Towards efficient and sustainable event generation for the HL-LHC

Jonathan Butterworth, Christian Gütschow et al

GridPP/Swift-HEP, Manchester

11 April 2025

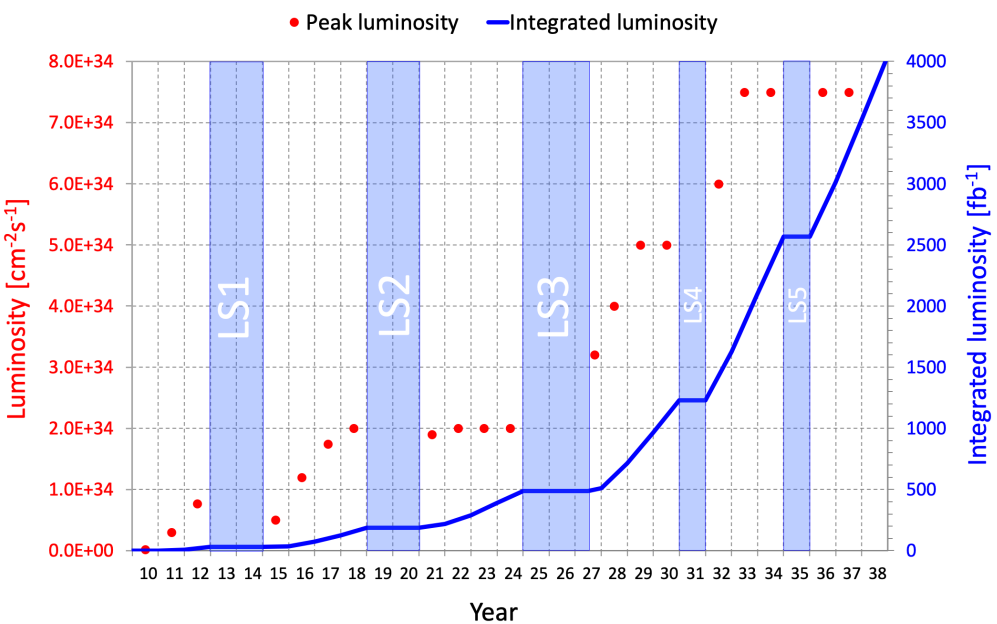
UCL ARC
ADVANCED RESEARCH COMPUTING

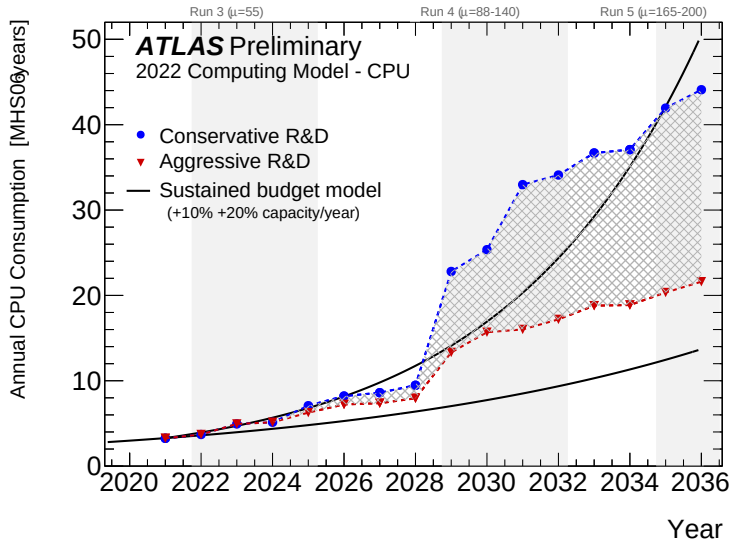
MCnet

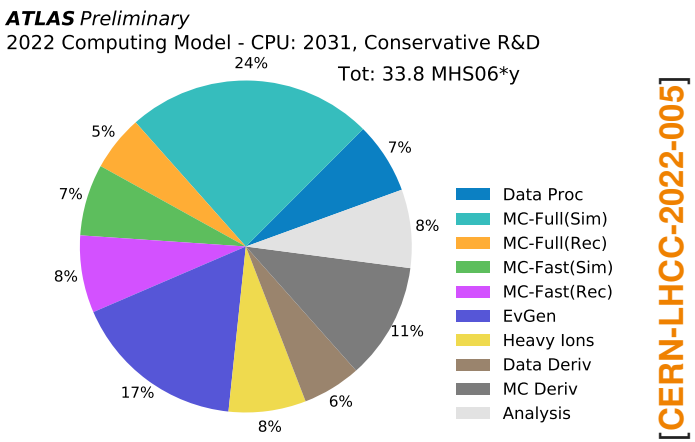
SWIFT HEP

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

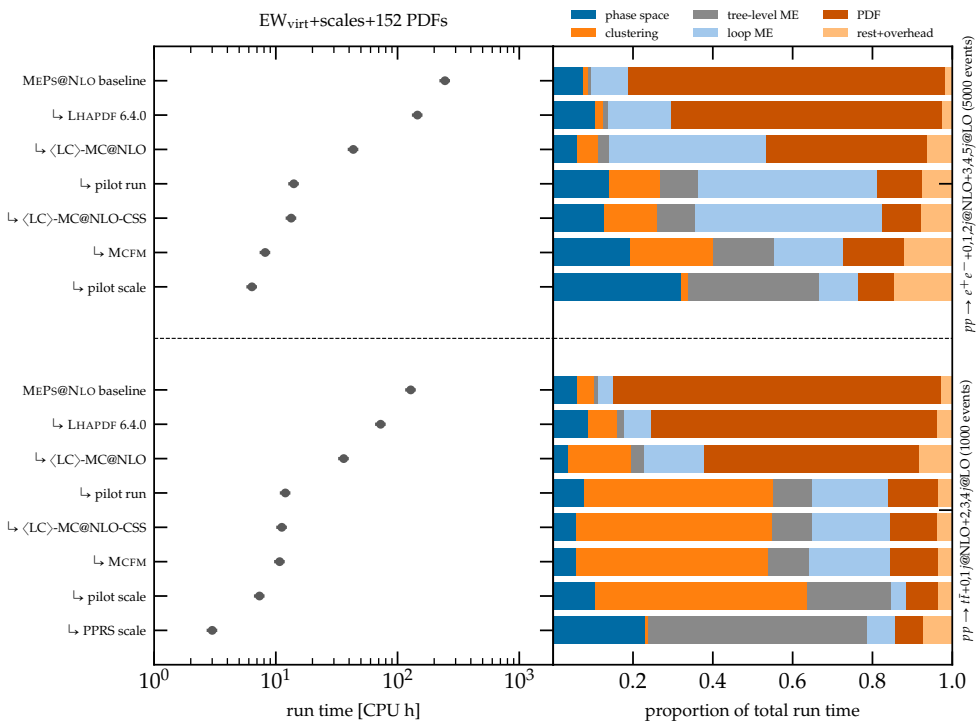






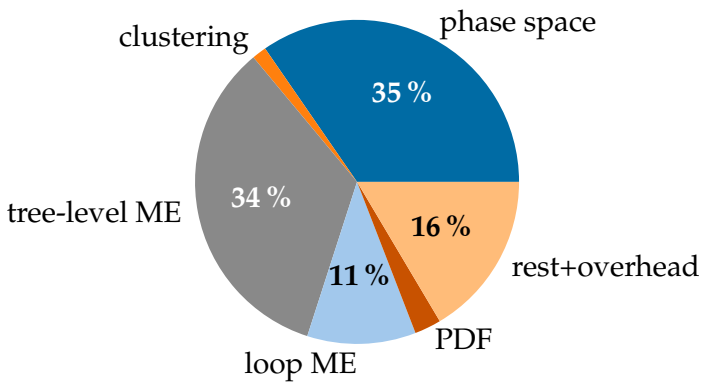
Targeted optimisation of CPU-based event generation

