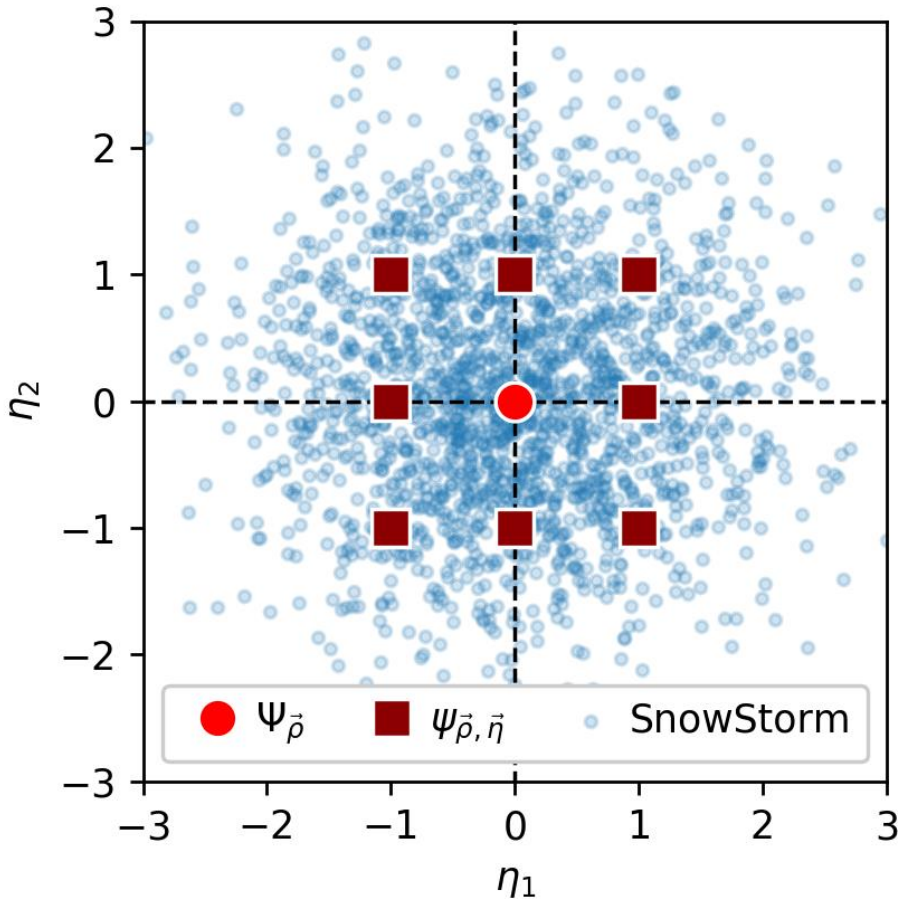
SnowStorm Simulation Chain – Overview

ERIK GANSTER, CHRISTIAN HAACK, JÖRAN STETTNER

Physics Institute III B
RWTH AACHEN UNIVERSITY

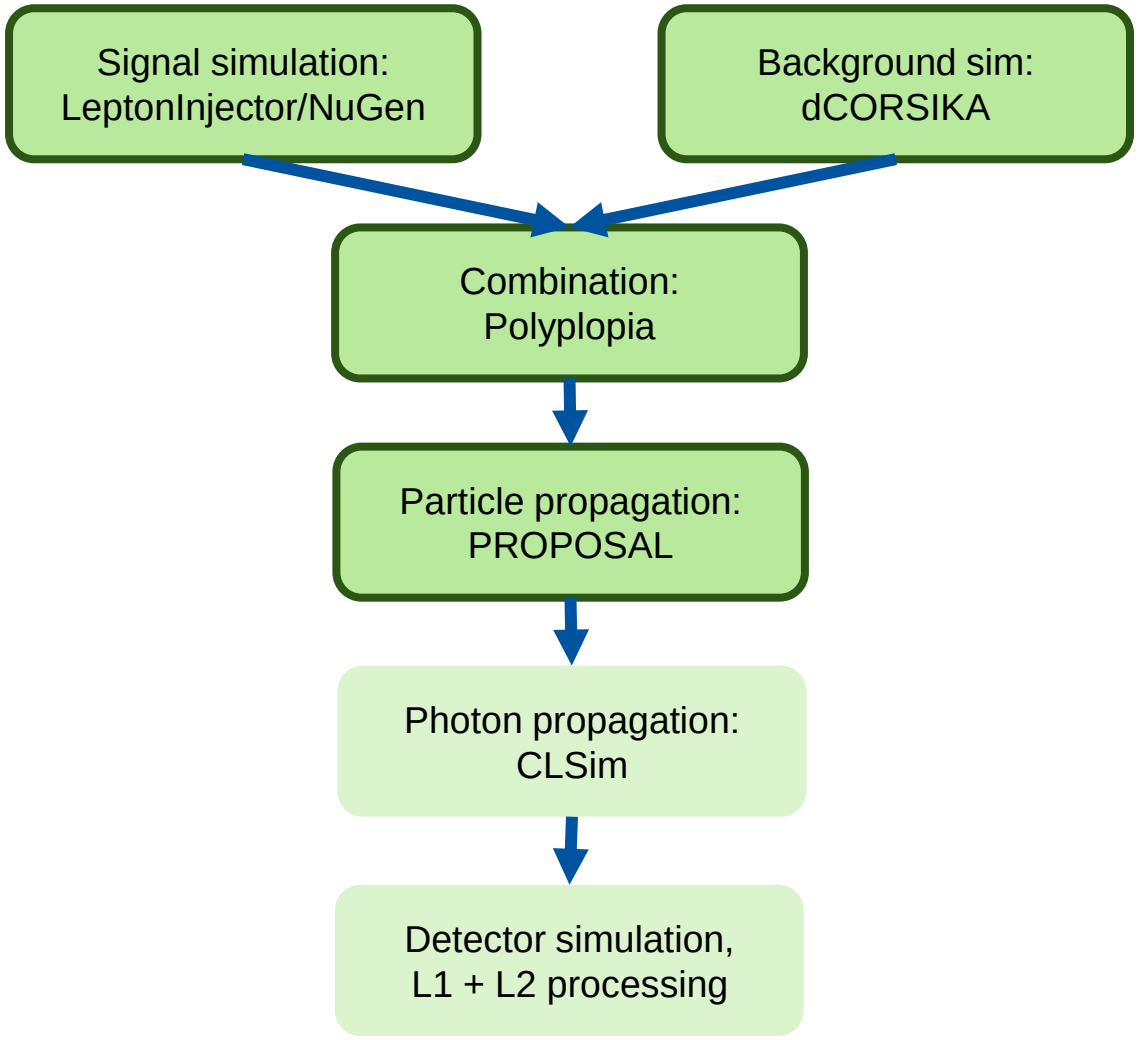
What is SnowStorm?

