Computation issues in Monte Carlo event generation

Andy Buckley, University of Glasgow

ACAT 2019
Saas Fee, 15 March 2019

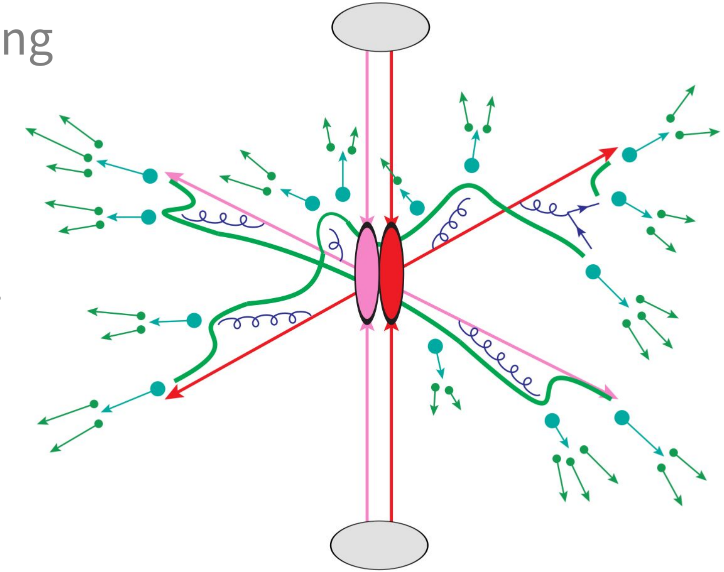
LHC event generation

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 → ~factorial explosion!*

Many issues in summary from HSF MC workshop: https://indico.cern.ch/event/751693/timetable/

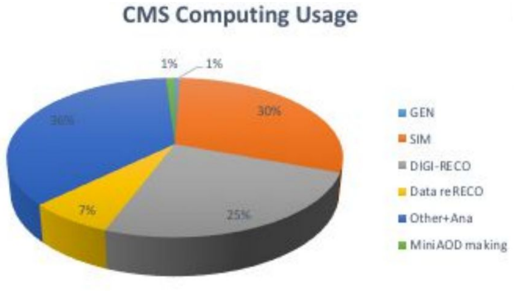


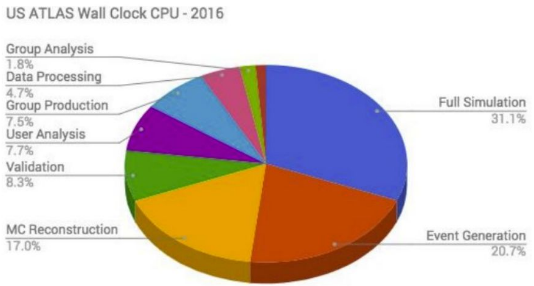
(Stefan Prestel)

Disclaimer: This is a top-down view: plenty of more “deep” QCD experts here. Many thanks to Stefan Hoeche, Holger Schulz, Keith Hamilton, Marek Schoenherr, Frank Siegert, Josh Macfayden, …

Evgen cost, present and future

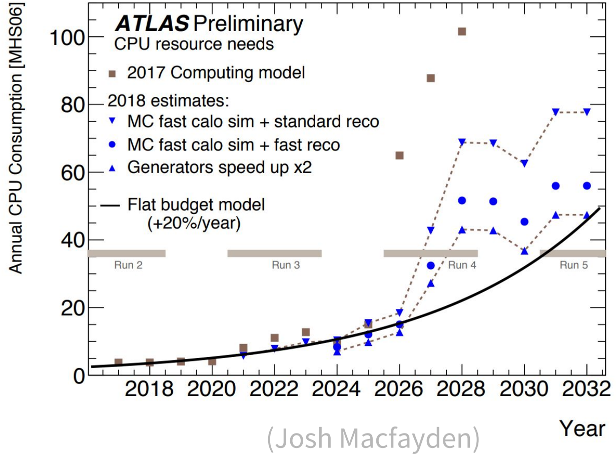
Cf. 2000, only leading-order shower/hadronisation MC available! (Alpgen, MadEvent, aMC@NLO in 2002, Sherpa & Powheg later). ~Trivial…





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!!

