SHERPA

Enrico Bothmann

Institute for Theoretical Physics
University of Göttingen

Event generators' and N(n)LO codes' acceleration
2023-11-13 at CERN

MCnet

Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Contents

- SHERPA: General purpose Monte Carlo event generator
- Includes parton-level event generators COMIX & AMEGIC plus all necessary particle-level simulation components
- This talk: performance and portability aspects

Table of contents

Why improve computing performance?

- High statistics at HL-LHC & excellent detector performance

  → Need for precise MCEG simulations

  → Poor MCEG performance can limit experimental success

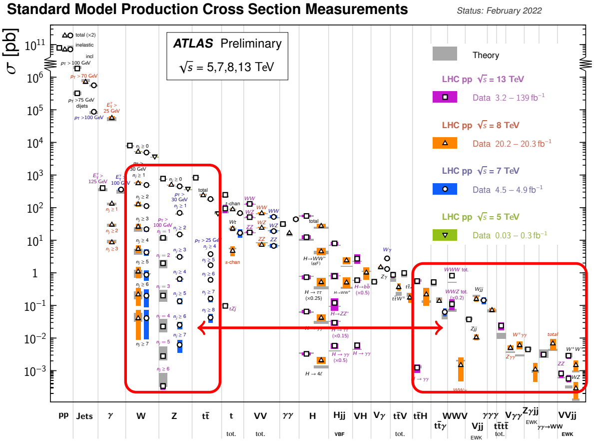
  [HSF Physics Event Generator WG] arXiv:2004.13687, arXiv:2109.14938

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.

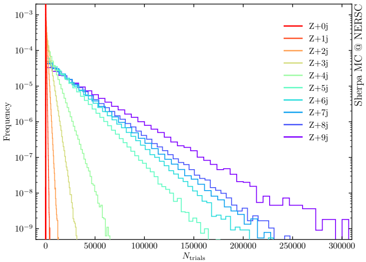
Which physics processes?



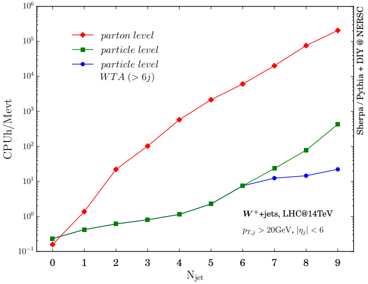
[ATLAS] https://twiki.cern.ch/twiki/bin/view/AtlasPublic/StandardModelPublicResults

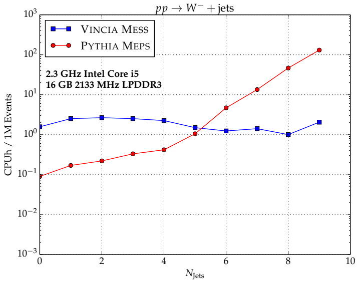
- Signals: High multiplicity but comparably low complexity
- Main backgrounds: High multiplicity and high complexity

- 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 and phase space (after extensive optimization & usage of analytic loop MEs [EB et al.] arXiv:2209.00843) → Chris' talk
- Reason: low unweighting efficiencies for high jet multiplicities [Höche,Prestel,Schulz] arXiv:1905.05120



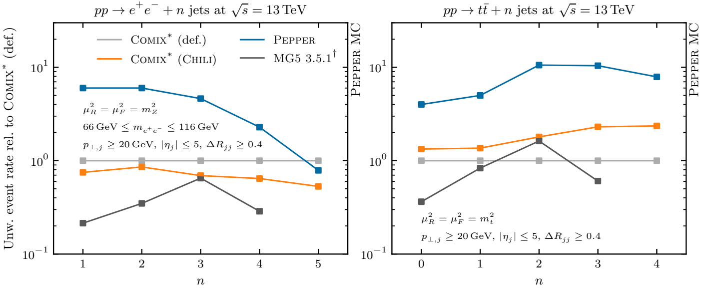
Timing distribution: scaling with multiplicity





- 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

Status quo: 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

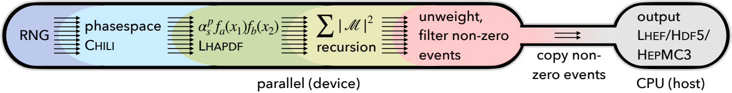
* Partonic processes split into to g/q groups (not SHERPA standard)

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

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

Portability solution

- Focus on high multiplicity ($e^+e^- + 4$, $t\bar{t} + 3$ or more jets) 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

