



University  
of Glasgow



# Challenges in Monte Carlo event generation

Andy Buckley, University of Glasgow

CHACAL 2024

Johannesburg, 23 January 2024



# LHC event generation

Evgen is simulation of the fundamental  $pp$  process

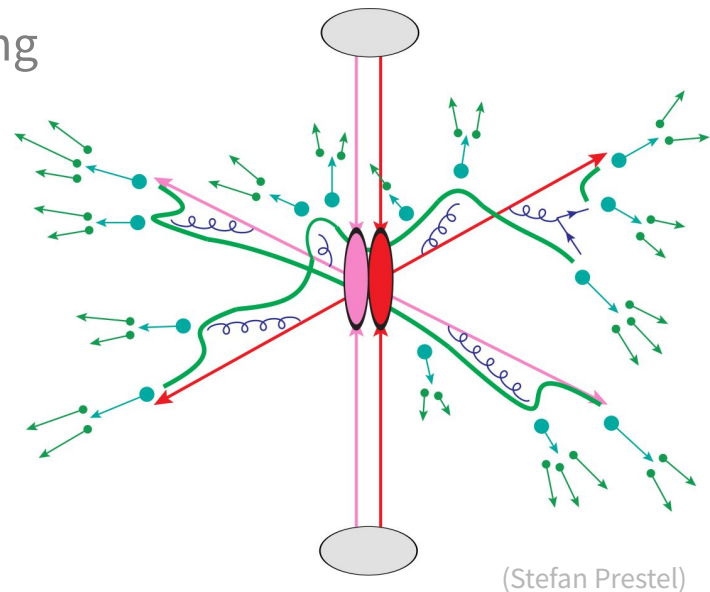
Focus on fully differential “SHG” codes, with dressing of hard matrix element by pQCD & pheno

In the LHC’s first decade, from tiny CPU cost to very expensive! cf. step-change in formal accuracy.

*While others simplified, MC  $\rightarrow$   $\sim$ factorial explosion!*

Many issues in summary from HSF MC workshop:

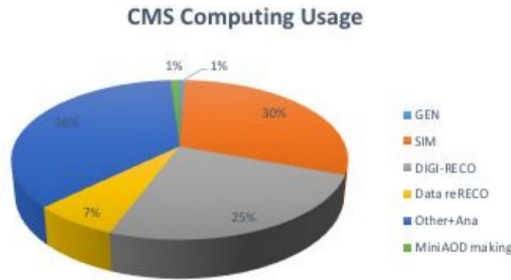
<https://indico.cern.ch/event/751693/timetable/>



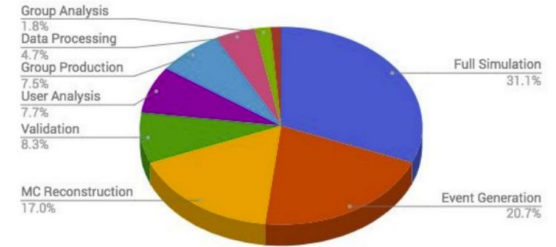
Thanks to Stefan Hoeche, Holger Schulz, Keith Hamilton, Marek Schoenherr, Frank Siegert, Chris Gutschow, ...

# Evgen cost, present and future

Cf. 2000, only leading-order shower/hadronisation MC available! (Alpgen, MadEvent, MC@NLO in 2002, Sherpa & Powheg, aMC@NLO later). ~Trivial...

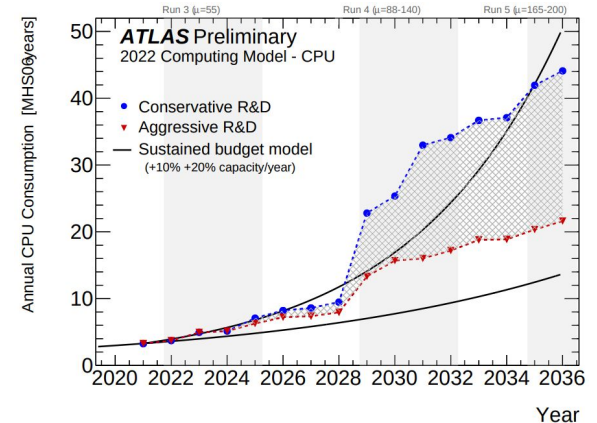


US ATLAS Wall Clock CPU - 2016



Big difference in ATLAS vs CMS time — reflects large use of CMS LO BSM. **For core SM, both 15-20%!!**

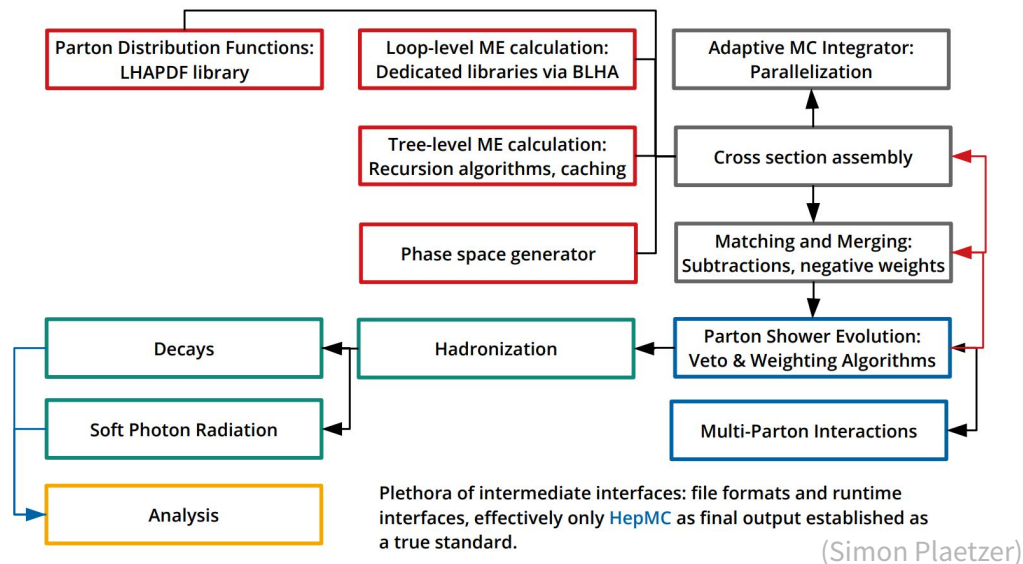
HL-LHC predictions very concerning: evgen leads CPU shortfall by x2. How come? Can't have HL-LHC physics impact dominated by MC stats systematics!!