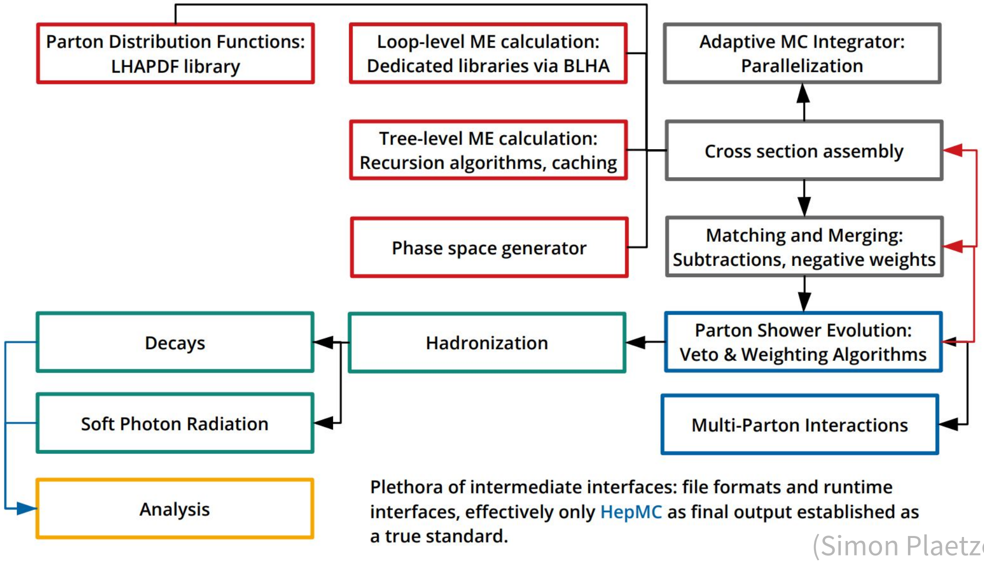


(Josh Macfayden)

SHG MC generator codes

Leading SHGs now all C++ based. Object orientation vs. performance?
Complex code (except Py8) → dev challenge

Typical modern SHG generator structure:



(Simon Plaetzer)

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

Event generation core: ME and phase space

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

$$\mathrm{d}\sigma \sim \mathrm{d}\sigma_{\mathrm{hard}}(Q) \times \mathrm{PS}(Q \to \mu) \times \mathrm{Had}(\mu \to \Lambda) \times \ldots$$

NB. factorisation is convenience, not reality!

$$\mathrm{d}\sigma_{\mathrm{hard}}(p_1, \ldots, p_n|Q) \sim |\mathcal{M}(p_1, \ldots, p_n)|^2 \mathrm{dPSP}(p_1, \ldots, p_n|Q)$$

$$\mathrm{d}\sigma = \mathrm{d}\sigma^{\mathrm{LO}} + \alpha_S(Q)\, \mathrm{d}\sigma^{\mathrm{NLO}}(Q) + \alpha_S^2(Q)\, \mathrm{d}\sigma^{\mathrm{NNLO}}(Q) + \ldots$$

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 → gridpacks, evgen embarassingly parallel

Phase space and loop integration

PS: $\int d^4p_1 \ldots d^4p_N \, |\mathcal{M}(\{p_i\})|^2 \, \delta p_1^2 \ldots \delta p_N^2 \delta^4 \left(p_1 + \ldots p_N\right)$ Loop: $\int d^dk_1 \ldots d^dk_{N_l} \, \frac{N(\{k_i\})}{\Pi_1^{N_d} D_i}$

Complexity of ME+PDF singularity structure makes both hard:

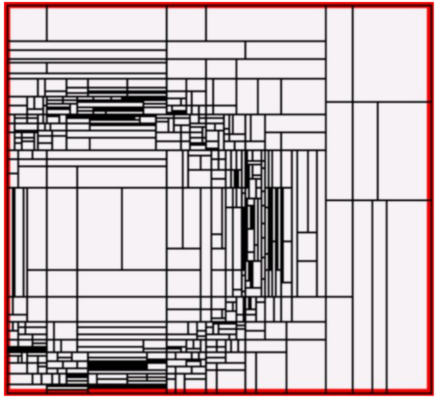
- **Loop** UV and IR singularities → large variance, poor convergence → *pre-generate*
- **PS** is the major scaling issue for higher order MC production, both in legs & loops:

(Valentin Hirschi — see talk)

	Tree-level MG5aMC [arXiv:1405.0301]		One-loop MadLoop [arXiv:1103.0621]
$d\bar{d} \to ZZ$	7 µs	x 10^2 →	0.6 ms
$d\bar{d} \to ZZg$	35 µs	x 10^3 →	38 ms
$d\bar{d} \to ZZgg$	220 µs	x 10^4 →	1200 ms

