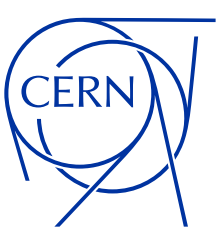
Event Generation at the HL-LHC with SHERPA and PEPPER

High Energy Phenomenology seminar series of Milan and Milan-Bicocca

Enrico Bothmann, CERN-IT, 2026-03-26



Introduction

- Very short reminder what event generation is about
- Part 1: Status and developments of physics with SHERPA 3
  - Perturbative physics: EW, polarised cross sections and NLL shower
  - ??? Beyond *pp*: Photon-induced processes, DIS/EIC and neutrino physics
  - Soft physics: Tuning and hadronisation uncertainties
- Part 2: Fast event generation in a heterogeneous computing environment
  - Portable hardware-accelerated parton-level calculations with PEPPER
  - ML and fast phase-space sampling with Normalizing Flows
  - Outlook on GPU accelerated loop calculations

Event generator ecosystem

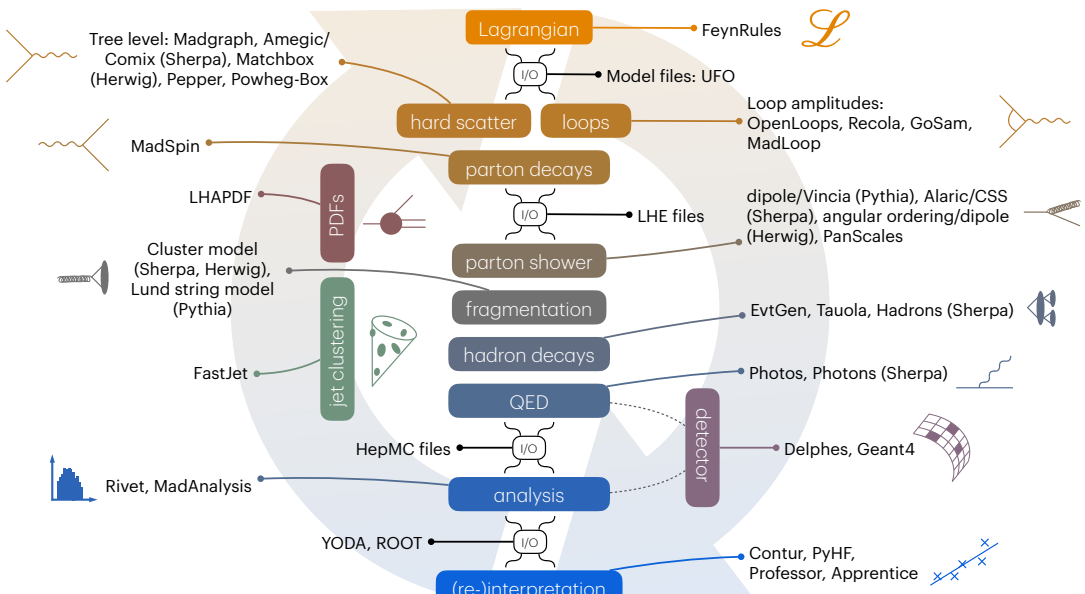


Figure by Ramon Winterhalder and Christian Gütschow

- Event generators: MC sampling to generate samples of fully differential simulated events for $\mathcal{L}_{\text{SM}}$ or most $\mathcal{L}_{\text{BSM}}$
- After adding detector simulation, direct statistical comparison with experimental samples possible
- Used for pheno studies, Collider/detector design, calibration, physics analyses, …

Monte-Carlo event building blocks

Fully automated state-of-the-art since Run-II:

- NLO QCD ME (+ approximate EW corrections)
- 2–3 NLO jet multiplicities merged (+ more at LO)
- matched with parton shower at (N)LL
- additional interactions between protons (MPI)
- hadronisation (cluster/string), hadron decays
- soft QED emissions (e.g. YFS resummed)

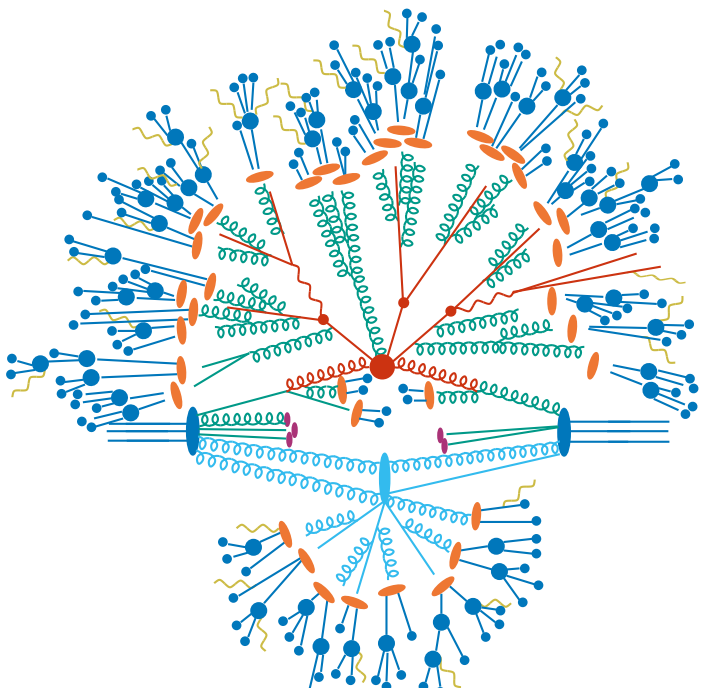


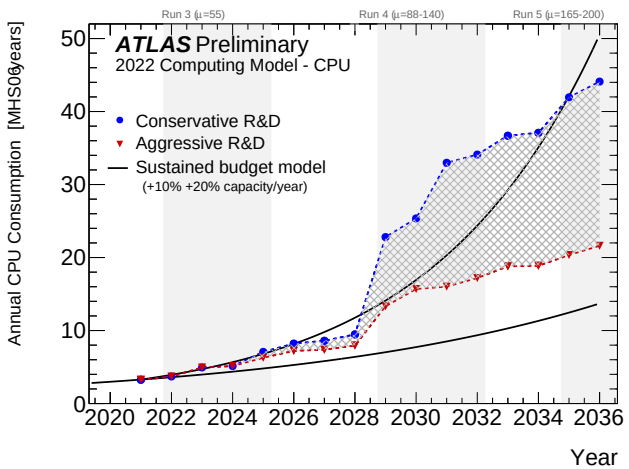
Figure by Frank Krauss, updated by Christian Gütschow