# SHG MC generator codes

Leading SHGs now all C++ based.  
Object-orientation vs. performance?  
Complex code → dev challenge

Typical modern SHG generator structure:



Factorised strategy for higher-order processes: generate process lib, use dedicated (MPI) phase-space integration run to optimise

# Event generation core: ME and phase space

MC generation rooted in integration and sampling of ME & phase space:

$$d\sigma \sim d\sigma_{\text{hard}}(Q) \times \text{PS}(Q \rightarrow \mu) \times \text{Had}(\mu \rightarrow \Lambda) \times \dots$$

*NB. factorisation is convenience, not reality!*

$$d\sigma_{\text{hard}}(p_1, \dots, p_n | Q) \sim |\mathcal{M}(p_1, \dots, p_n)|^2 d\text{PSP}(p_1, \dots, p_n | Q)$$

$$d\sigma = d\sigma^{\text{LO}} + \alpha_S(Q) d\sigma^{\text{NLO}}(Q) + \alpha_S^2(Q) d\sigma^{\text{NNLO}}(Q) + \dots$$

Efficient MC generation requires efficient sampling of ME over partonic phase space. Efficient = low rejection rates / high weights. Flatten integrand via change of variables: exploit physical singularities, e.g. multi-channel (MadEvent), VEGAS

Modern strategies: parallel run  $\rightarrow$  gridpacks, evgen embarassingly parallel

# Phase space and loop integration

$$\text{PS: } \int d^4 p_1 \dots d^4 p_N |\mathcal{M}(\{p_i\})|^2 \delta p_1^2 \dots \delta p_N^2 \delta^4(p_1 + \dots + p_N) \quad \text{Loop: } \int d^d k_1 \dots d^d k_{N_l} \frac{N(\{k_i\})}{\prod_1^{N_d} D_i}$$

Complexity of ME+PDF singularity structure makes both hard:

**Loop** UV and IR singularities → large variance, poor convergence → *pre-generate*

**PS** is the major scaling issue for higher order MC production, both in legs & loops:

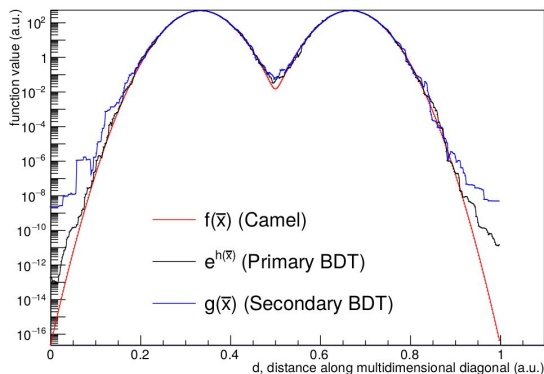
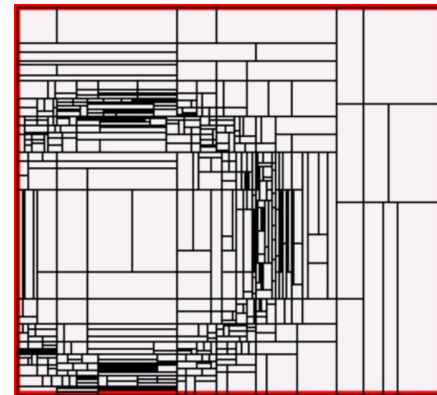
	Tree-level		One-loop
(Valentin Hirschi – see talk)	MG5aMC		MadLoop
	<a href="https://arxiv.org/abs/1405.0301">[arXiv:1405.0301]</a>		<a href="https://arxiv.org/abs/1103.0621">[arXiv:1103.0621]</a>
$d\bar{d} \rightarrow ZZ$	7 $\mu\text{s}$	$\times 10^2$ →	0.6 ms
$d\bar{d} \rightarrow ZZg$	35 $\mu\text{s}$	$\times 10^3$ →	38 ms
$d\bar{d} \rightarrow ZZgg$	220 $\mu\text{s}$	$\times 10^4$ →	1200 ms

**Ways beyond limitations of current adaptive sampling?**

# ML-assisted integration

Param transform equivalent to binning phase space for equal probability per bin. VEGAS = 1D projection, FOAM ~ 5D:

Bendavid [arXiv:1707.00028] builds on ideas for BDT as multidimension reweighter, extends with DNN:



Algorithm	# of Func. Evals	$\sigma_w / \langle w \rangle$	$\sigma_I / I$ (2e6 add. evts)
VEGAS	300,000	2.820	$\pm 2.0 \times 10^{-3}$
Foam	3,855,289	0.319	$\pm 2.3 \times 10^{-4}$
Generative BDT	300,000	0.082	$\pm 5.8 \times 10^{-5}$
Generative BDT (staged)	300,000	0.077	$\pm 5.4 \times 10^{-5}$
Generative DNN	294,912	0.083	$\pm 5.9 \times 10^{-5}$
Generative DNN (staged)	294,912	0.030	$\pm 2.1 \times 10^{-5}$