- Method for detector systematic treatment development by Ben Jones et.al. (paper, slides)
  - “traditional” approach: discrete MC sets for specific values of detector + ice parameters (red)
  - Snowstorm approach: continuous variation of nuisance parameters
  - (former Multisim)
- Ice/detector properties get perturbed/updated/sampled during simulation on an “event-to-event” basis
- Multiple nuisance parameters varied within a single MC set
- Based on the “standard” simulation chain with some tweaks to PhotonPropagation (CLSim)



SnowStorm Simulation Chain - Overview

- Signal + background simulation: NuGen/LeptonInjector and CORSIKA
  - NuGen used in first production
- Combination/Merging: Polyplopia
  - Directly merge signal+background primaries into a single I3MCTree level
  - No distinct signal/background MCTree
- Particle Propagation: PROPOSAL (and CMC)
  - Propagation of the MCTree (containing signal + background primaries) through the ice

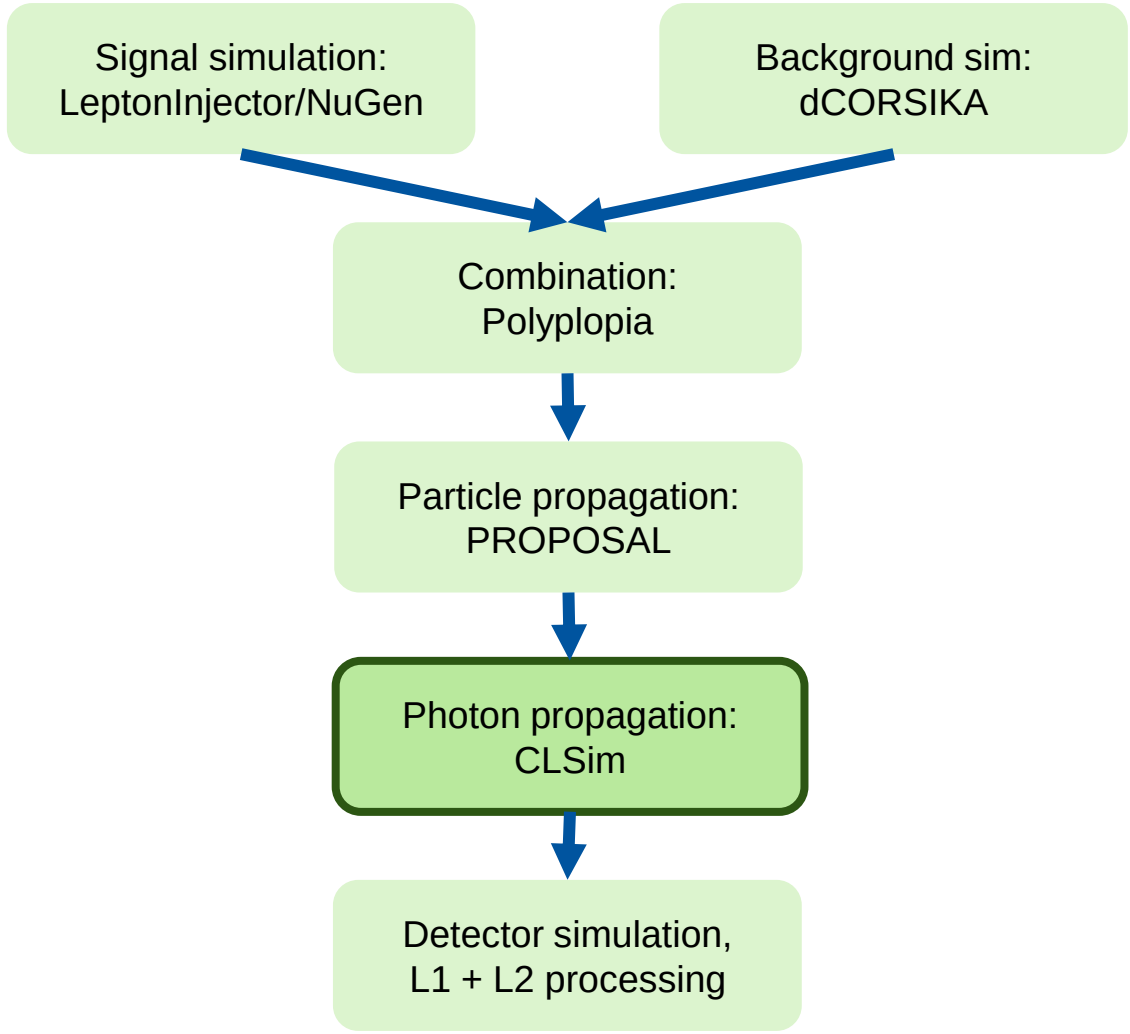


SnowStorm Simulation Chain - Overview

- Photon Propagation: CLSim
  - Instead of initializing a propagation kernel only once, update it with the detector + ice properties on an event-to-event basis by using the Snowstorm perturber
    → M(odel)-Frames

  - "Event-to-event basis": Update kernel after 10-1000 events (energy dependent) to minimize initialization overhead

  - The perturber applies parametrizations of ice/detector systematics by sampling a value for each and updates ("perturbs") the photon propagation kernel



SnowStorm Perturber - Parametrizations

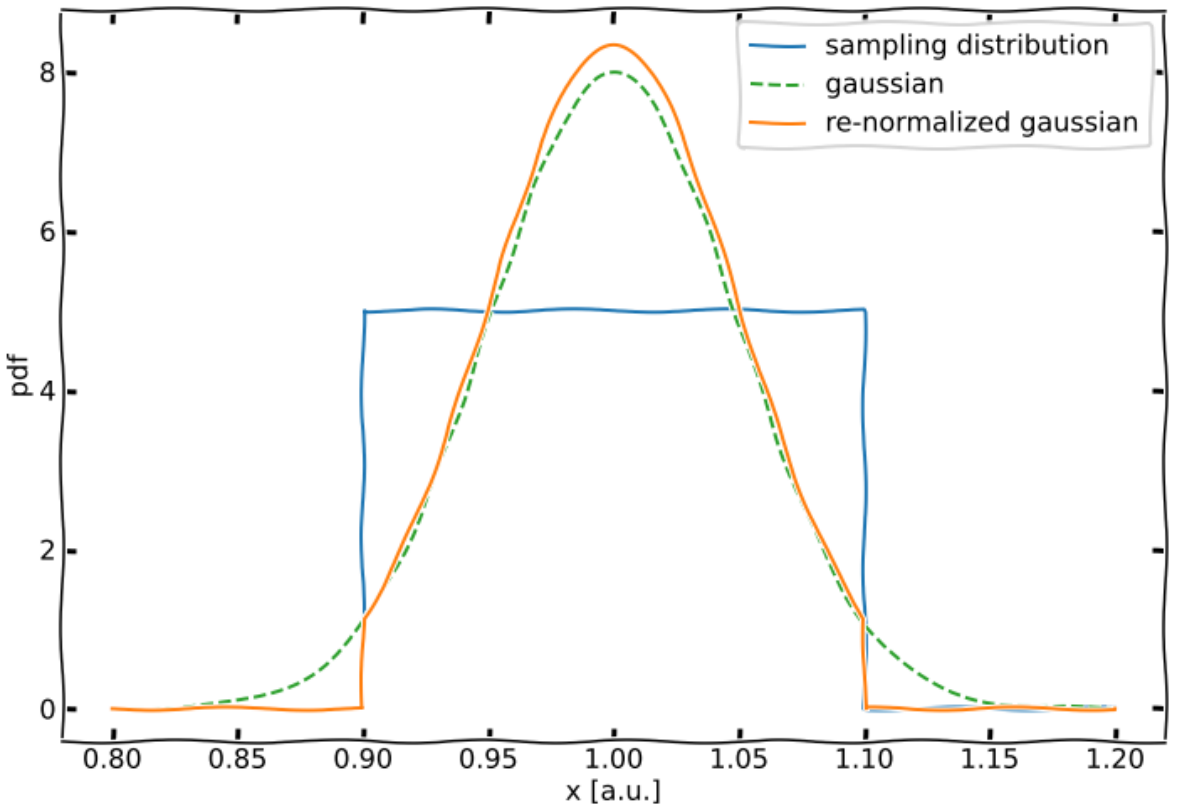
- Ice absorption/scattering:
  - Varied via the (depth dependent) absorption/scattering coefficients
- Anisotropy scale:
  - Scaling of anisotropy coefficients k1 and k2 (fixed axis)
- DOM efficiency:
  - Direct scaling of the DOMs wavelength acceptance
- HoleIce:
  - Changing the DOMs angular acceptance (but keeping the normalization)
- All perturbations are currently realized entirely by updating/rebuilding CLSim's photon propagation kernel