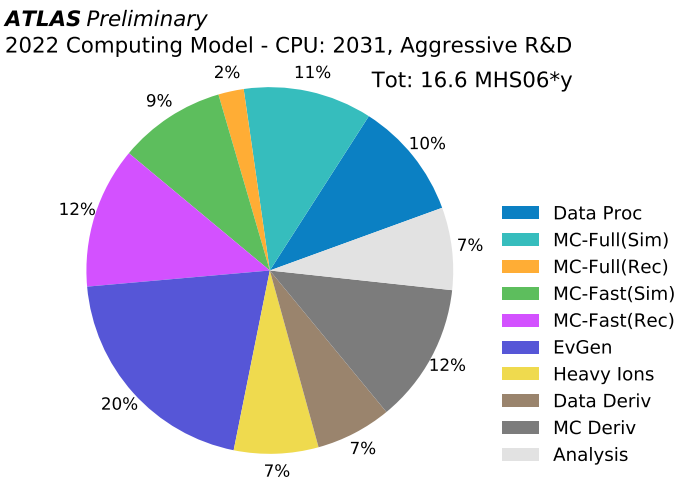
Precision wishlist: perturbative side

- Automated matching (and merging at lower order?) at NNLO in QCD
  - MINNLOPS, UN^{2}LOPS, GENEVA, … not fully generic and automated yet
- EW shower + automated matching and merging at NLO for QCD+EW … WIP
- Increased parton-shower accuracy beyond (N)LL
  - Toward full NLL: PANSCALES, ALARIC … almost there for pp collisions
  - Toward full (N)NLL (first: event shapes) [Dasgupta et al. 2406.02661]
  - Beyond leading colour (amplitude evolution) [Forshaw et al. 2505.13183]

Mostly mention this to get it out of the way … but will discuss ALARIC. Note some of the above represent huge increases in computational complexity.

The problem with performance at the HL-LHC





- Absolute numbers: O(10M–100M) CPU hours per year for “EvGen”
- Must address this so that detailed theory predictions exist to interpret data

Will discuss solutions for this (and future colliders) in Part 2.

Part 1: Physics with SHERPA 3

SHERPA overview

- A comprehensive modular event generation framework (400k LOC)
- Open-source development in Sherpa GitLab repo, also see the manual

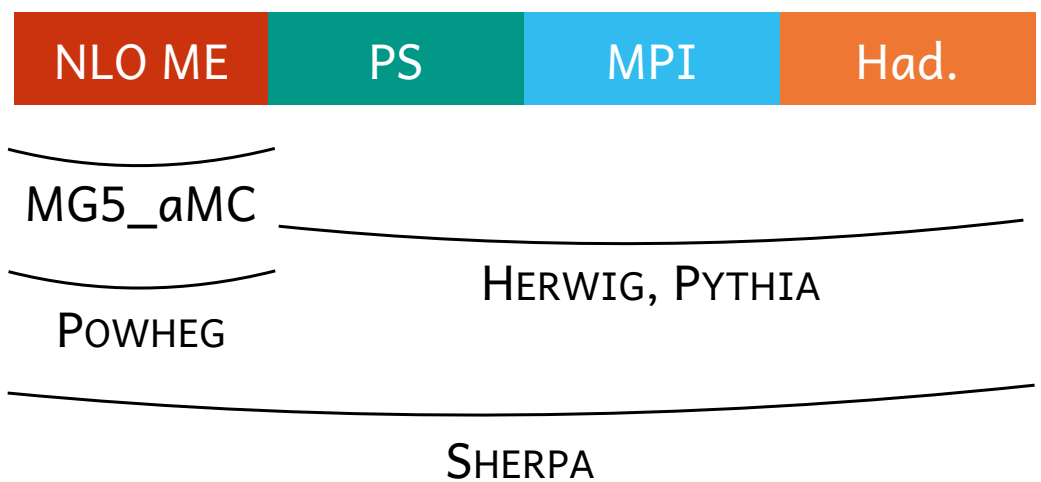
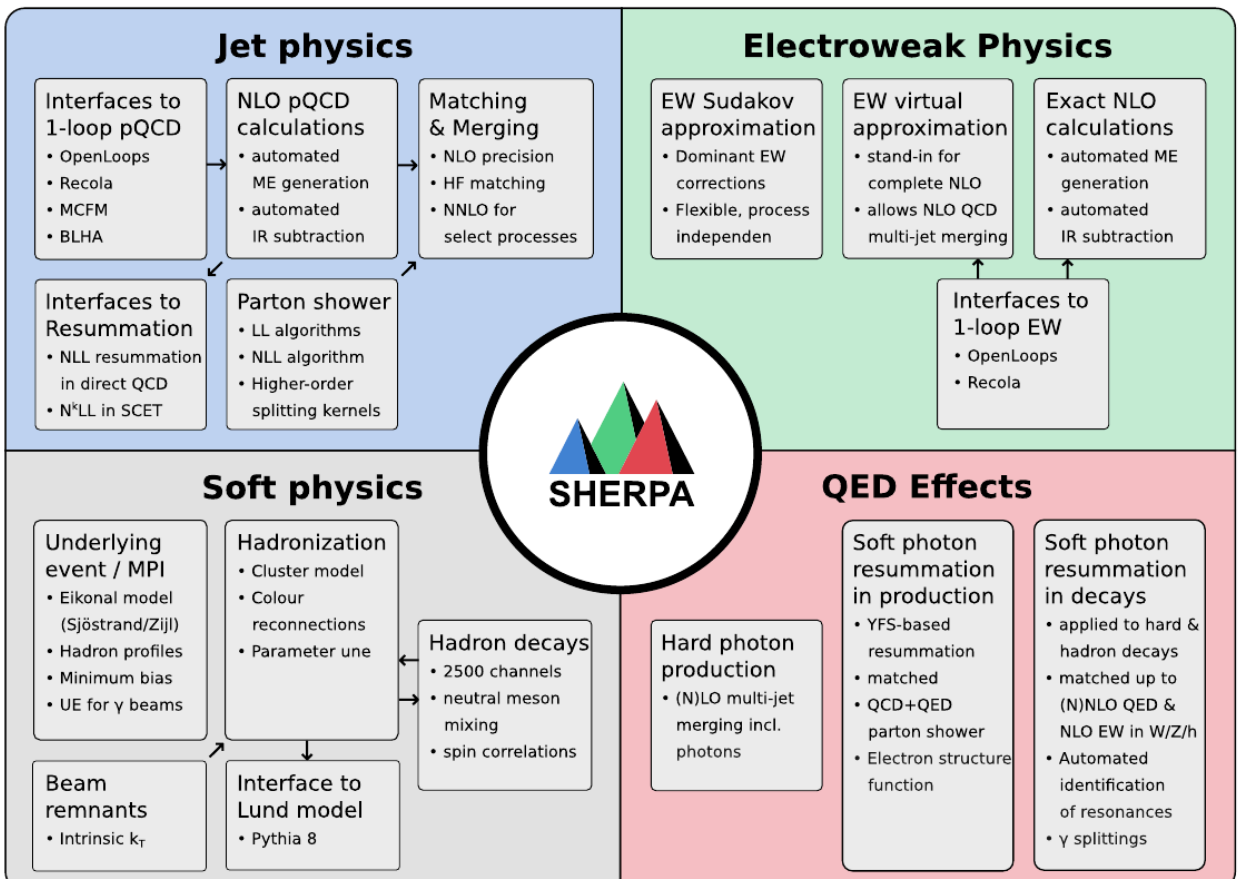
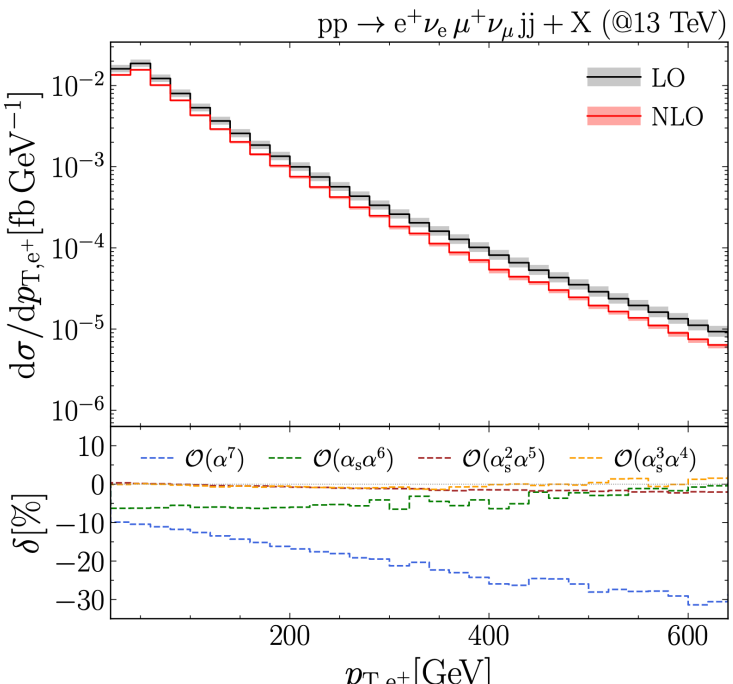


