PEPPER

A Portable Parton-Level Event Generator for the HL-LHC

Enrico Bothmann, Taylor Childers, Walter Giele,
Stefan Höche, Joshua Isaacson, Max Knobbe

Institute for Theoretical Physics
University of Göttingen

March 13th 2024
ACAT 2024, Stony Brook University, USA

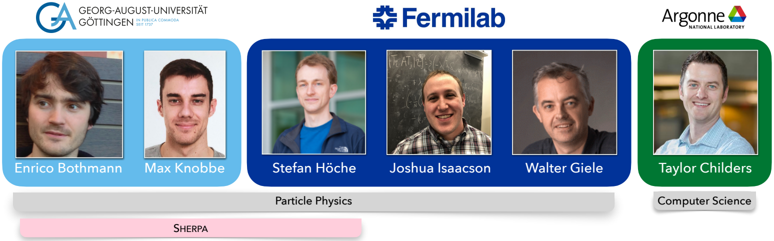
Funded by
Deutsche Forschungsgemeinschaft
German Research Foundation

About

About this Talk

- PEPPER: **P**ortable **E**ngine for the **P**roduction of **P**arton-level **E**vent **R**ecords
- General overview, discuss performance/portability aspects

About the Team



Introduction

Why improve computing performance?

- High statistics at HL-LHC & excellent detector performance

  $\rightarrow$ Need for precise MCEG simulations

  $\rightarrow$ Poor MCEG performance can limit experimental success

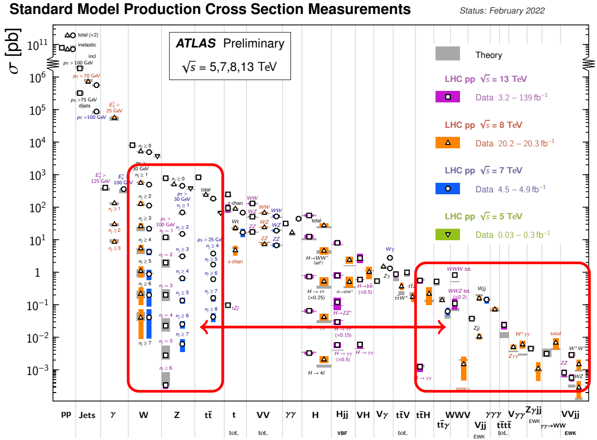
  [HSF Physics Event Generator WG] arXiv:2004.13687, arXiv:2109.14938

What dominates the computing budget?

- Which physics processes?
- Parton or particle level?
- Which final-state jet multiplicities?

In contrast to computers, human resources are scarce.
We can't afford to make incremental improvements.

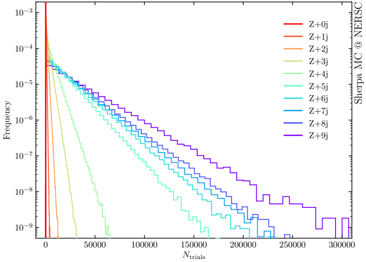
Which physics processes?



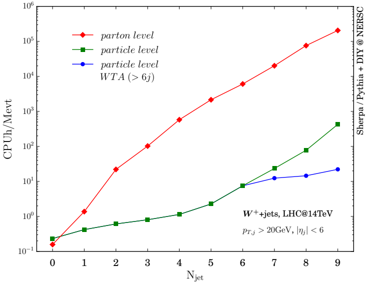
[ATLAS] https://twiki.cern.ch/twiki/bin/view/AtlasPublic/StandardModelPublicResults

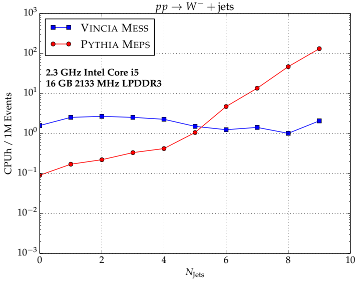
- Signals: High multiplicity but comparably low complexity
- Main backgrounds: High multiplicity and high complexity