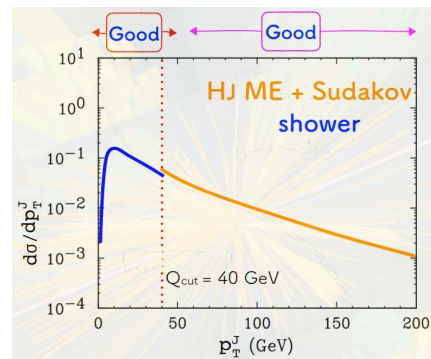
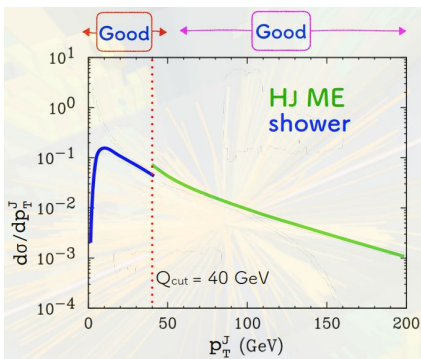
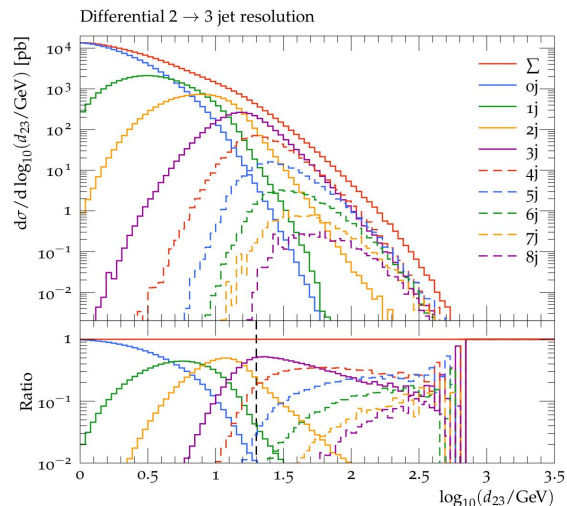
**Exciting possibility for integration with ME generators... via ML workflows?**

# Matching and merging

Last 10-20 years = avoiding double-counting, in a theoretically consistent way

ME/parton shower connection:  $N+1$  ME sample overlaps in phase space with  $N+PS$ . Slice phase-space between MEs and PS emissions to avoid conflict  
 → new freedoms: *merging scale*, *Sudakov ambiguities*

Shower needed in softer resummation regime. Preserve logarithmic accuracy of PS by e.g. cluster-based scale recalculation ⇒ CKKW, MiNLO

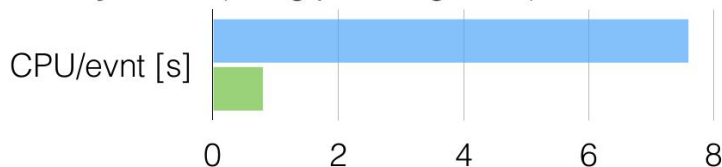


(Keith Hamilton)



# CPU implications of multileg match/merge

**W+0-2j@NLO** (using pre-integration)



Sherpa match/merge with CKKW:  
clustering to determine merging  
scale dominates CPU budget

⇒ improve x4 via an approximate  
“pilot scale” calculation (same as MG5)  
retrospectively recalcd & weighted

Sherpa LO/NLO merging performance:

## LO merging using COMIX

Process $W^-+$	0j	$\leq 1j$	$\leq 2j$	$\leq 3j$	$\leq 4j$
RAM Usage	39 MB	44 MB	49 MB	64 MB	173 MB
Initialization time	<1s	<1s	3s	22s	7m 7s
Startup time	<1s	<1s	<1s	<1s	2s
Integration time	25s	3m 19s	34m 8s	3h 12m	2d 17h
10k weighted evts	3m 24s	3m 51s	4m 2s	4m 4s	4m 21s
10k unweighed evts	3m 20s	4m 39s	11m 47s	35m 54s	4h 3m

## NLO merging using AMEGIC+BLACKHAT/COMIX (S/H-events)

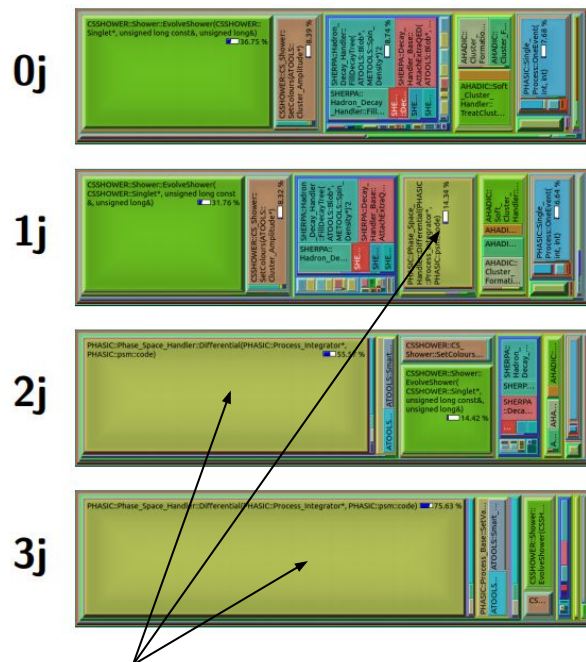
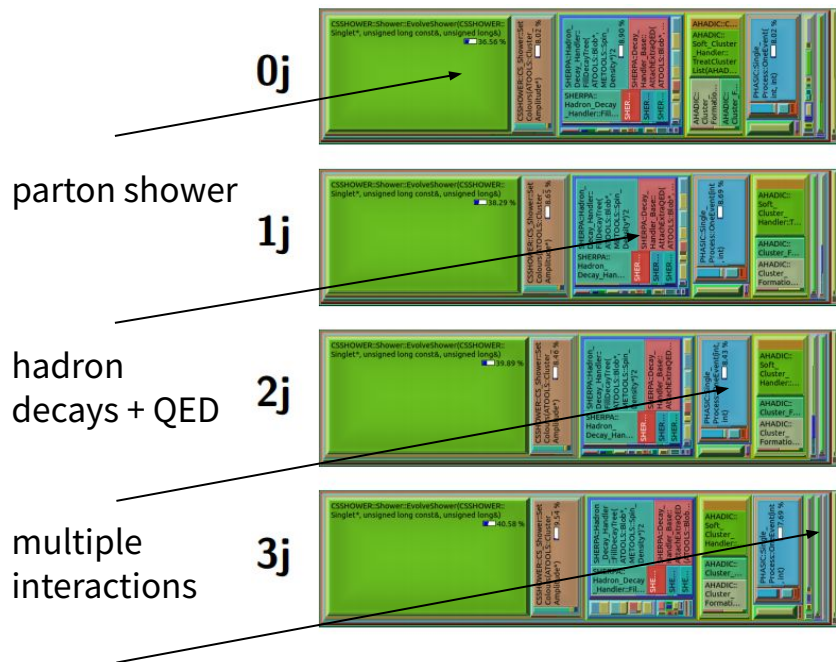
Process $W^-+$	0j	$\leq 1j$	$\leq 2j$
RAM Usage	51 MB	112 MB	572 MB
Initialization time	1s	20s	4m 6s
Startup time	<1s	2s	18s
Integration time	20m 48s	4h 45m	5d 23h
10k weighted evts	3m 58s	4m 38s	6m 48s
10k unweighted evts	4m 14s	4h 8m	24h 54m

# Origins of match/merge CPU (LO)

$W + < 4j$ :

Sherpa weighted

Sherpa unweighted



matrix element!

