Sherpa Report for LHCC Review

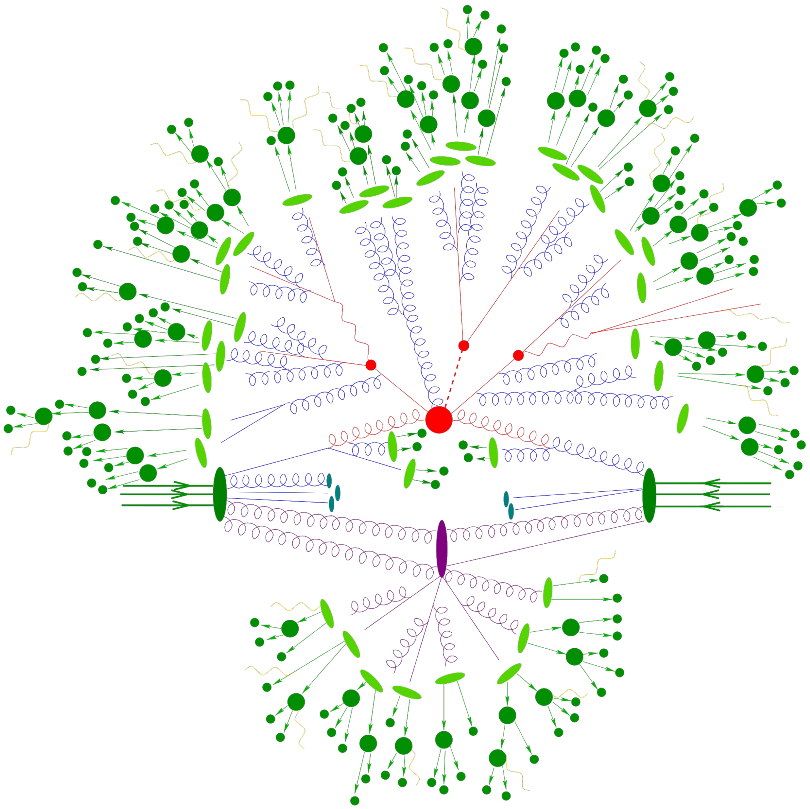
The Sherpa Team

Zoom Universe, 22/04/2021

The Sherpa framework

[Bothmann et al.] arXiv:1905.09127

- Hard interaction
  - LO, NLO QCD/EW*, NNLO QCD
  - ME generators Amegic & Comix
- Radiative corrections
  - Catani-Seymour based PS
  - Dire, YFS QED resummation
  - EW Sudakovs
- Multiple interactions
  - Sjöstrand-Zijl model
- Hadronization
  - Cluster hadronization model
- Hadron Decays
  - Phase space or EFTs,
  - YFS QED corrections