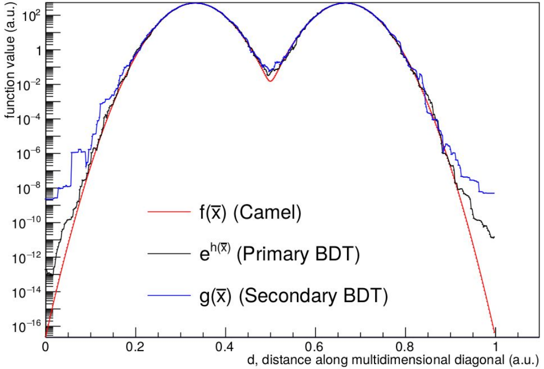
Ways beyond limitations of current adaptive sampling?

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:



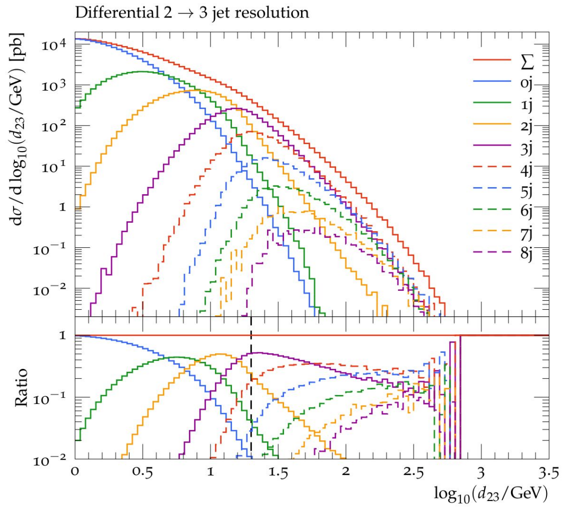
Algorithm	# of Func. Evals	$\sigma_w / < w >$	σ_I / I (2e6 add. evts)
VEGAS	300,000	2.820	$\pm 2.0 \times 10^{-3}$
Foam	3,855,289	0.319	$\pm 2.3 \times 10^{-4}$
Generative BDT	300,000	0.082	$\pm 5.8 \times 10^{-5}$
Generative BDT (staged)	300,000	0.077	$\pm 5.4 \times 10^{-5}$
Generative DNN	294,912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

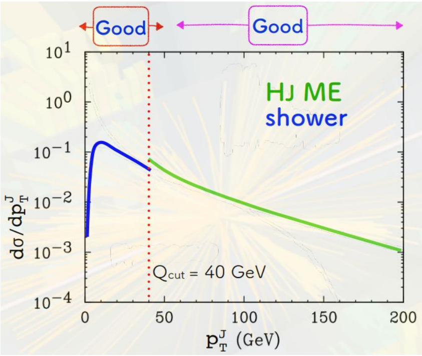
Matching and merging

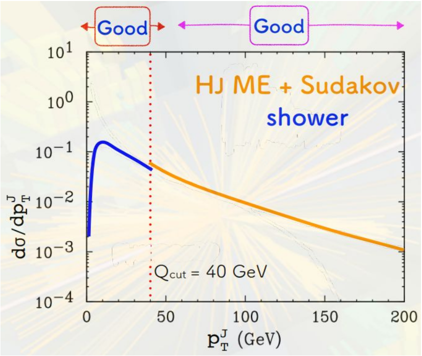
Last 10-20 years = avoiding double-counting, in a theoretically consistent way

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 recalc: CKKW, MiNLO.





(Keith Hamilton)

Potential ML gains in matching

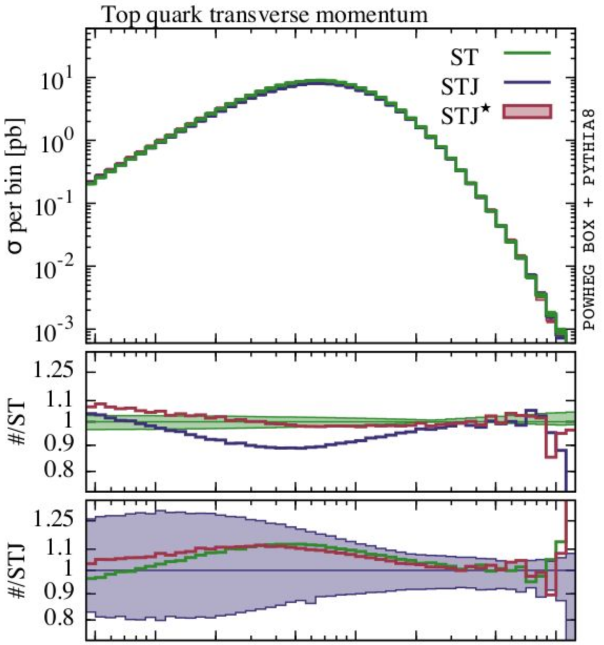
Novel use of neural nets in arXiv:1805.09855, to fit unknown higher-order resummation terms.

