Swift-HEP Generators: Status Update

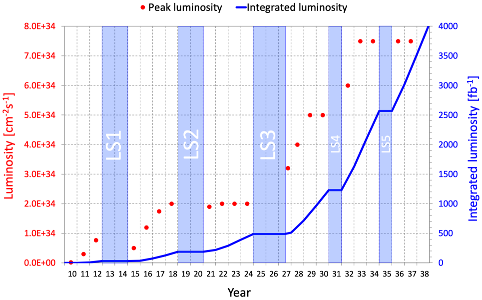
Enrico Bothmann (Göttingen), Andy Buckley (Glasgow),
Ilektra Christidi (UCL), Christian Gütschow (UCL),
Stefan Höche (FNAL), Max Knobbe (Göttingen),
Marek Schönherr (Durham)

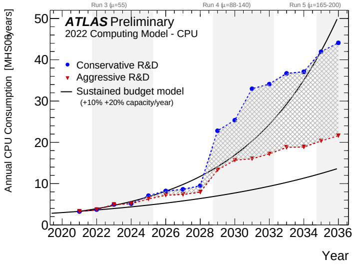
Swift-HEP workshop

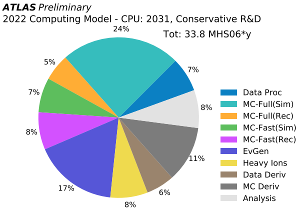
21 November 2023

Expected computing requirements

- ➜ projected evolution of computing resources sees cost of event generation on par with detector simulation
- ➜ LHC measurements in danger of being limited by Monte Carlo statistics







[CERN-LHCC-2022-005]

Systematic profiling

- Most event generation CPU spent on multi-leg NLO calculations [JHEP 08 (2022) 089]
  - used for main Standard Model processes
  - relevant to measurements and searches alike
  - extremely large event sample sizes

Systematic profiling

- Most event generation CPU spent on multi-leg NLO calculations [JHEP 08 (2022) 089]
  - used for main Standard Model processes
  - relevant to measurements and searches alike
  - extremely large event sample sizes
- Study CPU performance of MEPS@NLO calculations for $e^+e^- + 0, 1, 2j$@NLO$+3, 4, 5j$@LO and $t\bar{t} + 0, 1j$@NLO$+2, 3, 4j$@LO with Sherpa 2.2.11, OpenLoops 2.1.2 and LHAPDF 6.2.3 using VTune 2021.7.1

Systematic profiling

- Most event generation CPU spent on multi-leg NLO calculations [JHEP 08 (2022) 089]
  - used for main Standard Model processes
  - relevant to measurements and searches alike
  - extremely large event sample sizes
- Study CPU performance of MEPS@NLO calculations for $e^+e^- + 0, 1, 2j$@NLO$+3, 4, 5j$@LO and $t\bar{t} + 0, 1j$@NLO$+2, 3, 4j$@LO with Sherpa 2.2.11, OpenLoops 2.1.2 and LHAPDF 6.2.3 using VTune 2021.7.1
- performance dependence on the number of multiweights studied using different setups:
  - baseline MEPS@NLO (no variations)
  - + $\mathrm{EW}_{\mathrm{virt}}$ corrections
  - + 7-point variations of factorisation and renormalisation scales in matrix element and parton shower
  - + 100 (1000) NNPDF3.0nnlo replicas