- ATLAS' state-of-the-art SHERPA samples
  - $pp \to e^+e^- + 0, 1, 2$j@NLO $+ 3, 4, 5$j@LO
  - $pp \to t\bar{t} + 0, 1$j@NLO $+ 2, 3, 4$j@LO
- Majority of time (60–80 %) spent in tree-level matrix elements (ME) and phase space (after extensive optimization & usage of analytic loop MEs [EB et al.] arXiv:2209.00843)
- Reason: low unweighting efficiencies and expensive ME for high jet multiplicities [Höche,Prestel,Schulz] arXiv:1905.05120



Timing distribution: scaling with multiplicity





- Hard scattering simulation much more demanding than particle-level remainder [Höche,Prestel,Schulz] arXiv:1905.05120
- Complexity of merging ME&PS can be reduced to achieve linear scaling using sector showers [Brooks,Preuss] arXiv:2008.09468 so not a problem in principle

Algorithms: matrix element

Figure of merit

unweighted event generation throughput for highest jet multiplicity

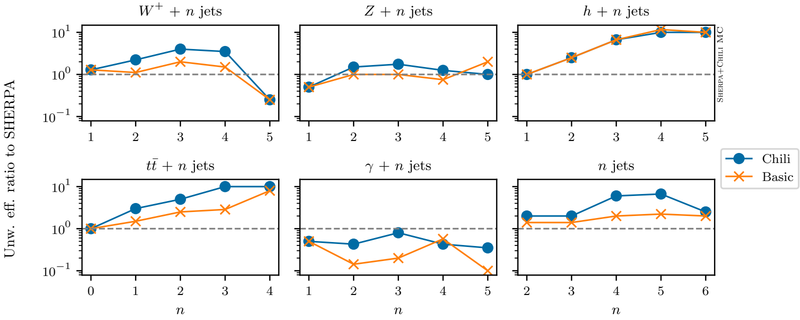
e.g. $pp \to e^+e^- + 5j$, $pp \to t\bar{t} + 4j$ or more

- Berends–Giele recursion for best multi-jet scaling behaviour
- Colour summing for lockstep GPU evaluation
  - Use minimal colour basis developed by amplitude community $\mathcal{O}((n-1)!^2) \to \mathcal{O}((n-2)!^2)$

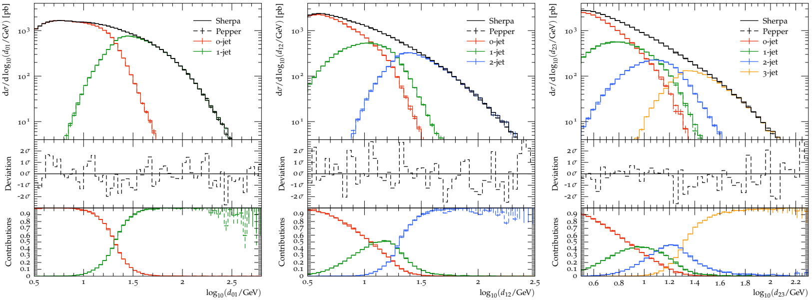
    [Melia] arXiv:1304.7809 arXiv:1312.0599 arXiv:1509.03297

    [Johansson,Ochirov] arXiv:1507.00332
  - Our implementation generalises it to $\ell\ell$+jets amplitudes
- Helicity sampling to avoid additional 2^n scaling

- CHILI phase-space generator uses simple (MCFM-inspired) structure: one t-channel + adjustable number of s channels
  [EB et al.] arXiv:2302.10449
  - Portable (ported builtin CHILI in PEPPER)
  - RAMBO-like speed
  - Efficiency on par with recursive COMIX phase-space
    - $\rightarrow$ Ideal to provide on-device ML training data for many jets