Diagram adapted from Marek Schönherr

SHERPA overview



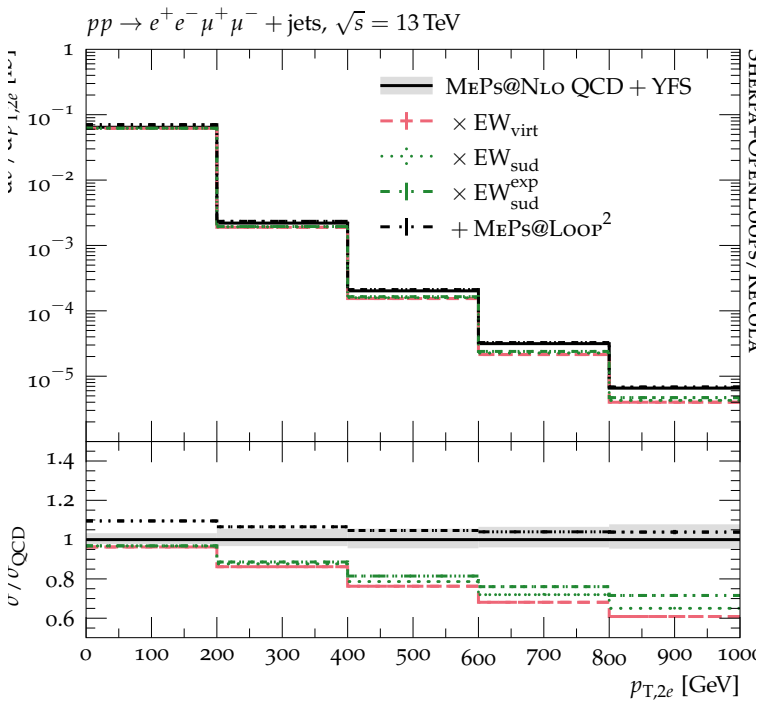
EW corrections in Sherpa 3 – fixed order



[Winterhalder et al. 2308.16716]

- NLO EW corrections increasingly important to reach precision targets of the LHC
- Can even exceed QCD corrections e.g. in VBS processes (plot)
- New in v3: NLO EW *fixed order* calculations [Schönherr 1712.07975]

EW corrections in Sherpa 3 – events



[EB et al. 2111.13453]

Approx. EW corrections for full particle-level multijet-merged SHERPA event generation

- For some years: EWvirt approximation (NLL+finite terms from virtual)
  [Kallweit et al. 1511.08692]
- New in v3: EWsudakov approximation (NLL terms only, no virtual needed)
  [EB, Napoletano 2006.14635] [EB et al. 2111.13453]
- Both neglect real-emission corrections
- Approx. taylored for high-energy limit (“EW Sudakov suppression”, e.g. large p_T)

Polarised intermediate particles

Probing the EW gauge sector & symmetry breaking (→ longitudinal polarisation of massive vector bosons)

Idea: use polarised MC predictions as templates for analyses with realistic final states

$$\mathrm{d}\frac{\sigma}{\mathrm{d}X} = f_L\,\mathrm{d}\frac{\sigma_L}{\mathrm{d}X} + f_R\,\mathrm{d}\frac{\sigma_R}{\mathrm{d}X} + f_0\,\mathrm{d}\frac{\sigma_0}{\mathrm{d}X} + f_{\text{int.}}\,\mathrm{d}\frac{\sigma_{\text{int.}}}{\mathrm{d}X}$$

- Explot longitudinal polarisation for EWSB studies
- Exploit SM-suppressed polarisation configurations for BSM studies

Lessons from previous analyses:

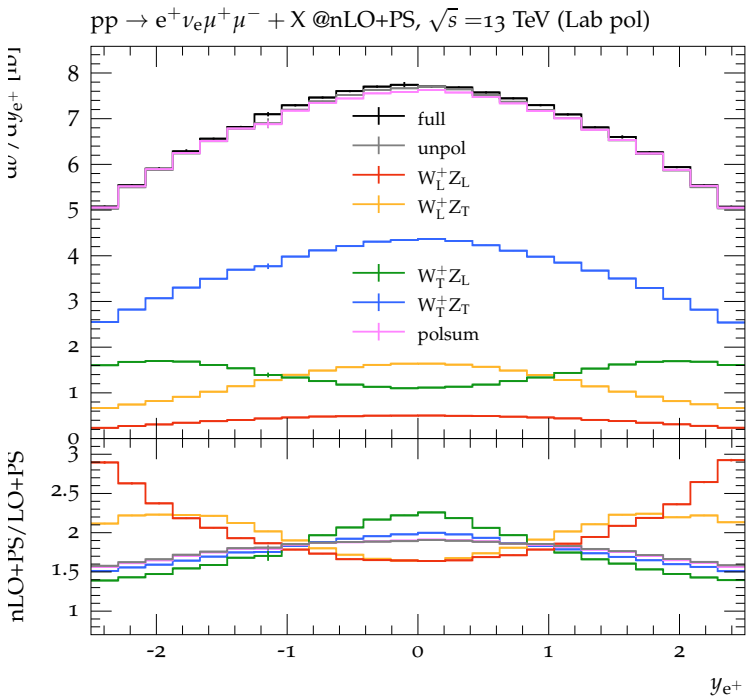
- Analytical projections not applicable in the presence of lepton cuts
- Exclusive predictions in MC event generators possible, ideally with HO corrections

Polarised intermediate particles

Caveats for polarised predictions:

- Polarisation basis is frame-dependent
- Interferences between different polarisations
- Polarisation only defined in production ⊗ propagator ⊗ decay factorisable amplitudes
  - Problem: non-resonant diagrams → no polarisation definition, but necessary for gauge invariance
  - Solution: appropriate approximations – gauge invariant options:
    - Pole approximation ((D)PA)
    - Narrow-width approximation (NWA)