(Marek Schoenherr, Stefan Hoeche)

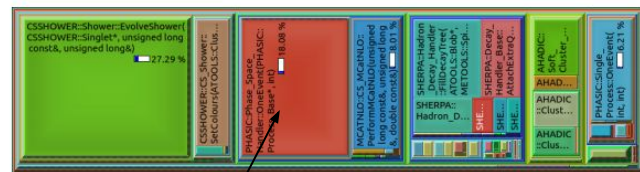
matrix elements & CKKW clustering

# Origins of match/merge CPU (NLO)

$W + \leq 2j$ : Sherpa weighted

Sherpa unweighted

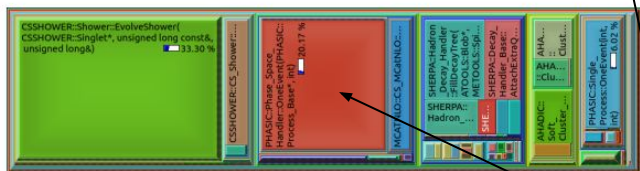
0j



1j



2j



(Marek Schoenherr, Stefan Hoeche)

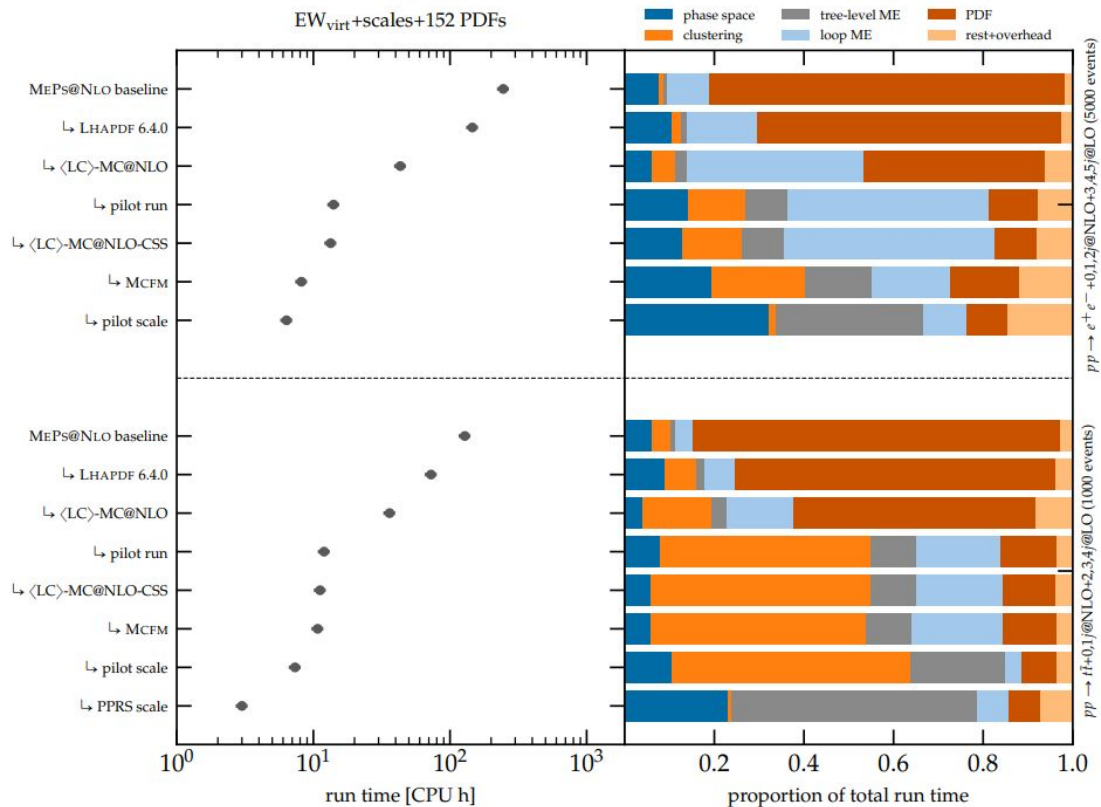
NLO matrix elements & matching (CKKW)

# Optimising Sherpa workflows

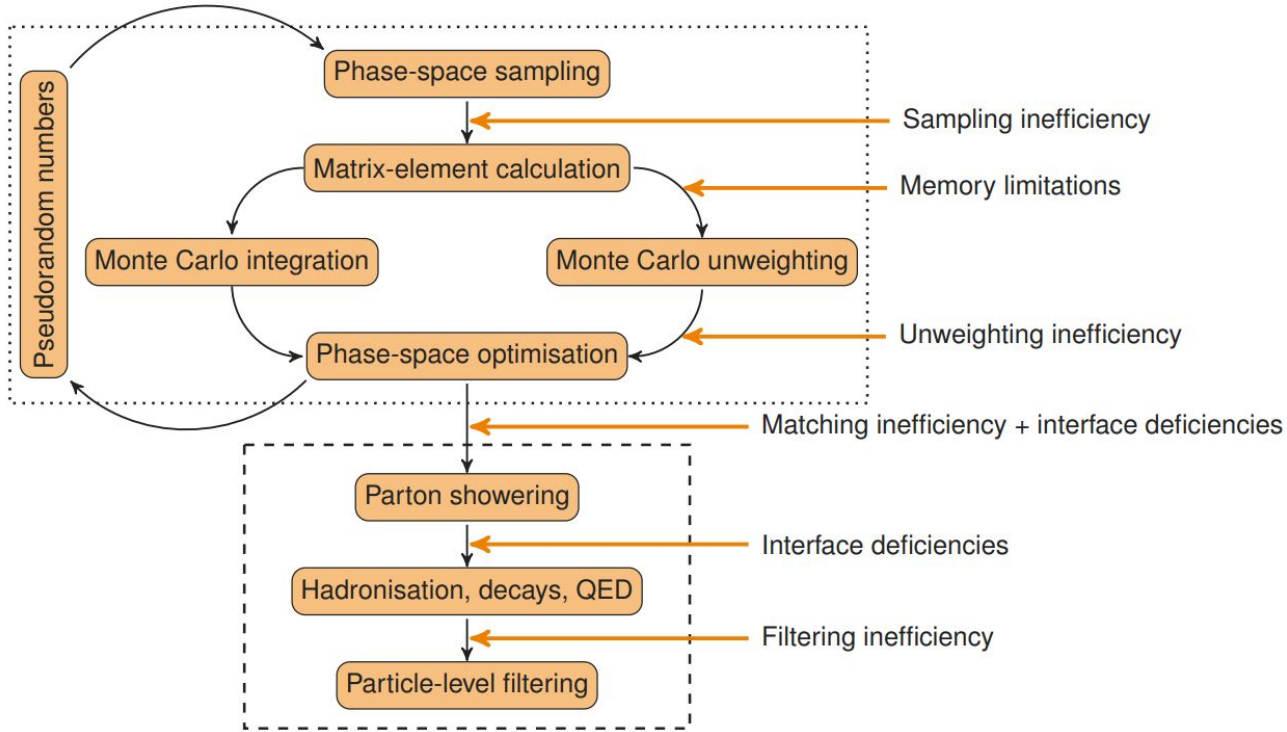
UK SWIFT-HEP computational efficiency project focused on Sherpa  $V$ +jets and  $t\bar{t}$ bar