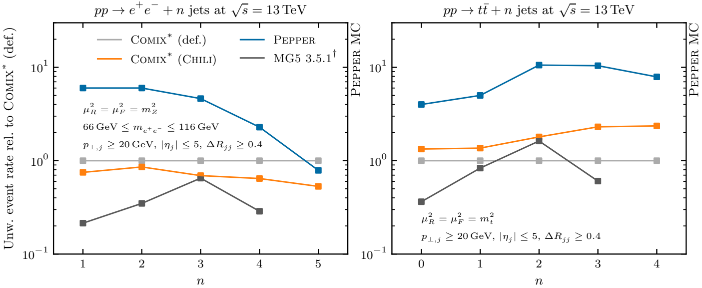


- PDF via LHAPDF, ported to CUDA and Kokkos
- Particle-level simulation via Sherpa or Pythia
  → LHEH5-based framework [EB et al.] arXiv:2309.13154 → Chris' talk



- Test of complete LHEH5-based simulation pipeline with Pepper+Sherpa [EB et al.] arXiv:2309.13154
- Additional 3× speed-up for ATLAS MEPS@NLO $pp \to e^+e^- +$ jets set-up → Sherpa v2.3.0 (Sep '23)

Baseline unweighted event generation performance



- Unweighted event throughput compared to COMIX*
- Constitutes baseline single-threaded performance of currently available competitive algorithms
- Novel standalone PEPPER performs better than COMIX, but PEPPER's real goal is portability [EB et al.] arXiv:2311.06198

Numbers generated on Intel Xeon E5-2650 v2

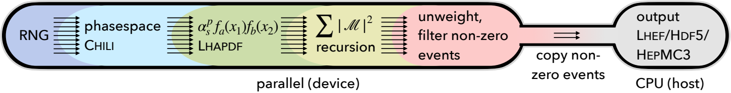
* Partonic processes split into to g/q groups (not SHERPA standard)

† Modified to match efficiency convention of [Gao et. al] arXiv:2001.10028

- Many computing vendors, heterogeneous architectures
- (Pre-)Exascale computing systems intentionally diverse

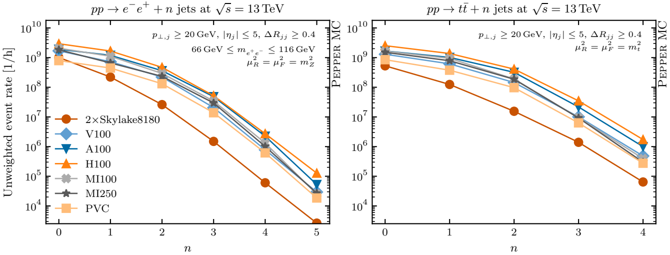
Portability is baked into PEPPER

- Focus on highest multi (e.g. $e^+e^- + 5$, $t\bar{t} + 4$)
  this is beyond small scale computing → WLCG / HPC
- 10–20 years ago: Homogeneous CPU+RAM architectures
- This is undergoing a big change (partly due to AI trends)
  - HPC moves to exascale era → scalability
  - GPU acceleration → portability
- PEPPER addresses both aspects with MPI, HDF5 and Kokkos
- PEPPER parallelises the entire parton-level event generation:



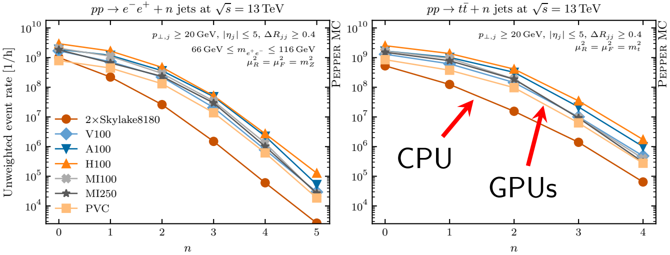
- Tested Xeon CPU, Intel/AMD/Nvidia GPU, HPC systems
  - ✓ Covers all (pre-)exascale architectures on previous slide
  - ✓ Scalable from a laptop to a Leadership Computing Facility

- Excellent performance across a wide range of architectures
- Portability provided by Kokkos: one code-base compiled for different architectures



MEvents / hour	2×Skylake8180	V100	A100	H100	MI100	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

- Excellent performance across a wide range of architectures
- Portability provided by Kokkos: one code-base compiled for different architectures