*via interfaces to 1-loop generators

General considerations

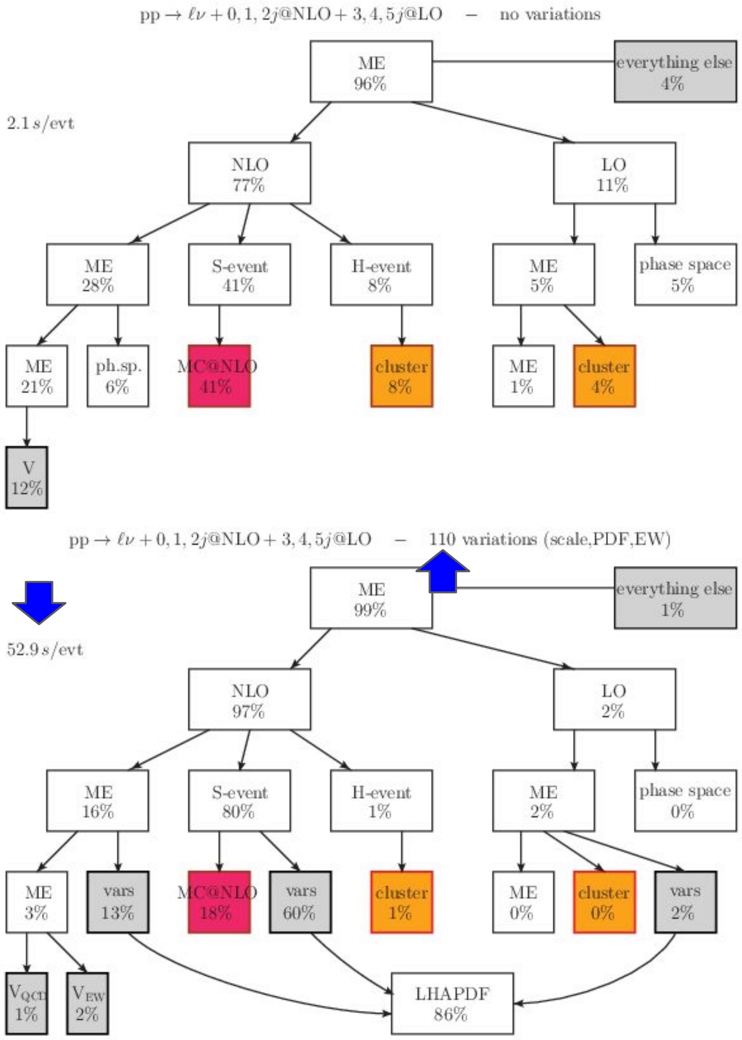
- Are there plans/funds in place to continue support through HL-LHC?
  - Dependent on funding agencies we will continue to have PDRA’s (the Sherpa posts at IPPP are seen as part of IPPP’s core mission and should be treated as “core” in the UK sense)
- What major physics updates do you foresee for HL-LHC?
  - NLO parton shower
  - NLO multi-jet merging in decays
  - Resonance-aware matching and NLO subtraction
  - Proper NNLO matching for important processes
  - Full SM parton shower and NLO EW matching
  - SHRiMPS (inclusive QCD scattering simulation)
  - Improved b- and c-decays
  - Fully validated SMEFT
- What major software updates are foreseen for HL-LHC? Main bottlenecks?
  - Restructuring of evgen framework for improved efficiency
  - Separation into ME & PS generator
  - GPU support in ME generation
- Are there issues/areas of work where help from HSF/expts may be needed?
  - Restructuring of framework
  - Porting of code to GPUs (ongoing, see later)

ME+PS generator specific considerations

- What are the current CPU performance bottlenecks & how being worked on?
  - Sherpa computes many quantities before unweighting that are only needed afterwards
    Mostly solved at this point (see next slides)
  - At the cost of accuracy some costly aspects of NLO evgen can be simplified
    (Ignore spin and color correlations. **This is optional and available in Sherpa 2.x.y**)
- What improvements in compute performance are expected for Run 4/5?
  - CPU and (to a lesser extent) memory consumption
    - Improvements to current algorithms will allow large speedup, but hard to predict
      (e.g. clustering algorithm has been sped up by factor 4 for Sherpa v3.x.y)
    - In any honest NNLO simulation, these factors will be overcompensated
      The method of choice is likely to sacrifice precision in a controlled manner
      (NLO example: drop color and spin correlations in PS matching)
    - Better phase-space sampling, e.g. cut efficiency
  - Negative weight fractions at NLO QCD and mitigation strategies
    - **Analyzed in detail in K. Danziger's Masters Thesis**
      https://cds.cern.ch/record/2715727

