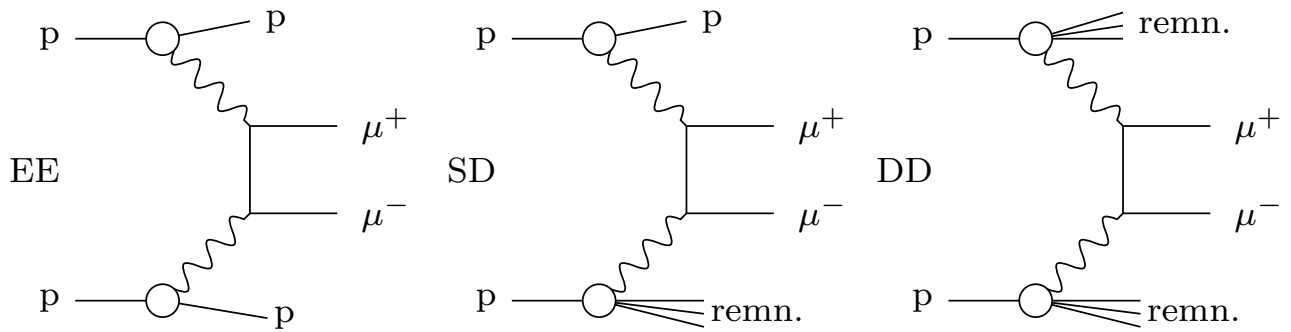
$\Rightarrow \eta = 0$ looks $\sim$ "high-E" scattering
$\eta \gg 0$ looks $\sim$ diffractive-like event



Future: Scatterings could eventually interact, e.g. to produce collectivity.

UPC can yield "clean" probes of non-perturbative & nuclear structure

For example in dimuon production:

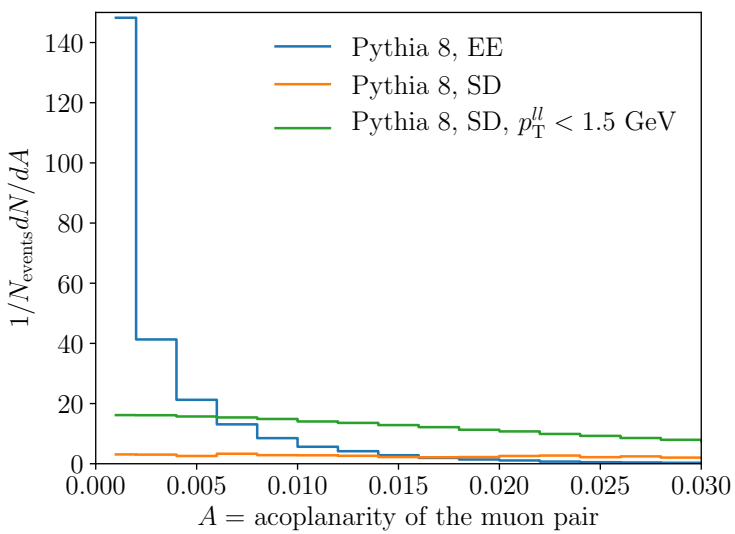


- Acoplanarity sensitive to initial γ virtuality, and requires model for direct production, single- and double diffraction.
- Final result in line with ATLAS [PLB 777 (2018) 303]

PYTHIA user can provide γ flux as a `PDF*`:

$$x f_{\gamma}^{\mathrm{p}}(x, Q^2) = \frac{\alpha}{2\pi} \frac{(1+(1-x)^2)}{Q^2} \frac{1}{(1+Q^2/(0.71\ \mathrm{GeV}^2))^4}$$

...see e.g. `main70.cc`

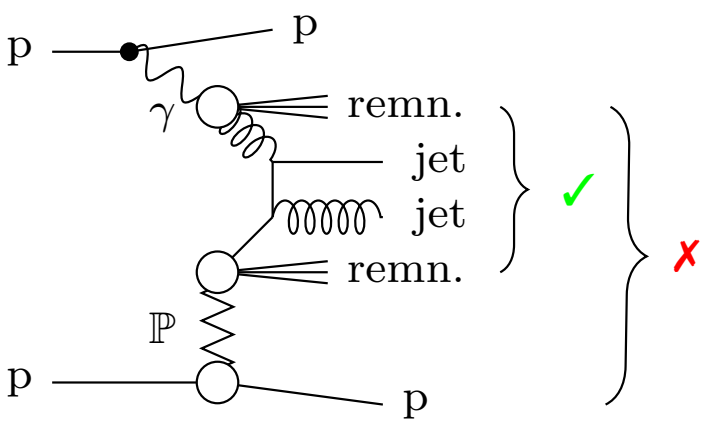


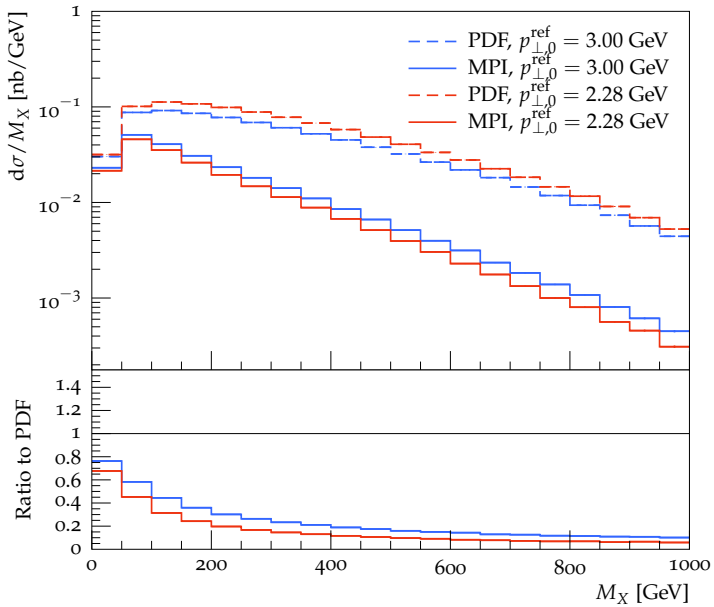
Physics capabilities: Ultra-peripheral collisions II

see https://arxiv.org/abs/1804.10373, https://arxiv.org/abs/1901.05261

Diffractive dijets in UPCs at LHC interpolate between small (HERA-like) and large (Tevatron-like) factorization breaking effects in hard diffraction.

$\Rightarrow$ Factorization breaking through dynamical rapidity gap survival





1. Select tentatively diffractive events based on diffractive PDFs

   `Diffraction:doHard = on, Diffraction:sampleType = 3,4`
2. Reject if MPIs in pp cloud the diffractive signal (but allow MPI in $\gamma\mathbb{P}$)

Summary

- PYTHIA 8.3 was released Oct. 30, 2019
  ...and includes many technical and administrational updates.
  ...we would value feedback on the new manual http://home.thep.lu.se/~torbjorn/pythia83html/Welcome.html, on using gitlab, docker containers and the new python interface!
- VINCIA and DIRE are now part of the core distribution, which will make use and comparisons much easier. Perturbative precision will continue to increase.
- Microscopic heavy-ion collision modeling major aspect of PYTHIA, addressing all things collective in pp, pA and AA collisions in one framework.
- Ultra-peripheral collision machinery can be applied to many processes at the LHC