Polarised intermediate particles – Sherpa 3 scheme



[Hoppe et al. 2310.14803]

Method [Hoppe et al. 2310.14803]

- Single, unpolarised simulation run, polarised cross sections as event weights, uses generic multi-weight implementation in SHERPA [EB et al. 1606.08753]
- Accuracy: nLO QCD+PS matching
  Virtual and ultra-soft and/or -collinear emission corrections not taken into account for polarisation cross sections

NLL-accurate parton shower development: ALARIC

- matching and merging at lepton colliders: 2507.22837
- at hadron colliders: 2404.14360

Photon-induced processes

Optional

DIS and EIC physics

Optional

- 2506.08994
- 2407.02456 (HERA)

Neutrino physics

Optional

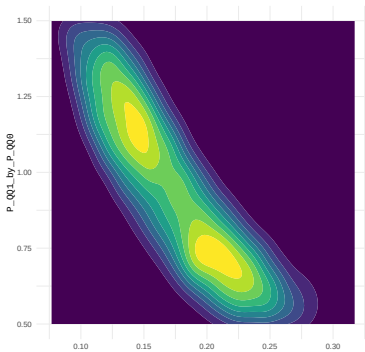
Soft physics: tuning and hadronisation uncertainties

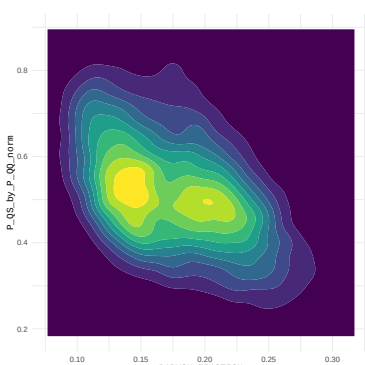
- Soft physics (confinement phase) not yet calculable from first principle
- Use instead phenomenological models (e.g. hadronisation: Cluster/Lund)
- These have model parameters → need to fit to data (“tuning”) (~20 params)
- So far no practical prescription for uncertainty of such models
- In their need, experiments sometimes take |Cluster – Lund| as the hadronisation “uncertainty” 😒

Soft physics: tuning and hadronisation uncertainties

Traditional tuning approach

- Initial scan $\mathcal{O}(1k)$ runs, then use interpolation polynomials and in parameters and do χ^2 minimisation, iterate narrowing ranges
- Result: best fit parameters with 1-σ errors, *assuming* Gaussians with single minimum
- Best fit parameters become generator defaults
- errors not used in practice:
  - can't afford any explicit reruns
  - while better than |Cluster - Lund|, still no straightforward interpretation





Soft physics: tuning and hadronisation uncertainties

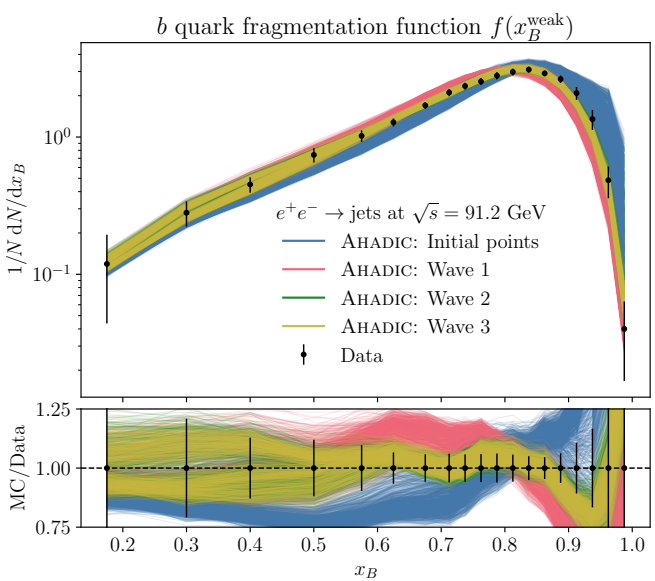
Idea: use History Matching (HM) for Tuning and robust uncertainty estimates
[Iskauskas, Knobbe, Krauss, Schumann 2602.22324]

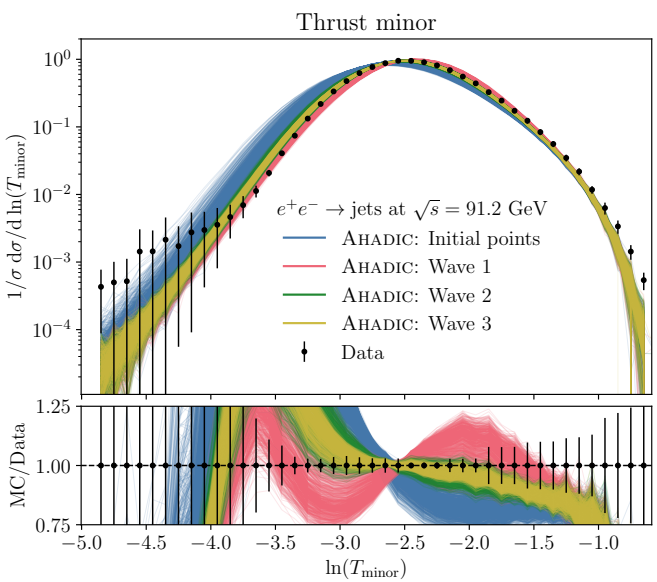
- HM identifies *non-implausible regions* of parameter space
- Generic solution for calibration of parameters of complex computer models, used in the oil industry, climate science, galaxy formation, atomic physics, epidemiology …
- implausibility measure similar to χ^2
- HM waves give rise to consecutive compression of valid volume

Soft physics: tuning and hadronisation uncertainties

[Iskauskas, Knobbe, Krauss, Schumann 2602.22324]

- Global calibration of 20 parameters using LEP1 data
- 3 HM waves, each $\mathcal{O}(1000)$ parameter points