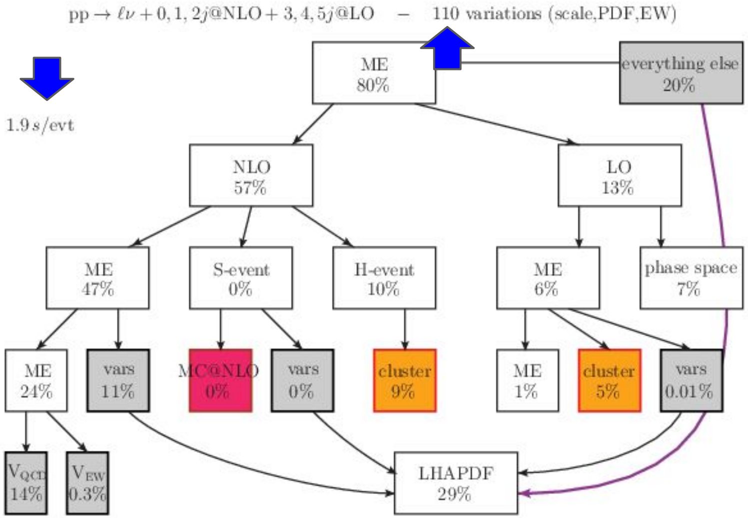
Performance analysis of v2.2.11

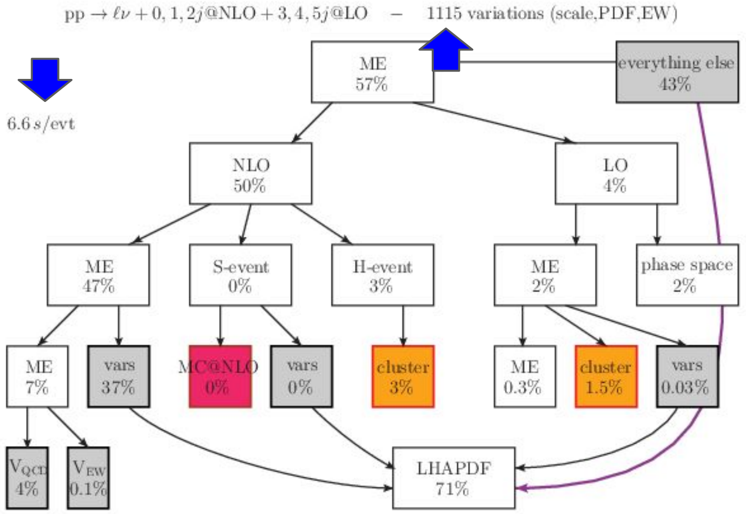
- Sherpa is written by theorists
  and not optimized for unweighting
- Code structure follows
  physical event structure
- w/o variations performance increases
  possible through physics compromises
  - unweighting of colour-/spin-correlated first emission costly, either neglect colour and spin correlations (also reduces neg. weights) or live with residual weight fluctuations
  - simplify parton shower clustering or apply after unweight generates addtl. correction terms of higher order but physics modeling may be acceptable
- w/ variations performance increases
  possible without physics compromises
  - too many quantities computed when program flow in the correct physics module, but still before unweighting, needs to be restructured
  - excessive use of LHAPDF through inefficient call structure
- Gains also possible from use of analytics



Performance analysis as of now

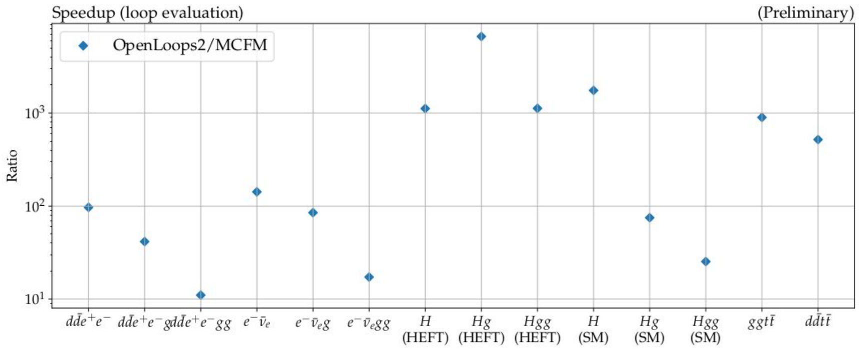
- Re-arranged program flow
  in computation of weight variations
- Added option for traditional MC@NLO
  (no color or spin correlations in S events)
- EW corrections computed after unweight
- **Factor ~28 overall speedup**
  for 110 scale/PDF variations,
  Factor ~1.4 without variations
- Remaining bottlenecks
  - Evaluation of 1-loop ME (see next slide)
  - Parton-shower clustering
    (factor 4 speedup expected in v3.x.y)
  - Math library (pow, exp and log functions)
- Improvements will be made available
  on short timescale as v2.3.0