MEvents / hour	2×Skylake8180	V100	A100	H100	MI100	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

Portability: Aurora example

- Estimate "roughly 330 billion [leptonically decaying V+jets] events" required for HL-LHC [ATLAS] arXiv:2112.09588
  - "**Sherpa 2.2.11 setup would exceed budget by 16%**"
  - Assume all 330 billion events are Z+4j
    Production cost at parton-level would be:
    - 240M CPUh Comix @ Intel E5-2650 v2 CPU
    - 380k GPUh Pepper @ Nvidia A100 →

This would be 8h on Aurora (with PVC)

Conclusions and outlook

Status

- ✓ Portable parton-level multi-jet event generator PEPPER
  Achieves scalability from a laptop to a Leadership Computing Facility

Outlook

- Use synergies PEPPER/CHILI $\leftrightarrow$ on-device training $\leftrightarrow$ ML
- Add more processes to PEPPER, work towards NLO

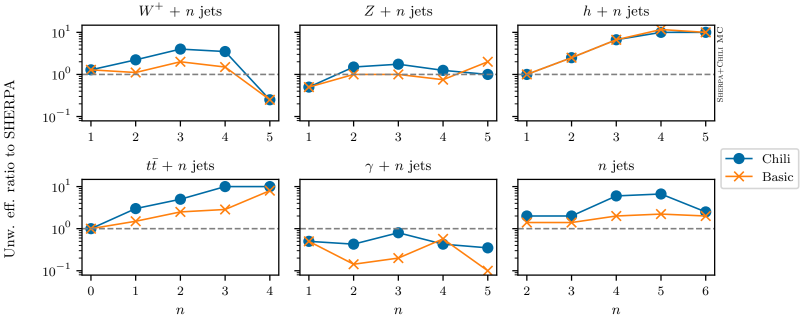
Discussion points

- Regularly updated per-process event generation cost data from ATLAS & CMS? (time/energy/money/. . .)
- Can we get together and establish HPC/GPU workflows with hep-ex & LCFs? (Usability $\leftrightsquigarrow$ Flexibility, Portability . . .)
- Expected adoption of HPC resources by LHC computing?

5 Backup

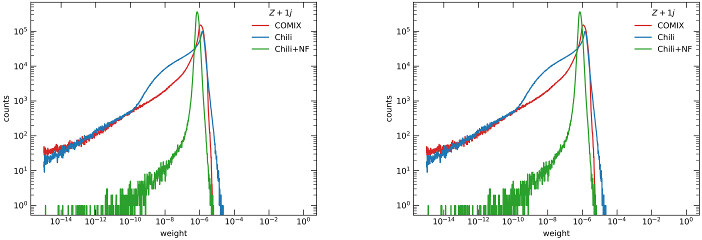
Simple & portable phase space

- CHILI phase-space generator uses simple (MCFM-inspired) structure: one t-channel + adjustable number of s channels
  [EB et al.] arXiv:2302.10449
  - Portable (e.g. basic or "mild" CHILI in PEPPER)
  - RAMBO-like speed
- Performance of basic CHILI (single channel) on par with recursive phase-space in COMIX see also figure on slide 9



Simple & portable phase space: applications & NIS

- What to do with CHILI? (stand-alone library ↗)
  - Simplicity & portability → used in PEPPER v1 as default
  - Speed → public parton-level SHERPA version for HPC ↗
- One/few channels + speed + portability → good fit for ML
- Example: Neural Importance Sampling
  Proof-of-principle [EB et al.] arXiv:2001.05478
  i-Flow+SHERPA [Isaacson et al.] arXiv:2001.10028 arXiv:2001.05486
  MADNIS [Heimel et al.] arXiv:2212.06172 arXiv:2311.01548 → Theo's talk
- CHILI+MADNIS quick'n'dirty [EB et al.] arXiv:2302.10449



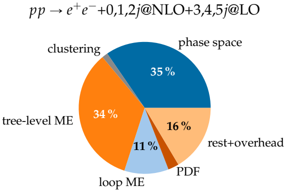
- Future: SHERPA v3.x, on-device NN training with PEPPER