Systematic profiling

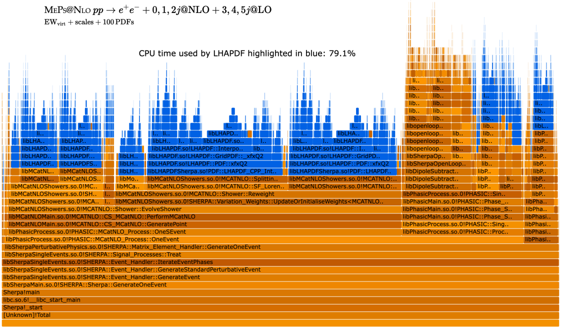
- Most event generation CPU spent on multi-leg NLO calculations [JHEP 08 (2022) 089]
  - used for main Standard Model processes
  - relevant to measurements and searches alike
  - extremely large event sample sizes
- Study CPU performance of MEPS@NLO calculations for
  $e^+e^- + 0, 1, 2j$@NLO+$3, 4, 5j$@LO and $t\bar{t} + 0, 1j$@NLO+$2, 3, 4j$@LO
  with Sherpa 2.2.11, OpenLoops 2.1.2 and LHAPDF 6.2.3 using VTune 2021.7.1
- performance dependence on the number of multiweights studied using different setups:
  - baseline MEPS@NLO (no variations)
  - + EW_{virt} corrections
  - + 7-point variations of factorisation and renormalisation scales in matrix element and parton shower
  - + 100 (1000) NNPDF3.0nnlo replicas
- detailed write-up presented in [EPJC 82 (2022) 12]

Initial profiling exercises

- first generator CPU profiling done by Tim Martin suggested per-event CPU dominated by LHAPDF

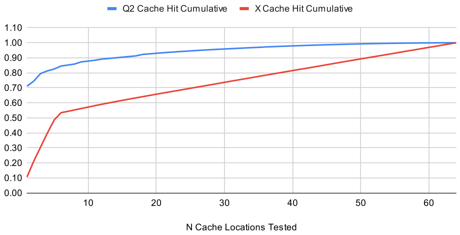
- graph shows PDF calls highlighted in blue (using LHAPDF 6.2.3)

- maybe not completely surprising: multiweights originally not designed with hundreds of variations in mind [**EPJC 76 (2016) 11**]



- explore two approaches in parallel: make LHAPDF faster and rework LHAPDF call strategy

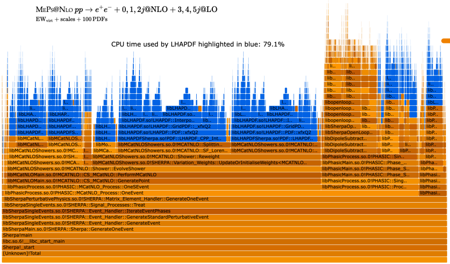
Improving LHAPDF

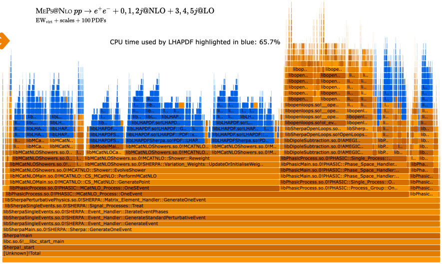


- first PDF-grid cache introduced in v6.3.0
  - rendered ineffective by PDF-call strategy used in Sherpa
  - nevertheless useful as case study
- follow-up release v6.4.0 with improved interpolation logic
  - revised cache implementation with improved memory layout (but well-matched call strategy in the generator still crucial)
  - pre-computation of shared coefficients of the interpolation polynomial along (x, Q^2) grid lines
  - results in factor 3 speed-up for single flavour computations
  - can achieve factor 10 speed-up when combining with multi-flavour caching

Impact of new LHAPDF

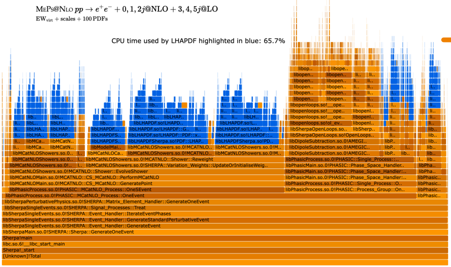
- ATLAS V+jets setup **overall 30% faster** using new LHAPDF release
  - switching from old ATLAS production default v6.2.3 to new v6.4.0 release