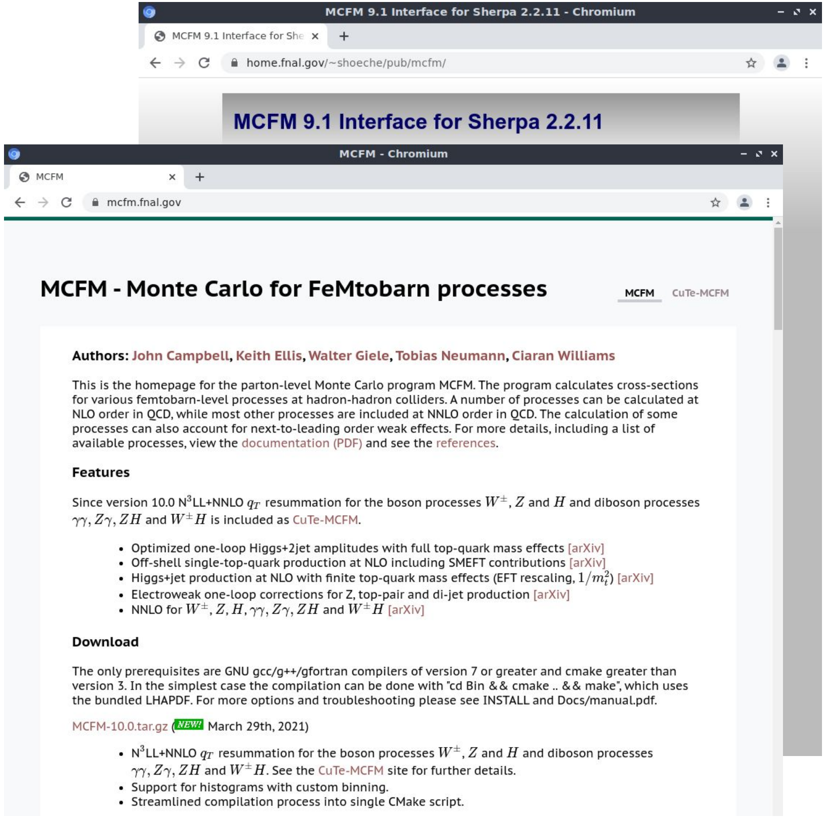




Efficient evaluation of QCD virtual corrections

- New BLHA-type interface to MCFM 9.1 [Campbell,Preuss,SH] TBP
- Currently supported processes
  - pp→e^+e^- + ≤ 2 partons $\mathcal{O}(\alpha)=2$, $\mathcal{O}(\alpha_s)=n_{parton}+1$
  - pp→νν + ≤ 2 partons $\mathcal{O}(\alpha)=2$, $\mathcal{O}(\alpha_s)=n_{parton}+1$
  - pp→e^-ν + ≤ 2 partons (any CKM) $\mathcal{O}(\alpha)=2$, $\mathcal{O}(\alpha_s)=n_{parton}+1$
  - pp→e^+ν + ≤ 2 partons (any CKM) $\mathcal{O}(\alpha)=2$, $\mathcal{O}(\alpha_s)=n_{parton}+1$
  - pp→h + ≤ 2 partons (HEFT/SM) $\mathcal{O}(\alpha)=1$, $\mathcal{O}(\alpha_s)=n_{parton}+3/2$
  - pp→tt $\mathcal{O}(\alpha)=0$, $\mathcal{O}(\alpha_s)=3$
- Example speedup of loop evaluation





- Sherpa plugin available for v2.2.11+

Improving unweighting efficiency with Neural Networks

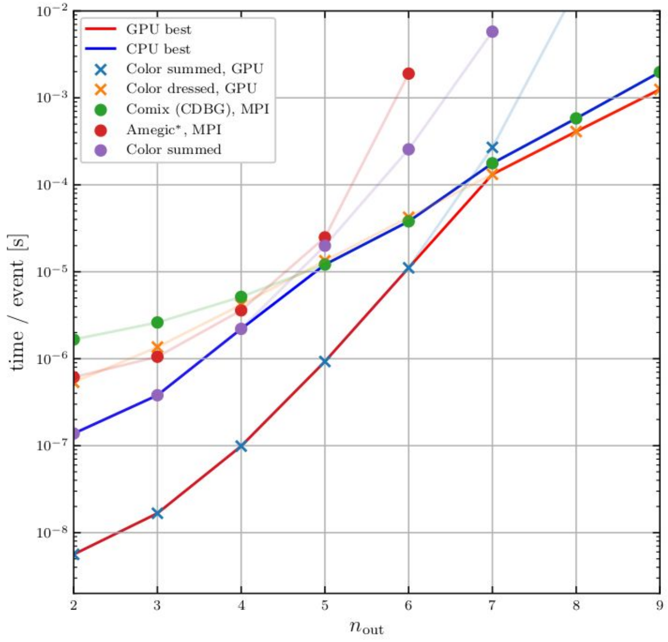
[Bothmann,Janßen,Knobbe,Schmale,Schumann] arXiv:2001.05478
[Gao,Isaacson,Krause,Schulz,SH] arXiv:2001.05486

- Variable transformation technique:
  - Learn integrand to improve importance sampling, similar to Vegas
  - Insufficient training yields high uncertainties, but no bias
  - Events generated from scratch with no pre-existing sample required
- Reality check for the most relevant processes