(Some) Components of a MC Computation

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

Components we need to consider:

- Tree-level Matrix elements
- Phase space generation
- PDF's



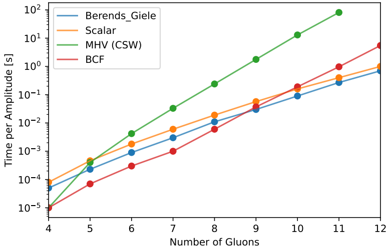
→ Large portion of MC time spend in ME + PS

[EB et al.] arXiv:2209.00843 → Chris' talk

The Amplitudes

- Strategies to compute tree-level amplitudes
  1. Berends-Giele like recursion
  2. Scalar
  3. MHV (CSW)
  4. BCF
- Rely on performance studies from early 2000's
  hep-ph/0602204,hep-ph/0607057
- We are interested in best scaling behaviour / performance for multi-jet processes

⇒ **Choice: Berends-Giele recursion**



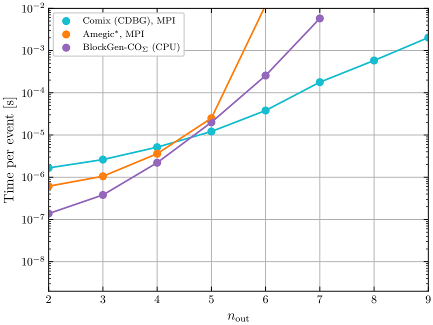
Based on numbers from hep-ph/0602204

Benchmark performance for gluon-only Color-treatment:

- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- Color-summing scales worse than color-dressing, but faster up to roughly 5-6 outgoing jets
- Caveat: Color-sampling comes with penalty factor from slower convergence

⇒ **Algorithmic choice: Sum colors** Helicity-treatment:

- Picture less clear, still allow multiple options

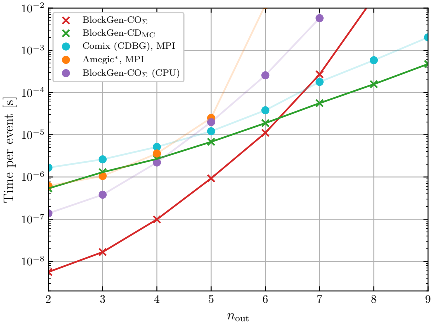


Benchmark performance for gluon-only Color-treatment:

- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- Color-summing scales worse than color-dressing, but faster up to roughly 5-6 outgoing jets
- Caveat: Color-sampling comes with penalty factor from slower convergence

⇒ **Algorithmic choice: Sum colors** Helicity-treatment:

- Picture less clear, still allow multiple options



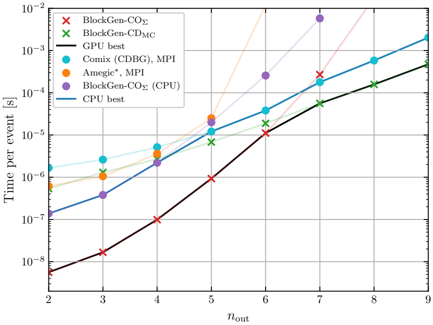
The Color & Helicity Sum [EB, Giele, Höche, Isaacson, Max Knobbe, 2106.06507]

Benchmark performance for gluon-only Color-treatment:

- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- Color-summing scales worse than color-dressing, but faster up to roughly 5-6 outgoing jets
- Caveat: Color-sampling comes with penalty factor from slower convergence

⇒ **Algorithmic choice: Sum colors** Helicity-treatment:

- Picture less clear, still allow multiple options

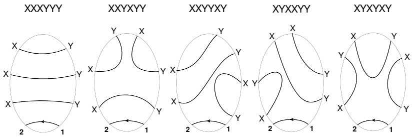


From Gluon-only to V+Jets

- Introduce spinors (Weyl for massless, Dirac for massive particles)
- Add more general QCD three point vertices
- Straight-forward for helicity-sum and Berends-Giele recursion
- First time in a code aimed for production: use minimal QCD color-basis $\{A(1, 2, \sigma), \sigma \in \mathrm{Dyck}\}$