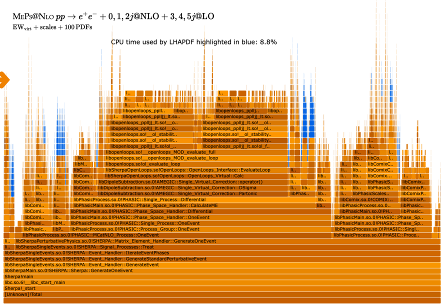




Internal restructuring and pilot run

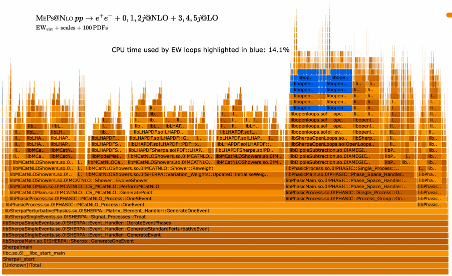
- perform the unweighting using a minimal setup and once an event is accepted, rewind RNG state and re-calculate accepted event using all the bells and whistles
- **achieves factor 5 speed improvement** for ATLAS setup (using LHAPDF 6.4.0 yields additional 6% speed-up)
- pilot run reduces CPU spent on evaluating PDFs to below 10%



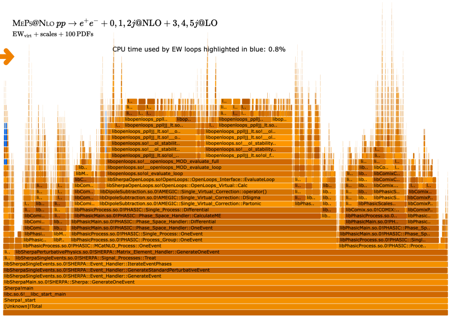


Internal restructuring in Sherpa 2.2.12: the pilot run

- ➜ CPU spent on calculating EW one-loop amplitudes going from 19% down to 0.8% when using the pilot run with the ATLAS V+jets setup
- ➜ nevertheless, $\sim$40% of the CPU still spent on calculating QCD loops

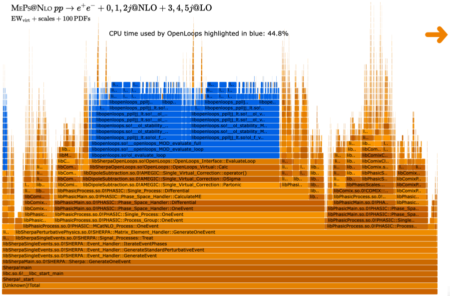


➜

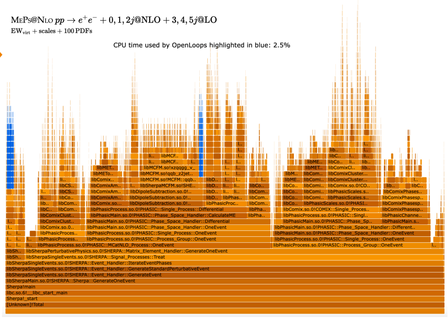


Analytic vs numerical QCD loop amplitudes

- ➜ employ analytic one-loop amplitudes (if available) in the pilot run using Sherpa-MCFM interface [**EPJC 81 (2021) 12**]
- ➜ yields **additional ∼35% speed improvement** for the V+jets setup



