GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Institut für Theoretische Physik

Performance of modern color decompositions for standard candle LHC tree amplitudes

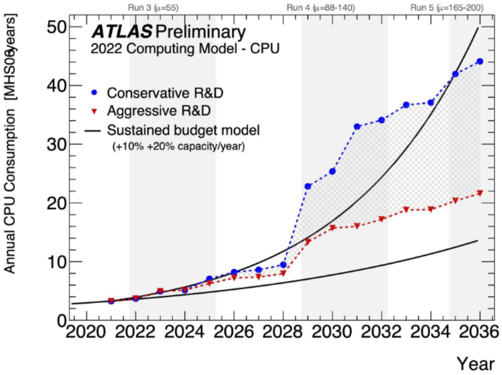
Max Knobbe
ACAT-2022

In collaboration with:
E. Bothmann, T. Childers, W. Giele, S. Höche, J. Isaacson, R. Wang

MCnet

Gefördert durch
DFG Deutsche Forschungsgemeinschaft

Introduction / Motivation



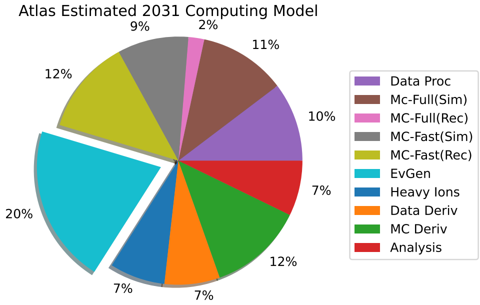


Figure and numbers taken from [CERN-LHCC-2022-005]

- Computing needs are predicted to grow faster than available resources
  $\rightarrow$ Computing budget might limit physics outcome
- Sizeable part of CPU budget spend on event generation (roughly 20%)
- Our strategy, dedicated rewrite to tackle major bottle-necks
  $\rightarrow$ Identify major bottle-necks first!

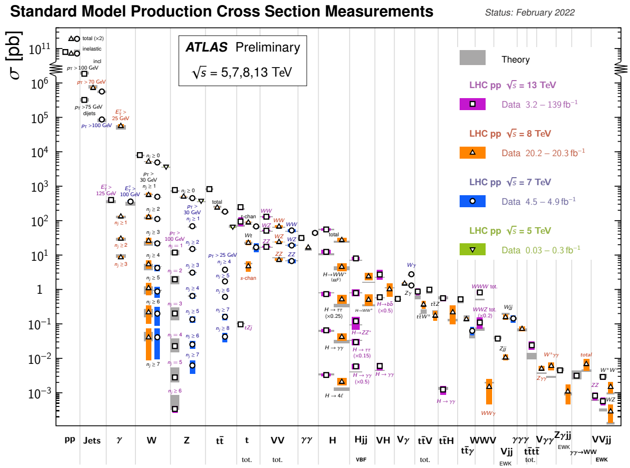
Boiling down the Problem

Expensive MC Samples: V+Jets with many jets

- Background to essential analysis(es), e.g. Higgs-boson and top-quark measurements
  cf. [2112.09588]
- Large production cross-section → large MC samples
- To reduce significant portion of MC budget, ensure to be efficient for these processes

Guiding Principles for the following discussion:

1. Good performance for bottleneck processes (for now: V+Jets,ttbar), in particular for many jets!
2. Deployable in modern architectures
3. Useful integration in existing MC tool-chain



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

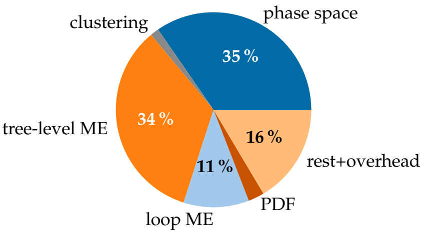
(Some) Components of a MC Computation

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

- Large portion of MC time spend in ME + PS
  cf. [2209.00843] C. Gütschow's talk
- In this talk: Re-think ME stratgy
- Goal: Develop efficient strategy for the different components
- Naive treatment of helicity/color sum scales terribly with increase of multiplicity
-
$$|\mathcal{M}_{ab\to X_n}(1, \ldots, n)|^2 = \sum_{\text{helicity}} \sum_{\text{color}} A(p_1, \ldots, p_n) A(p_1, \ldots, p_n)^\dagger$$

  Three major components we have to take care of are
  1. The helicity sum
  2. The amplitudes
  3. The color sum

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



cf. Talk by C. Gütschow, or [2209.00843]

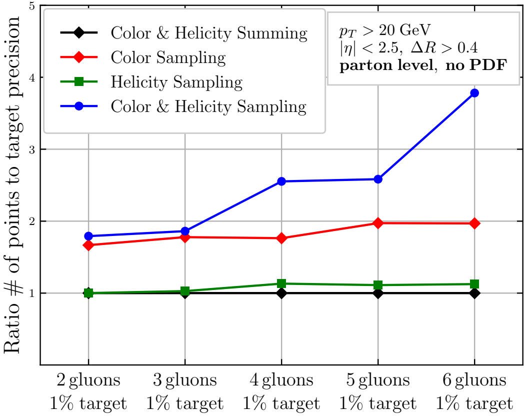
1^{st} Component: The Helicity Sum

Investigate impact of helicity-summing vs. sampling on convergence

$\rightarrow$ How many points to we have to compute to make up for MC estimate?