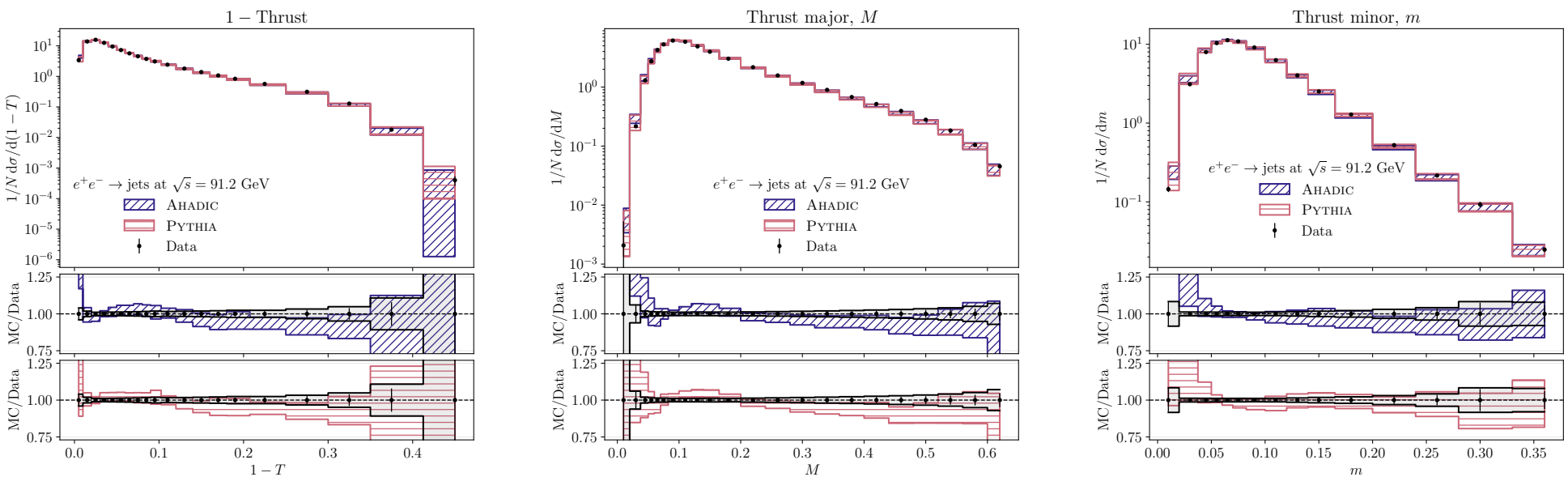




Soft physics: tuning and hadronisation uncertainties

[Iskauskas, Knobbe, Krauss, Schumann 2602.22324]

- Estimate model uncertainty by final wave (800 members)

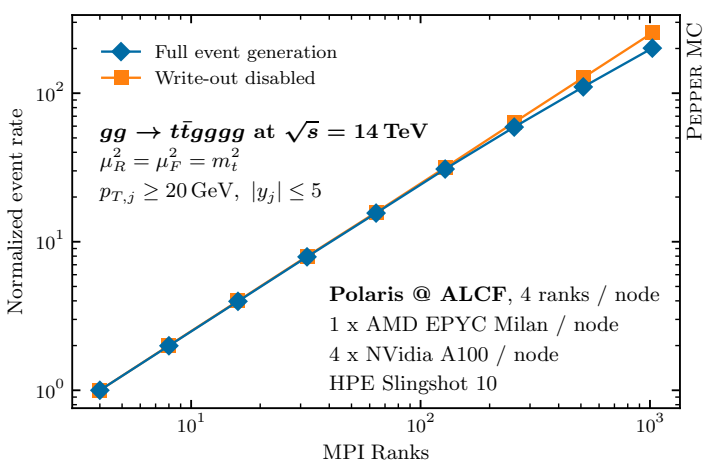


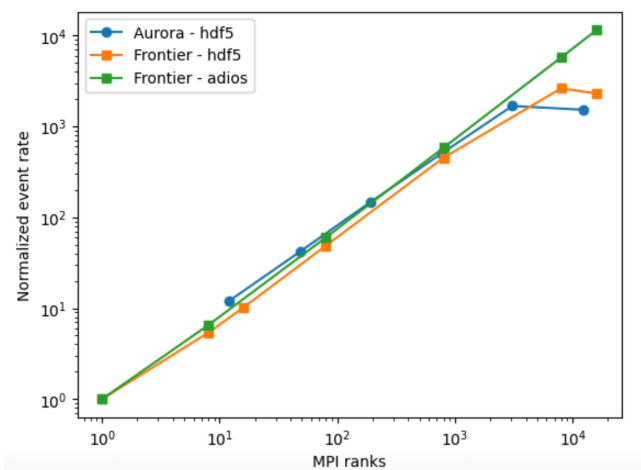
- Still can't rerun in practice → next step: combine with reweighting

Part 2: Accelerated event generation

Invitation: Portable event generation at 0.5 Exascale

- Left: Scaling to 1k Nvidia A100 nodes on Polaris [EB et al. 2311.06198]
- Right: Scaling to 10k AMD MI250x nodes on Frontier → 0.5 ExaFLOPs, 20% of Frontier, 50M unweighted Z+5j events/min *Max Knobbe (FNAL), Ana Gainaru (ORNL)*





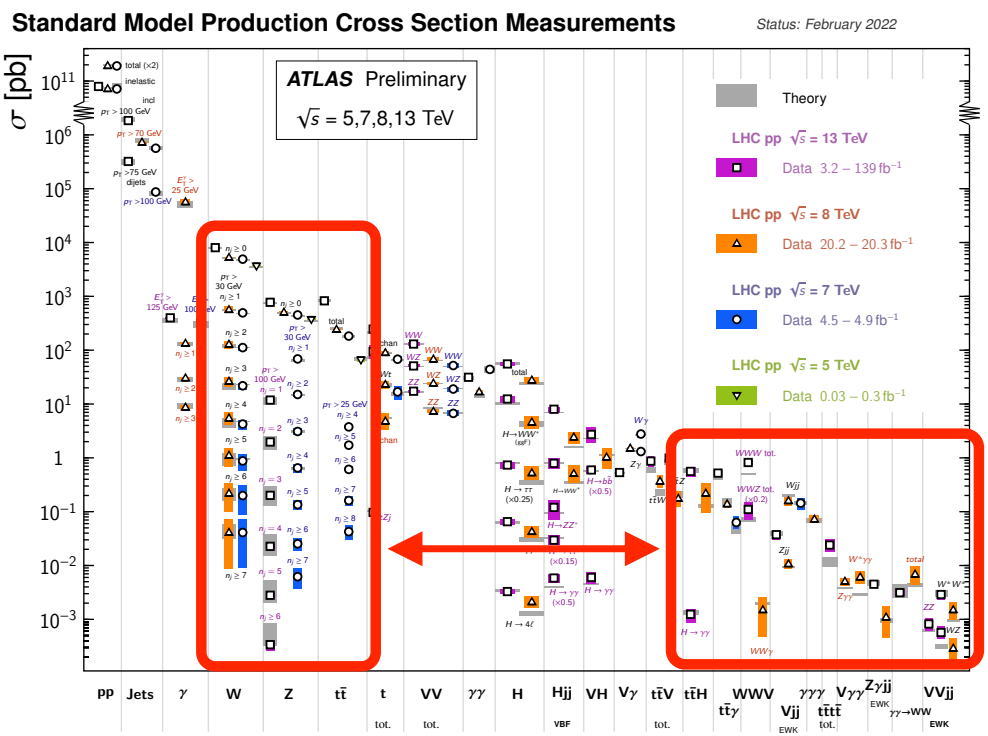
- Single node: 1.7M unweighted $t\bar{t} + 4j$ events on 1 × H100 Nvidia GPU vs. 0.06M on 2 Intel Xeon Platinum 8180M Server CPUs (56 cores)

Why improve event generation efficiency?

- Keep up with high statistics at HL-LHC & excellent detector performance
  - ATLAS/CMS Projections: computing requirements > budget during HL-LHC
  - ~20 % event generation → poor efficiency can limit experimental success
- Improve physics description
  - Extend physics reach (1–2 additional jets to keep up with stats & energies)
  - Better description of high-multiplicity inter-jet correlations to differentiate New Physics from BSM backgrounds
- Keep up with hardware/HPC trends

What to improve? What dominates computing budgets?