Systematic/Parametrization	Default Sampling Distribution	Sampling Range
(IceWavePlusModes	*MultivariateNormal*	*2x12 parameters)*
Scattering	uniform	[0.9, 1.1]
Absorption	uniform	[0.9, 1.1]
AnisotropyScale	uniform	[0.0, 2.0] (= 0-15%)
DOMEfficiency	uniform	[0.9, 1.1]
HoleIceForward_Unified	uniform	p0 [-1.0, +1.0] p1 [-0.2, +0.2]

SnowStorm MC – Towards the Use in an Analysis

- SnowStorm: Continuous variation of nuisance parameters (detector systematics) instead of multiple discrete sets for specific values

- ➢ Reweighting of the same simulation set to different distributions/values of systematic parameters on analysis level
- ➢ A single SnowStorm sets replaces both, baseline + systematic sets
- ➢ Easily expandable: just need to “expand” the sampling distribution to account for the new sets in the reweighting
- ➢ No need to use all SnowStorm systematics within the set: marginalizing (=no reweighting) for systematics you don’t want to use



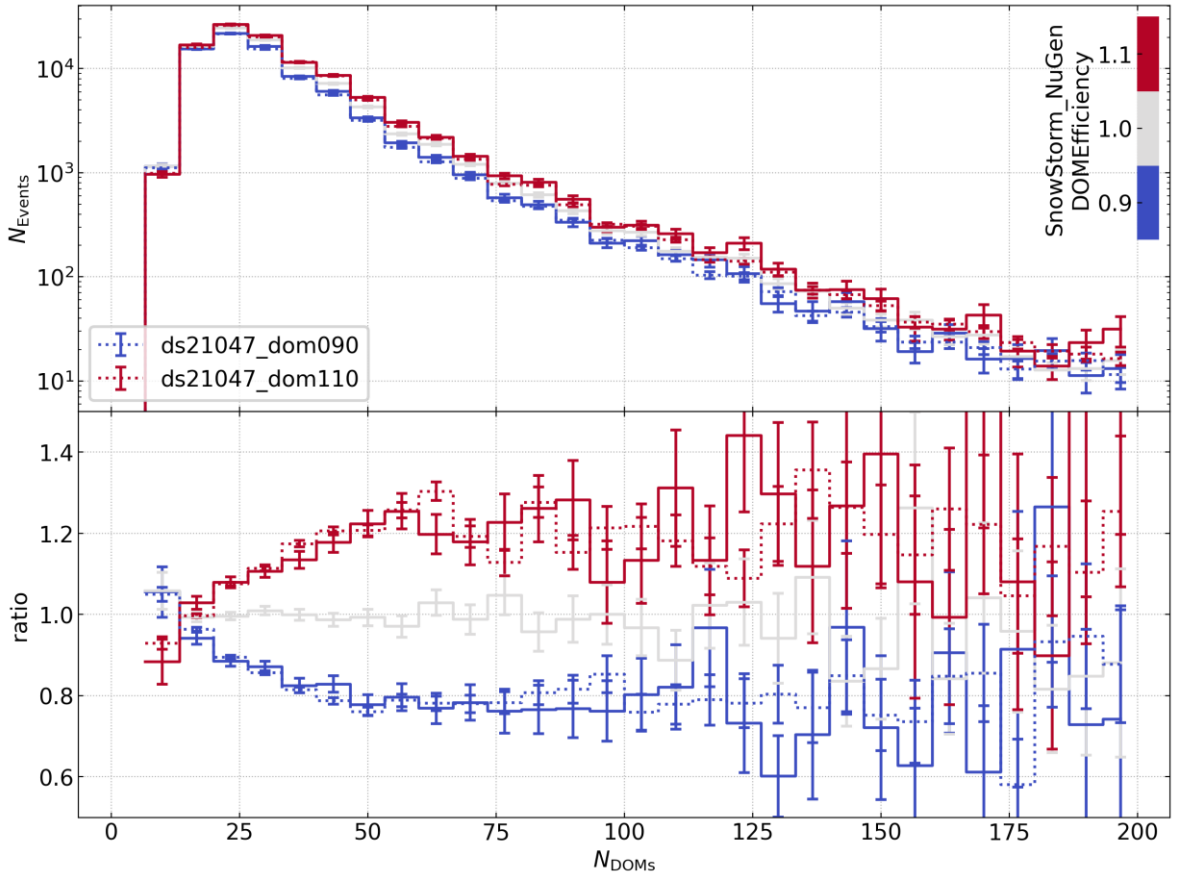
SnowStorm – Production Status and Proof of Concept

- First Snowstorm MC runnin gon iceprod
  - NuGen NuMu already done, NuE and NuTau ongoing

➢ Reweighted SnowStorm sets can reproduce discrete systematic sets:
  - Solid lines: SnowStorm NuMu, reweighted using a Gaussian with $\mu = 0.9; 1.0; 1.1$ and $\sigma = 0.02$
  - Dashed lines: “old/traditional” systematic sets with DOMEff 0.9/1.1

➢ Marginalizing all other systematics at the same time

N_{DOMs} : Number of hit DOMs for different DOM Effs.



