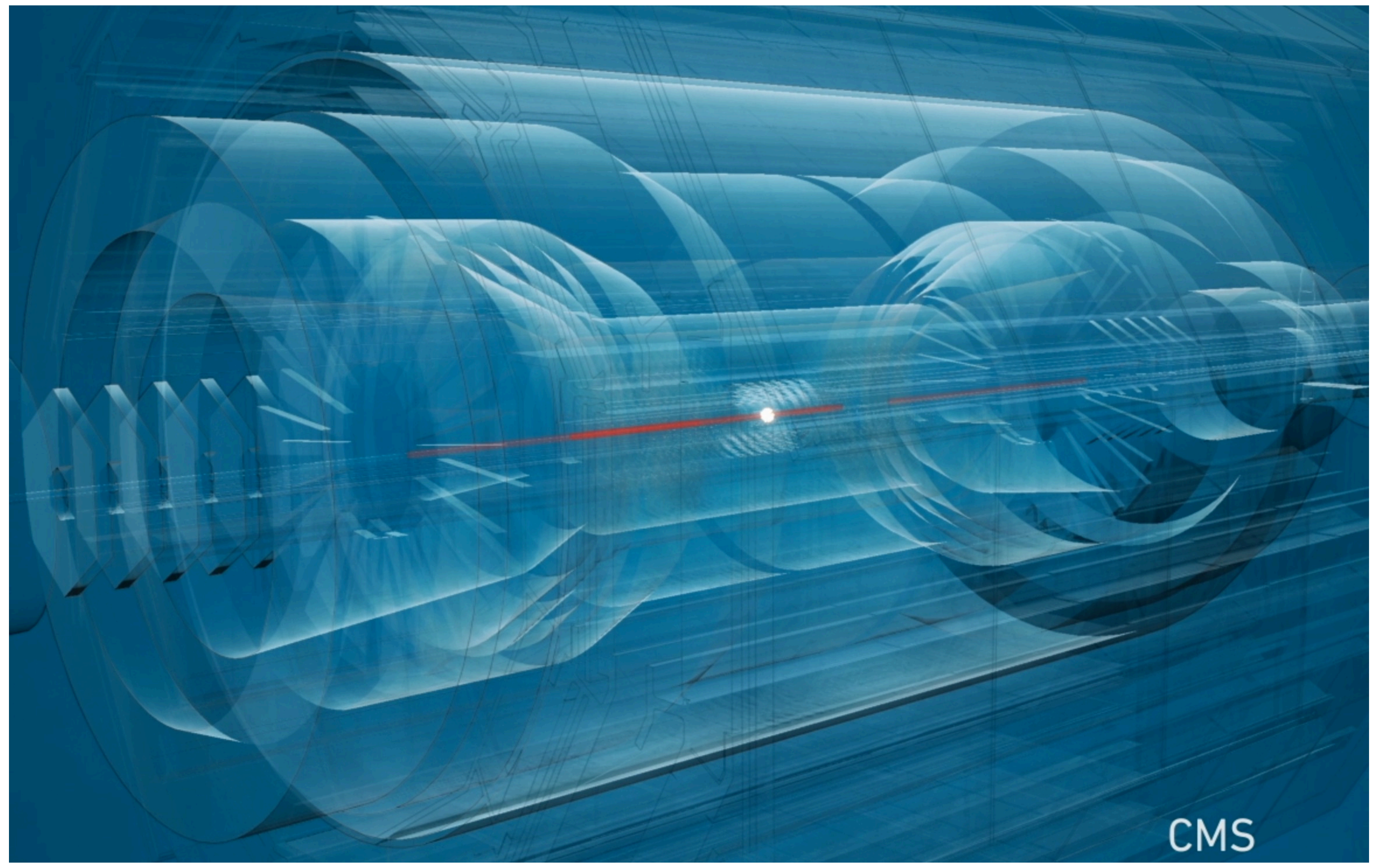


Experimental Needs

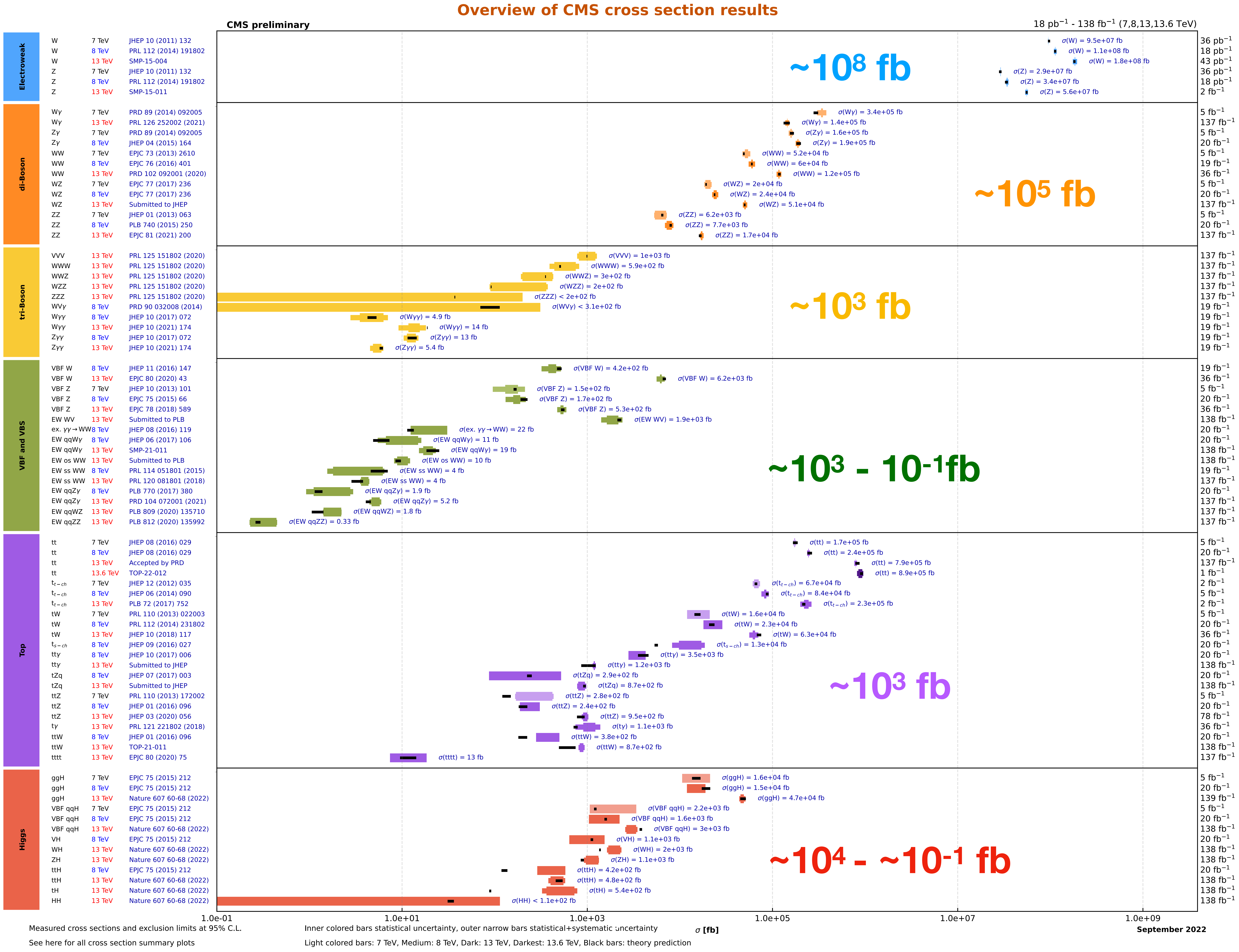
MC net Meeting

Saptaparna Bhattacharya

DESY, Northwestern University, Humboldt Fellow

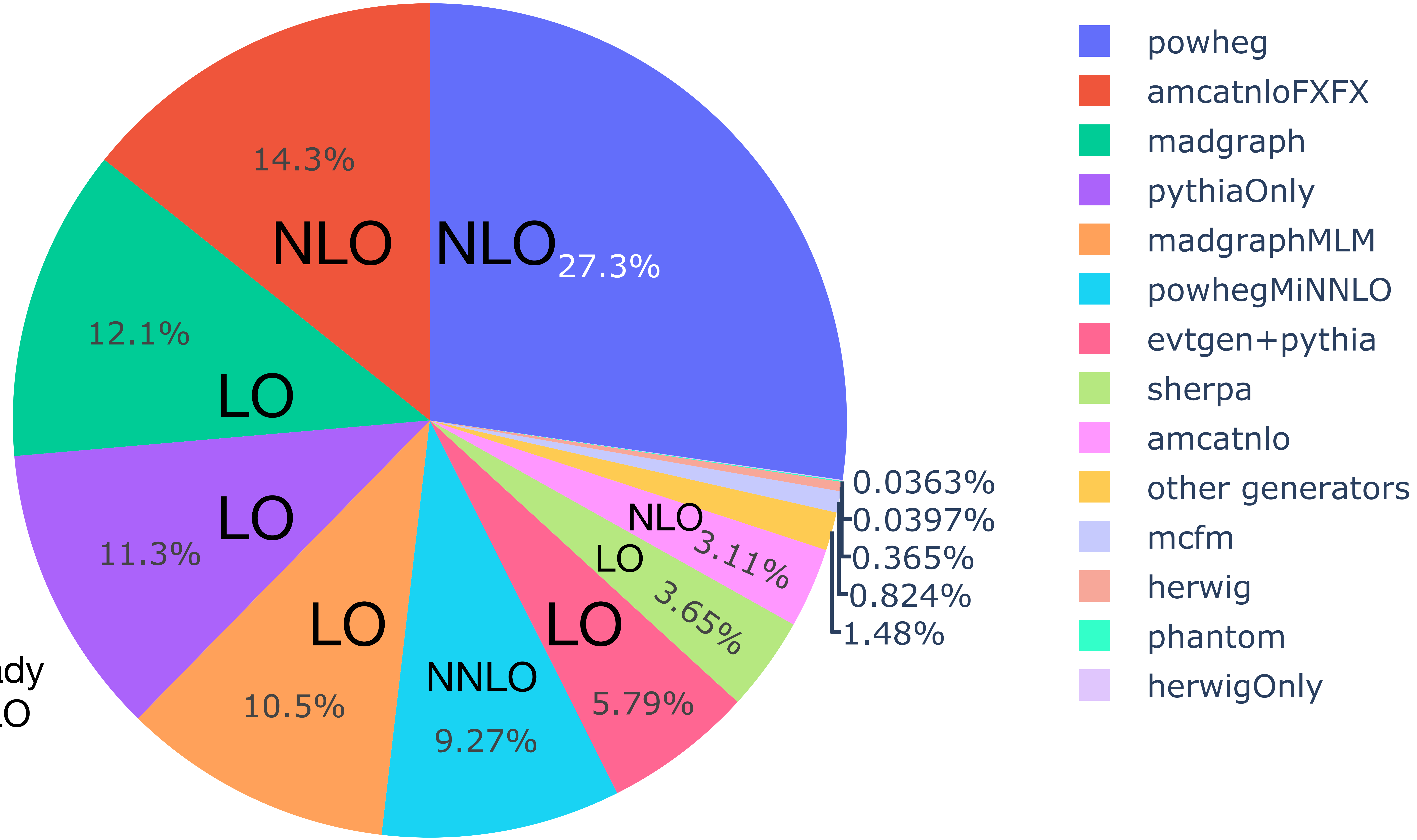
Spans 10 orders of magnitude

Spans 14 orders of magnitude when inelastic cross sections are taken into account



Generator usage split by events

- NNLO: 9.27%
- NLO: 44.71%
- LO: 46.03%



- Most of the events already generated at least at NLO accuracy
- Major consumers of Madgraph_aMC@NLO