➜

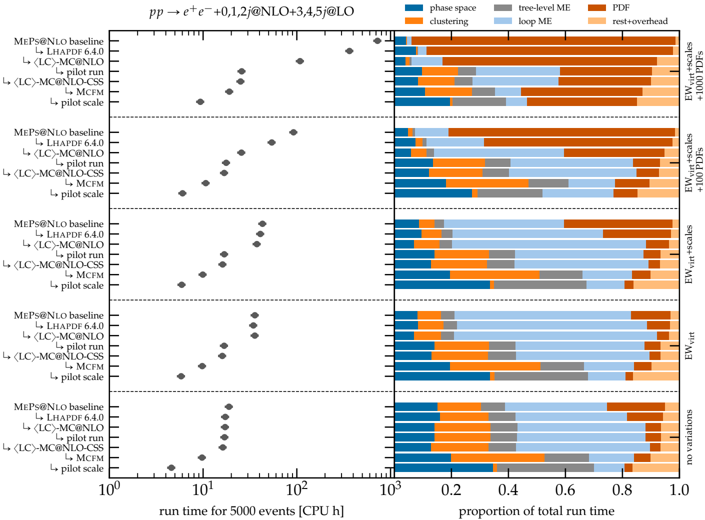


Full suite of improvements

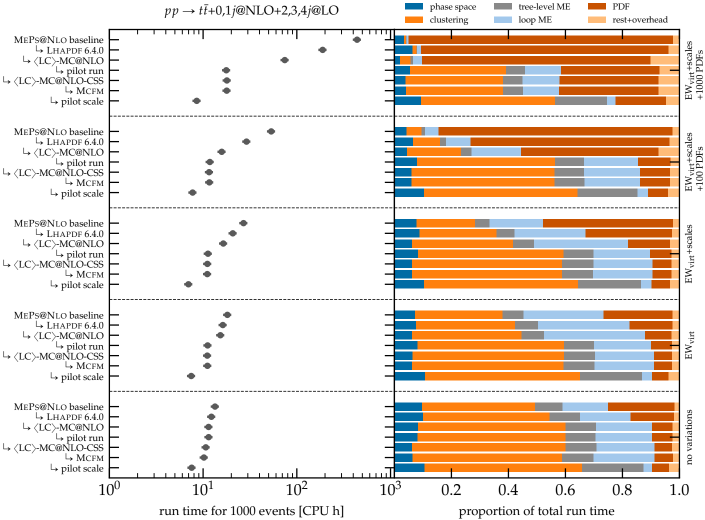
- study the impact of different improvements sequentially:
  - improved interpolation strategies in LHAPDF (6.2.3 $\rightarrow$ 6.4.0)
  - replace full-colour spin-correlated S-MC@NLO algortihm with leading-colour spin-averaged $\langle LC \rangle$-MC@NLO (`NLO_CSS_PSMODE` 0 $\rightarrow$ 1)
    - this disables subleading colour corrections in the parton shower
  - introduce pilot run in Sherpa (2.2.11 $\rightarrow$ 2.2.12)
  - defer leading-colour MC@NLO until after the unweighting (`NLO_CSS_PSMODE` 1 $\rightarrow$ 2)
  - use analytic one-loop amplitudes from MCFM in pilot run
  - use a simplified pilot scale for the unweighting

cumulative speed-ups for:	$pp \rightarrow e^+e^-$ + jets			$pp \rightarrow t\bar{t}$ + jets		
	runtime [CPU h/5k events]			runtime [CPU h/1k events]		
setup variant	old	new	speed-up	old	new	speed-up
no variations	20 h	5 h	4×	15 h	8 h	2×
EW_{virt}	35 h	5 h	6×	20 h	8 h	2×
EW_{virt} +scales	45 h	5 h	7×	25 h	8 h	4×
EW_{virt} +scales+100 PDFs	90 h	5 h	15×	55 h	8 h	7×
EW_{virt} +scales+1000 PDFs	725 h	8 h	78×	440 h	9 h	51×