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



[EPJ C82 (2022) 12]

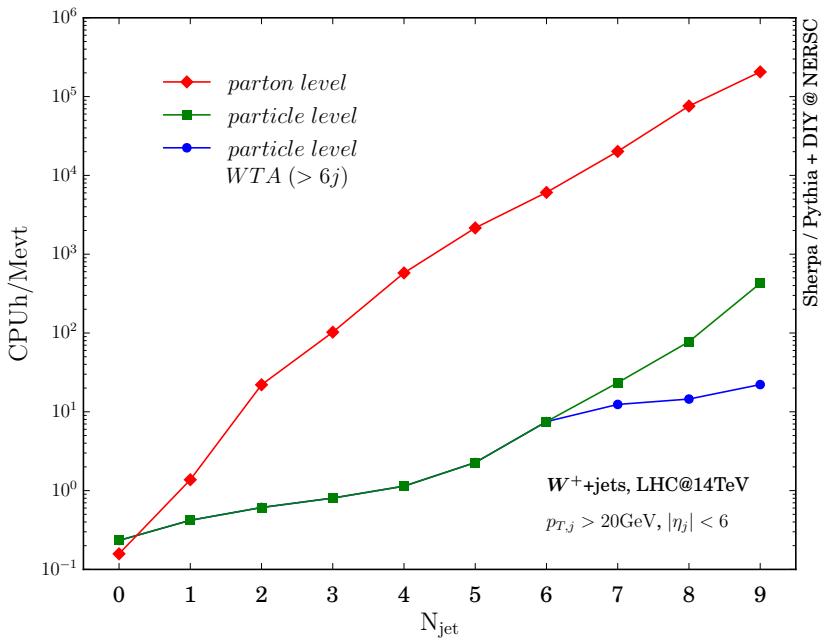
$pp \to e^+e^-$+0,1,2j@NLO+3,4,5j@LO

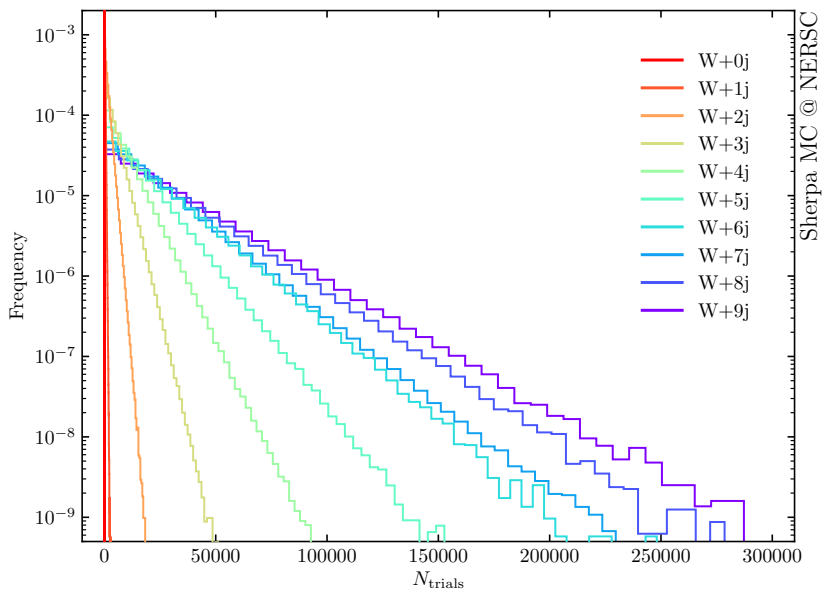


- CPU consumption **overall improved by factors of $\times$39 and $\times$43** for V+jets and $t\bar{t}$+jets

Parton vs particle level

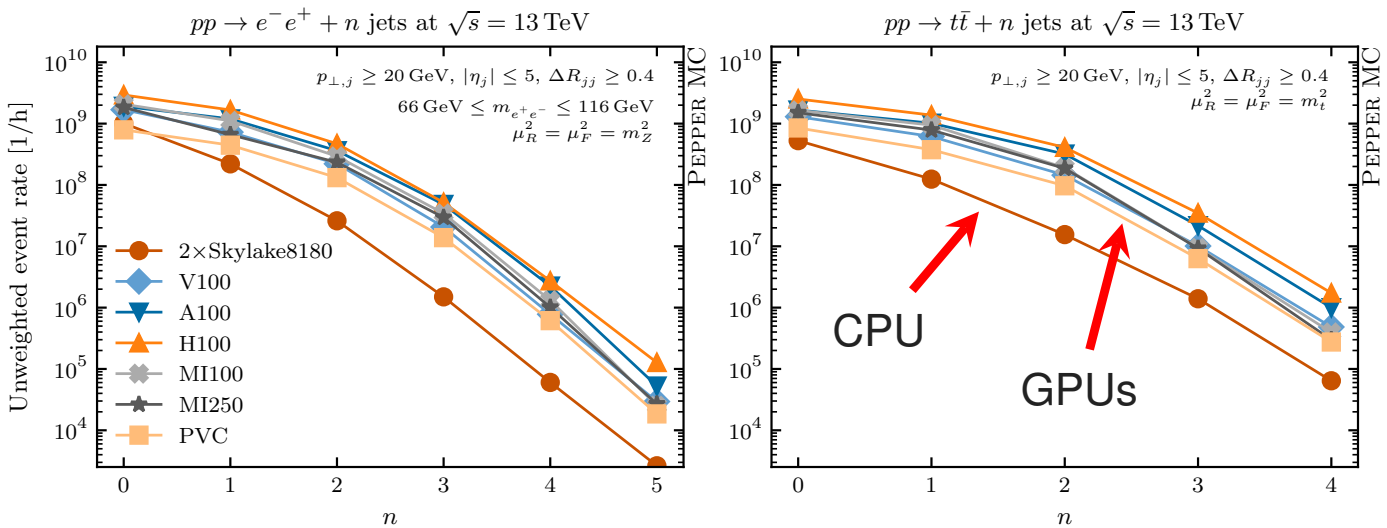
- Scaling of parton- and particle level analysed in [PRD 100 (2019) 1]
- cost of showering matrix elements with extra emissions **dominated by parton level**
  - number of diagrams grows factorially with every additional emission
    (at best exponentially when exploiting recursions a la COMIX)
- low-multiplicity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





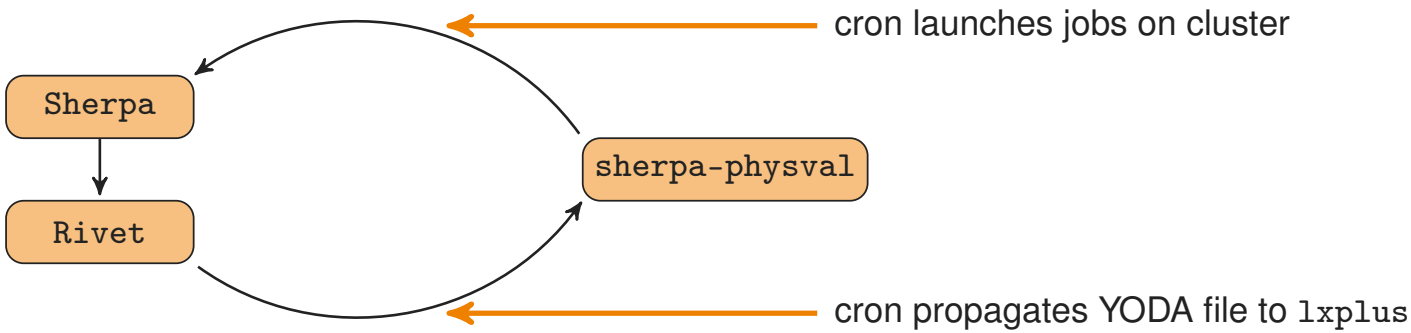
Pepper: a portable ME generator for the HL-LHC