Bottlenecks in LHAPDF, use of PDF multiweights, scale calculations, etc.

Being smarter about what's calculated pre-unweighting  
→ 40x speed improvements!



# More bottlenecks



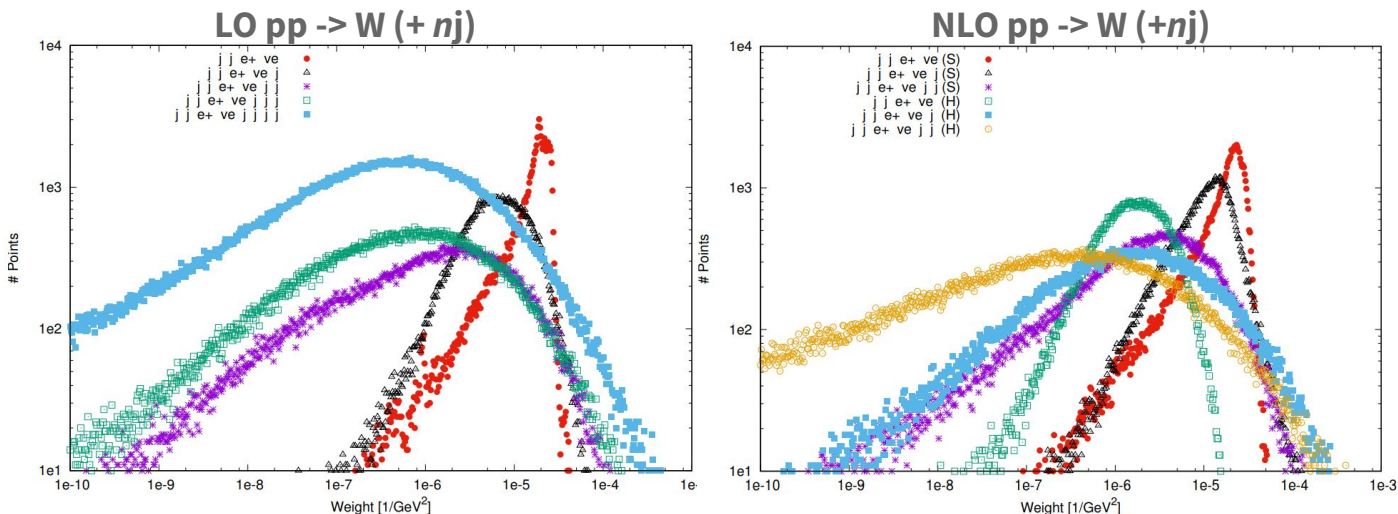
(Chris Gutschow)

Holistic project to improve.  
Community tools/formats like LHE and HepMC are major issues

⇒ release of HDF5 parallel-I/O format, LHEH5 [arXiv:2309.13154]

# Unweighting – the symptom

Origin of *weights* in shortcomings of phase-space proposal density functions: weight variances explode as multiplicities increase



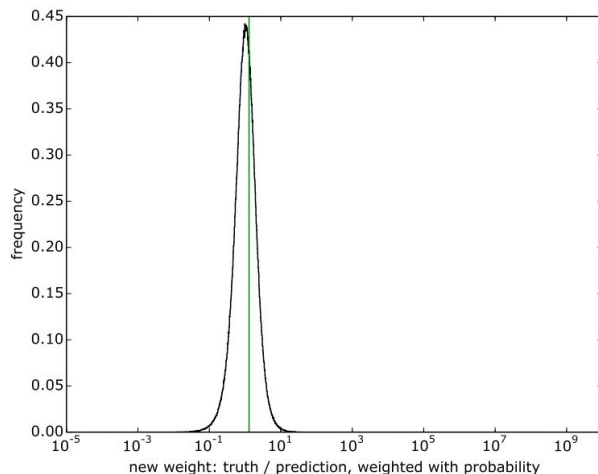
Problems from misplaced divergences, late discoveries  $\rightarrow$  spikes, and -ve weights  
MC@NLO formalism  $\sim$  25% -ve weight fraction  $\Rightarrow$  factor 2 in stat loss

# Unweighting — ML phase-space functions?

Unweighting efficiency relates to the inefficiency of sampling the phase space in the first place: perfect integration proposals would give uniform weights, with no wastage. Importance sampling proposal density too complicated?

Many studies, largely from Sherpa collaboration, on using machine-learning methods for better proposals.

Current status: ML proposals can work well at low multiplicities... but still break down for the expensive events where improvement's needed!



