SnowStorm

Physics
Institute III B

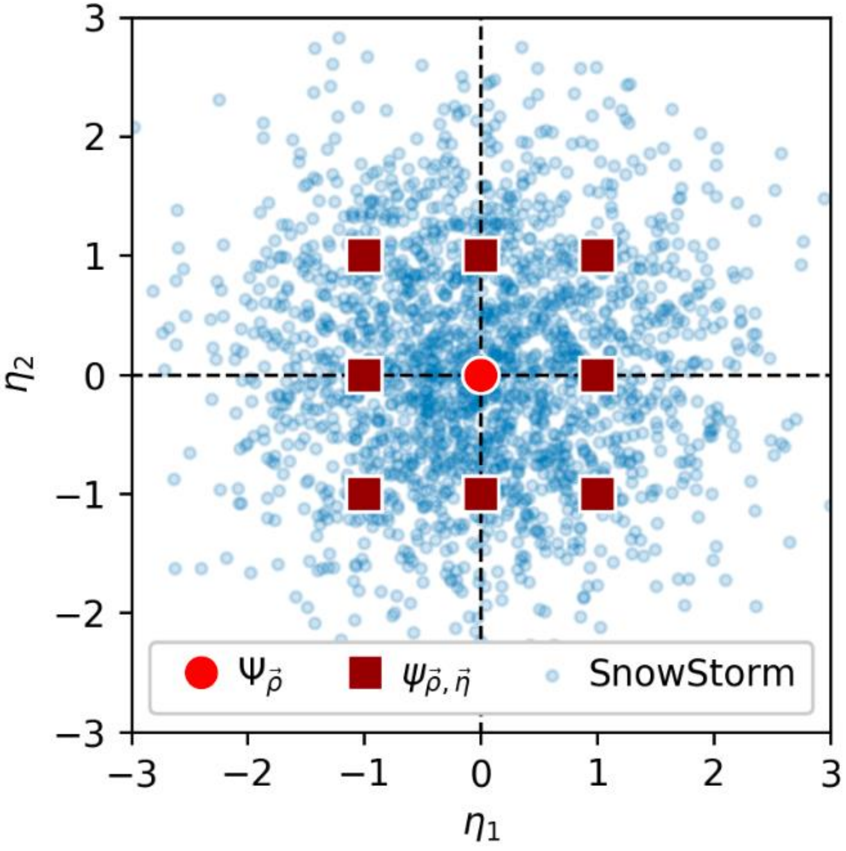
RWTH AACHEN
UNIVERSITY

Reminder: What is SnowStorm?

- Continuous variation of nuisance parameters
  - Instead of generating multiple simulation sets for some specific choices/combinations of nuisance parameters, generate a single SnwoStorm event ensemble
  - In the ensemble, a different combination of the nuisance parameters is chosen based on their allowed phase space
  - More detailed method overview:
    - SnowStorm paper
    - Ben’s talk at the Spring Collaboration Meeting 2020 (slides)

- Multiple nuisance parameters/detector systematics can be included within a single SnowStorm set at the same time

- Just a single snowstorm simulation set is needed in the later analysis



SnowStorm – Available Parametrizations

- Currently there are 6 (7) “snowstormable” systematics implemented
- Parametrizations are gathered in a “SnowStorm perturber” which applies them during photon propagation
- Nuisance parameters → detector/medium properties