Breakdown of CPU budget in V+jets

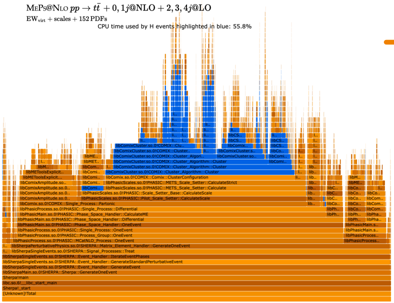


Breakdown of CPU budget in $t\bar{t}$+jets

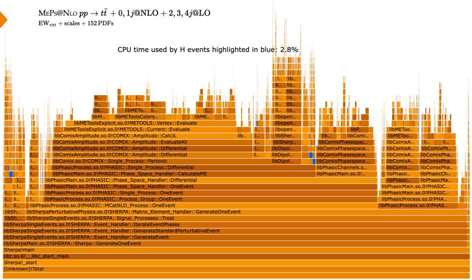


Cluster-independent scale definition

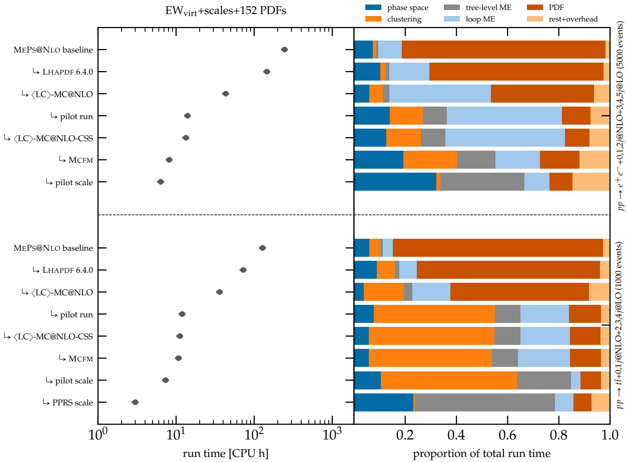
- ➜ employ clustering-independent scale definition ($H'_\mathrm{T}/2$) for $\mathcal{H}$-events in $t\bar{t}$+jets (already used in V+jets baseline setup)
- ➜ yields **additional factor 2 speed-up** of the overall run time



➜



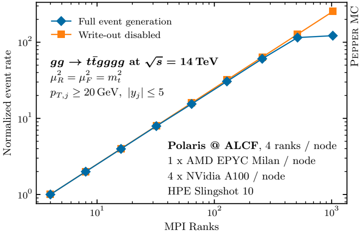
Case study: latest ATLAS baseline configuration

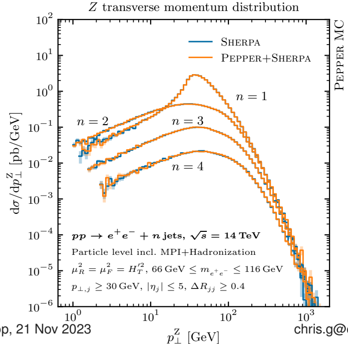


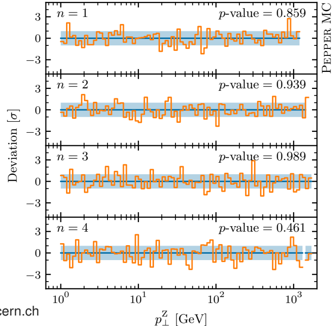
➜ CPU consumption **overall improved by factors of ×39 and ×43** for V+jets and $t\bar{t}$+jets

Interlude: Sherpa with GPU acceleration

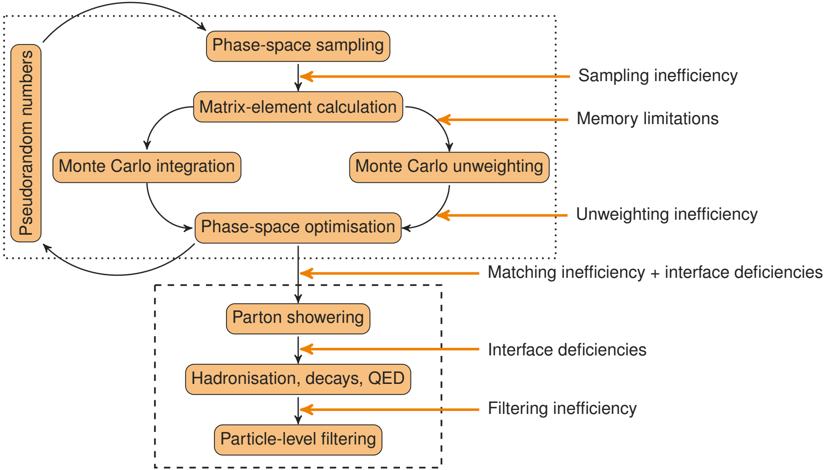
- First production-ready release of a portable GPU-accelerated LO-matrix-element generator Pepper and corresponding phase-space generator Chili → [arXiv:2311.06198]
- With this, ATLAS' estimated 300B V+jets events needed for HL-LHC could be run on Frontier (4h), Aurora (6h), Leonardo (8h), Lumi (15h)