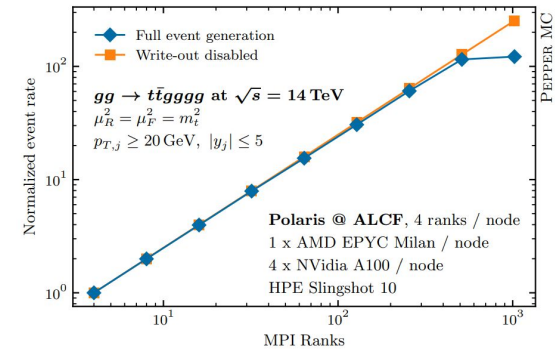
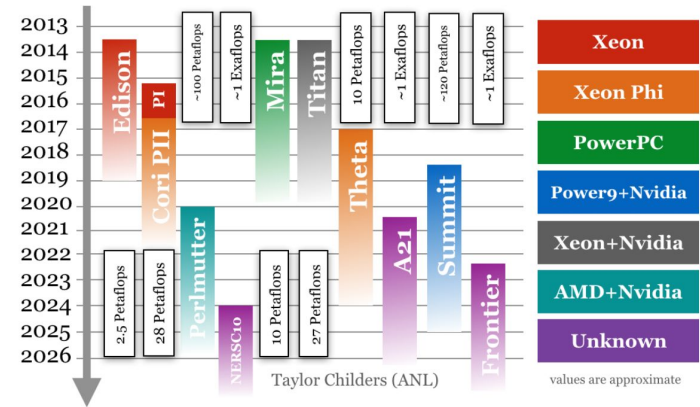
# New compute architectures

Many ~2000-2010 assumptions being challenged: for LHC production purposes, which dominate resource use, single-core Grid assumed. ATLAS → “on-the-fly” for Run2!

All change, at least for MEs: availability of (US) HPC facilities... push on MPI, vectorisation, GPUs. Can evgen ME production use this?

MG5 LO started ~2009: little interest, recent HSF action. Sherpa Chili+Pepper GPU LO gen production release 2023 [arXiv:2311.06198]

Evolution of US DOE HPC Systems





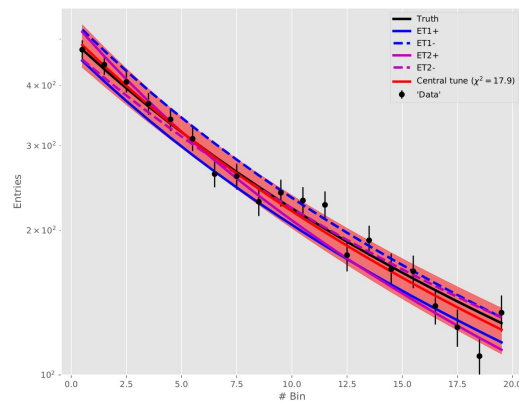
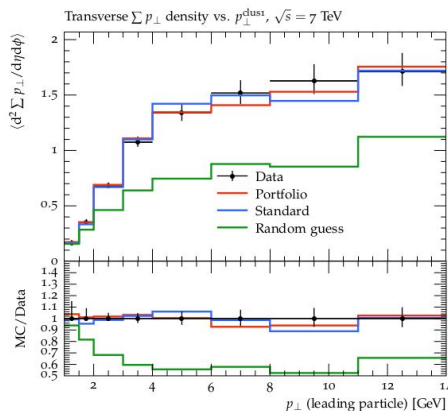
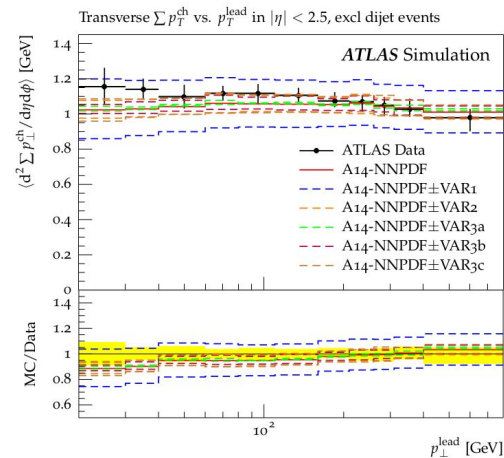
# Generator tuning, too

ME and MEPS matching is the sexy stuff!

**Exclusive event generation is useless without “dirty details”:** from partons to hadrons, hadronisation, MPI...

These models have tenuous physical underpinning: need tuning to data (cf. PDFs!)

Main tuning machinery is Professor interpolation fitter: “approximate computation” but just polynomials!  
Recent extensions: portfolio opt, Pade rationals, NNs. *Added value?*



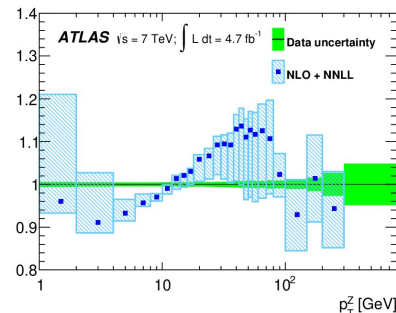
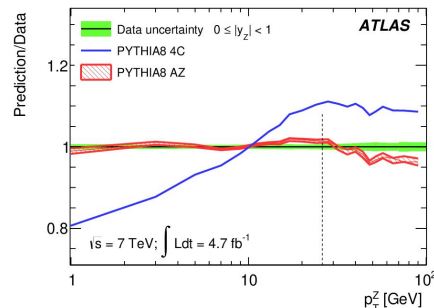
# Low-hanging fruit?

As well as high-tech solutions, pragmatism helps: *how little can we get away with?*

Two modes of MC usage: theory-test, and “data fitting”. Often just need the latter.

**Formal precision != accuracy!**

Shower “flexibility” sometimes useful...



**Reweighting of LO?!** Differential  $K$ -factors to NNLO/NLO, beyond-leading order/colour showers. Could use BDT/DNN methods. *Actual physics impact?*

**(Partonic) event-sharing:** commonly accessible repository of matrix-element event samples between expts (and pheno), allowing different showers & variations

# Sociology

Have to mention the social factors:  
perverse incentives lurk in the background!