Computing challenges ahead…

Computing Budgets

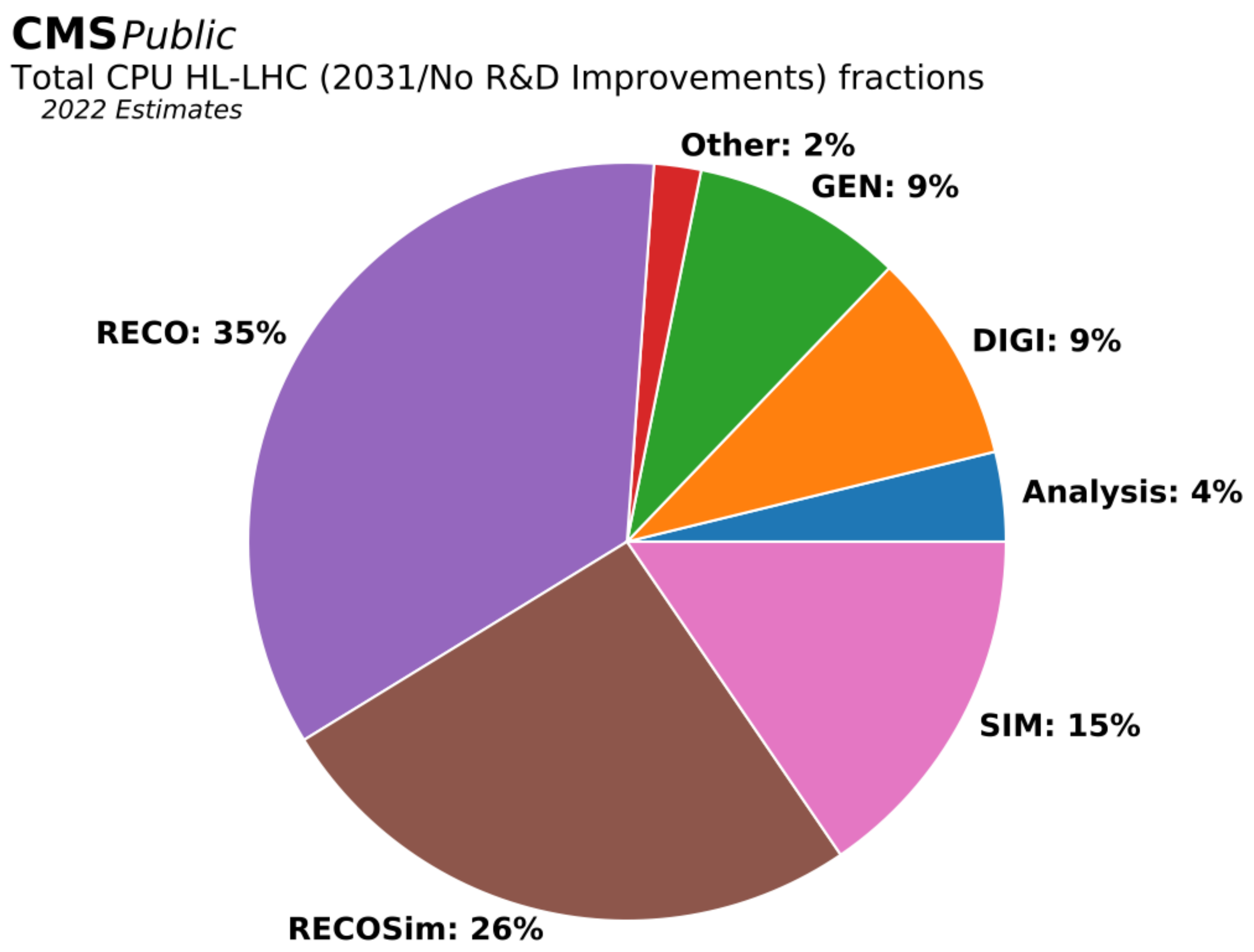
- The precision program comes at a cost of computing resources
- Current event generation step accounts for 9% of the total CPU footprint
- Expected to increase as physics processes are generated at high precision
- Current area of focus:
  - Implement multithreading for all workflows ✅
  - Making large gridpacks read-only ✅
  - Reduction of negative weights
  - GPU off-loading

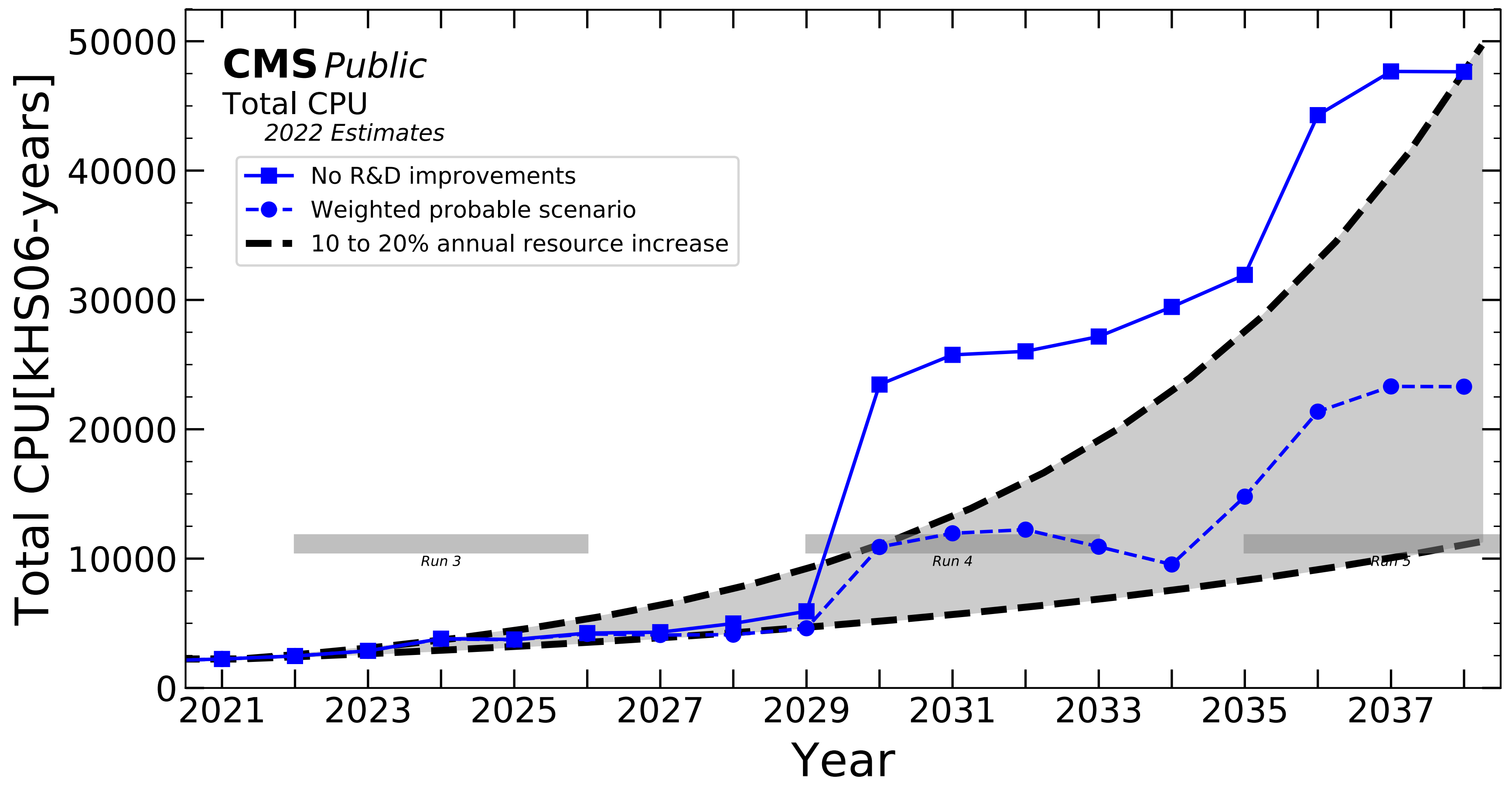
CPU Efficiency

Physics based improvement

Complete or nearing completion indicated with: ✅

https://twiki.cern.ch/twiki/bin/view/CMSPublic/CMSOfflineComputingResults





Need to focus on computing improvements now…

- As the workflows become more complicated and higher order corrections are implemented, it is foreseeable that event generators will start costing more in terms of CPU usage
  - To this end, over the last few years, several improvements have been explored and implemented
    - Multithreading at the event generator level (talk given in HSF meeting)
    - Mitigation of negative weights
    - Synergy with Madgraph4GPU group

LO vs. NLO — the problem of negative weights

- While next-to-leading order (NLO) often provides best description of data
  - Samples suffer from large fraction of negative weights
    - Lead to statistical dilution and computational costs for downstream processing
  - For tails of distributions, often easier to generate samples with additional partons at leading order (LO)
- Recommend analysts to exercise caution and decide what is needed for their analysis