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

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

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

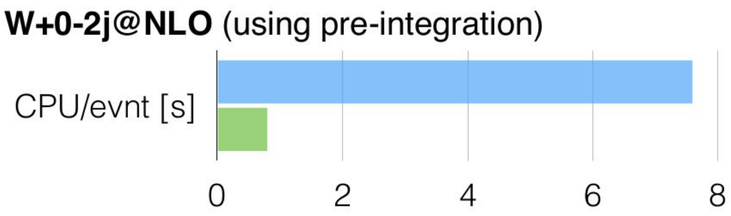
Fit A_2 with NN ML-based tuning of degrees of freedom; test universality at 8 TeV!



$$\ln \delta\Delta(y_{12}) = -2 \int_{y_{12}}^{Q^2_{bt}} \frac{dq^2}{q^2} \, \bar{\alpha}^2_{\rm S} \, \mathcal{A}_2(\Phi) \, \ln \frac{Q^2_{bt}}{q^2}$$

$$\mathcal{L} = \sum_{i=1}^{N_{\rm bins}} \left[\sum_{j=1}^{N} w^{\rm ST}_{i,j} - \sum_{k=1}^{N'} w^{\rm STJ}_{i,k} \, e^{\tilde{A}_2(\Phi_i)\, \mathcal{G}_2(\lambda)} \right]^2$$

CPU implications of multileg match/merge



Sherpa match/merge with CKKW: clustering to determine merging scale dominates CPU budget; improve x4 via same approx as MG5

Sherpa LO/NLO merging performance:

LO merging using COMIX

Process W^-+	0j	≤1j	≤2j	≤3j	≤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 unweighted evts	3m 20s	4m 39s	11m 47s	35m 54s	4h 3m

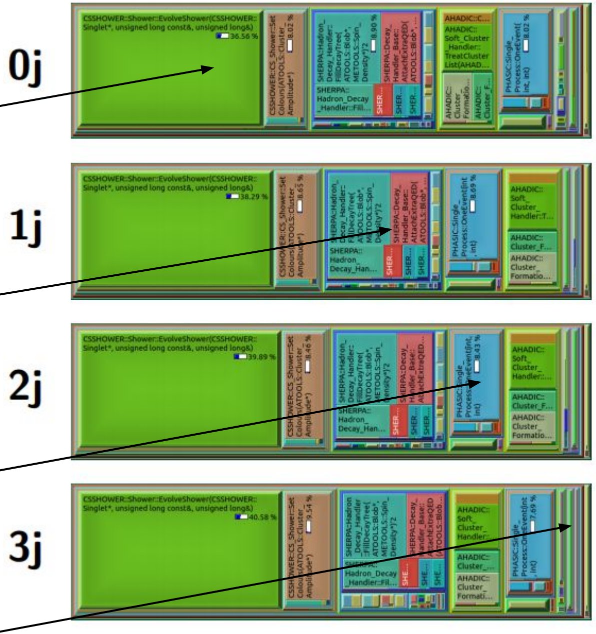
NLO merging using AMEGIC+BLACKHAT/COMIX (S/H-events)

Process W^-+	0j	≤1j	≤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



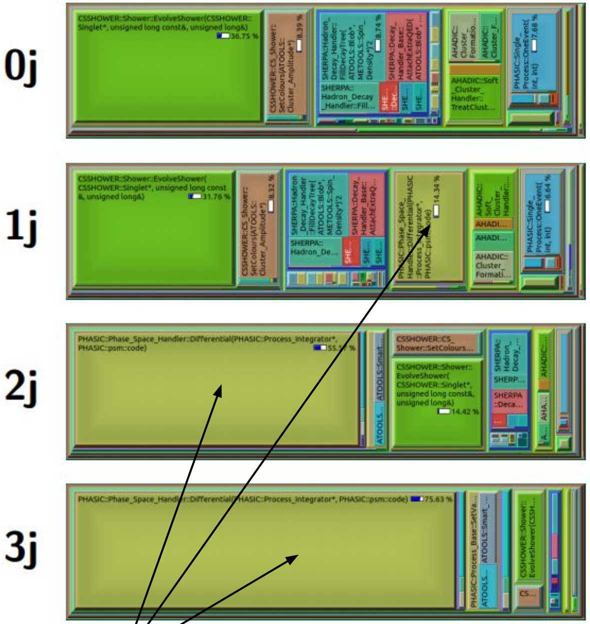
parton shower

hadron
decays + QED

multiple
interactions

matrix element!