Which processes are relevant computationally?



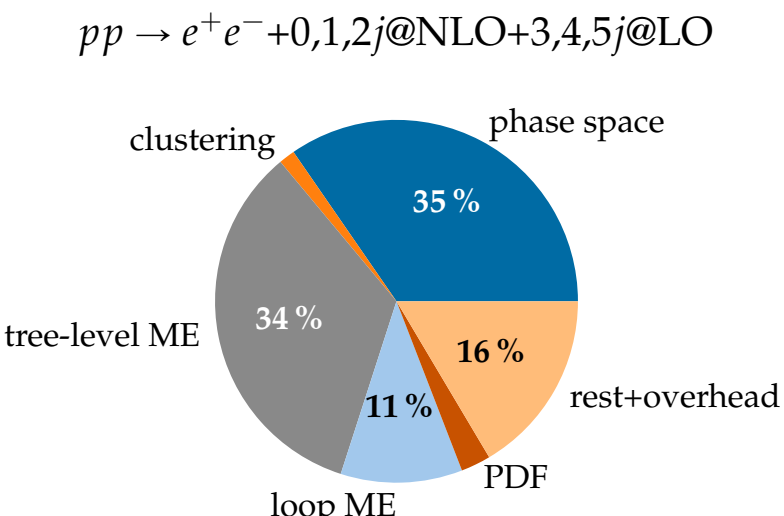
https://twiki.cern.ch/twiki/bin/view/AtlasPublic/StandardModelPublicResults

- Signals (on the right): High multiplicity, but low complexity/stats
- Main backgrounds (on the left): High multiplicity *and* high complexity/stats

Which multiplicities at what perturbative order?

Heavy hitter background samples (here: ATLAS SHERPA setups)

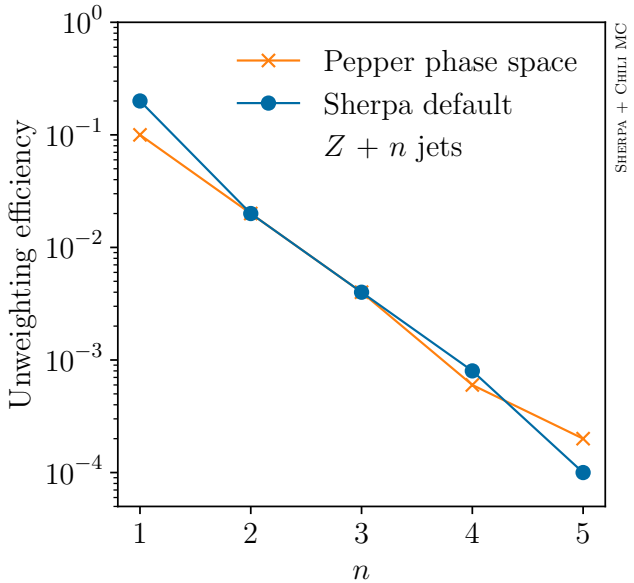
- $pp \to e^+e^- + 0,1,2j$ @ NLO + $3,4,5j$ @ LO
- $pp \to t\bar{t} + 0,1j$ @ NLO + $2,3,4j$ @ LO
- Unweighted events needed for downstream processing
- 60–80% time spent in tree-level ME and their phase space, after optimising and using analytic loop MEs [EB et al. 2209.00843]
- Reason: low unweighting efficiencies & expensive ME for high jet multis n
- Highest 1–2 jet multiplicities contribute most



Which multiplicities at what perturbative order?

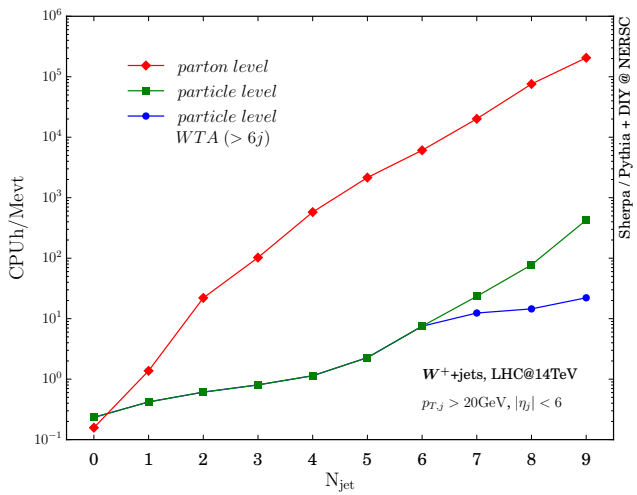
Heavy hitter background samples (here: ATLAS SHERPA setups)

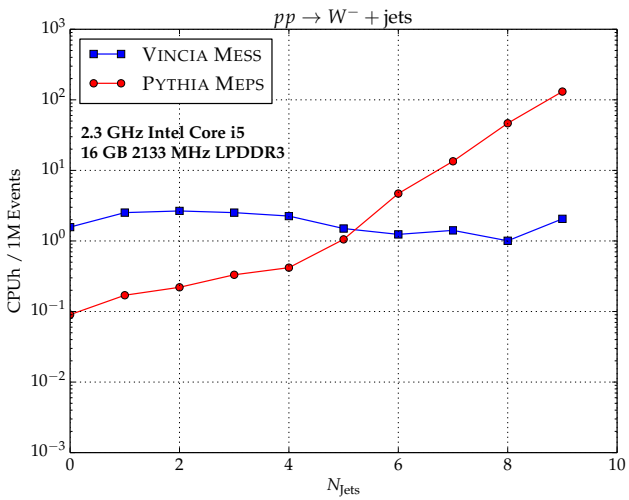
- $pp \to e^+e^- + 0, 1, 2j$ @ NLO + $3, 4, 5j$ @ LO
- $pp \to t\bar{t} + 0, 1j$ @ NLO + $2, 3, 4j$ @ LO
- Unweighted events needed for downstream processing
- 60–80% time spent in tree-level ME and their phase space, after optimising and using analytic loop MEs [EB et al. 2209.00843]
- Reason: low unweighting efficiencies & expensive ME for high jet multis n
- Highest 1–2 jet multiplicities contribute most



[EB et al. 2302.10449]

Parton or particle level?





- Hard scattering simulation much more demanding than particle-level remainder [Höche et al. 1905.05120]
- Complexity of multi-jet merging can be reduced to achieve linear scaling using sector showers [Brooks, Preuss 2008.09468]
  - Reconstructing shower histories not a problem in principle

PEPPER project aims and preconditions

Identified bottleneck, so define figure of merit for PEPPER project:

- Unweighted parton-level event throughput
  for highest relevant jet multiplicity of heavy-hitter processes
  - e.g. $pp \to e^+e^- + 5j$, $pp \to t\bar{t} + 4j$ ($+1j$ for HL-LHC era?)