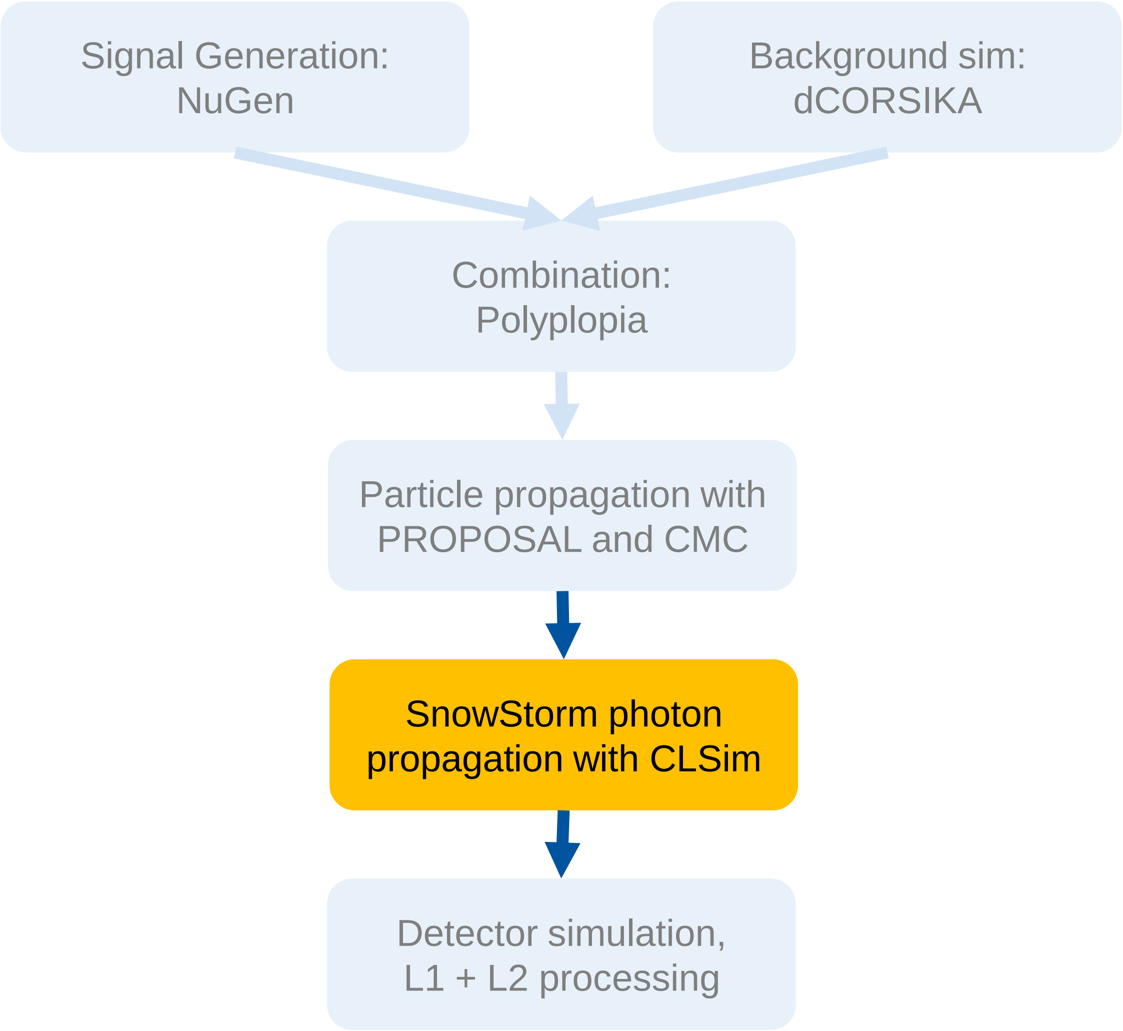
Systematic/Parametrization	Parameter(s)
IceWavePlusModes	12 amplitude + 12 phase shifts
Scattering	Global scaling
Absorption	Global scaling
AnisotropyScale	Strength scaling
DOMEfficiency	Global scaling
HoleIceForward_Unified	p0, p1
HoleIceForward_MSU	p1, p2

Table: Overview of all available systematic perturbations for SnowStorm

SnowStorm Simulation Chain

- The “SnowStorm magic” happens during photon propagation:
  - Application of the SnowStorm perturber for varying the detector + ice model parameters according to pre-defined parametrizations
  - Currently only supports CLSim