- Computing architectures undergoing a big change (partly due to AI trends)
  - HPC moves to exascale era, which requires scalability; GPU acceleration requires portability
- PEPPER addresses both aspects with **MPI**, **HDF5** and **Kokkos**
  - PEPPER **parallelises** the entire parton-level event generation
  - Excellent performance across a wide range of architectures

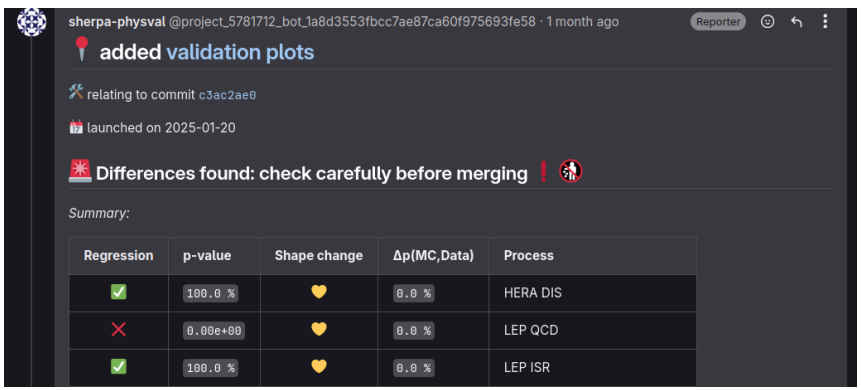


- **Successful DiRAC RAC proposal** for a GPU-accelerated V+jets sample with Pepper to be made available for ATLAS, CMS and the wider HEP community via CERN OpenData

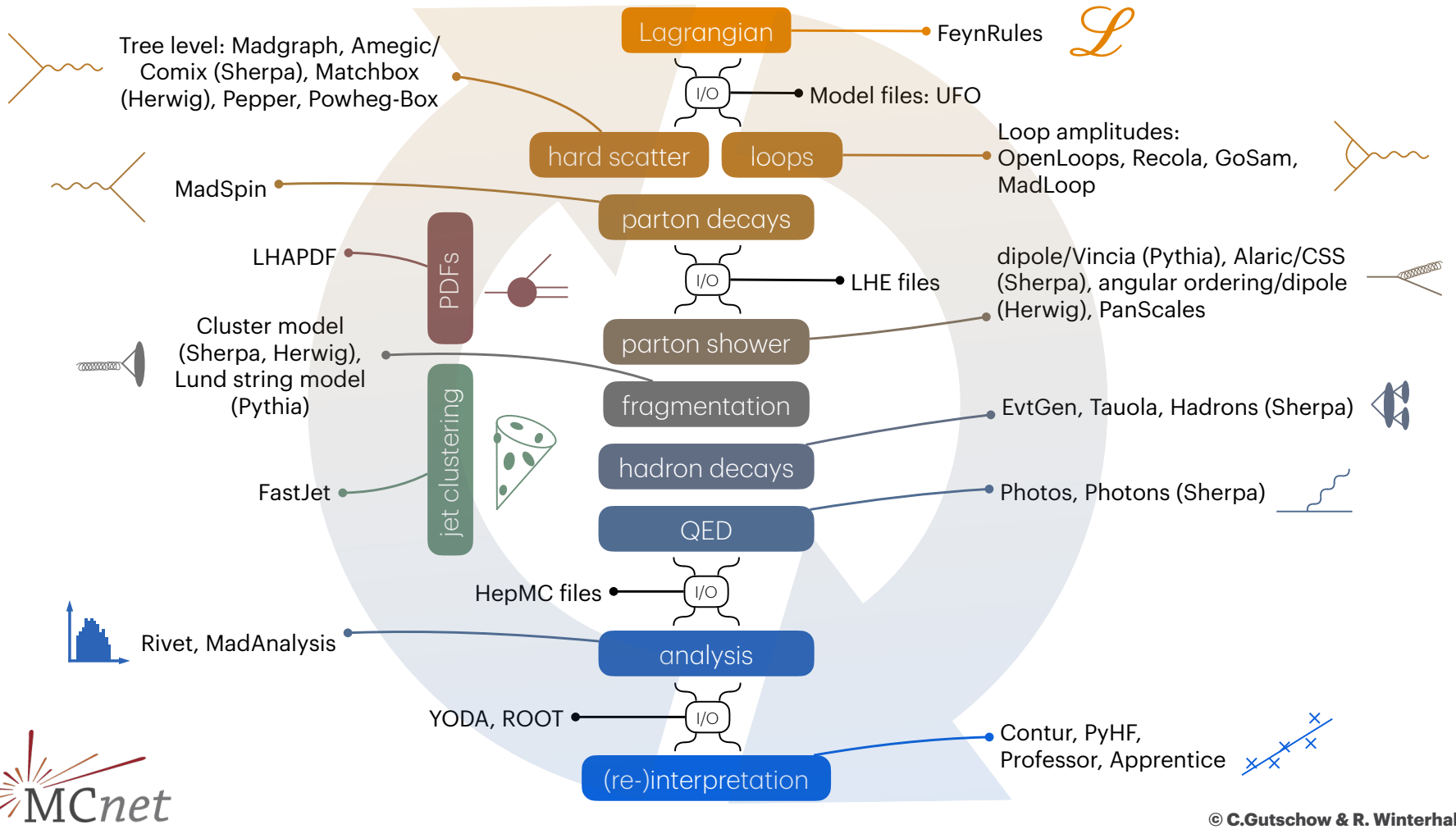
Fully automated physics validation in Sherpa



- Cron jobs at UCL, Göttingen and Durham run large-scale physics validation
- Plots automatically collated on EOS-hosted [validation page]
- Summary pushed to open MR associated with the branch