unweighting efficiency $\langle w \rangle / w_{\max}$		LO QCD $n=0$	$n=1$	$n=2$	$n=3$	$n=4$	NLO QCD (RS) $n=0$	$n=1$
$W^+ + n$ jets	Sherpa	$2.5 \cdot 10^{-1}$	$3.4 \cdot 10^{-2}$	$6.7 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$	$6.6 \cdot 10^{-4}$	$6.5 \cdot 10^{-2}$	$2.9 \cdot 10^{-3}$
	NN+NF	$5.8 \cdot 10^{-1}$	$1.2 \cdot 10^{-1}$	$8.8 \cdot 10^{-3}$	$1.6 \cdot 10^{-3}$	$8.9 \cdot 10^{-4}$	$1.7 \cdot 10^{-1}$	$4.0 \cdot 10^{-3}$
	Gain	2.3	3.6	1.3	0.99	1.4	2.7	1.4
$W^- + n$ jets	Sherpa	$2.4 \cdot 10^{-1}$	$3.9 \cdot 10^{-2}$	$8.4 \cdot 10^{-3}$	$1.7 \cdot 10^{-3}$	$8.8 \cdot 10^{-4}$	$6.0 \cdot 10^{-2}$	$3.3 \cdot 10^{-3}$
	NN+NF	$6.2 \cdot 10^{-1}$	$1.3 \cdot 10^{-1}$	$1.2 \cdot 10^{-2}$	$2.3 \cdot 10^{-3}$	$9.8 \cdot 10^{-4}$	$1.6 \cdot 10^{-1}$	$3.8 \cdot 10^{-3}$
	Gain	2.6	3.2	1.5	1.4	1.17	2.8	1.2
$Z + n$ jets	Sherpa	$4.3 \cdot 10^{-1}$	$4.3 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$2.7 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	$1.1 \cdot 10^{-1}$	$4.9 \cdot 10^{-3}$
	NN+NF	$5.1 \cdot 10^{-1}$	$1.1 \cdot 10^{-1}$	$1.3 \cdot 10^{-2}$	$2.6 \cdot 10^{-3}$		$1.8 \cdot 10^{-3}$	$4.9 \cdot 10^{-3}$
	Gain	1.2	2.6	1.1	0.97		1.7	1.0

Towards a Matrix Element generator for GPUs

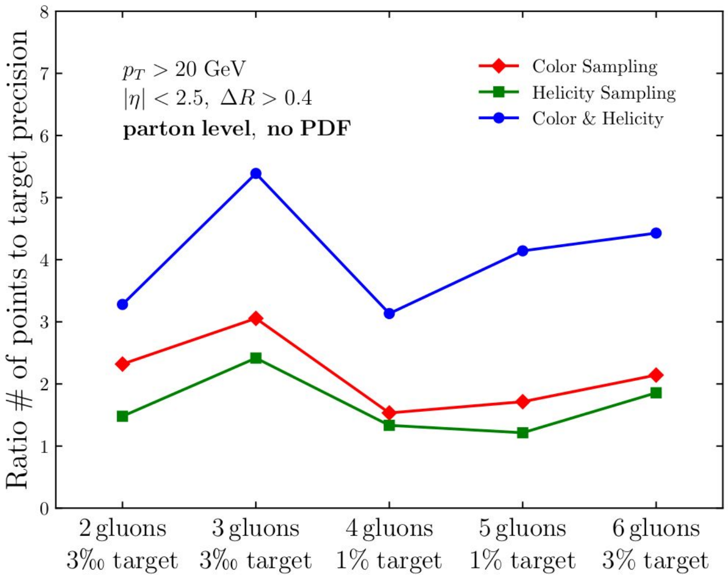
- Design study for new GPU ME generator
  [Bothmann,Giele,SH,Isaacson,Knobbe] TBP
- Currently limited to all-gluon amplitudes at arbitrary multiplicity
- Good performance compared to TESS
  [Giele,Stavenga,Winter] arXiv:1002.3446
- Exploration of color summation/sampling and helicity summation/sampling
  - Color sum preferred on GPU
    (Compute bound but memory lean)
  - High-multiplicity computation memory limited (color matrix) but practically likely irrelevant
  - Minor overhead (~factor 2) for helicity sampling but major timing improvement
- To be generalized with help of Kokkos
  [Bothmann,Childers,Giele,SH,Isaacson,Knobbe]



ME evaluation time per phase-space point
Measured on E5-2690 and V100

Design choice for GPU generator

- Helicity sampling proves beneficial for computation of matrix elements
  [Draggiotis,Kleiss,Papadopoulos] hep-ph/9807207
  - Memory lean algorithm
  - Predictable memory access pattern
- For color sampling we have two options
  - Memory intensive algorithm with predictable access pattern (hep-ph/9807207)
  - Memory lean algorithm with random memory access pattern (Comix)
- There is a “penalty” to sampling because the phase-space integral will converge slower due to addtl. integration variables
  [Gleisberg,SH] arXiv:0808.3674
- But this overhead factor does not scale!



Overhead for helicity and color sampling, measured with Comix as function of # points to target precision