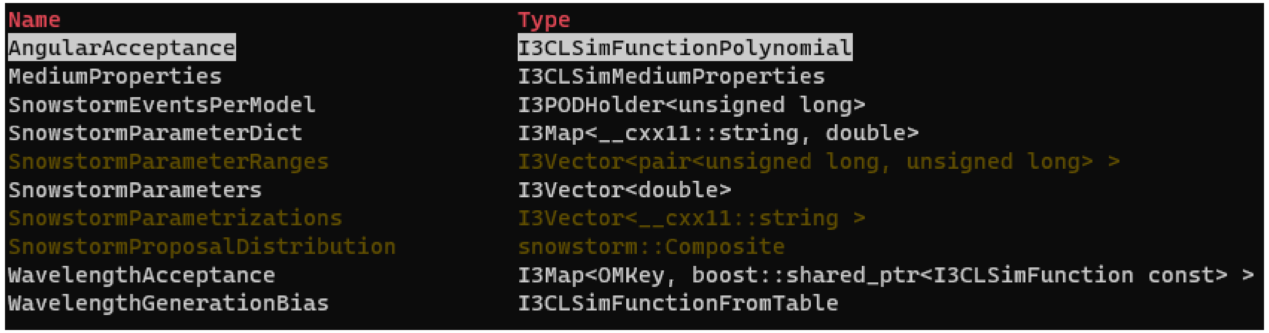
- Software locations:
  - `snowstorm` software project in icetray/main
    - Parametrizations + perturber
  - `SnowSuite` script collection in simprod-scripts
    - Includes the script for running the actual photon propagation step
  - SnowStorm software documentation


Signal Generation:
NuGen
Background sim:
dCORSIKA
Combination:
Polyplopia
Particle propagation with PROPOSAL and CMC
SnowStorm photon propagation with CLSim
Detector simulation, L1 + L2 processing

“Standard” Photon Propagation – How It Works

- Initializing the photon propagator with the detector + ice-model parameters to use
- Initialize photon propagation kernel (GPU)
- Start the photon propagation
- Continue until all events/frames are processed

SnowStorm Photon Propagation – How It Works

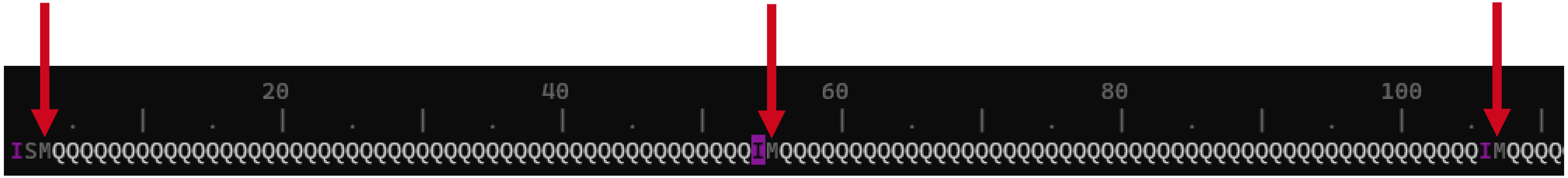


```
Name                                  Type
AngularAcceptance                     I3CLSimFunctionPolynomial
MediumProperties                      I3CLSimMediumProperties
SnowstormEventsPerModel               I3PODHolder<unsigned long>
SnowstormParameterDict                I3Map<__cxx11::string, double>
SnowstormParameterRanges              I3Vector<pair<unsigned long, unsigned long> >
SnowstormParameters                   I3Vector<double>
SnowstormParametrizations             I3Vector<__cxx11::string >
SnowstormProposalDistribution         snowstorm::Composite
WavelengthAcceptance                  I3Map<OMKey, boost::shared_ptr<I3CLSimFunction const> >
WavelengthGenerationBias              I3CLSimFunctionFromTable
```