  [T. Melia 1304.7809 & 1312.0599 & 1509.03297; H. Johansson, A. Ochirov, 1507.00332]

  → Allows to fix one fermion line, remaining permutations are given by Dyck-Words
  → Four particle Dyck Words: ()(), (())
  → Significantly fewer amplitudes to compute
- Include EW particles after QCD basis has been set up



[1304.7809]

Phase space generator: CHILI [SciPost Phys. 15 (2023) 169]

- Differential phase space element for an n-particle final state

$$\mathrm{d}\Phi_n(a,b;1,\ldots,n) = \left[\prod_{i=1}^{n} \frac{\mathrm{d}^3\vec{p}_i}{(2\pi)^3\,2E_i}\right] (2\pi)^4\delta^{(4)}\left(p_a+p_b-\sum_{i=1}^{n} p_i\right) .$$

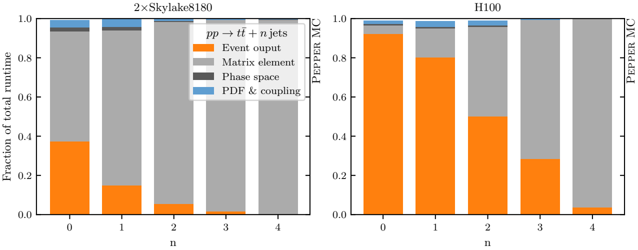
- Standard factorization formula

$$\mathrm{d}\Phi_n(a,b;1,\ldots,n) =$$

$$\mathrm{d}\Phi_{n-m+1}(a,b;\{1,..,m\},m{+}1,\ldots,n)\,\frac{\mathrm{d}s_{\{1,..,m\}}}{2\pi}\,\mathrm{d}\Phi_m(\{1,..,m\};1,\ldots,m) .$$

- Use t-channel + adjustable number of s-channels
- Basic strategy: use single t-channel and only add s-channel resonances when required
  $\rightarrow$ easy to combine with Vegas
  $\rightarrow$ lean implementation allows for portability

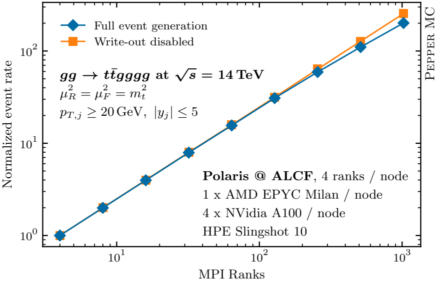
Timing details

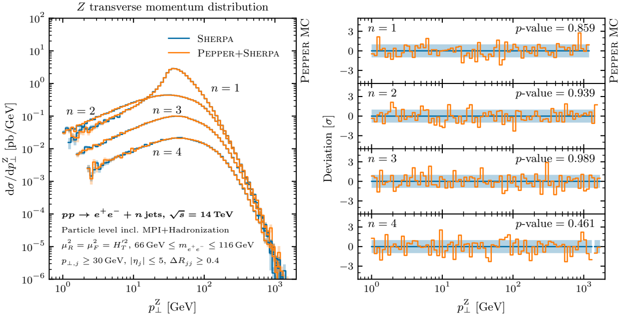


- Lower multiplicities are limited by write-out speed
  $\rightarrow$ No more need for computing improvements, but faster I/O
- Computing becomes relevant component only for large multiplicities

Scalability

- Test scalability for up to 1024×A100's
- Scaling violation due to I/O problems
  → currently investigated at ALCF
- Equivalent technology to [2309.13154]
  → established scaling to up to 16k threads at NERSC
- These problems don't show up for a couple of nodes or low data volume
  → important benchmark





- Validated against SHERPA
  → for $V + j$, $t\bar{t}$+j, single multiplicity and multi-jet merged
- Writeout of HDF5 files, processable via SHERPA & PYTHIA