Sherpa unweighted

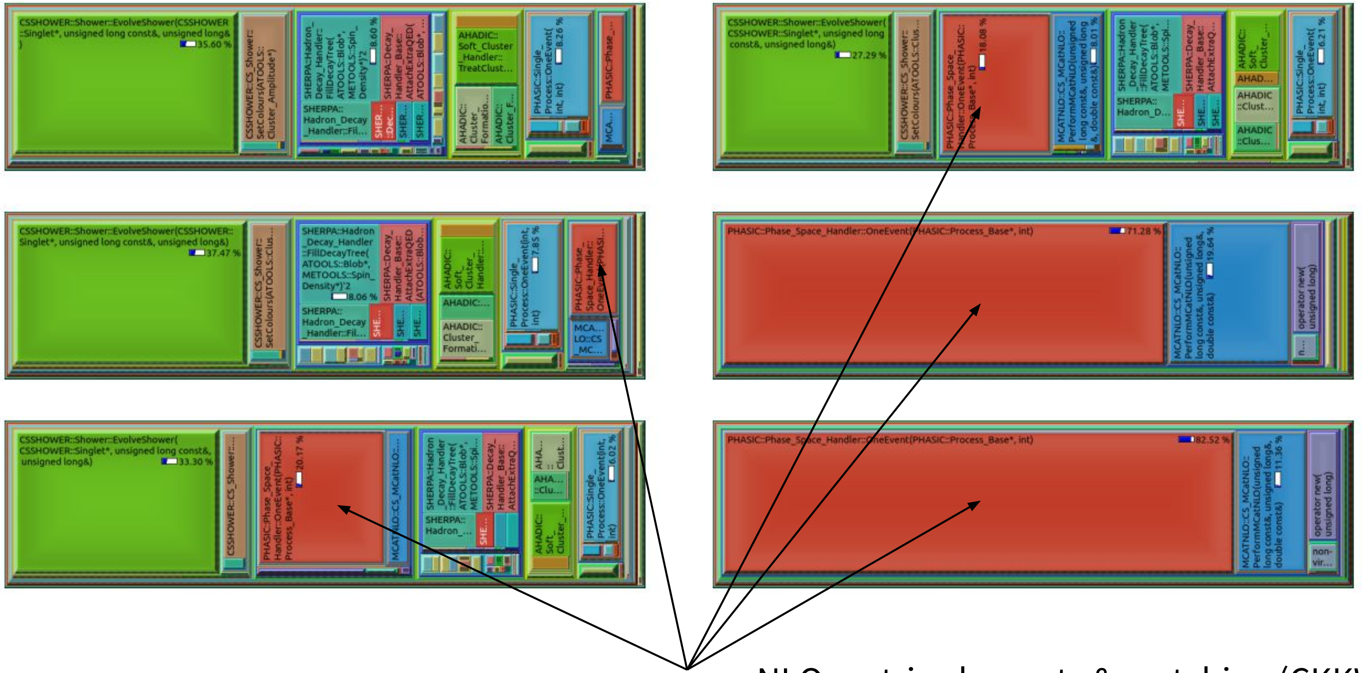


matrix elements & CKKW clustering

(Marek Schoenherr, Stefan Hoeche)

Origins of match/merge CPU (NLO)

W + < 2j: Sherpa weighted Sherpa unweighted



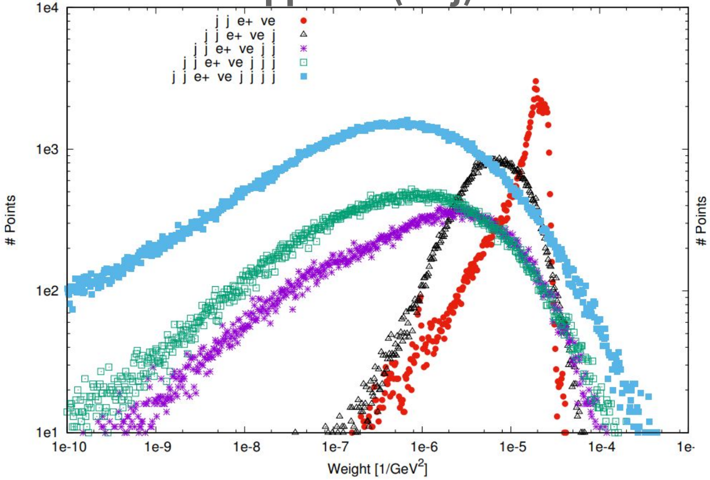
NLO matrix elements & matching (CKKW)