Rate of negative events

$pp \to e^+e^-$	6.9%	(1.3)
$pp \to e^+\nu_e$	7.2%	(1.4)
$pp \to H$	10.4%	(1.6)
$pp \to Hb\bar{b}$	40.3%	(27)
$pp \to W^+j$	21.7%	(3.1)
$pp \to W^+t\bar{t}$	16.2%	(2.2)
$pp \to t\bar{t}$	23.0%	(3.4)

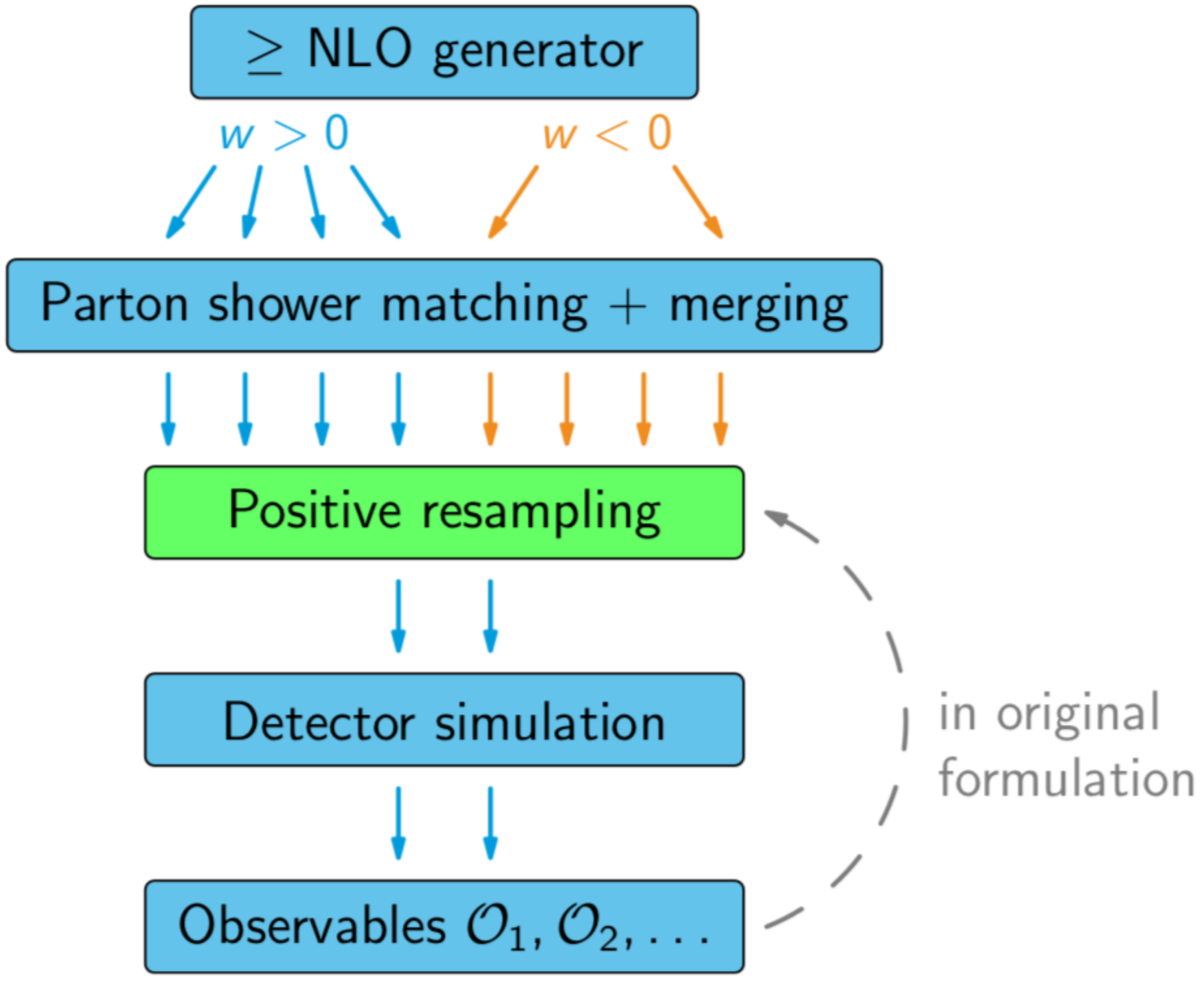
Cost In sample size

$$c(f) = \frac{1}{(1-2f)^2}$$

From Olivier Mattelaer's talk

Handling negatively weighted events

- Handling of negatively weighted events currently pursued with the generator theory community
- New and improved positive resampling and neural network resampling introduced by Andreas Maier and Jeppe Andersen
  - Implements redistribution of weights without incurring any bias and affecting an observable: leads to dramatic reduction in negative weight fraction
- First implementation in place → validation in progress



https://doi.org/10.1140/epjc/s10052-020-08548-w

Madgraph4GPU

Stephan Hageboeck, Stefan Roiser,
Andrea Valassi, Olivier Mattelaer

- The CMS generators group is working closely with CERN computing and theorists to take advantage of developments toward a GPU based version of Madgraph
- Matrix element calculations are offloaded to GPUs
- Overall execution is still dominated by the Fortran part of the computation
- Currently in discussions on implementing a GPU-based event generation in central workflows