SnowStorm – Production Status and Proof of Concept

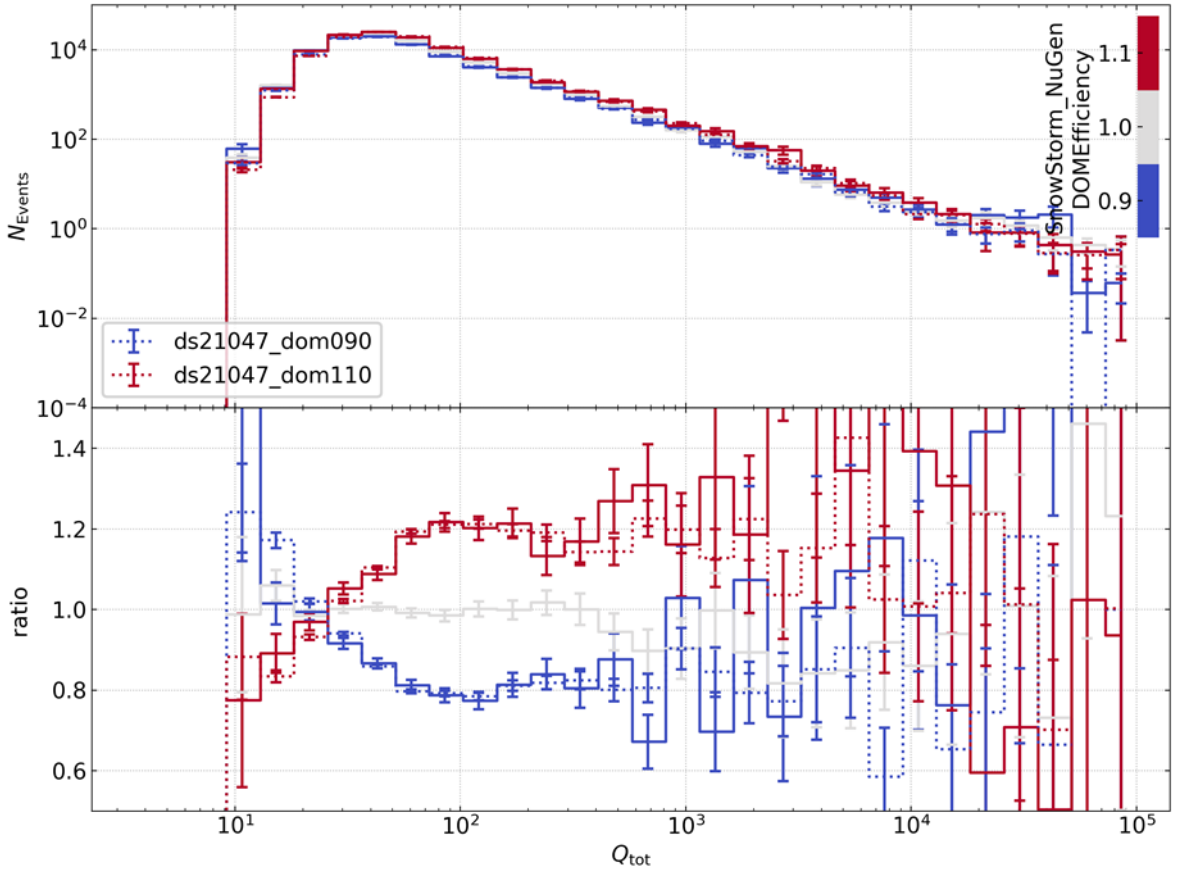
- First Snowstorm MC runnin gon iceprod
  - NuGen NuMu already done, NuE and NuTau ongoing

➢ Reweighted SnowStorm sets can reproduce discrete systematic sets:
  - Solid lines: SnowStorm NuMu, reweighted using a Gaussian with $\mu = 0.9; 1.0; 1.1$ and $\sigma = 0.02$
  - Dashed lines: “old/traditional” systematic sets with DOMEff 0.9/1.1

➢ Marginalizing all other systematics at the same time

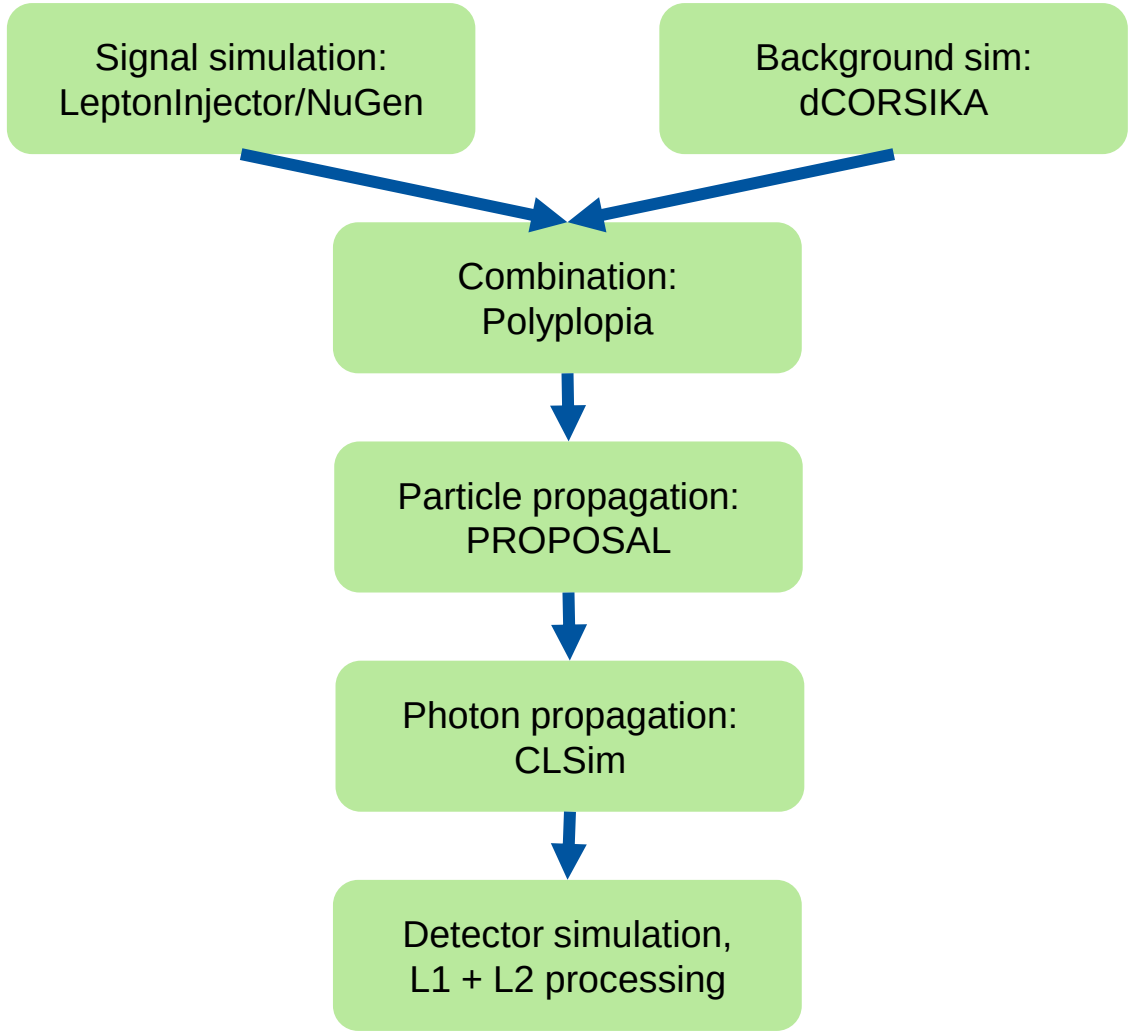
- First production is aiming for analysis framework/tool development:
  - Update analysis tolls/frameworks to Snowstorm MC
  - Development of new/advanced analysis techniques
  - Verify those new techniques/tools by using the available discrete simulation sets

Qtot: Total charge per event for different DOM Effs.



SnowStorm Simulation Chain – Summary Part I

- ✓ Snowstorm simulation chain production right now
  - ✓ **py3-v4.1.1** with combo.V01-00-02
- ✓ Proof of concept for reweighting SnowStorm systematics
  - ✓ Can recover discrete systematic sets
  - ✓ Marginalizing works as well
- ✓ snowstorm project (in combo) contains the perturber + systematic parametrizations
- ✓ Simulation (chain) scripts located in: `simprod-scripts/resources/scripts/SnowSuite`
- ✓ Thanks to everyone involved in the development!