MC developers are theorists. Technical evgen logistics deeply immersed, but incompatible with theory funding and career paths. **On HSF radar: experiments supplying MC effort directly?**

“Other” programs in the HEP MC ecosystem also performance critical: notably **HepMC** and **LHAPDF**. Some action on the latter through performance-oriented funded projects

## Contents

1	Introduction	2
2	Software and Computing Challenges	5
3	Programme of Work	11
3.1	Physics Generators	11
3.2	Detector Simulation	15
3.3	Software Trigger and Event Reconstruction	23
3.4	Data Analysis and Interpretation	27
3.5	Machine Learning	31
3.6	Data Organisation, Management and Access	36
3.7	Facilities and Distributed Computing	41
3.8	Data-Flow Processing Framework	44
3.9	Conditions Data	47
3.10	Visualisation	50
3.11	Software Development, Deployment, Validation and Verification	53
3.12	Data and Software Preservation	57
3.13	Security	60
4	Training and Careers	65
4.1	Training Challenges	65
4.2	Possible Directions for Training	66
4.3	Career Support and Recognition	68
5	Conclusions	68

# Summary

MC event generation has undergone a sea-change during LHC 1,2

Huge leaps in formal accuracy of fully exclusive predictions: (N)NLO+PS  
*But also huge leaps in CPU demands!*

Not yet a super-active area for innovative data-sci work! Familiarity, reward, ...  
⇒ Main MC/experiment interactions on physics, not tech  
⇒ Requires expertise in “both worlds”: hard to find!

Experiments also need to think hard about what they really *need*.  
*Approximate methods may be more appropriate, and ML ↷ better approximations*

HSF: <https://arxiv.org/abs/1712.06982>

MCnet++: <https://arxiv.org/abs/1902.01674>

# Backup

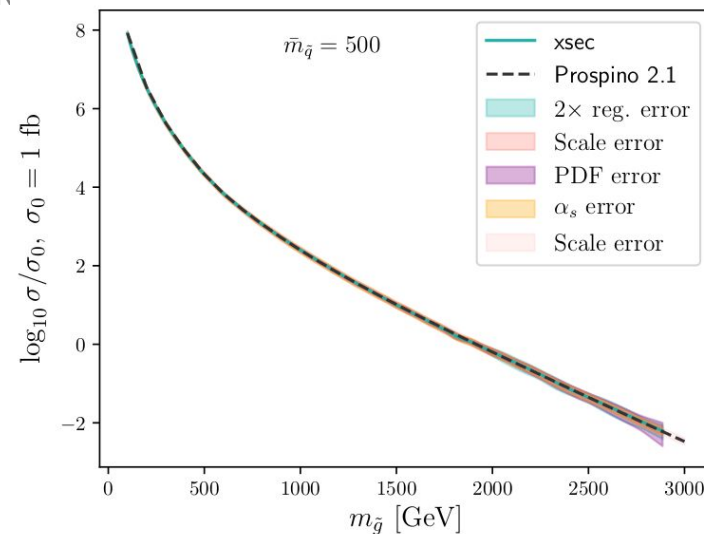
# Generator systematics

Perturbative QCD  $\rightarrow$  truncation of perturbative expansion in  $\alpha_s$ . Unphysical dependence on scale  $\mu_R$  — reduced but not eliminated by higher orders

Also PDF uncertainties: nucleon momentum structure, with fit errors. Trivial to reweight at LO, complex at NLO+.

Uncertainties as weights or gen-specific coeffs

(Relative) systematics may be easier to learn than differential cross-sections. Prelim SUSY total-xsec DGP interpolation incl systematics [Raklev, v.d.Abeele]:



# Potential ML gains in matching

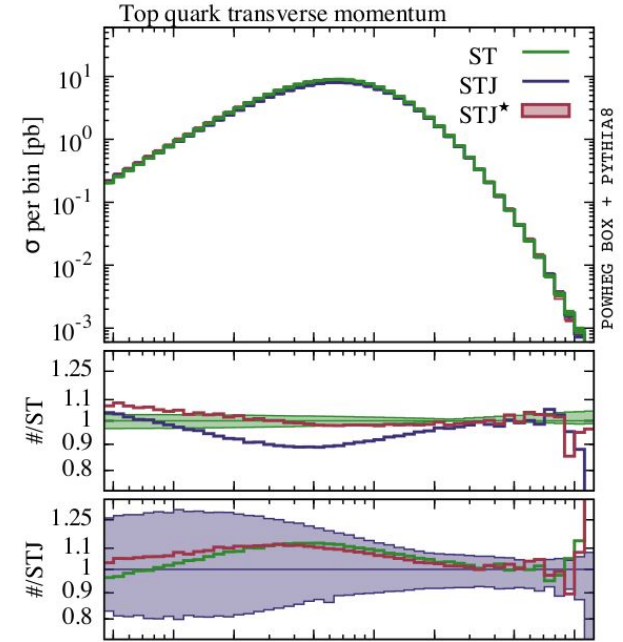
Novel use of neural nets in arXiv:1805.09855, to fit unknown higher-order resummation terms.

Calculation is NLO-matched single-top + jet (STJ) in the POWHEG-MiNLO formalism: enhance fixed-order calculation with matched NLL Sudakov form factor:

$$d\sigma_{\mathcal{M}} = \Delta(y_{12}) \left[ d\sigma_{\text{NLO}}^{\text{STJ}} - \Delta(y_{12})|_{\bar{\alpha}_S} d\sigma_{\text{LO}}^{\text{STJ}} \right]$$

But this spoils the NLO accuracy of ST! Fix at *NNLL*...

Fit  $A_2$  with NN ML-based tuning of degrees of freedom; test universality at 8 TeV!



$$\ln \delta\Delta(y_{12}) = -2 \int_{y_{12}}^{Q_{bt}^2} \frac{dq^2}{q^2} \bar{\alpha}_S^2 \mathcal{A}_2(\Phi) \ln \frac{Q_{bt}^2}{q^2} \quad \mathcal{L} = \sum_{i=1}^{N_{\text{bins}}} \left[ \sum_{j=1}^N w_{i,j}^{\text{ST}} - \sum_{k=1}^{N'} w_{i,k}^{\text{STJ}} e^{\tilde{\mathcal{A}}_2(\Phi_i)} \mathcal{G}_2(\lambda) \right]^2$$

# Potential ML gains in matching

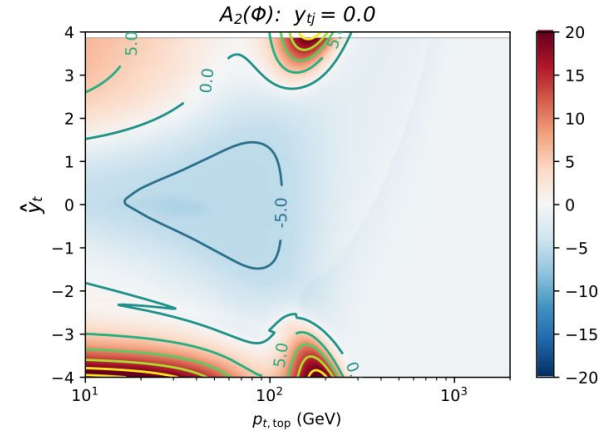
Novel use of neural nets in arXiv:1805.09855, to fit unknown higher-order resummation terms.