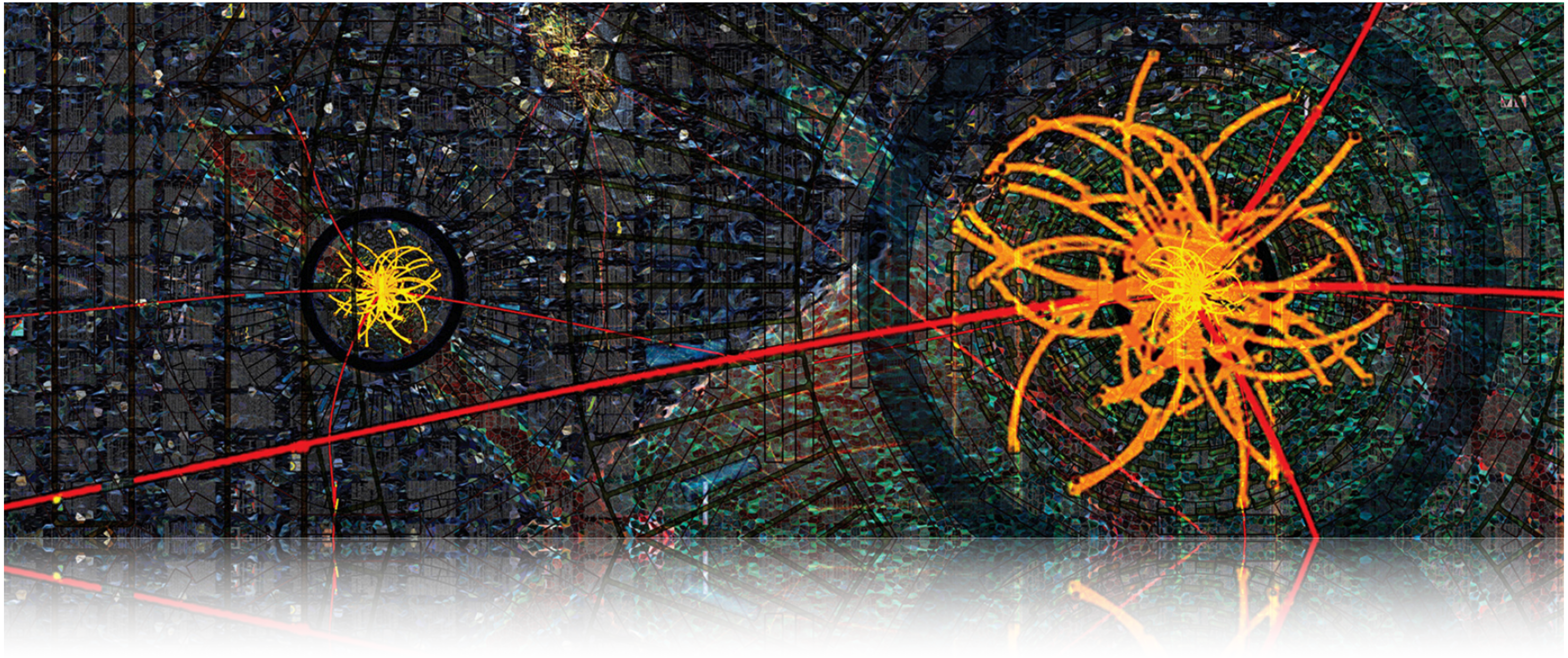
		madevent		
CUDA grid size		8192		
$gg \to t\bar{t}gg$	MEs precision	$t_{TOT} = t_{Mad} + t_{MEs}$ [sec]	N_{events}/t_{TOT} [events/sec]	N_{events}/t_{MEs} [MEs/sec]
Fortran	double	55.4 = 2.4 + 53.0	1.63E3 (=1.0)	1.70E3 (=1.0)
CUDA	double	2.9 = 2.6 + 0.35	3.06E4 (x18.8)	2.60E5 (x152)
CUDA	float	2.8 = 2.6 + 0.24	3.24E4 (x19.9)	3.83E5 (x225)

NVidia V100, Cuda 11.7, gcc 11.2

		madevent		
CUDA grid size		8192		
$gg \to t\bar{t}ggg$	MEs precision	$t_{TOT} = t_{Mad} + t_{MEs}$ [sec]	N_{events}/t_{TOT} [events/sec]	N_{events}/t_{MEs} [MEs/sec]
Fortran	double	1228.2 = 5.0 + 1223.2	7.34E1 (=1.0)	7.37E1 (=1.0)
CUDA	double	19.6 = 7.4 + 12.1	4.61E3 (x63)	7.44E3 (x100)
CUDA	float	11.7 = 6.2 + 5.4	7.73E3 (x105)	1.66E4 (x224)
CUDA	mixed	16.5 = 7.0 + 9.6	5.45E3 (x74)	9.43E3 (x128)

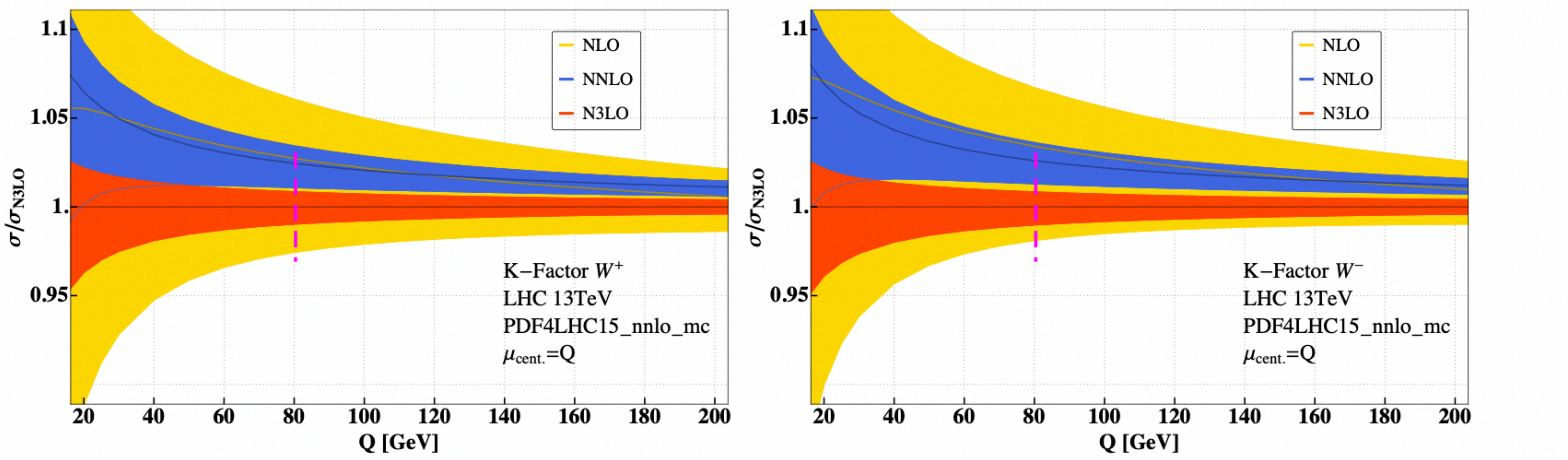
NVidia V100, Cuda 11.7, gcc 11.2

Modeling of physics processes



Uncertainties for Matrix Element predictions

- Conventional method of estimating scale uncertainties (obtaining NNLO if we calculate up to NLO) exist
  - Unfortunately uncertainty band has no statistical interpretation