Additional goals of PEPPER:

- Be portable to achieve efficiency on today's heterogenous hardware
- Try new algorithms and see how they work with modern hardware
- Sustainable software development with knowledge transfer in mind
- Create a simple tool that can be used with Sherpa (+ Pythia),
  but also standalone, e.g. for ML studies

PEPPER amplitudes

- Berends–Giele recursion for best multi-jet scaling behaviour
  Based on early 2000s performance studies: [Dinsdale et al. hep-ph/0602204], [Duhr et al. hep-ph/0607057]
- Colour summing to allow for lockstep evaluation on GPU-like hardware
- Combine with minimal colour basis for general QCD amplitudes

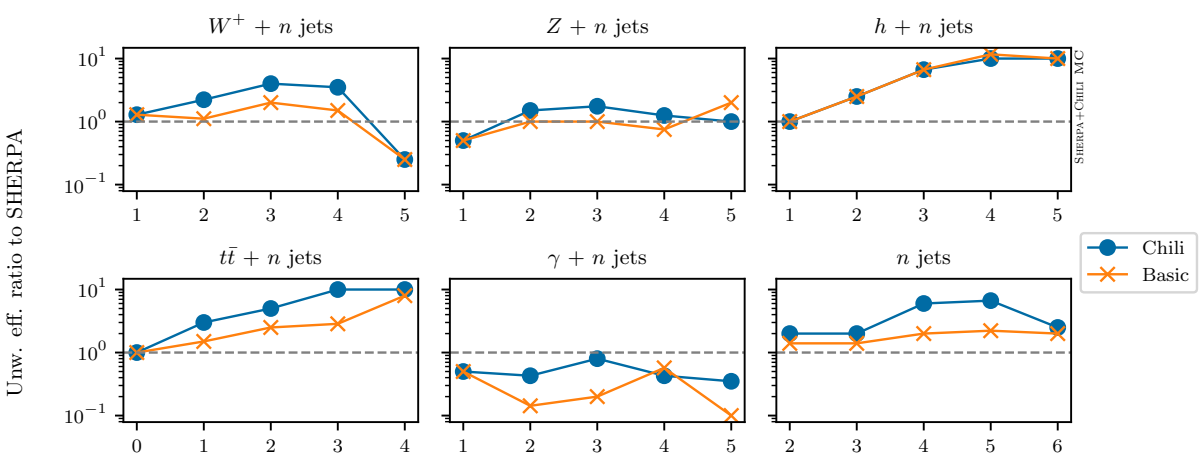
$$\mathcal{O}\left((n-1)!^2\right) \to \mathcal{O}\left((n-2)!^2\right) \to \mathcal{O}\left(\frac{(n-2)!^2}{k!}\right)$$

[Melia 1304.7809] [Melia 1312.0599] [Melia 1509.03297] [Johansson, Ochirov 1507.00332]

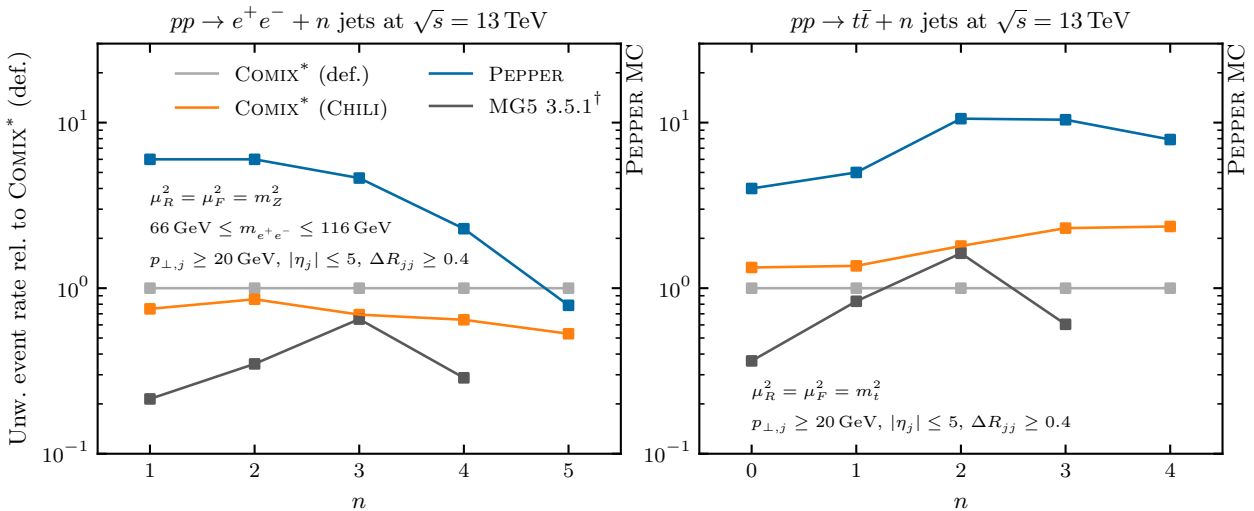
  - Combined for first time with Berends–Giele recursion
  - Generalised in our implementation for e^+e^- + jets
- Helicity sampling to avoid additional 2^n scaling

PEPPER phase space: CHILI

- CHILI phase-space sampling inspired by MCFM-like structure: one t-channel + adjustable number of s-channels [EB et al. 2302.10449]
- Simple and thus easily ported for parallel architectures
- RAMBO-like speed, efficiency for heavy hitter processes on par with complex recursive COMIX phase space in SHERPA:



Baseline performance comparison (single-threaded)



- Unweighted event throughput compared to COMIX* generator in SHERPA [EB et al. 2311.06198]
- Satisfying performance, but the real goal is portability

Numbers generated on Intel Xeon E5-2650 v2

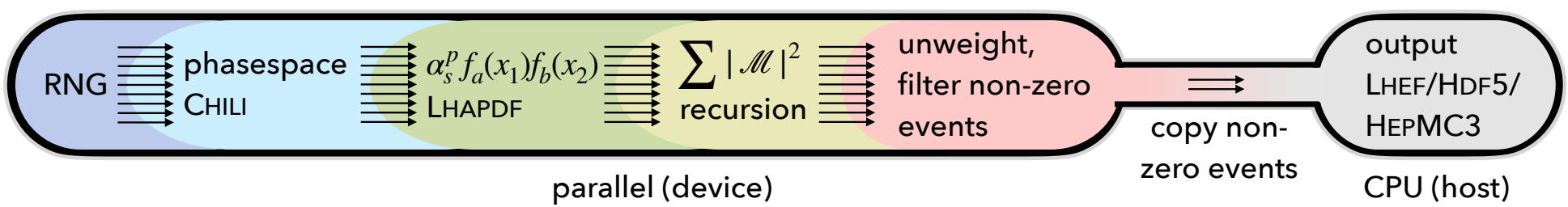
- COMIX*: Partonic processes split into g/q groups (not SHERPA standard)
- MG5†: Modified to match unweighting efficiency convention

Why portability?