SnowStorm MC

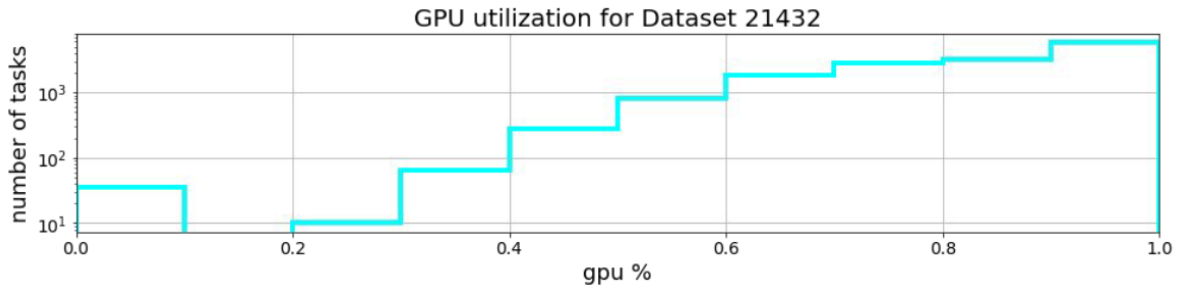
Resource usage/efficiency

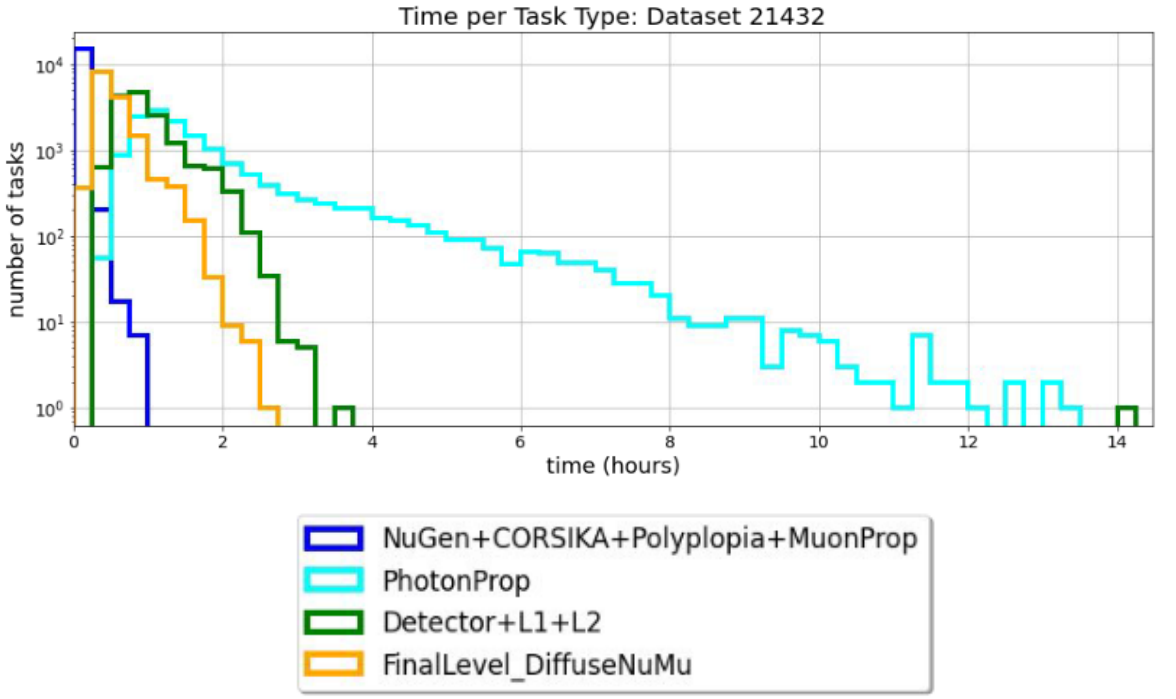
SnowStorm – Resource Usage/Efficiency

- David showed some plots on the resource usage of the SnowStorm NuMu set yesterday

- Before submitting the SnowStorm sets to iceprod we spent quite some time to use the available resources as efficient as possible:
  - Optimized jobs/tasks for an average runtime of 2h
  - Reduce CLSims initialization overhead by tweaking the number of M frames per job
  - Split the simulation into 3 energy ranges with different nJobs, nEvents, spectral index, nMFrames
    - 1e2 – 1e4, 1e4 – 1e6, 1e6 – 1e8 GeV
  - Submitted (multiple) small benchmarks sets before to test those settings

- However: we encountered some problems trying to create an „on average“ resource efficient job, especially in terms of task runtime and GPU utilization



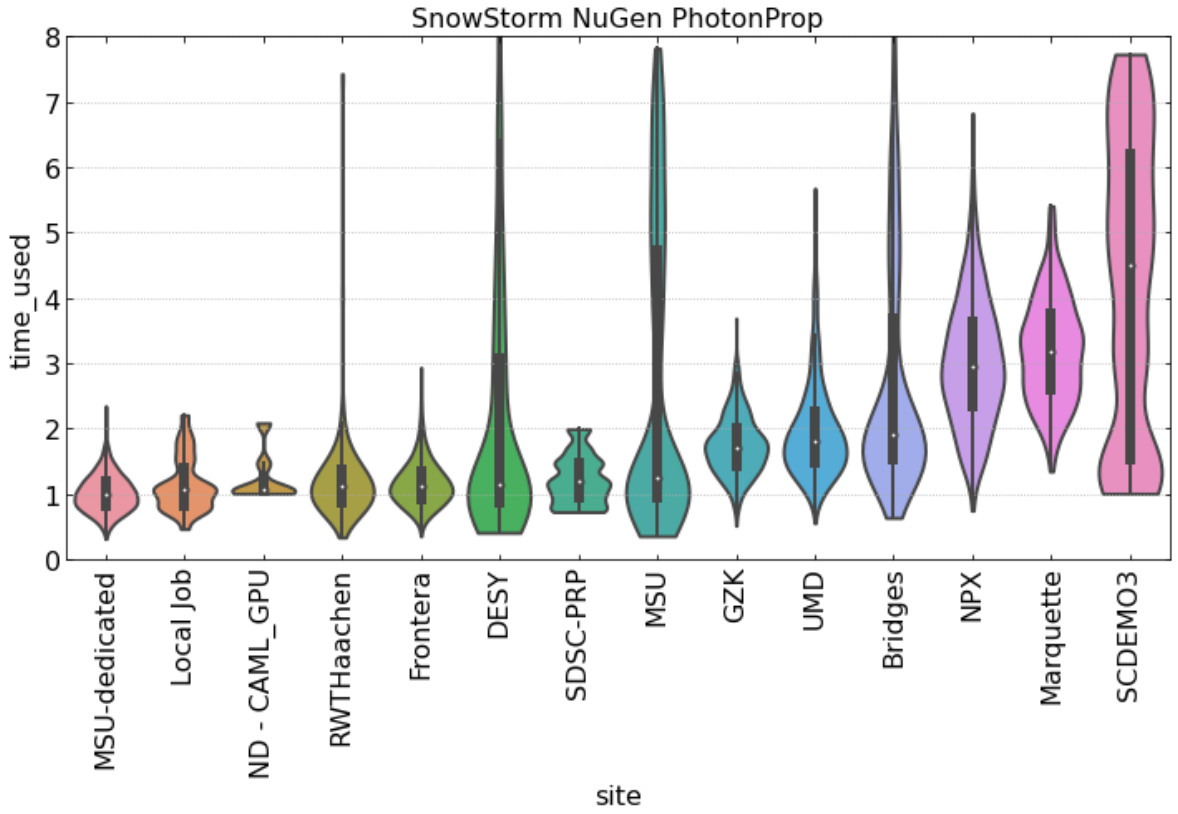


SnowStorm – Runtime of PhotonPropagation Task

- We have a variety of different GPUs in the pyglidein pool:
  - NPX/Marquette: GeForce GTX 980/1080
  - Bridges: Tesla K80
  - DESY: Tesla K20m, K80, P4, RTX2080 Ti
  - MSU: Tesla K80, V100
  - MSU-dedicated: GeForce RTX 2080 Ti