1. Initializing the photon propagator (CLSim) with the baseline detector and ice-model parameters

2. Initialize the perturber with the parametrizations to be applied

3. Apply the perturber and update the detector + medium properties
   - Store the updated parameters in an M-frame

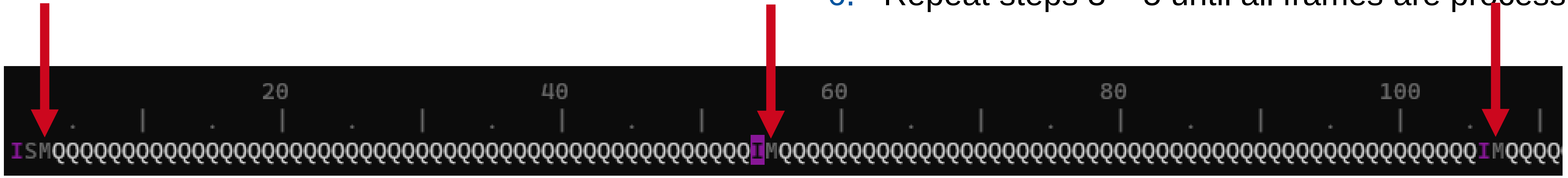


```
                  20                  40                  60                  80                 100
    .    |    .    |    .    |    .    |    .    |    .    |    .    |    .    |    .    |    .    |    .    |
ISMQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQIMQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQIMQQQQ
```

SnowStorm Photon Propagation – How It Works

- Initializing the photon propagator with the ice-model and parameter to use
- Initialize photon propagation kernel
- Start the photon propagation
- Continue until all events/frames are processed

1. Initializing the photon propagator (CLSim) with the baseline detector and ice-model parameters
2. Initialize the perturber with the parametrizations to be applied
3. Apply the perturber and update the detector + medium properties
   - Store the updated parameters in an M-frame
4. Setup (update) the photon propagation kernel using the updated/perturbed parameters
5. Process a pre-defined number of frames
6. Repeat steps 3 – 5 until all frames are processed



```
                  20                  40                  60                  80                 100
    .    |    .    |    .    |    .    |    .    |    .    |    .    |    .    |    .    |    .    |    .    |
ISMQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQIMQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQQIMQQQQ
```

SnowStorm – Perturber

- The perturber is a wrapper to apply multiple parametrizations one after another

- Gathers all configured parametrizations with their individual sampling distributions/boundaries

- If called:
  - Draw a new sample/value (or multiple) for each parametrization
  - Apply the parametrization using the sampled value(s) to update the detector/medium properties in the M frame

- Bookkeeping of sampled/used values (also in the M frame)

➢ Not limited to CLSim

Systematic/Parametrization	Default Sampling Distribution	Sampling Range
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]

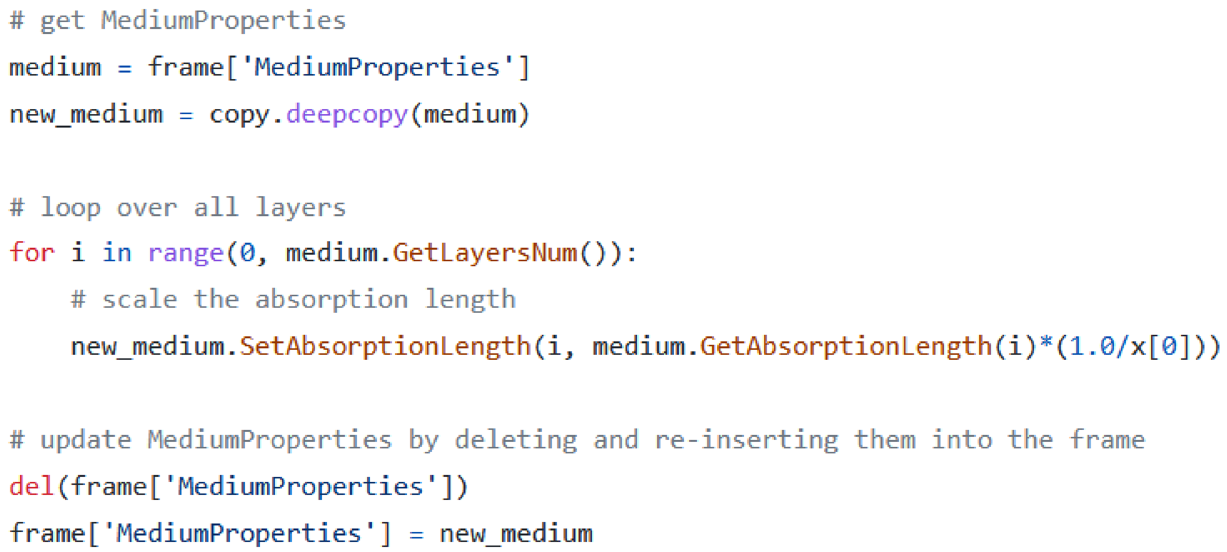
Parametrizations + sampling ranges used for the first snowstorm production sets