Portability: example 1

- Polaris testbed to prepare scientific applications for exascale: 560 nodes, 1×CPU & 4×A100 GPUs each, Slingshot 10
  - On a single node, utilizing a single A100, PEPPER generates and stores 320M unweighted $pp \to t\bar{t}jj$ events per hour
  - Ideal scaling up to 1/4 of the entire system for $gg \to t\bar{t}gggg$
  - Limited only by I/O (50% of runtime in $pp \to t\bar{t}jj$)

Portability: example 2

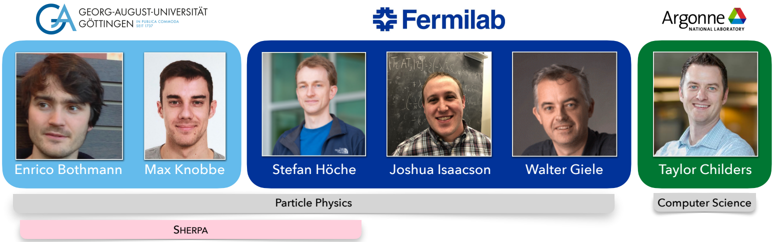
- Estimate "roughly 330 billion [leptonically decaying V+jets] events" required for HL-LHC [ATLAS] arXiv:2112.09588
  - "**Sherpa 2.2.11 setup would exceed budget by 16%**"
  - Assume all 330 billion events are Z+4j
    Production cost at parton-level would be:
    - 240M CPUh COMIX @ Intel E5-2650 v2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

This would be 8d on Polaris, or 6h on Aurora

Portable parton-level simulations with PEPPER

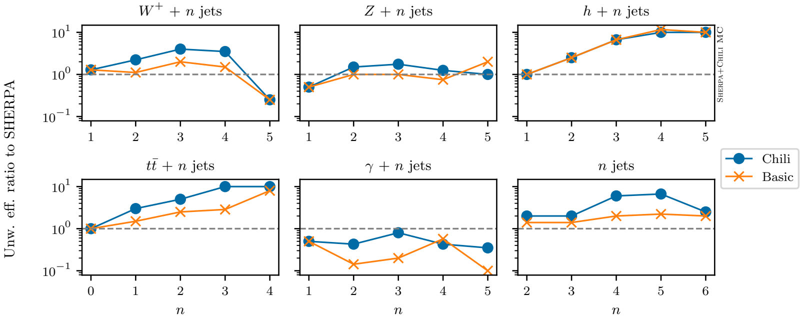
✓ Can target NVidia, AMD, Intel GPUs; HPC-ready

- Ideal to provide on-device ML training data for many jets
- Particle-level simulation via SHERPA or PYTHIA
  → LHEH5-based framework [EB et al.] arXiv:2309.13154 → Chris' talk
- All details on PEPPER [EB et al.] arXiv:2311.06198 → Max' talk
- v1 release on GitLab



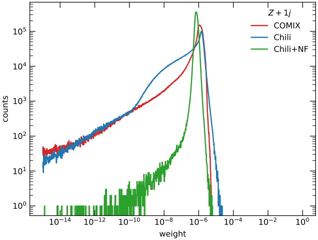
Simple & portable phase space

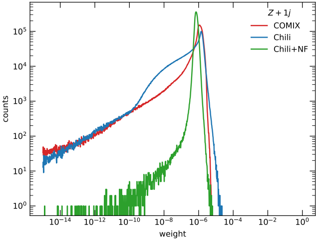
- CHILI phase-space generator uses simple (MCFM-inspired) structure: one t-channel + adjustable number of s channels
  [EB et al.] arXiv:2302.10449 → Max' talk
  - Portable (e.g. basic or "mild" CHILI in PEPPER)
  - RAMBO-like speed
- Performance of basic CHILI (single channel) on par with recursive phase-space in COMIX see also figure on slide 6



Simple & portable phase space: applications & NIS

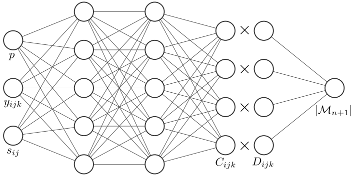
- What to do with CHILI? (stand-alone library ↗)
  - Simplicity & portability → used in PEPPER v1 as default
  - Speed → public parton-level SHERPA version for HPC ↗
- One/few channels + speed + portability → good fit for ML
- Example: Neural Importance Sampling
  Proof-of-principle [EB et al.] arXiv:2001.05478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