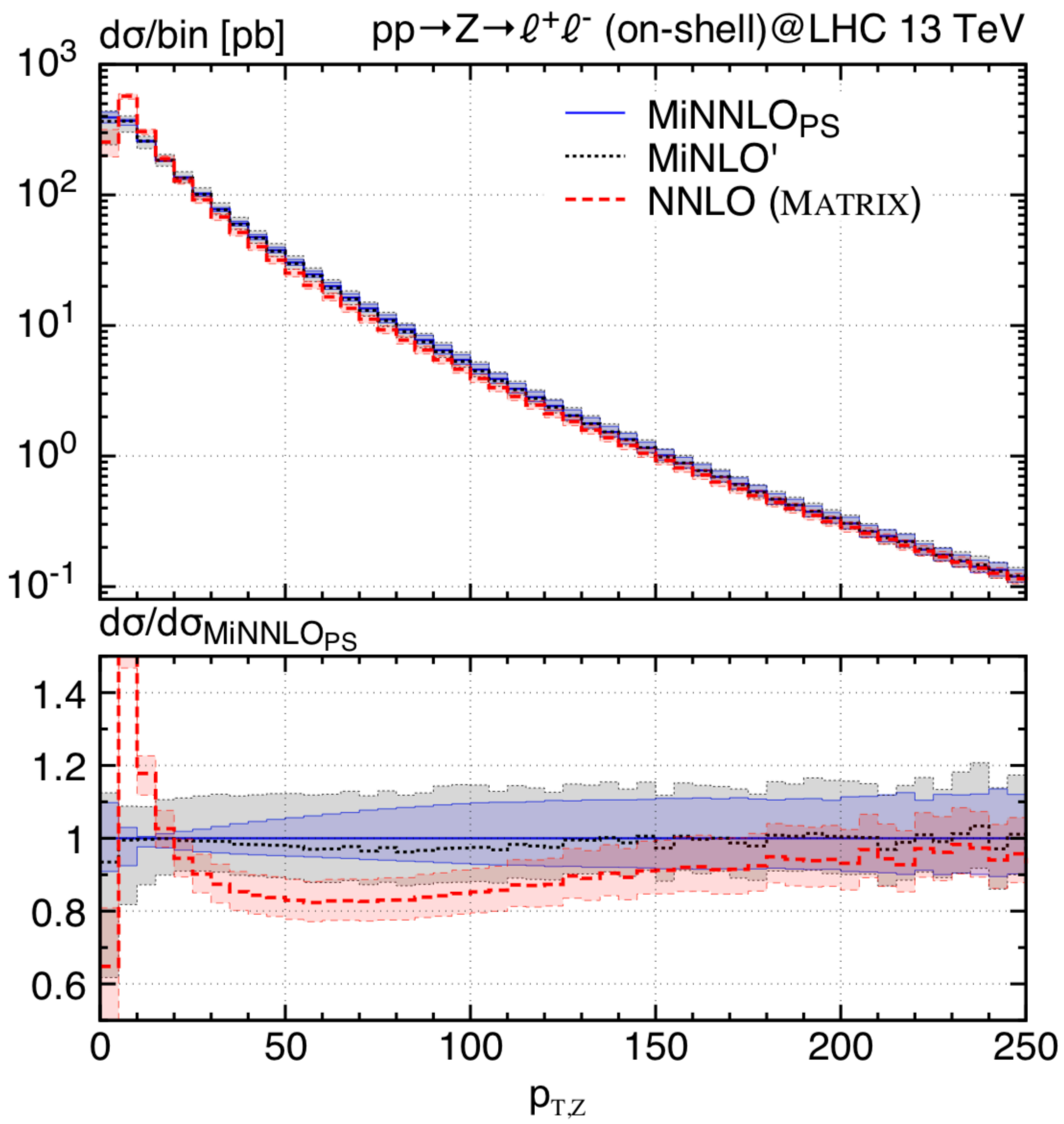


N3LO computation lies outside NNLO band obtained by varying scales

arXiv:2007.13313

Sample generation at the precision frontier

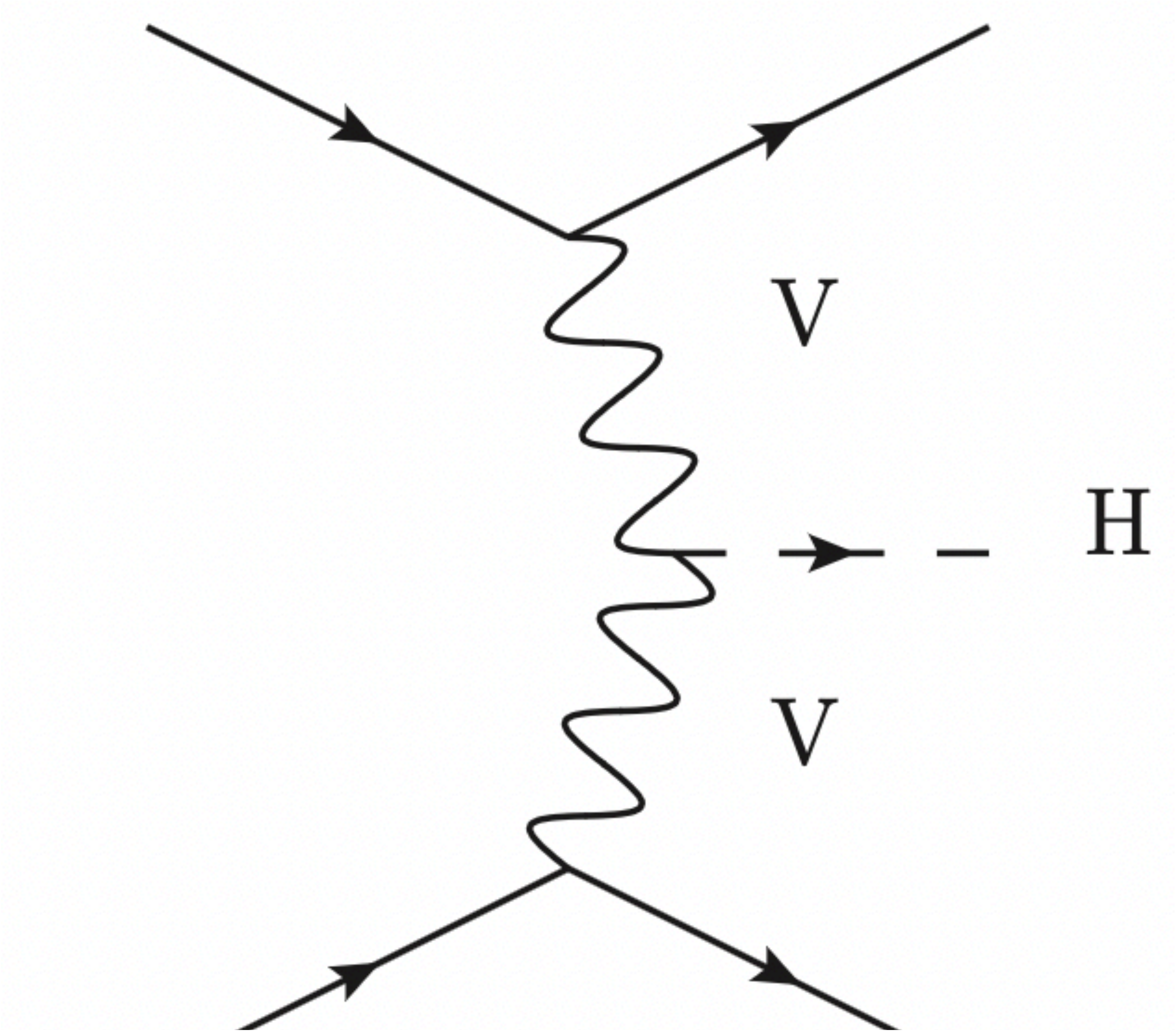
- MiNNLO (arXiv:1908.06987) sample generated at next-to-next-leading-order (NNLO) accuracy + parton shower (NNLO+PS)
- NNLO generation enabled by merging jet multiplicities with custom scale choice
- Large sample size required to match sample size of data
- Low fraction of negative weights
- Extensive validation performed
- However, MiNNLO is computationally expensive