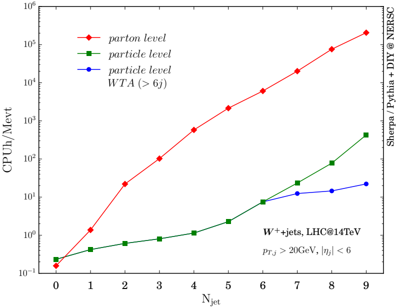
Other bottlenecks

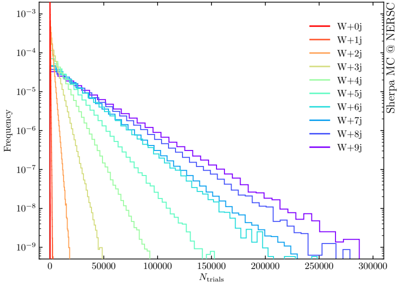


→ Lack of active development on infrastructure tools (LHE, HepMC, ...) set to become a major bottleneck going forward

Parton vs particle level

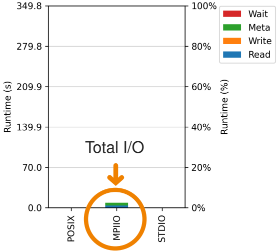
- Scaling of parton- and particle level analysed in [**PRD 100 (2019) 1**]
- cost of showering matrix elements with extra emissions **dominated by parton level**
  - number of diagrams grows factorially with every additional emission (at best exponentially when exploiting recursions a la `COMIX`)
- low-multiplicity matrix elements cheaper to regenerate entirely than to store on disk

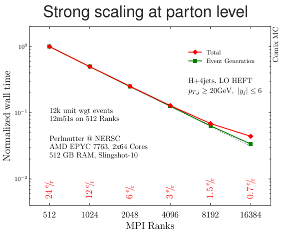


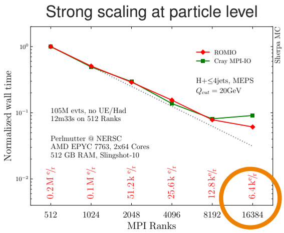


Introducing LHEH5

→ new efficient LHE-like data format based on HDF5+HighFive proposed in [arXiv:2309.13154]







105M events / min

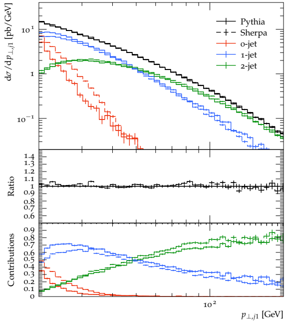
→ overall I/O time reduced to below 1s per rank

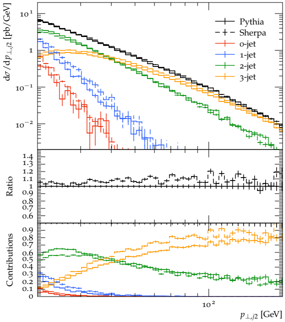
→ time spent in I/O operations less than 5% when reading 128.85 GiB

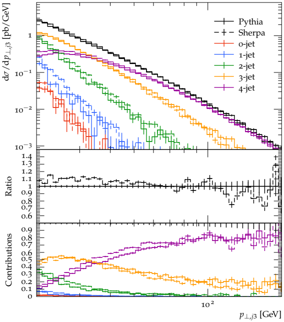
→ ideal for accessing back-fill queues at large computing centres

More robust uncertainty estimates

- LHEH5 inputs are already supported by both Sherpa and Pythia!
- 10 % uncertainty seen in Z+jets due to different algorithmic choices in the parton showers







Future event generation workflows

- ➜ Approach 1: produce parton-level samples centrally with input from the MC developers, provide them in a shared space for all experiments
  - ➜ experiments run their preferred shower setup (✔)
  - ➜ allows for affordable plug & play between different models (✔)
  - ➜ lowers cost threshold for reproducing larger setups after some time if need be (✔)
  - ➜ requires more storage for parton-level events (✘)
  - ➜ new infrastructure needs to be set up and maintained (✘)