(Marek Schoenherr, Stefan Hoeche)

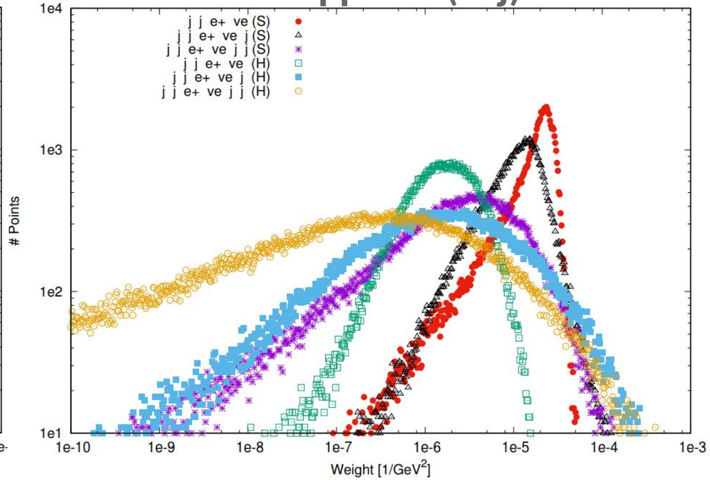
Unweighting — the symptom

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

LO pp -> W (+ *n*j)



NLO pp -> W (+*n*j)



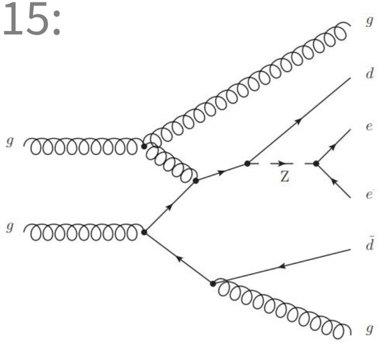
Problems from discovered “new max” → spikes, and -ve weights

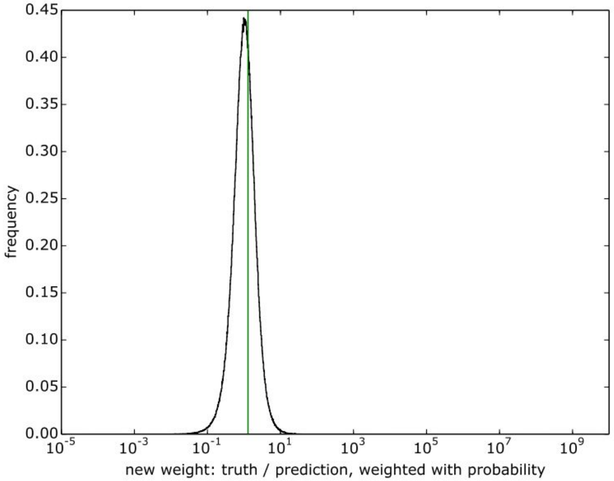
MC@NLO formalism ~ 25% -ve weight fraction ⇒ factor 2 in stat loss

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 3*n* features





Try 3 bases, with "LHC physics" momenta best.

750x speed-up, but x3 weight mismodelling

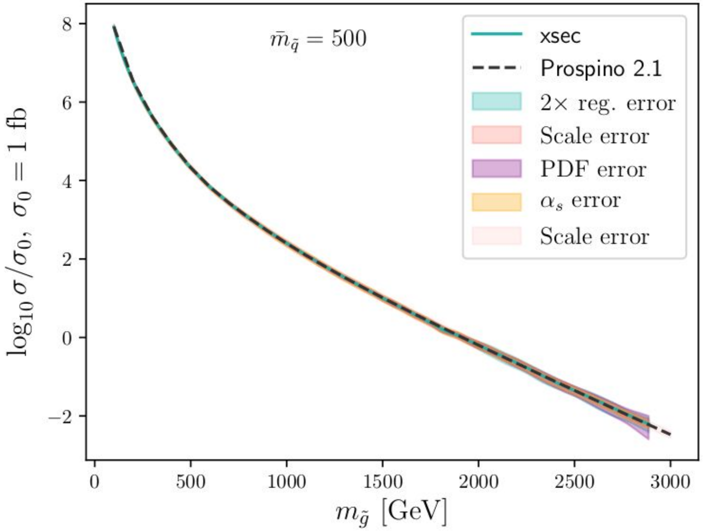
Generator systematics

Perturbative QCD → truncation of perturbative expansion in α_s. Unphysical dependence on scale μ_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]:

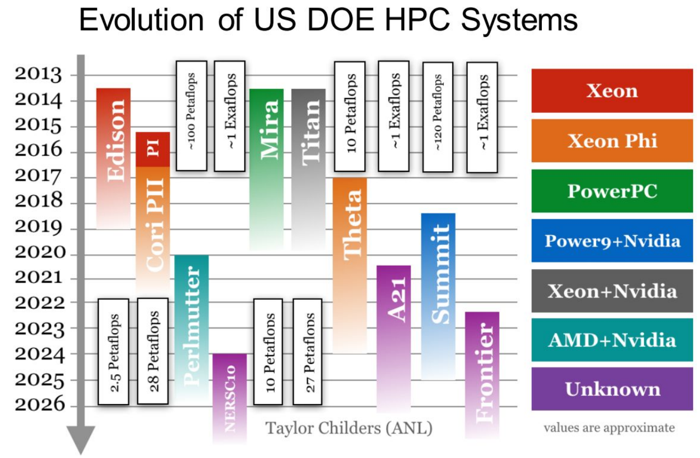


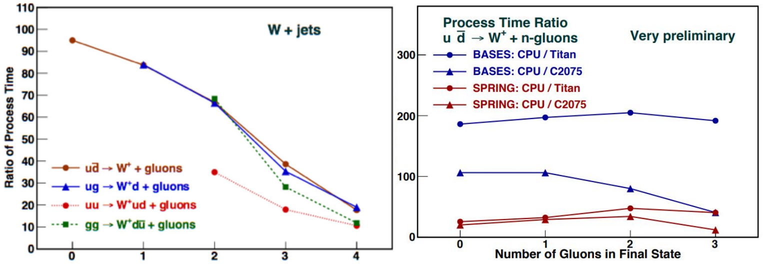
Modern compute architectures

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

GPU and similar architectures will dominate. Can evgen ME production use this? MG5 *LO* started ~2009, all SM in 2013. Little interest? *Automation via ML workflows?*

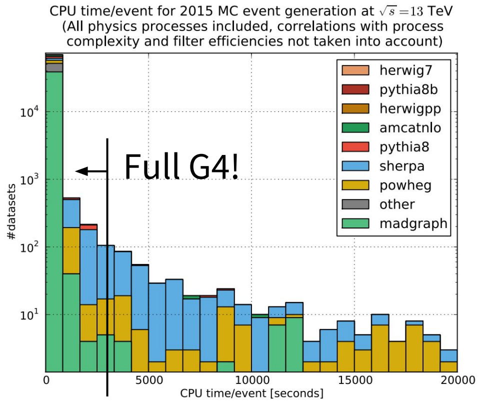




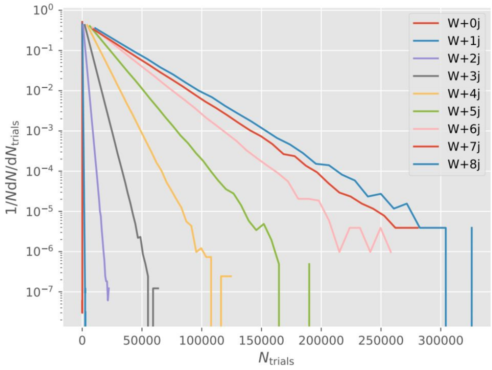
Why is ATLAS MC more expensive than CMS?

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?