MC tools ecosystem



© C.Gutschow & R. Winterhalder

MC support tools updates

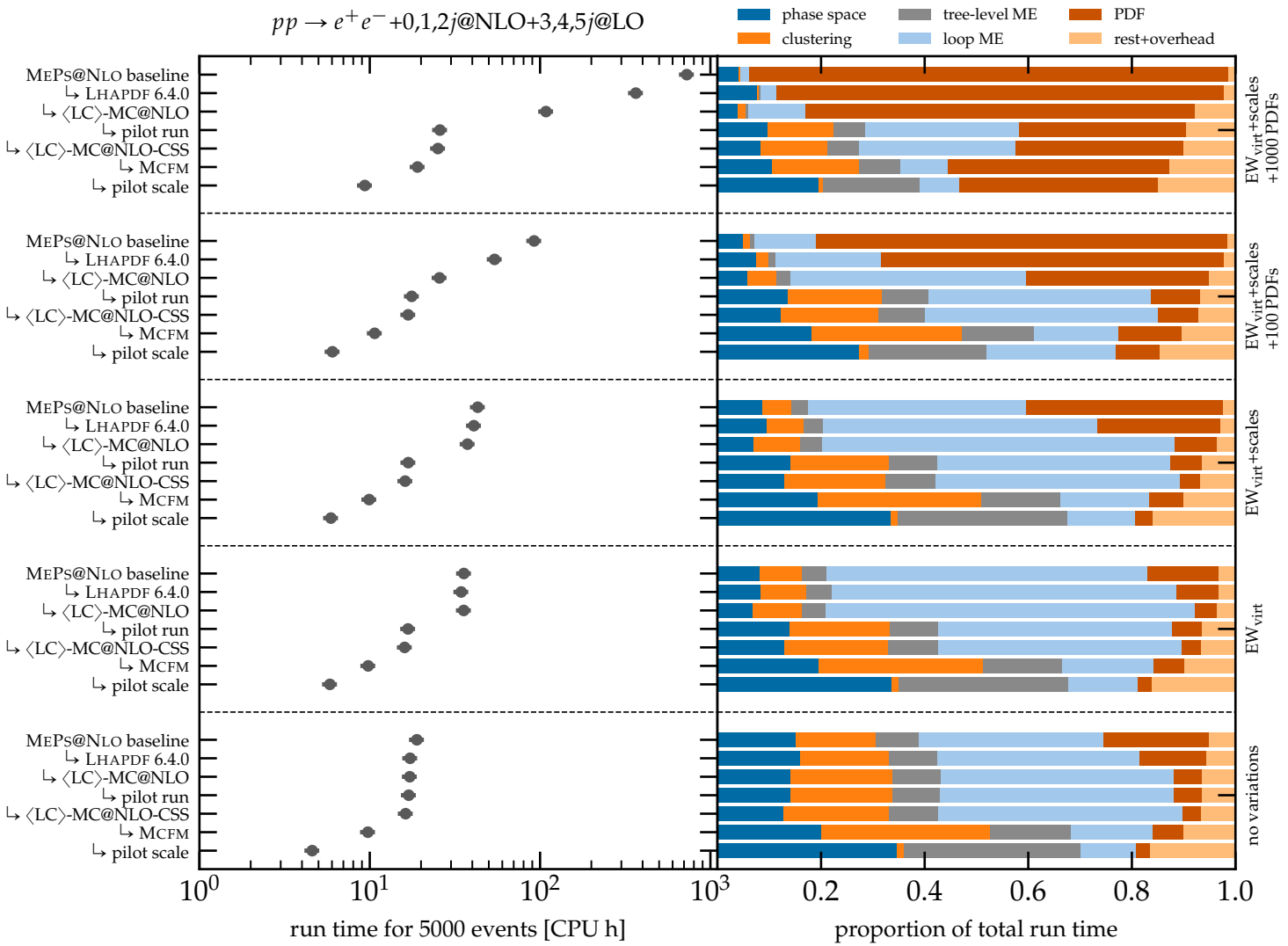
- new minor releases for YODA (v2.1) and Rivet (v4.1)
  - YODA: new HDF5 format for better HPC support/ML interfacing
  - Rivet: new BSM-oriented API to reduce threshold for preservation of search-type analyses
- various tools: automated unit-test coverage reporting as part of CI pipelines
- LHAPDF: several features in development
  - GPU: Kokkos interpolations being tested+optimised, release in summer
  - 2D spline interpolation by Glasgow MSci: prototype done, aim at N3LO calcs where higher-order smoothness matters; needs optimisation, stabilisation and code/API merging
  - flexibility: refactor to split ipol measures from algs, allow multidim extension: applications in pion/photon PDFs, nuclear corrections, frag functions, neutrino flux functions, ...
- just last week: another [**IPPP workshop**] on Monte Carlo support tools (supported by CoSeC)
- focus will now shift towards systematic profiling of HepMC
- General theme: make event generation maximally HPC friendly and develop the MC infrastructure to deliver the best possible calculations and robust uncertainties

Summary

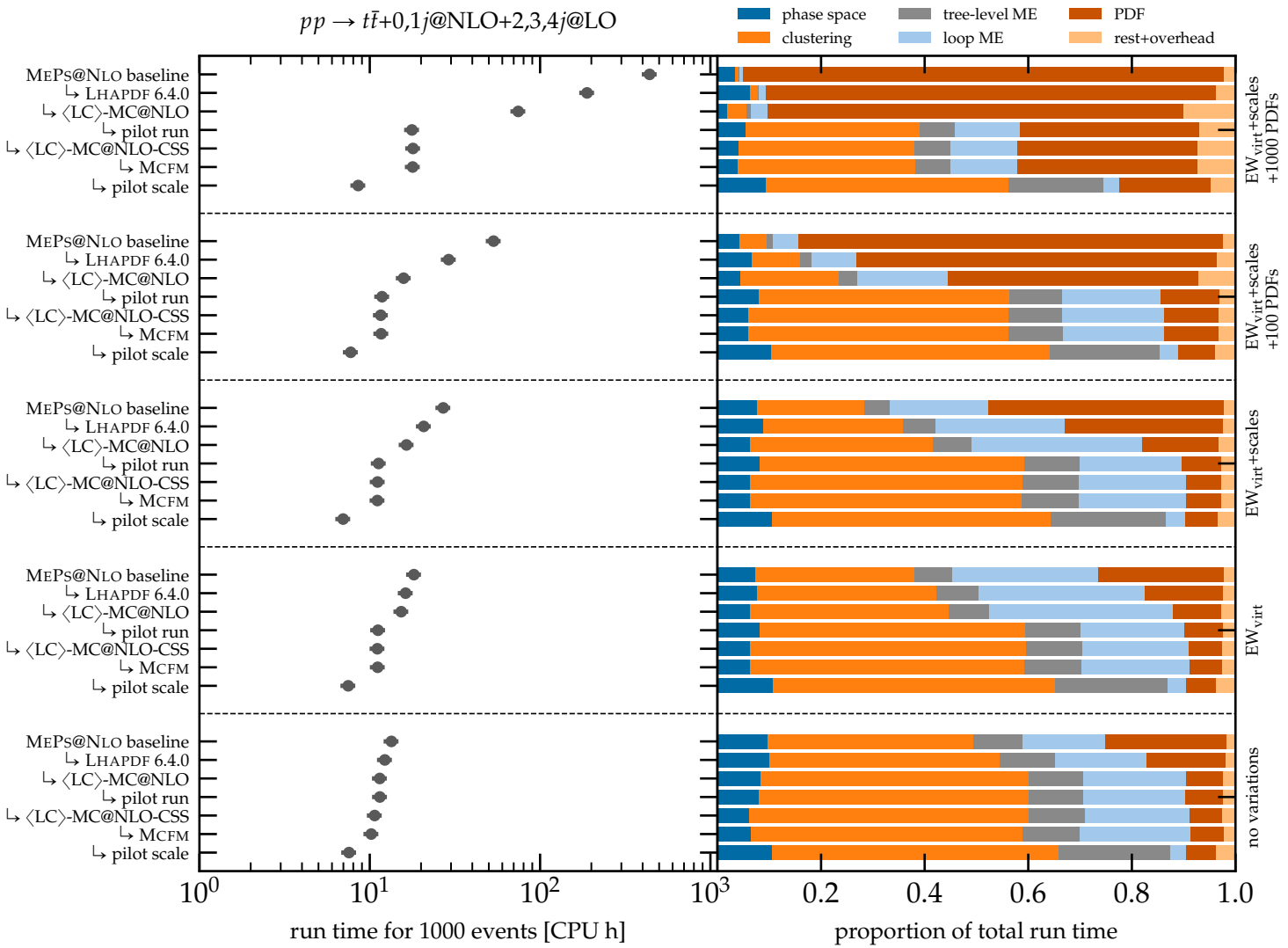
- computing cost of traditional CPU-based multi-leg event generation significantly reduced by factor 40–80 following dedicated profiling [EPJ C82 (2022) 12]
- first production-ready portable LO event generator allows to incorporate GPU resources into high-precision simulations [SciPost Phys 15 (2023) 4]
- this constitutes the first realistic and sustainable approach towards large scale event generation in the HL-LHC era
- well positioned to exploit Leadership Computing Facilities across the world
  - DiRAC just allocated some resources for a proof of principle
- focus now shifting towards better calculations and more robust uncertainty estimates $\Rightarrow$ **MORE AND BETTER PHYSICS FROM HL-LHC ++**