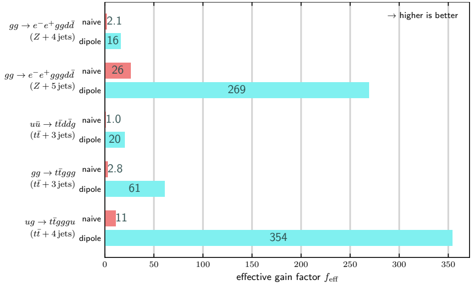




- Future: SHERPA v3.x, on-device NN training with PEPPER

- Replace $|\mathcal{M}|^2$ with fast ML surrogate → Daniel's talk
- Use second unweighting step to correct to exact $|\mathcal{M}|^2$
  [Danziger et al.] arXiv:2109.11964
- Train linear coefficients C_{ijk} of dipole terms D_{ijk}
  [Janßen et al.] arXiv:2301.13562



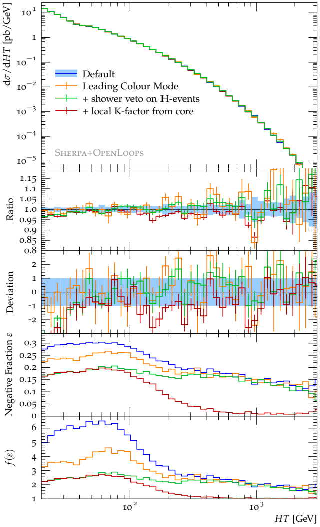


SHERPA negative weight fractions

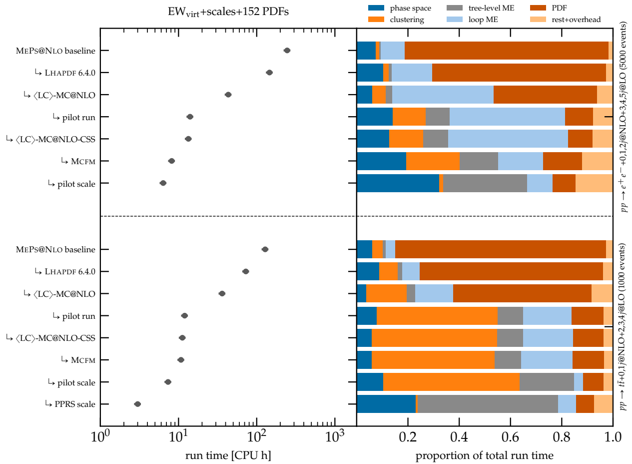
[Danziger Höche Siegert] arXiv:2110.15211

[ATLAS] arXiv:2112.09588

- Explored three methods to reduce negative weight fraction in SHERPA
  1. S-MC@NLO $\rightarrow$ MC@NLO $=$ leading colour & no spin corr.
  2. Include jet veto on $\mathbb{H}$ events, as originally formulated in [Höche et al.] arXiv:1207.5030, see also [Frederix et al.] arXiv:2002.12716
  3. Use local K-factor in NLO$\rightarrow$LO merging from core configuration instead of highest multiplicity
- SHERPA v2.2.8 (Sep '19)

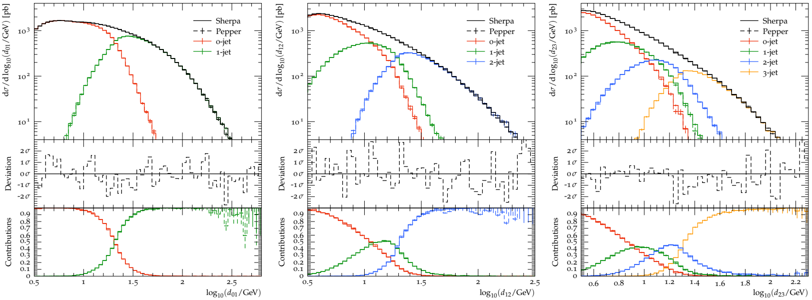


SHERPA performance improvements



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

→ 39× speed-up for ATLAS MEPS@NLO $pp \to e^+e^- +$ jets set-up
→ 43× speed-up for ATLAS MEPS@NLO $pp \to t\bar{t} +$ jets set-up
SHERPA v2.2.13 (Nov '22)



- 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)

Conclusions and outlook

Status

- ✓ Accelerated & scalable version of SHERPA → Chris' talk (Tue)
- ✓ Portable parton-level event generator PEPPER → Max' talk (Tue)
  Achieves scalability from a laptop to a Leadership Computing Facility

Outlook

- Production-ready surrogate unweighting
- Use synergies PEPPER/CHILI ↔ on-device training ↔ 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 ⇜⇝ Flexibility, Portability ...)
- Expected adoption of HPC resources by LHC computing?