➢ Huge differences in the task runtime between the sites, ranging from 1 – 4.5h (on average)

➢ Difficult to optimize job/task runtime for multiple sites at once

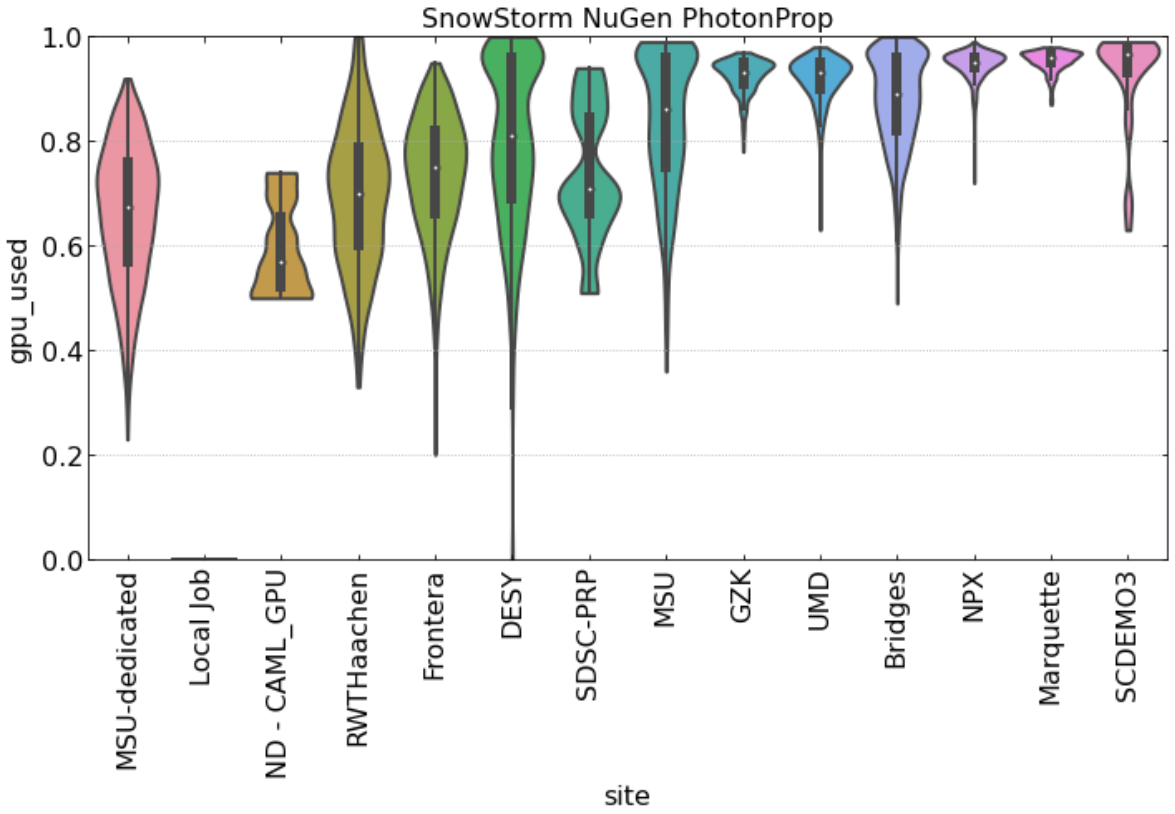


SnowStorm – GPU Utilization of PhotonPropagation Task

- GPU utilization for different sites

➢ Good utilization: UMD, NPX, Bridges
➢ Bad utilization: MSU, DESY
➢ Dreadful utilization: Frontera, MSU-dedicated

- For Snowstorm we tried to optimize the sets as best as possible:
  - Highlights differences between the sites
  - Problems more visible as for other sets



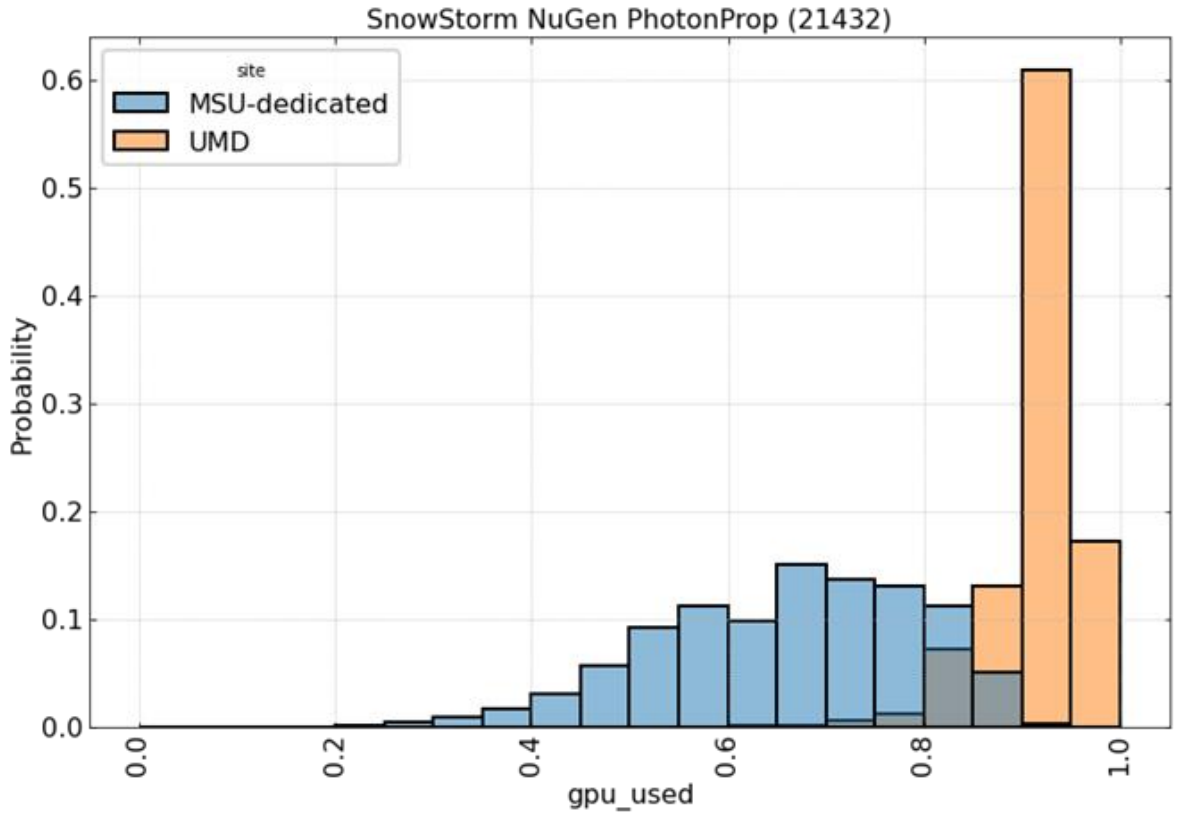
SnowStorm – GPU Utilization of PhotonPropagation Task

- GPU utilization for different sites

- Good utilization: UMD, NPX, Bridges
- Bad utilization: MSU, DESY
- Dreadful utilization: Frontera, MSU-dedicated

➢ It‘s impossible to make the same job to utilize >90% GPU on all sites
  – At least on „user level“ by only tweaking nEvents, energy range, sepctral index, etc.