Vector Boson Fusion modeling

- Large theoretical uncertainties associated with VBF modeling
- Behavior of third jet merits investigation → sensitive to influence of PS modeling
- Sources of possible variations arise from choice of generator, matching scheme, shower MC program, recoil scheme

https://arxiv.org/pdf/2003.12435.pdf



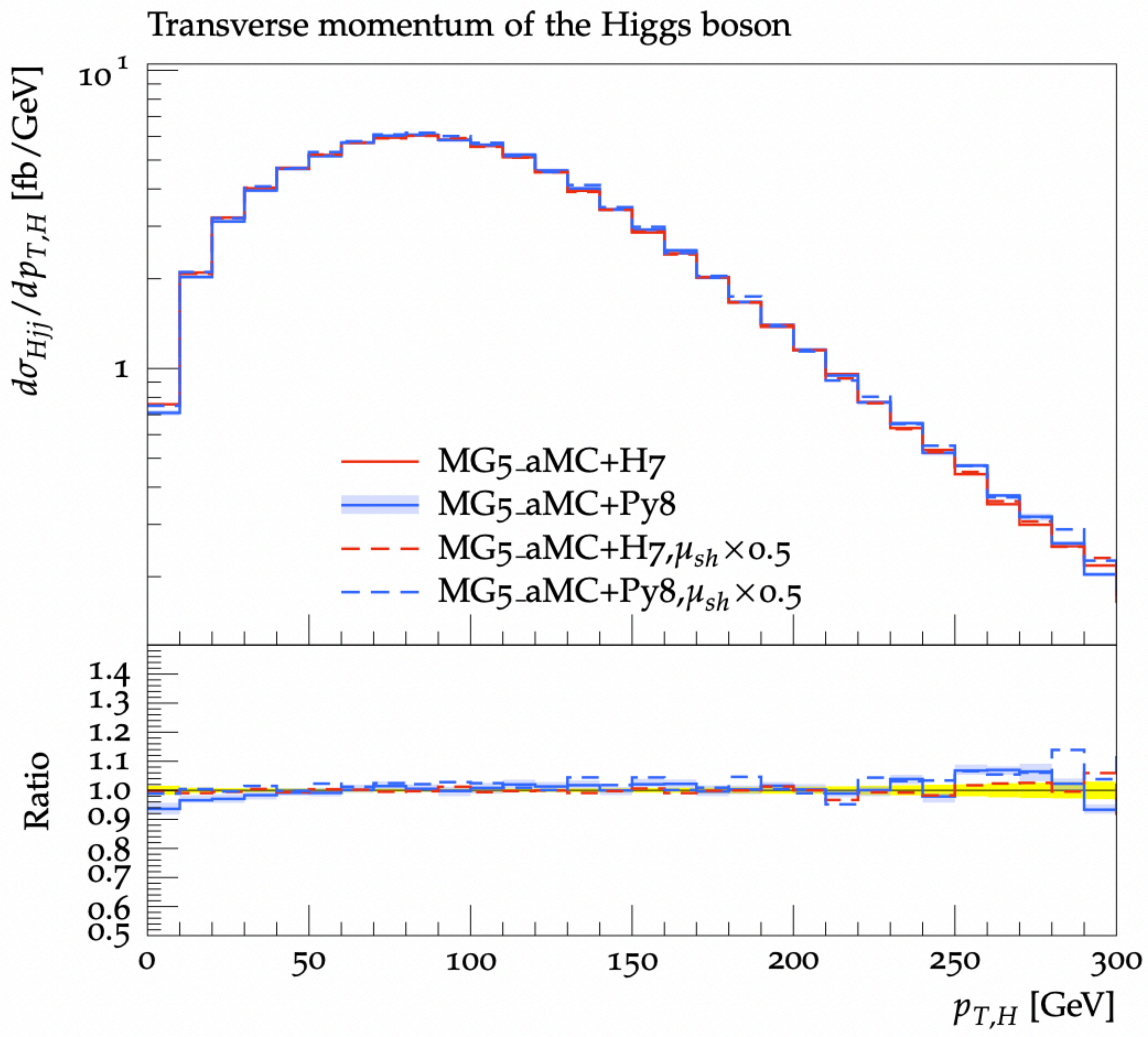
generator	matching	SMC	shower recoil	used in comparison
VBFNLO+Herwig7/Matchbox	$\oplus$	HERWIG 7.1.5	global ($\tilde{q}$) / local (dipole)	✓($\tilde{q}$)
HJets+Herwig7/Matchbox	$\oplus$	HERWIG 7.1.5	global ($\tilde{q}$) / local (dipole)	
MadGraph5_aMC@NLO 2.6.1	$\oplus$	HERWIG 7.1.2	global	✓
MadGraph5_aMC@NLO 2.6.1	$\oplus$	PYTHIA 8.230	global	
POWHEG-BOX V2	$\otimes$	PYTHIA 8.240	local (dipole)	✓
POWHEG-BOX V2	$\otimes$	PYTHIA 8.240	global	
POWHEG-BOX V2	$\otimes$	HERWIG 7.1.4	global ($\tilde{q}$)	

Vector Boson Fusion modeling

- Inclusive (in the number of jets) observable — p_T of the Higgs modeled well by all generators