- Compare convergence in realistic setup
  $\rightarrow$ Recursive PS-Sampler, ...
- Helicity sampling comes with close to zero additional points
- Loss in precision increases with multiplicity for color and helicity sampling

$\Rightarrow$ **Algorithmic choice: Sample helicities**

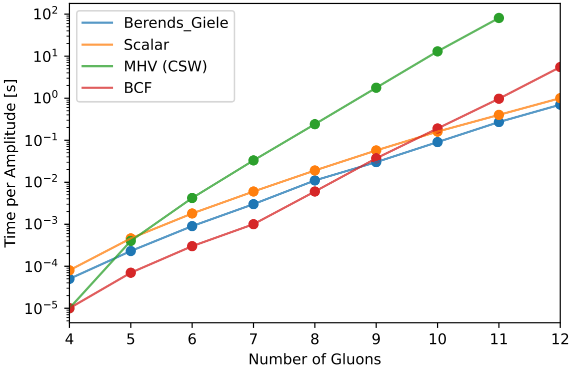


[Bothmann, Giele, Höche, Isaacson, MK, 2106.06507]

2^{nd} Component: The Amplitudes

- Different strategies to compute tree-level amplitudes efficiently
  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 (guiding principles)

Best scaling option is the Berends-Giele recursion
⇒ **Algorithmic choice: Berends-Giele recursion**

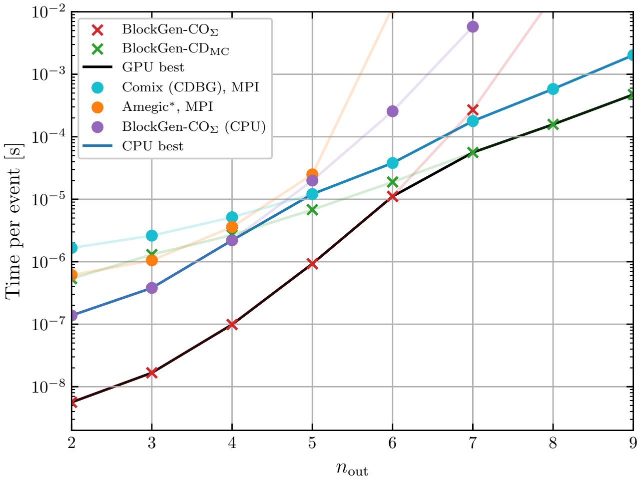


Based on numbers from hep-ph/0602204

NB: BCF-like recursions show potential for intermediate multiplicites, dominant effect from (N)MHV amplitudes that can be included in current generator

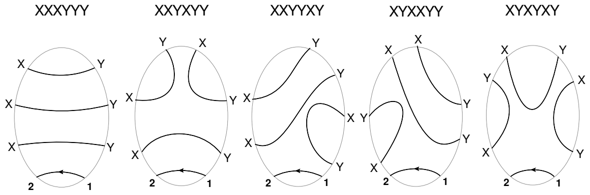
- Benchmark performance for gluon-only process
- Relevant test, since as-many-gluon-as-possible amplitudes make up largest portion of computing time for jet-processes
- Compare different color treatments: color-dressing/summing/sampling
- Color-sampled algorithms scale similar to color-summed approaches
- 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**



New results: 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
  [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
  → Significantly less amplitudes to compute
- Need to treat process groups (next slide)



[1304.7809]

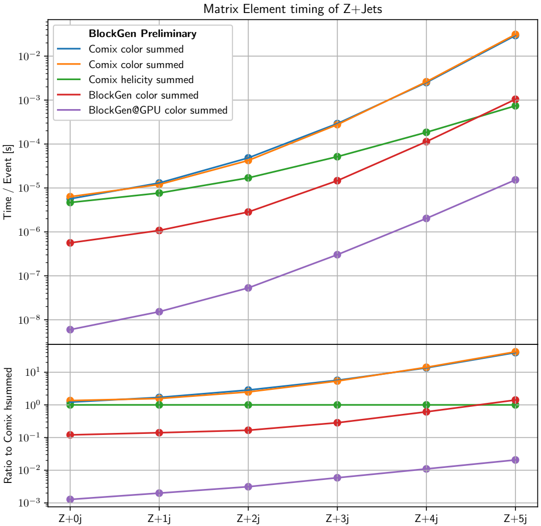
New Results: Include Vector-bosons

- Straight-forward for helicity-sum and Berends-Giele recursion
- Introduces helicity dependant vertices / coupling
- But: can be quite complicated, if number of EW-vertices is not restricted
  $\rightarrow$ Start with lowest order first
- Only remaining caveat: Color ordering introduced color-ordered vertices, amplitudes
  $\rightarrow$ Carefull when attaching EW lines to QCD ones