➢ Distribute jobs/tasks e.g. energy dependent to different sites?

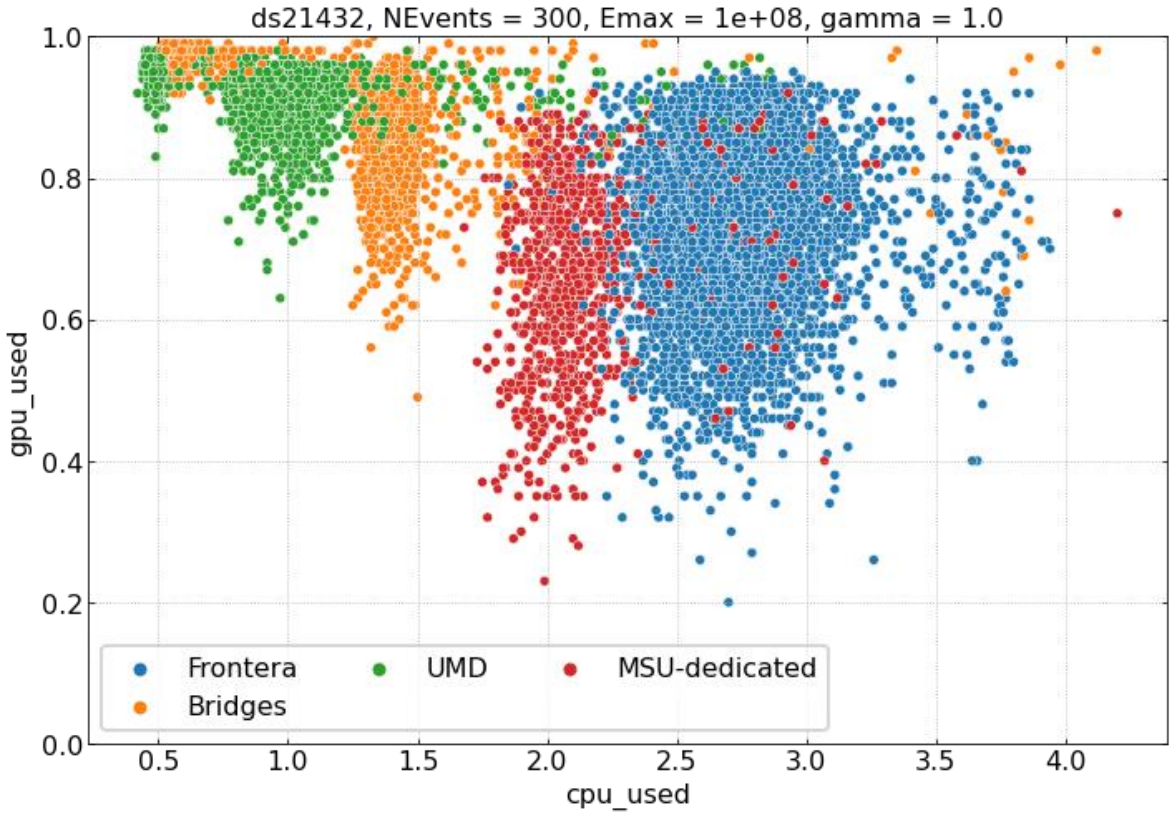


SnowStorm – Resource Efficiency/Utilization

- Not only the GPU utilization strongly varies between the sites, also the CPU usage…

➢ CPU bottleneck when CLSim runs on fast GPUs?
  ➢ Especially for high energy events (this set)

➢ Overhead of re-initializing the propagation kernel multiple times?
  ➢ Overhead strongly depends on GPU speed…
  ➢ Can be reduced be reducing the number of M-frames within a job
  ➢ However: Snowstorm method depends on sampling different ice properties, we can‘t reduce the overhead infinitely without loosing the physics…



SnowStorm Simulation Chain – Summary Part II

- ✓ Snowstorm simulation chain running and in production
  - ✓ **py3-v4.1.1** with combo.V01-00-02
- ✓ Proof of concept for reweighting SnowStorm systematics
  - ✓ Can recover discrete systematic sets
  - ✓ Marginalizing works as well
- ✓ First optimizations for resource efficiency applied
  - – Observed huge differences between computing clusters/sites the jobs run
  - – Difficult to design an “on average” efficient job for all sites
- ➢ We can still improve resource utilization/efficiency
  - ➢ Workshop’s coding sessions

SnowStorm Simulation Chain – Outlook

- ✓ Snowstorm simulation chain running and in production
  - ✓ **py3-v4.1.1** with combo.V01-00-02
- ✓ Proof of concept for reweighting SnowStorm systematics
  - ✓ Can recover discrete systematic sets
  - ✓ Marginalizing works as well
- ✓ First optimizations for resource efficiency applied
  - – Observed huge differences between computing clusters/sites the jobs run
  - – Difficult to design an “average” efficient job for all sites
- ➢ We can still improve resource utilization/efficiency
  - ➢ Workshop’s coding sessions

- ➢ Further optimize resource utilization/efficiency
- ➢ make Snowstorm SpiceBFRv1 compatible
  - ➢ Current baseline icemodel is Spice3.2.1
  - ➢ Use the updated “Snowstorm CLSim interface” (Alexander H.)
- ➢ DOMEff and HoleIce can be modified after the actual photon propagation
  - ➢ 3 parameters can be sampled for “free”, without additional GPU time
  - ➢ Also increases GPU utilization: less models needed
  - ➢ Need to store I3PhotonSeriesMap, at least temporarily
- ➢ Optimize different energy ranges for different sites and distribute jobs accordingly?

Appendix

SnowStorm MC – Example: FinalLevel DiffuseNuMu

MC TrueEnergy for different DOM Effs.