Backup

Breakdown of CPU budget in V+jets



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

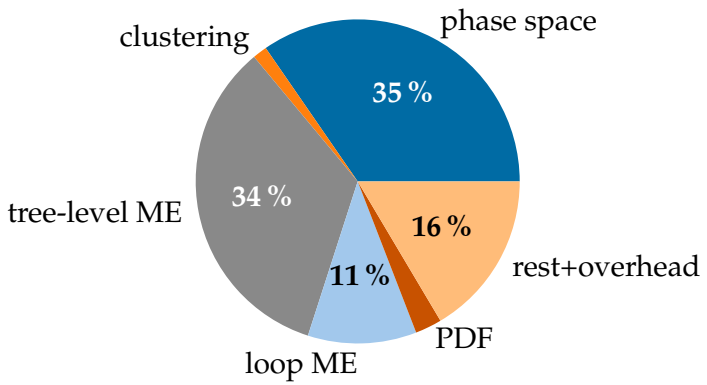


Parton-level event generation

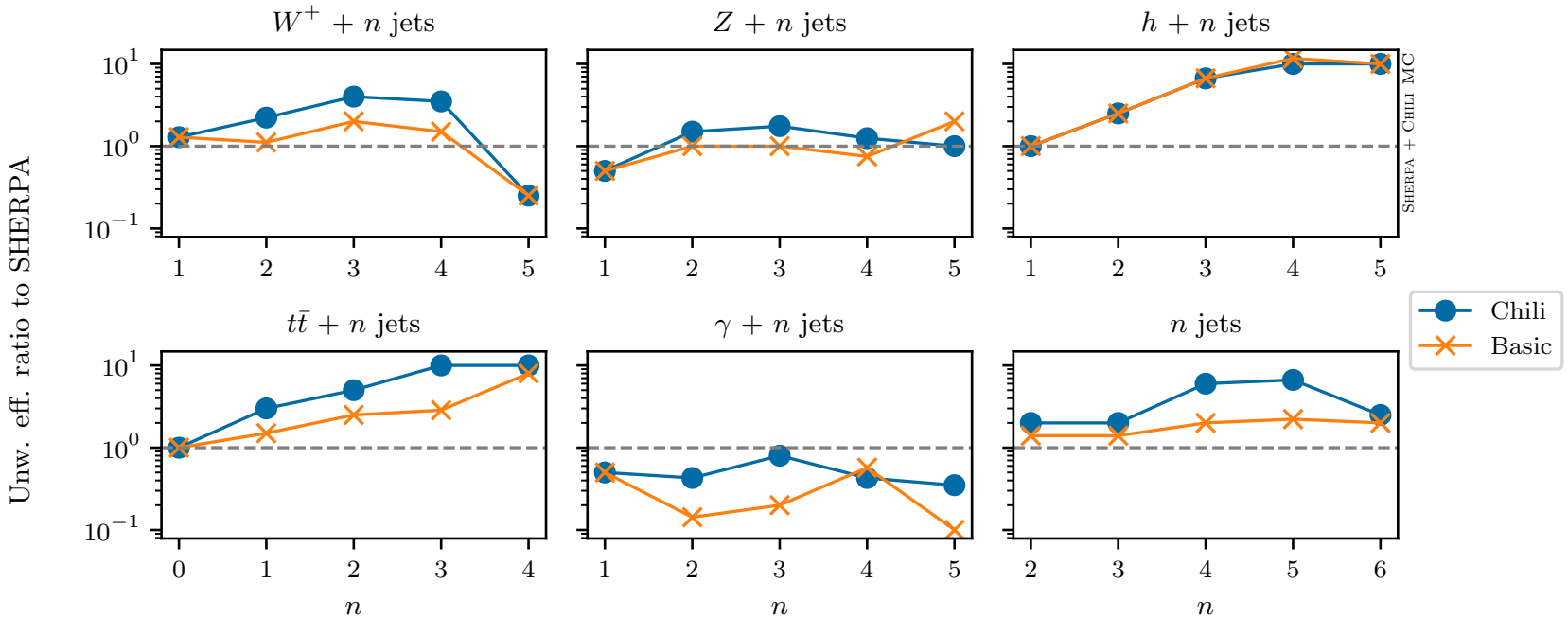
$$\sigma_{pp\to X_n} = \sum_{ab} \int dx_a dx_b d\phi_n \times f_a(x_a, \mu_F^2) f_b(x_b, \mu_F^2) \times |\mathcal{M}_{ab\to X_n}|^2 \times \Theta(p_1, \ldots, p_n)$$

- Large portion of MC time spend in ME+PS
- Components we need to consider:
  - Phase space generation → CHILI
  - PDFs → LHAPDF
  - Tree-level matrix elements → PEPPER

$pp \to e^+e^-$+0,1,2j@NLO+3,4,5j@LO



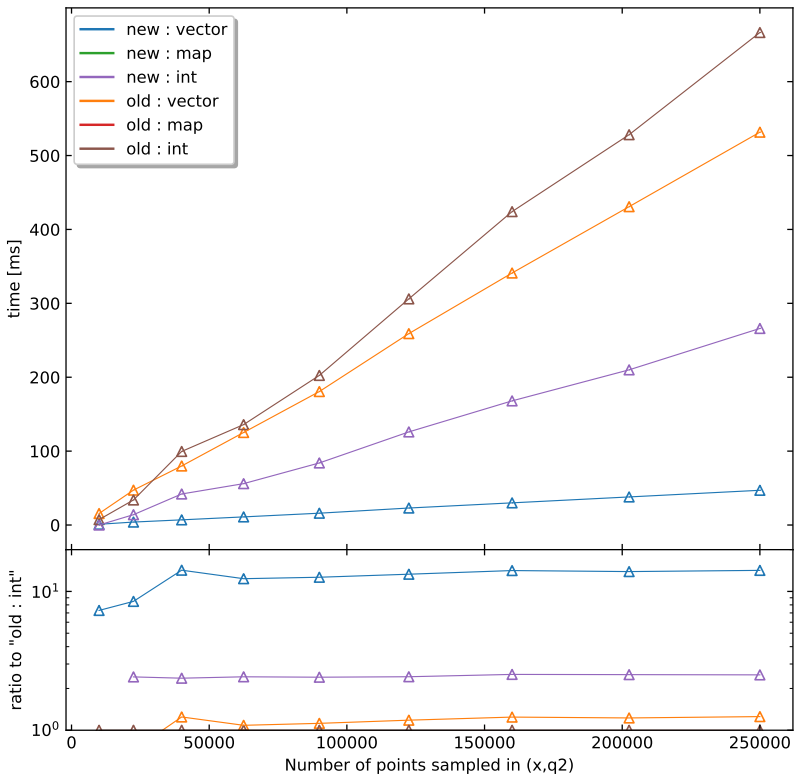
Chili performance compared to SHERPA default