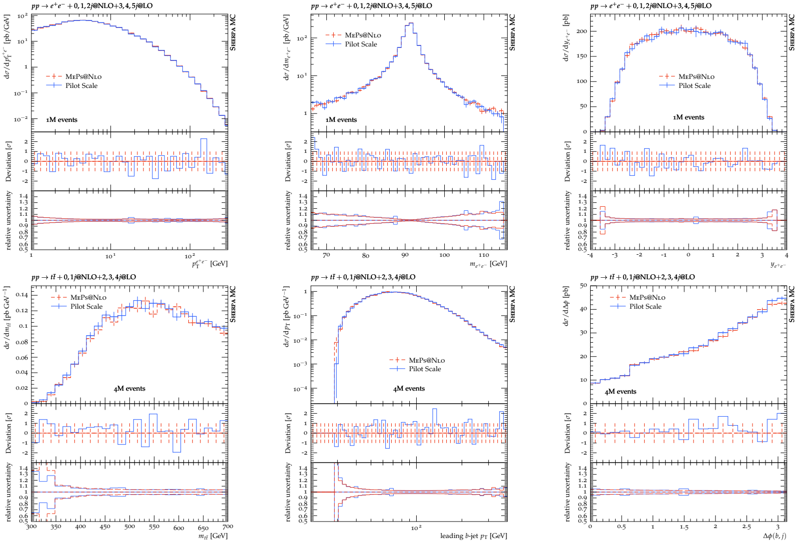
Future event generation workflows

- Approach 1: produce parton-level samples centrally with input from the MC developers, provide them in a shared space for all experiments
  - experiments run their preferred shower setup (✔)
  - allows for affordable plug & play between different models (✔)
  - lowers cost threshold for reproducing larger setups after some time if need be (✔)
  - requires more storage for parton-level events (✘)
  - new infrastructure needs to be set up and maintained (✘)
- Approach 2: run everything in one go, harnessing heterogeneous resources, possibly with in-memory transfer of GPU-accelerated calculation components
  - no intermediate storage for parton level events needed (✔)
  - minimal infrastructure changes required (✔)
  - parton-level events continue to cost twice as strictly necessary (✘)
  - regenerating larger setups from scratch will become painful (✘)

Summary

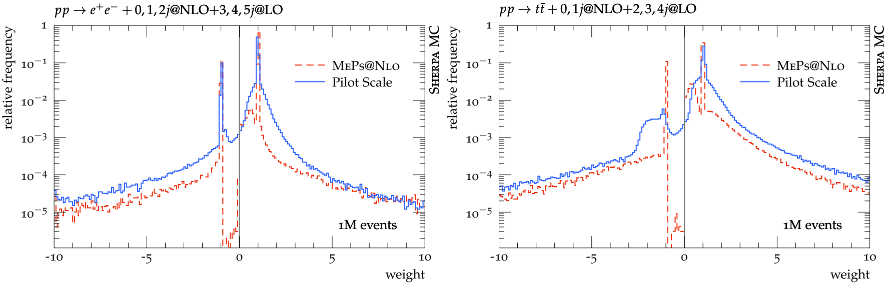
- factor 40–80 speed-up following dedicated profiling of ATLAS multi-leg NLO setups
  - LHAPDF 6.4 release series brings major performance improvements with noticeable impact on overall event-generation run time
  - introduction of pilot run in Sherpa brings a factor 5 improvement
  - using analytic QCD loop amplitudes in the unweighting brings another factor 1.5
- achieves major factor-10 milestone set by HSF Generators group
- new `LHEH5` format allows for efficient parton-level event generation
  - facilitates more robust uncertainty estimates of parton-shower effects
  - additional factor 3–6 speed-up for traditional grid resources
- seeing latest performance improvements reflected in up-to-date projections from the experiments paramount for defining appropriate objectives going forward

Comparison of MEPS@NLO vs Pilot Scale strategy



Weight distribution for pilot scale

➜ weight distributions for partially unweighted events after matching and merging:



➜ second unweighting would reduce the efficieny by less than factor 2 for large N_{events}