- Many computing vendors (Nvidia, Intel, AMD), heterogeneous architectures
- (Pre-)Exascale computing systems intentionally diverse
- Portable generator projects besides PEPPER: CUDACPP for MADGRAPH5 [Hageböck et al. 2507.21039], MadFlow [Carrazza et al. 2106.10279]

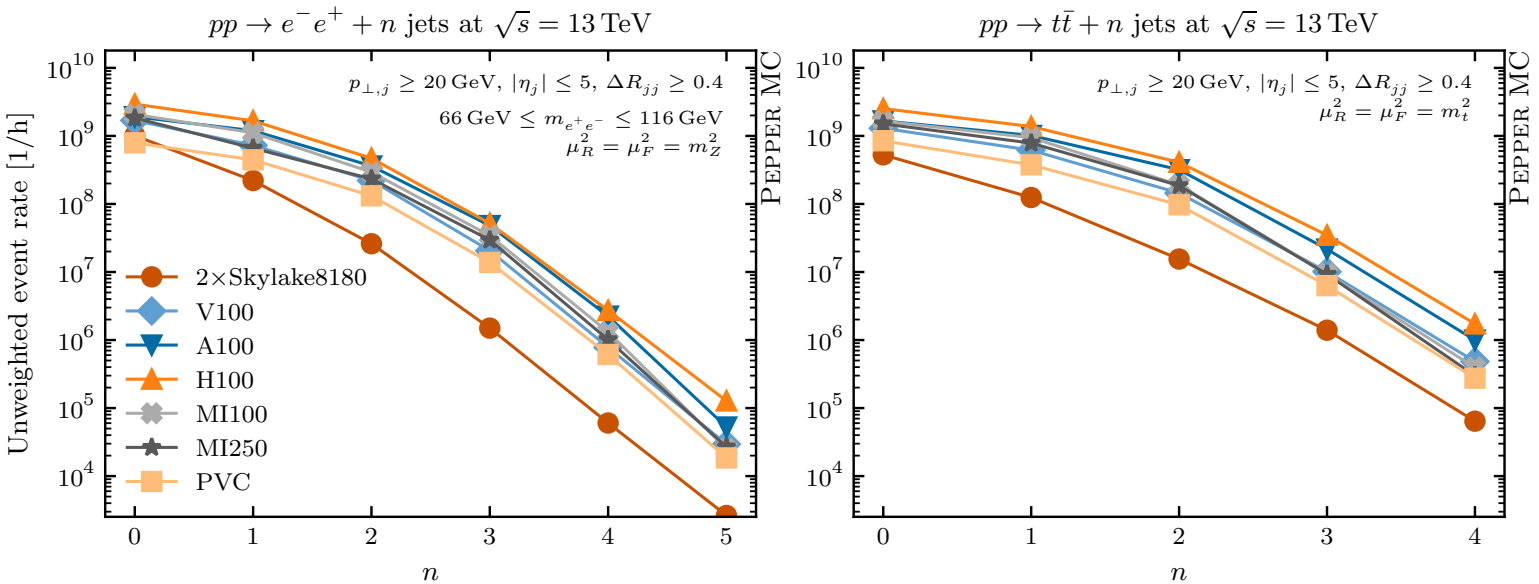
Portability

- To study algorithms & allow for quick progress, writting from scratch with parallel execution in mind (SoA, device/host memory, coalescent reads/writes, lock-step processing)
- Except for I/O, the entire simulation chain generates events in parallel:



- KOKKOS parallelisation abstractions to support OpenMP, Nvidia, AMD, Intel GPU, CPU vector instructions
- MPI & HDF5 to minimise unwanted file I/O on HPC clusters

Results



MEvents / hour	2×Skylake8180	V100	A100	H100	MI100	½×MI250	PVC
$pp \to t\bar{t} + 4j$	0.06	0.5	1.0	1.7	0.4	0.3	0.3
$pp \to e^- e^+ + 5j$	0.003	0.03	0.05	0.1	0.03	0.03	0.02

Toolchain integration

- Pepper writes out HDF5-based LHEH5 parton-level event samples
- Particle-level simulation via SHERPA or PYTHIA
  - Validated and benchmarked in [EB et al. 2309.13154]
- With SHERPA, also multi-jet merged event generation supported:

Machine Learning: platform

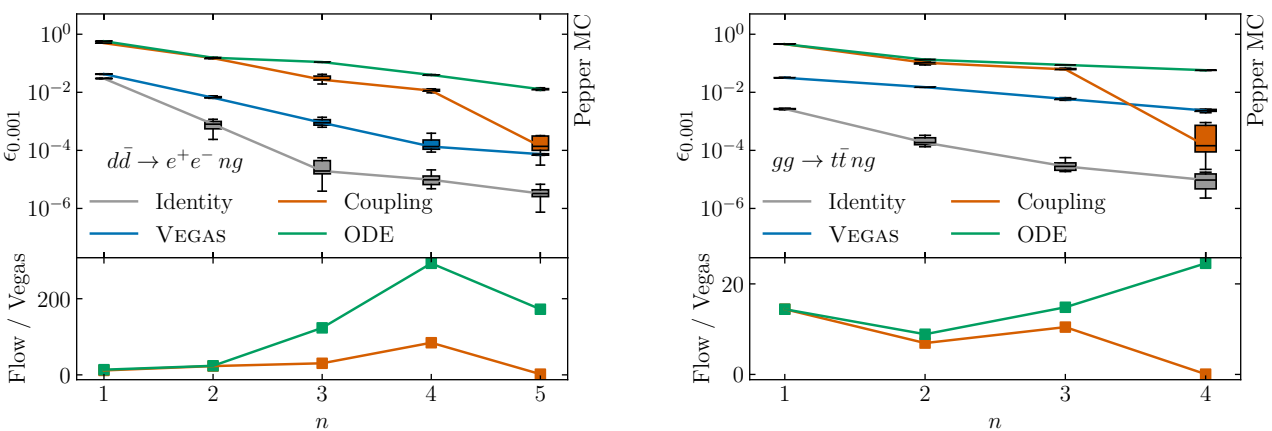
- GPU accelerated event generation enables very fast on-device training and exploration of new ML methods, e.g. surrogates, samplers, …
- "Pyper" API by CERN summer student Carla Lopez avoids CPU ↔ GPU copies
  - PyTorch example available to pass an array of random numbers to PEPPER and get back an array of $|\mathcal{M}|^2$ values, never leaving the GPU

- Also older file-based interfaces available to read/write random numbers, momenta, $|\mathcal{M}|^2$ values, phase-space weights, …

Machine Learning: training an efficient sampler

[Bothmann, Knobbe, Janßen 2506.18987]

- Phase-space sampling with helicity-conditioned Continuous Normalizing Flow trained with Flow Matching ("ODE Flows")



- efficiency improvements over VEGAS up to 198 ($e^+e^- + 4g$) / 25 ($t\bar{t} + 4g$)
- ODE flow scales better to higher multiplicities/dimensionalities

Machine Learning: training an efficient sampler

[Bothmann, Knobbe, Janßen 2506.18987]

- Fantastic efficiencies, but ODE Flows too slow to evaluate: for each sampled point, integrate learned vector field to transform from latent to target space
- Idea: Use RegFlow to combine strengths of efficient ODE flows and fast Coupling Normalizing Flows by training the latter with the former
- With RegFlow-trained Coupling Flows, we get walltime speed-ups of the unweighted parton-level event generation of 12 ($e^+e^- + 4g$) / 8 ($t\bar{t} + 4g$)
- Next step: add conditioning on flavour groups and study physical processes

TODO

- Other SHERPA-related ML work: …
- Mention Carrazza's work etc.
- Mention Ramon' work etc.
- Plug MCnet School 2026
- Don't forget energy plots