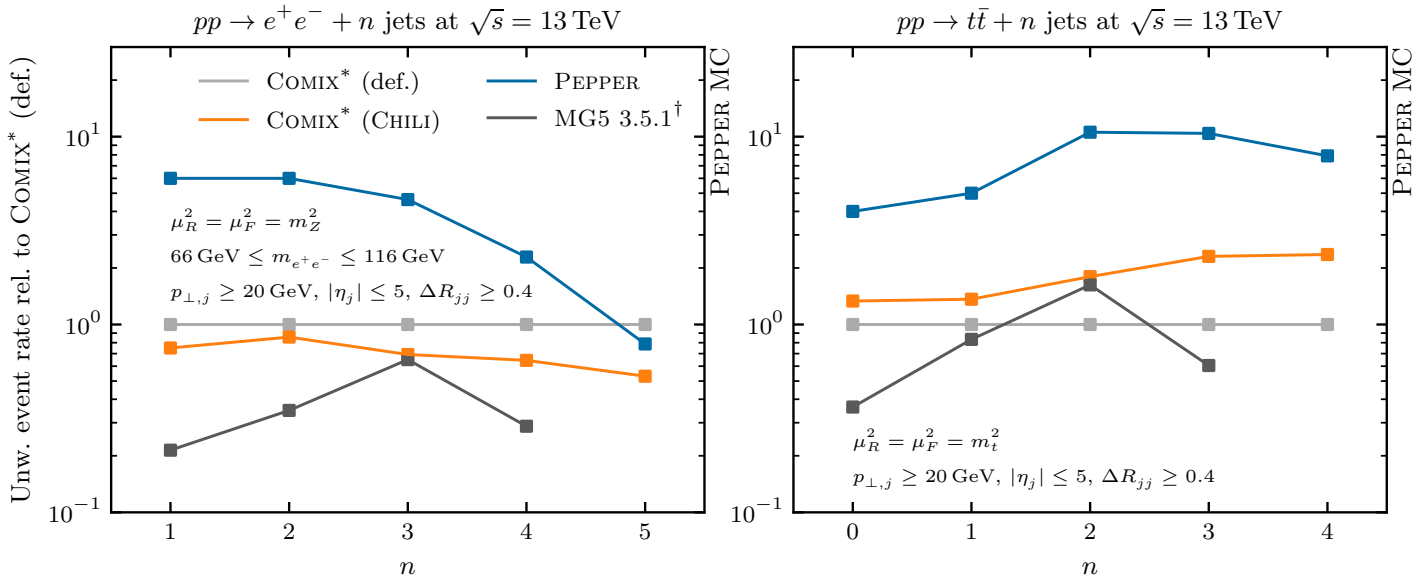
- CHILI phase-space generator uses simple (MCFM-inspired) structure: one t-channel + adjustable number of s channels [SciPost Phys 15 (2023) 4]
  - Portable (ported built-in CHILI in PEPPER)
  - Efficiency on par with recursive COMIX phase-space

Making LHAPDF GPU-ready

- Byproduct of porting exercise
  - PDF evaluations not critical
  - copying of data is
- Huge performance gains
  - 3-10x speedup per PDF evaluation
  - more possible by changing MC workflows
- OpenMPI used for efficient initialisation
  - constant init time vs. infinite init time [PRD 100 (2019) 1]
- Added CUDA + Kokkos interface/version
  - excellent computing performance / accuracy
  - portable version used for the remaining talk



PEPPER + CHILI: baseline CPU performance



- First complete CHILI + PEPPER benchmark
  - Unweighted event throughput compared to COMIX*
  - Constitutes baseline single-threaded performance of currently available competitive algorithms
  - Standalone PEPPER performs better than COMIX, but PEPPER's real goal is portability

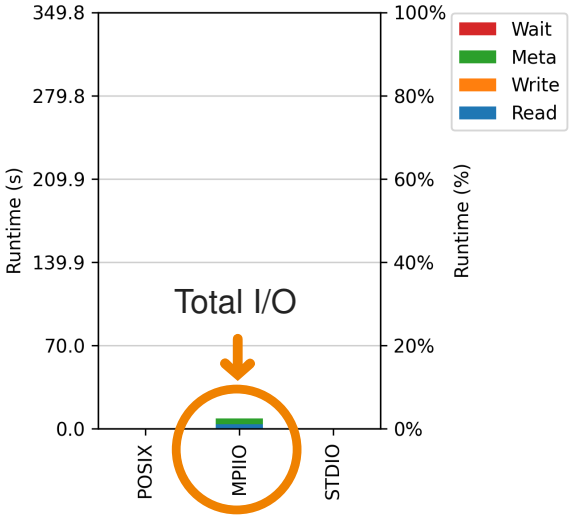
Numbers generated on Intel Xeon E5-2650 v2
* Partonic processes split into to *g*/*q* groups (not SHERPA default)
† Modified to match efficiency convention of [PRD 101 (2020) 7]

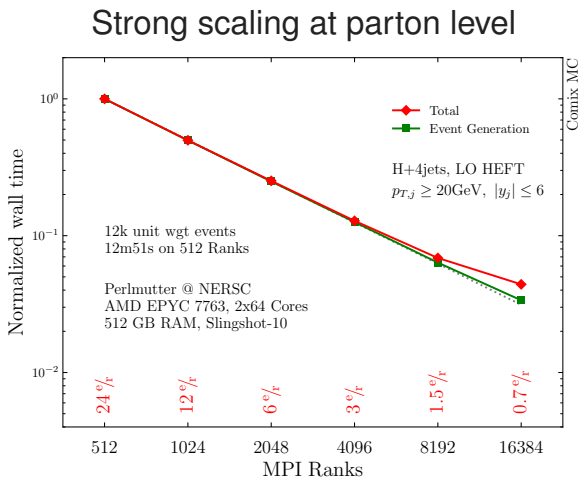
Benchmarking with state-of-the-art event generators

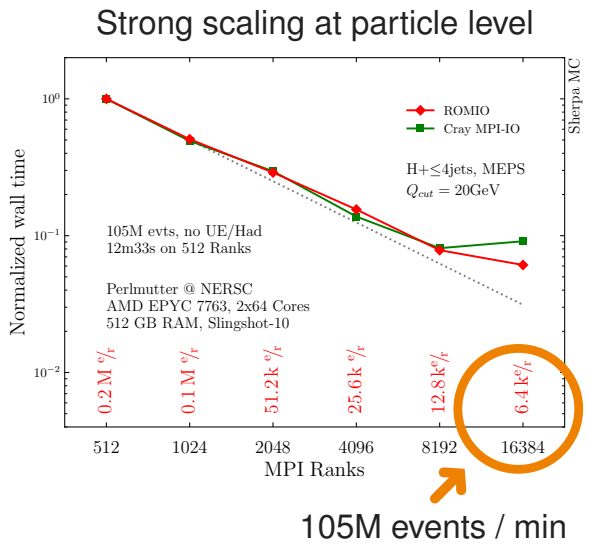
- comparison of SHERPA's COMIX with PEPPER+CHILI, on a single core Intel(R) Core(TM) i3-8300 CPU at 3.70GHz with 8MB L3 cache.
- samples generated with a given target for the total cross section uncertainty ("Tot. unc.")
- "Speed-up" gives the walltime gain factor of PEPPER+CHILI vs. COMIX
- PEPPER+CHILI: $Z + 0, 1j$ generated using helicity summing, while the higher ones use helicity sampling, thereby achieving the best possible performance in each case
- factorial scaling in PEPPER causes COMIX to win at very high multiplicities

Process	Tot. unc. [%]	SHERPA (COMIX)			PEPPER+CHILI			Speed-up
		Walltime [s]	Mem. (USS) [MB]	Eff. [%]	Walltime [s]	Mem. (USS) [MB]	Eff. [%]	
Z+0j	0.089	68	62	22	10	40	43	6.8
Z+1j	0.19	76	66	5.3	31	33	10	2.5
Z+2j	0.99	92	64	0.28	10	35	1.4	9.2
Z+3j	3.8	95	65	0.037	36	43	0.097	2.6
Z+4j	14	122	115	0.0050	71	133	0.016	1.7