Results

Relevant benchmark process: $pp \to Z[ee] + nj$

- Realistic setup for ME computations
  $\to$ Include all sub-processes, no PDF
  $\to$ Dominated by as-many-as-possible
- Compare dedicated C++ Version with dedicated Cuda version
- PS: Rambo
- CPU: i3-8300 CPU @ 3.70GHz, GPU: Tesla V100S
- Excellent performance compared to current Sherpa-Standard
- Almost three orders of magnitude for lower multiplicities (single CPU core vs full GPU)



Phase-Space generation

Basic building blocks:

- T-channel phase-space generator from MCFM [Figy, Giele 1806.09678]
- S-channel phase-space factorization and decays [Byckling, Kajantie NPB9(1969)568]
- Variable number of s-channel decays (user-defined) to eliminate exponential scaling

Performance within Sherpa, compared to Comix [Gleisberg, Höche 0808.3674]

Process /	Default PS		New PS	
MC accu	Time	# pts	Time	# pts
W+1j / 1‰	4m 52s	10.3M	2m 32s	3.10M
W+2j / 3‰	17m 12s	5.52M	13m 52s	2.53M
W+3j / 1%	46m 24s	7.48M	20m 16s	1.15M

Xeon® E5-2650v2, timing for ME & PS, # points before cuts
$\sqrt{s} = 14$TeV, anti-k_T, $p_{T,j} = 30$GeV, $\mu_{R/F} = H'_T/2$

Process /	Default PS		New PS	
MC accu	Time	# pts	Time	# pts
H+1j / 1‰	2m 20s	1.83M	1m 36s	1.50M
H+2j / 3‰	4m 36s	2.32M	4m 4s	0.71M
H+3j / 1%	18m 12s	2.32M	12m 56s	0.63M

Xeon® E5-2650v2, timing for ME & PS, # points before cuts, H on-shell
$\sqrt{s} = 14$TeV, anti-k_T, $p_{T,j} = 30$GeV, $\mu_{R/F} = H_{T,m}/2$

Process /	Default PS		New PS	
MC accu	Time	# pts	Time	# pts
$t\bar{t}$+0j / 1‰	4m 38s	3.15M	4m 0s	3.59M
$t\bar{t}$+1j / 3‰	3m 12s	1.38M	3m 4s	1.47M
$t\bar{t}$+2j / 1%	11m 58s	1.47M	11m 20s	0.89M

Xeon® E5-2650v2, timing for ME & PS, # points before cuts
$\sqrt{s} = 14$TeV, anti-k_T, $p_{T,j} = 30$GeV, $\mu_{R/F} = H_{T,m}/2$

Process /	Default PS		New PS	
MC accu	Time	# pts	Time	# pts
2j / 1‰	12m 48s	2.98M	7m 44s	1.80M
3j / 3‰	22m 48s	6.80M	23m 12s	2.39M
4j / 1%	1h 25m	6.95M	50m 24s	0.91M

Xeon® E5-2650v2, timing for ME & PS, # points before cuts
$\sqrt{s} = 14$TeV, anti-k_T, $p_{T,j} = 30$GeV, $\mu_{R/F} = H_T/2$

Conclusion / Outlook

- BlockGen matrix-elements show excellent performance on both CPU and GPU
  - Helps to reduce CPU time consumption but also enables more/complicated computations and use of modern GPU data-centers
- CPU Speed-up $\mathcal{O}(10)$, GPU Speed-up $\mathcal{O}(100)$
  - Color-summing/helicity sampling good choice for intermediate to high multiplicities
  - No process-specific optimisations $\rightarrow$ straight-forward extension to more processes
  - Potentially need some more fine-tuning of Cuda version
- Eventual goal: parton-level generator that delivers seed events to SHERPA for further processing
  - Proposed workflow, (ME,PS)@GPU, (Shower,Hadronisation,MPI,..,)@CPU using HDF5 I/O [Höche, Prestel, Schulz 1905.05120]
  - Need to complete combination with PS-Integrator
  - Add unweighting & HDF5 event output

    $\rightarrow$ Nearing usable product, natural inclusion in current toolchain