Calculation is NLO-matched single-top + jet (STJ) in the POWHEG-MiNLO formalism: enhance fixed-order calculation with matched NLL Sudakov form factor:

$$d\sigma_{\mathcal{M}} = \Delta(y_{12}) \left[ d\sigma_{\text{NLO}}^{\text{STJ}} - \Delta(y_{12})|_{\bar{\alpha}_S} d\sigma_{\text{LO}}^{\text{STJ}} \right]$$

But this spoils the NLO accuracy of ST! Fix at NNLL...

Fit  $A_2$  with NN ML-based tuning of degrees of freedom; test universality at 8 TeV!



$$\ln \delta\Delta(y_{12}) = -2 \int_{y_{12}}^{Q_{bt}^2} \frac{dq^2}{q^2} \bar{\alpha}_S^2 \mathcal{A}_2(\Phi) \ln \frac{Q_{bt}^2}{q^2} \quad \mathcal{L} = \sum_{i=1}^{N_{\text{bins}}} \left[ \sum_{j=1}^N w_{i,j}^{\text{ST}} - \sum_{k=1}^{N'} w_{i,k}^{\text{STJ}} e^{\tilde{\mathcal{A}}_2(\Phi_i)} \mathcal{G}_2(\lambda) \right]^2 \quad 4$$

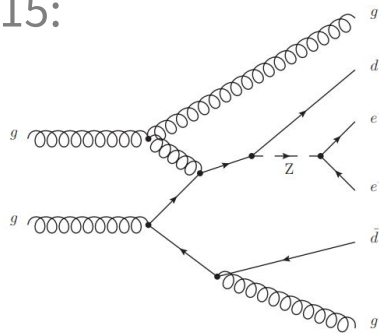


# Unweighting — more phase-space ML?

Unweighting efficiency relates to the inefficiency of sampling the phase space in the first place: perfect integration proposals would give uniform weights, with no wastage. Importance sampling proposal density too complicated?

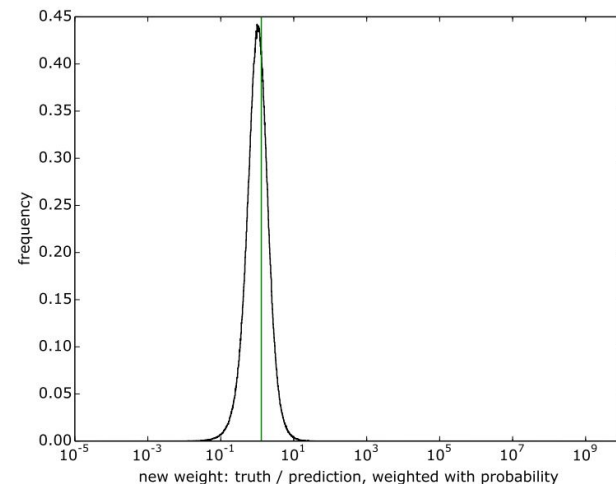
Krause & Siegert MSc study, 2015:

parametrise  $Z+jjgg$  weight function in  $3n$  features



Try 3 bases, with “LHC physics” momenta best.

750x speed-up, but x3 weight mismodelling



# Why is ATLAS MC more expensive than CMS?

Numbers aren't quite fair: CMS 1% vs ATLAS 20%...  
on a very different balance of samples. But still:

**Sherpa monolithic mode vs MadGraph factorised.**

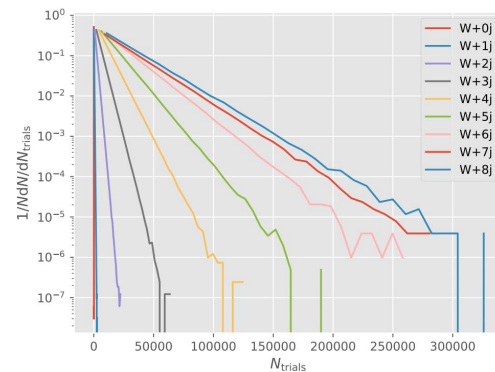
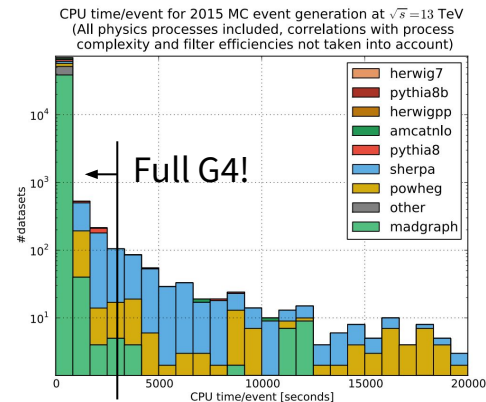
Convenient, but at high multiplicities, runs ~as  
slow as the (lowest rate) highest multiplicity!

*See next slide...*

NB. Not all about MEs: flavour *filtering* is also costly.

Need to account for all sources of  $b$  and  $c$

Reuse possible? Multiple streams? In-MC hooks?



# Making Sherpa HPC-friendly

Factorise Sherpa mode parton multiplicities

Split to HPC-friendly new ME interchange format based on HDF5 vs LHE. Incl some Sherpa-specifics

LO ME scales worse than shower/matching, etc. — ratio plateaus for > 3 jets. Hybrid gen strategy: use HPC resources → high-mult MEs?

From zero to 8-jet analysis with 100M events in 25 mins!! Also viable on smaller scale concurrency?