New LHE-H5 format





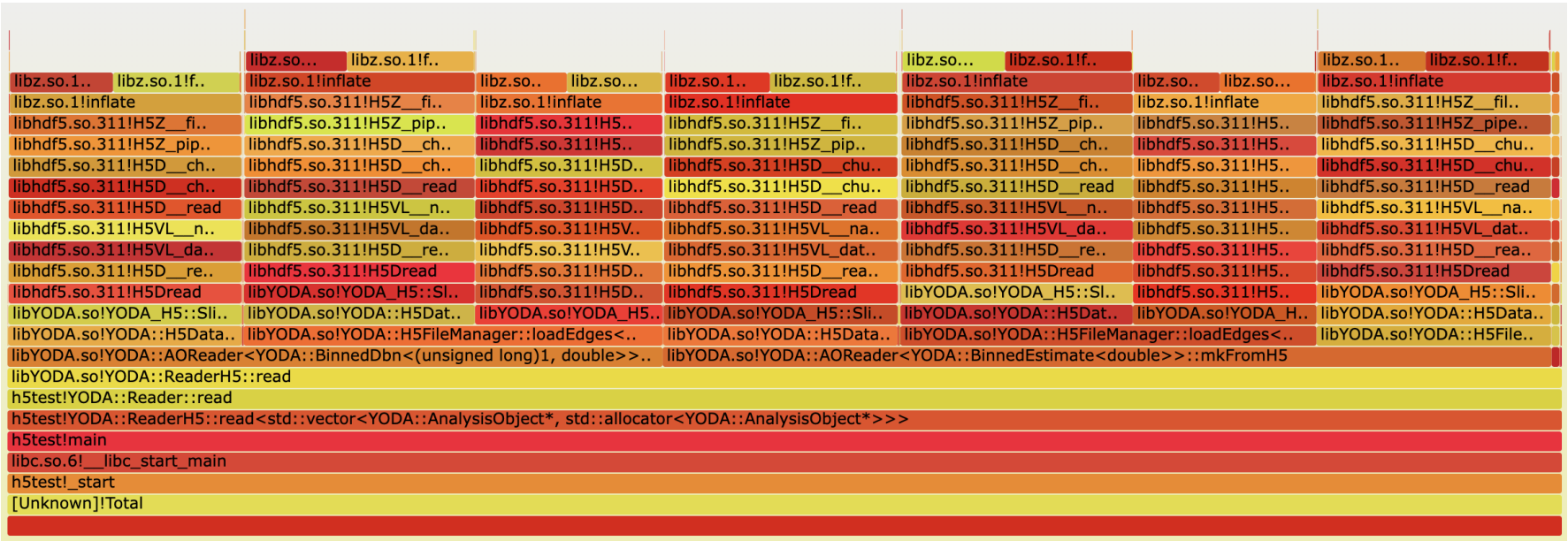


- new efficient LHE-like data format based on HDF5+HighFive proposed in [PRD 109 (2024) 1]
- overall I/O time reduced to below 1s per rank
  - time spent in I/O operations less than 5% when reading 128.85 GiB
- ideal for accessing back-fill queues at large computing centres
- already support by Sherpa and Pythia

New YODA-H5 format

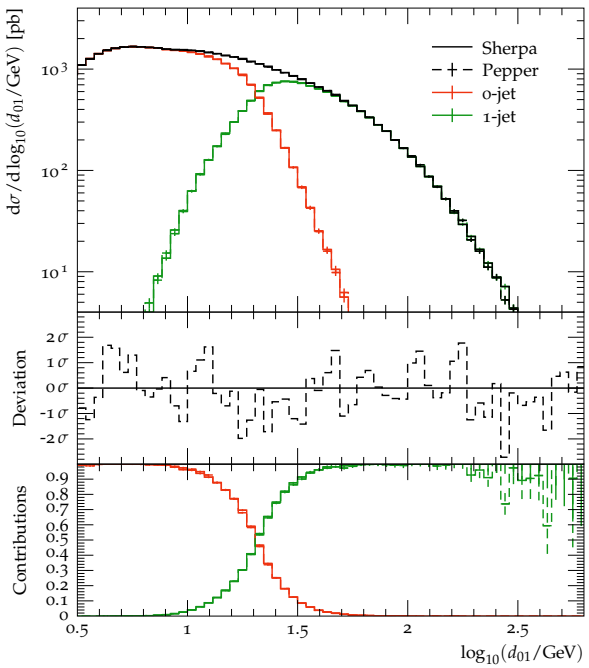
- provides an alternative for applications where ASCII is currently a bottleneck
- integer-factor speed-ups for I/O operations; nicely balanced profiles

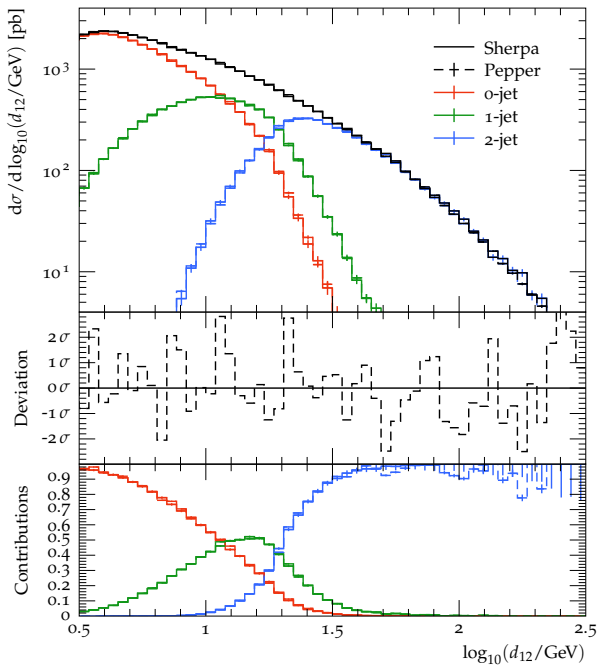
Raw ASCII size	ASCII/H5 exec times excl. compression	ASCII/H5 exec times incl. compression	gz vs raw h5
$\mathcal{O}$(5kB)	0.36	0.35	0.51
$\mathcal{O}$(3MB)	2.69	1.03	6.34
$\mathcal{O}$(33MB)	2.15	0.41	4.80

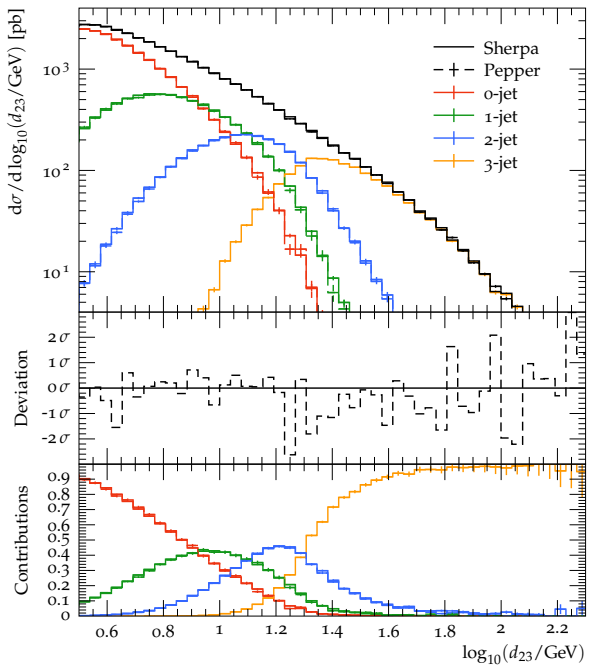


Tool-chain integration

- validated for standard candle processes (Z+jets shown) at various multiplicities
- can mix and match generators to reduced computing time to the absolute minimum required for event simulation