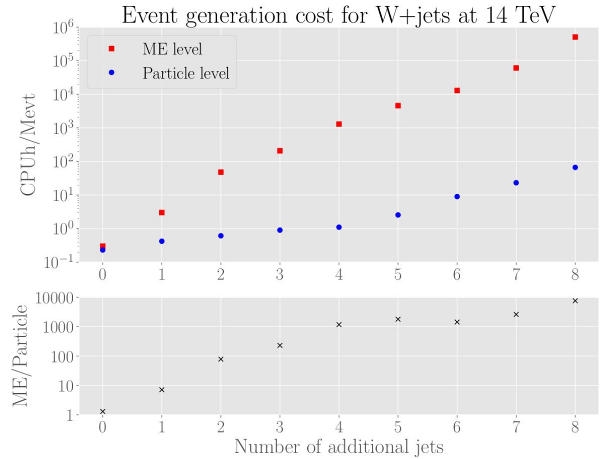
Making Sherpa HPC-friendly

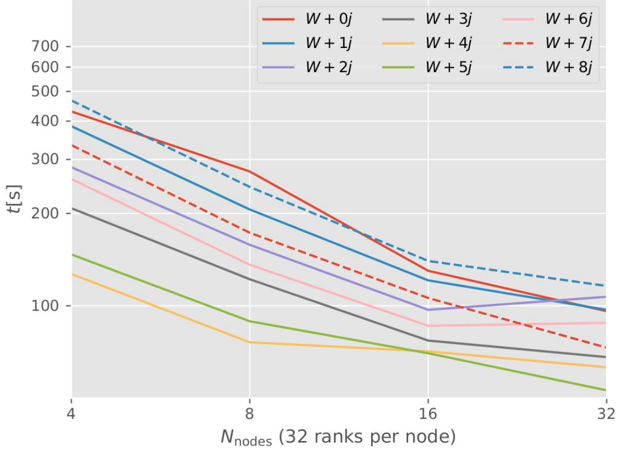
Factorise Sherpa mode parton multiplicities

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?





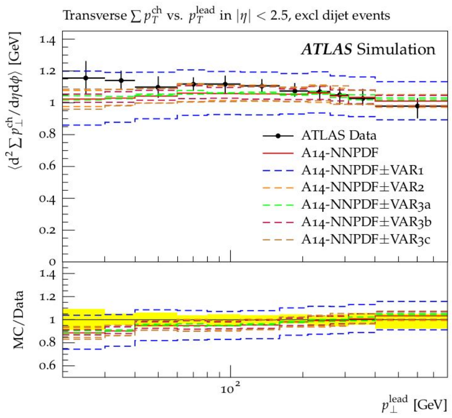
Generator tuning, too

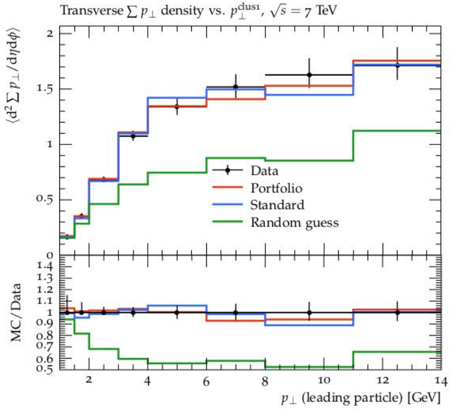
ME and MEPS matching is the sexy stuff!

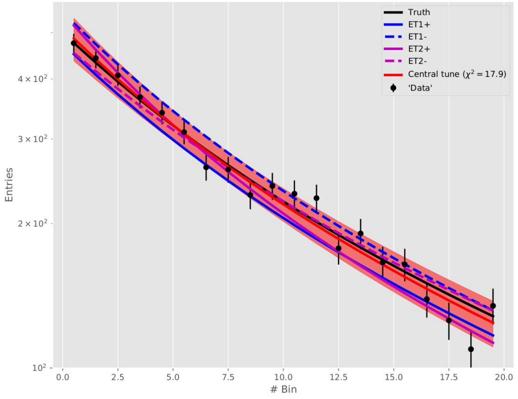
Exclusive event generation is useless without “dirty details”: from partons to hadrons, hadronisation, MPI…

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?*



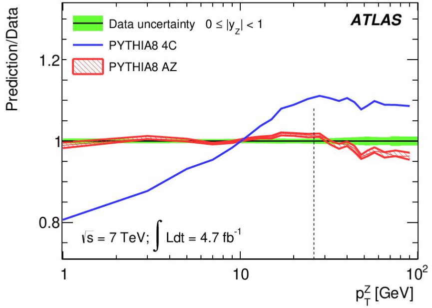


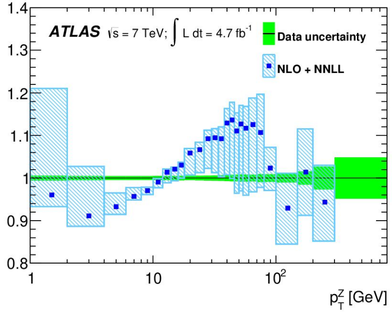


Low-hanging fruit?

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 to supply MC tech effort directly?**

“Other” programs in the HEP MC ecosystem also performance critical: notably **LHAPDF** parton densities. Re-engineered 2014 with speed gains in some use-modes… but *unfunded & ~dormant*

Contents

HEP Software Foundation

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"

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