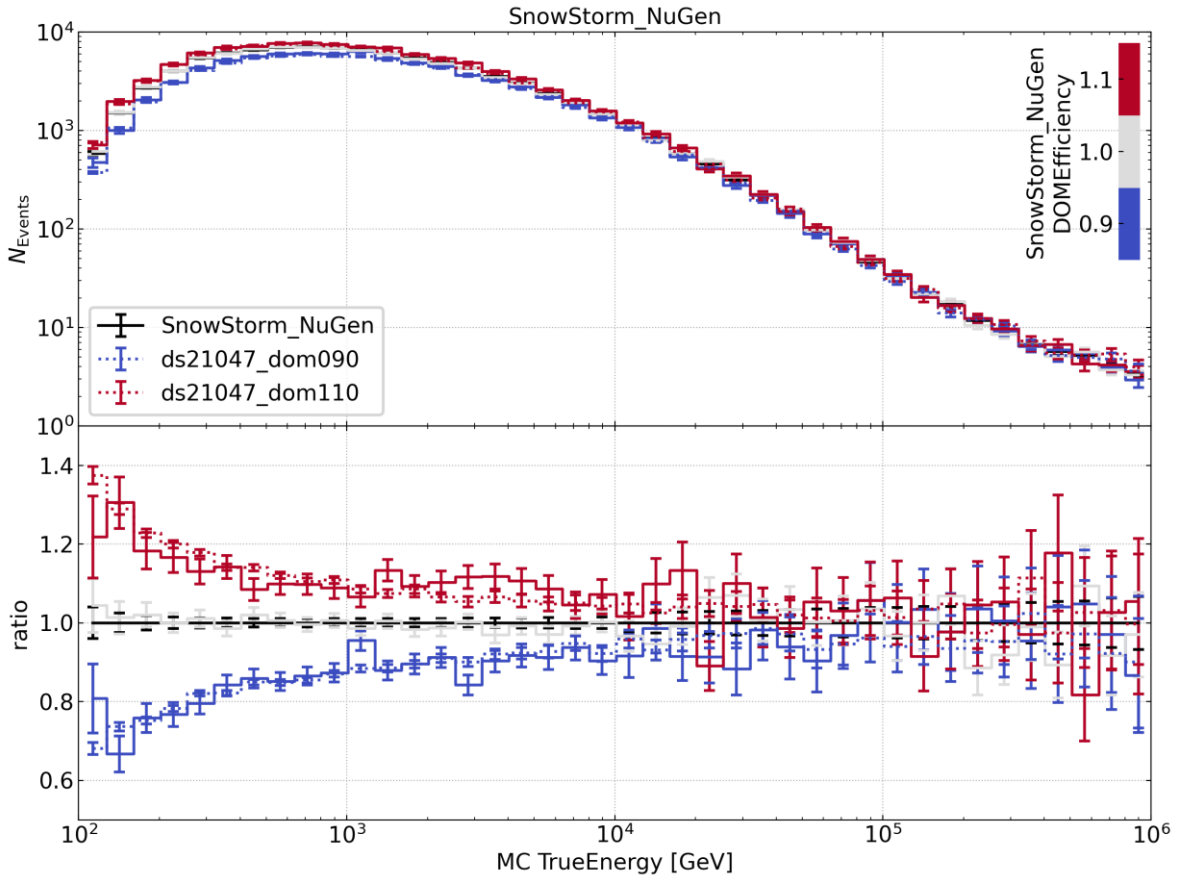
- Solid lines: SnowStorm NuMu, reweighted using a Gaussian with $\mu = 0.9; 1.0; 1.1$ and $\sigma = 0.02$
- Full SnowStorm set as baseline for the ratio

- Dashed lines: “old/traditional” systematic sets with DOMEff 0.9/1.1 (1.0 set as baseline in the ratio)

➢ Reweighted SnowStorm sets can reproduce discrete systematic sets

➢ Marginalizing all other systematics at the same time

MCPrimaryEnergy



SnowStorm MC – Example: FinalLevel DiffuseNuMu

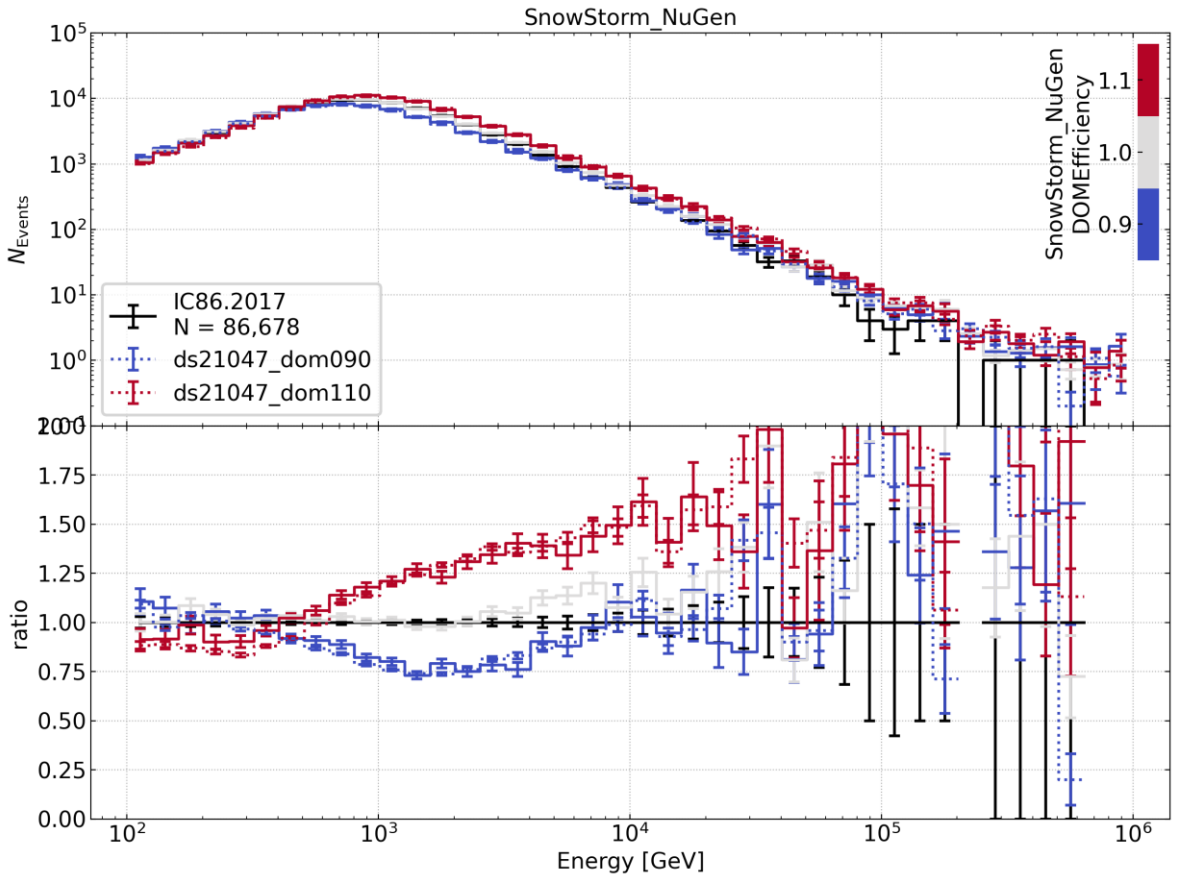
Truncated energy for different DOM Effs.

- Solid lines: SnowStorm NuMu, reweighted using a Gaussian with $\mu = 0.9; 1.0; 1.1$ and $\sigma = 0.02$
- Dashed lines: “old/traditional” systematic sets with DOMEff 0.9/1.1 (1.0 set as baseline in the ratio)
- *IC86.2017 data as baseline*

➢ Reweighted SnowStorm sets can reproduce discrete systematic sets

➢ Marginalizing all other systematics at the same time

Truncated Energy



SnowStorm MC – Example: FinalLevel DiffuseNuMu

SplineMPEIC Zenith for different DOM Effs.

- Solid lines: SnowStorm NuMu, reweighted using a Gaussian with $\mu = 0.9; 1.0; 1.1$ and $\sigma = 0.02$
- Dashed lines: “old/traditional” systematic sets with DOMEff 0.9/1.1 (1.0 set as baseline in the ratio)
- *IC86.2017 data as baseline*

➢ Reweighted SnowStorm sets can reproduce discrete systematic sets

➢ Marginalizing all other systematics at the same time

SplineMPEIC Zenith