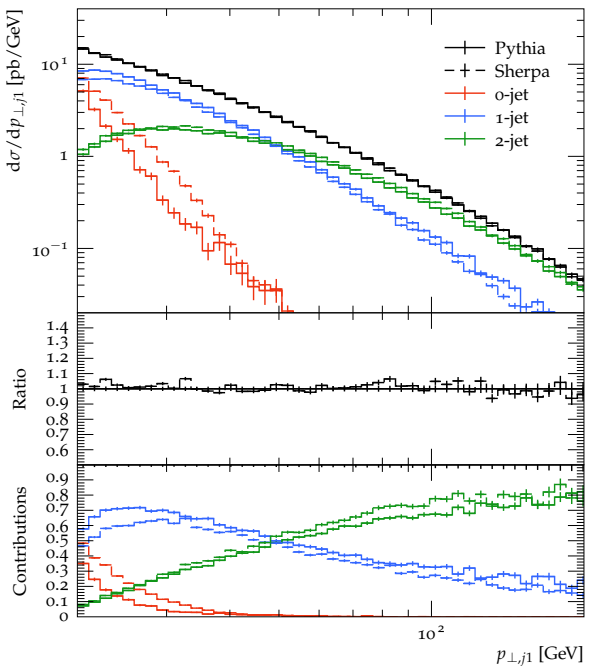


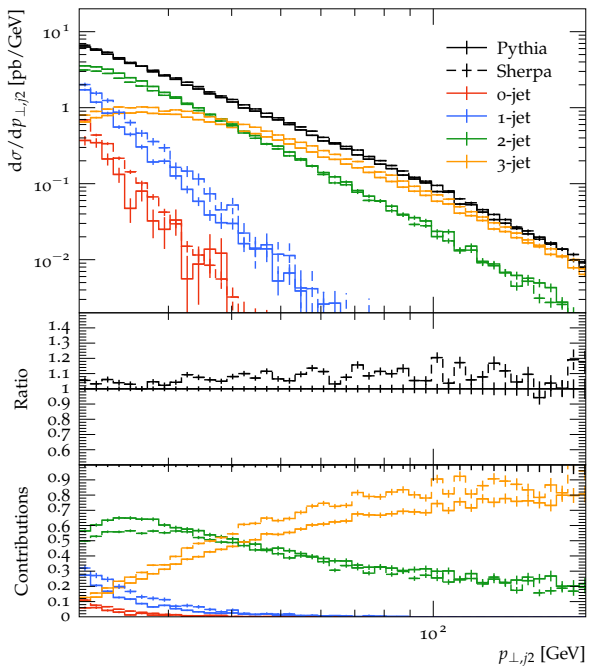


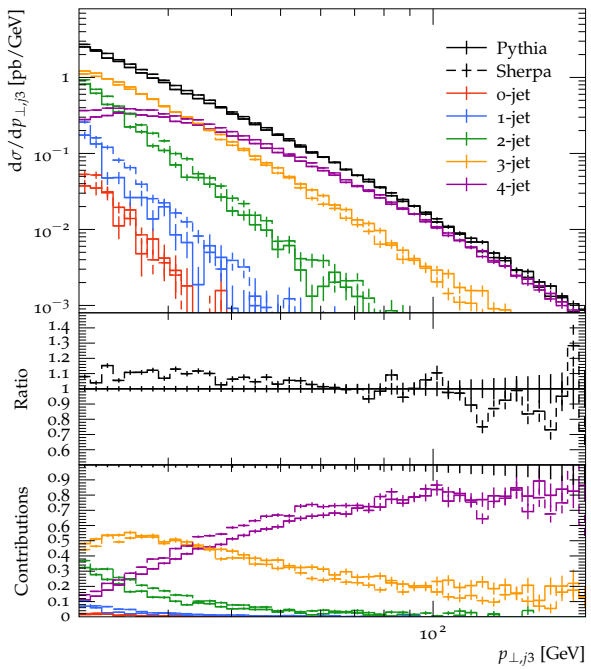
[PRD 109 (2024) 1]

More robust uncertainty estimates

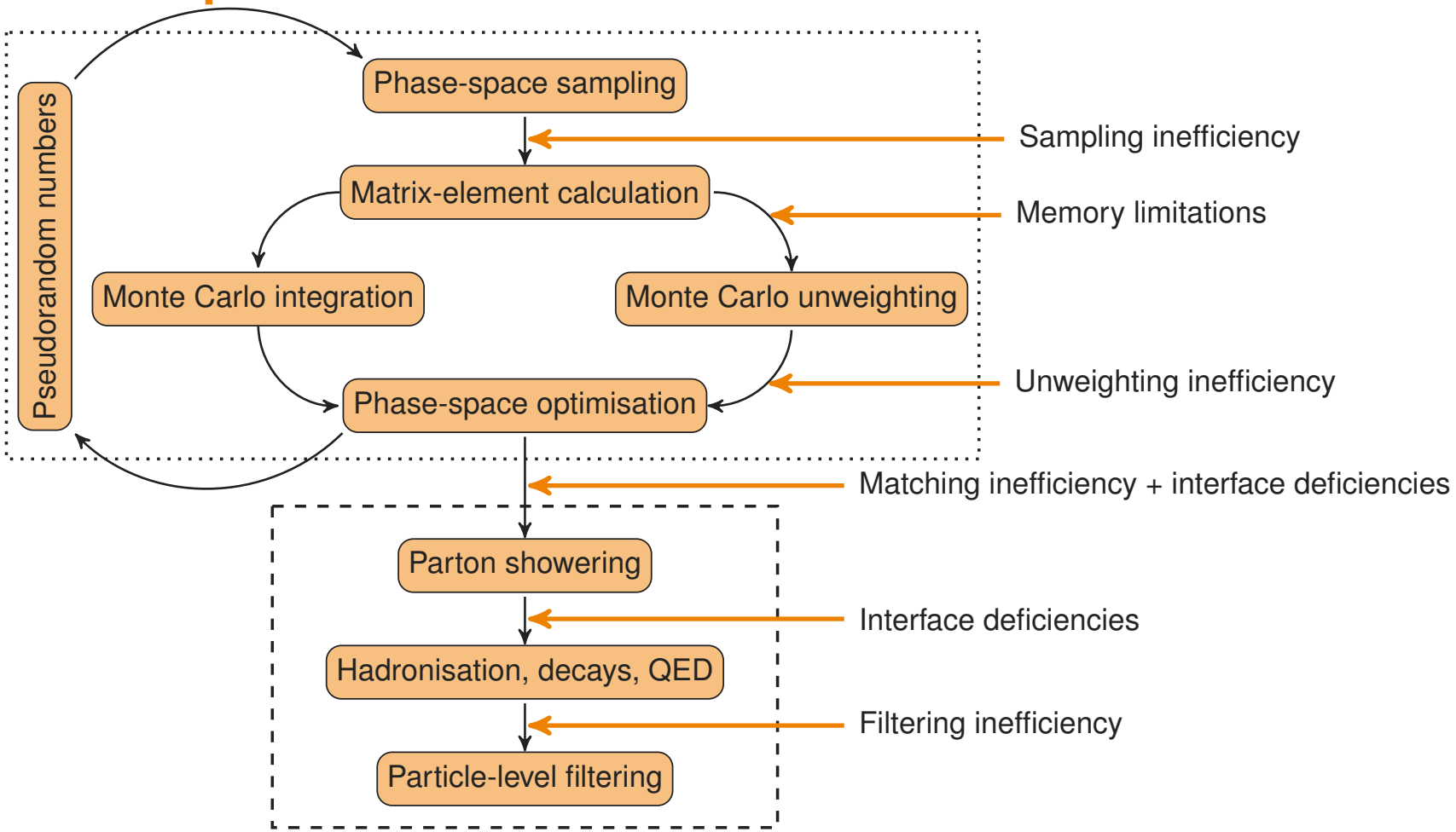
- LHEH5 enables efficient substitution of various parts in the event generation chain
  - already supported by both Sherpa and Pythia!
- 10 % uncertainty seen in Z+jets due to different algorithmic choices in the parton showers







Typical MC production chain



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