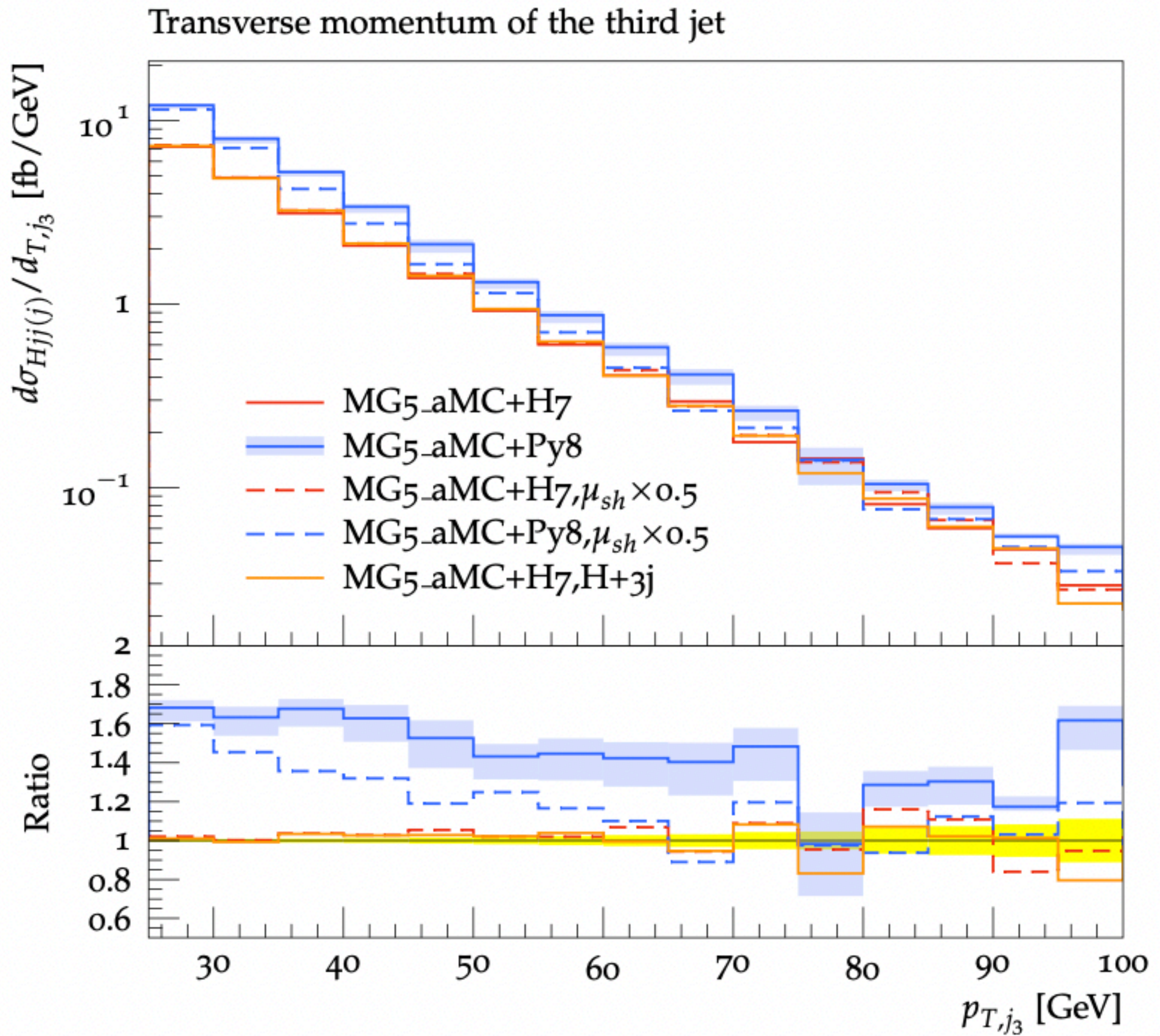
- Modeling differences in jet bins almost exactly compensate

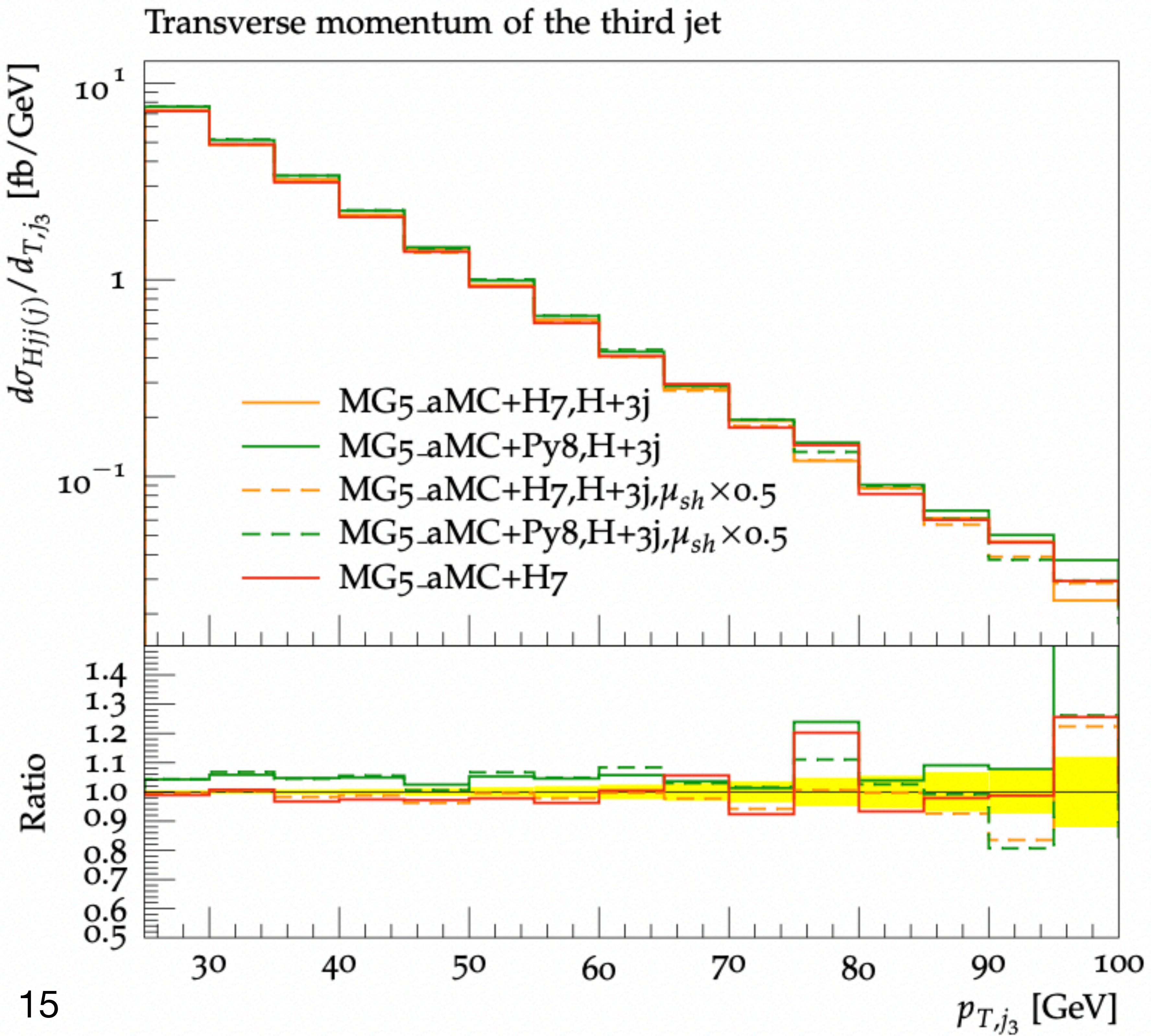
- Little to no dependence on choice of renormalization, factorization and shower scales



Vector Boson Fusion modeling

- Looking at the third jet:
  - Global dipole recoil scheme clearly incorrect
    - In CMS, samples generated with local dipole recoil scheme
    - Can we do better?





What we need in CMS

- Improvements in generator workflows *crucial* to usher in the precision era of the LHC
- Specific issues:
  - Handling large gridpacks efficiently — currently I/O is a bottleneck and the read-only functionality is the solution
  - Multithreading
  - GPU offloading
  - Reduction of negatively weighted events in NLO workflows
- Improved physics modeling in specific areas of phase space
  - In many cases improved physics modeling comes at the cost of computing resources
- With the challenges of the High Luminosity LHC in mind, it would be good to structure generator workflows to be compliant with modern software engineering and design standards, so they can be scaled on $\mathcal{O}(100k)$ cores
  - Challenges arise from the vast number of external packages that are run → could benefit from further optimization
- Reduction of theoretical uncertainties in precision measurements → perhaps the topic of a future meeting?