SnowStorm – Parametrization(s)

- Parametrize the effect of e.g., a global absorption scaling
- Do the actual perturbation/transformation of the detector/medium properties
- Use an API to access + modify the internal photon propagator functions used for modelling the detector/medium properties
  - I3CLSimFunction(s)

➢ Snowstorm parametrizations currently only implemented for CLSim
  ➢ Should be extendable to PPC if it provides a similar interface

Example of the global absorption scaling:



```
# get MediumProperties
medium = frame['MediumProperties']
new_medium = copy.deepcopy(medium)

# loop over all layers
for i in range(0, medium.GetLayersNum()):
    # scale the absorption length
    new_medium.SetAbsorptionLength(i, medium.GetAbsorptionLength(i)*(1.0/x[0]))

# update MediumProperties by deleting and re-inserting them into the frame
del(frame['MediumProperties'])
frame['MediumProperties'] = new_medium
```

SnowStorm – Parametrizations Overview

- IceWavePlusModes: icewave ice-model using Fourier decomposition to vary the per-layer scattering and absorption lengths (paper)
  - Located in `ice-models/python/icewave`

- Ice absorption/scattering:
  - Global scaling of the per-layer absorption/scattering coefficients (lengths scaled by 1/x)

- Anisotropy scale:
  - Scaling of anisotropy coefficients k1 and k2 (fixed anisotropy axis)

- DOM efficiency:
  - Direct scaling of the DOMs wavelength acceptance
  - Multiple options here, tried to choose the “most solid one”

- HoleIce (MSU + Unified model):
  - Change of the DOMs angular acceptance function (but keeping the normalization constant)

Systematic	Parameter(s)
IceWavePlusModes	12 amplitude + 12 phase shifts
Scattering	c_{scat}
Absorption	c_{abs}
AnisotropyScale	$c_{anisotropy}$
DOMEfficiency	ϵ_{opt}
HoleIceForward_Unified	p0, p1
HoleIceForward_MSU	p1, p2

Table: Overview of all available systematic perturbations for SnowStorm

SnowStorm “Requirements”

Big thanks to Jakob van Santen and Claudio for their help!

- Internal functions for modelling the ice + detector properties need to be accessible via an API
  - modifiable for perturbations
- Serializable detector + medium properties
  - Book-keeping
- Re–initializable photon propagation kernel
  - This is done after each bunch of frames is processed with a specific ice-model/detector perturbation: reduce re-initialization overhead as much as possible
- Ideally there should be only one (or one preferred) way for a parametrization to modify the medium/detector properties
  - Chosen such that it will not conflict with other variations

➢ The perturber just applies the parametrizations and writes updated detector/medium properties to the M frame
  - ➢ The actual re-initialization is done within the executable script which “just” access these information

Summary & Outlook

- Internal functions for modelling the ice + detector properties need to be accessible via an API
  - modifiable for perturbations
- Serializable detector + medium properties
  - Book-keeping
- Re–initializable photon propagation kernel
  - This is done after each bunch of frames is processed with a specific ice-model/detector perturbation: reduce re-initialization overhead as much as possible
- Ideally there should be only one (or one preferred) way for a parametrization to modify the medium/detector properties
  - Chosen such that it will not conflict with other variations

- ➢ SnowStorm MC production using CLSim available in icetray since combo.V01-00-00
- ➢ So far only tested/run with Spice3.2.1 as baseline ice-model
  - ➢ Existing parametrizations should also work with new ice-models
- ➢ CLSim Interface for varying the birefringence strength of SpiceBFRv1/2 exist (Alexander H.)
  - ➢ No parametrization yet
- ➢ Perturber + Parametrizations should be extendable to PPC if a similar API for accessing the medium properties and updating/reinitializing the propagation kernel exists

Appendix

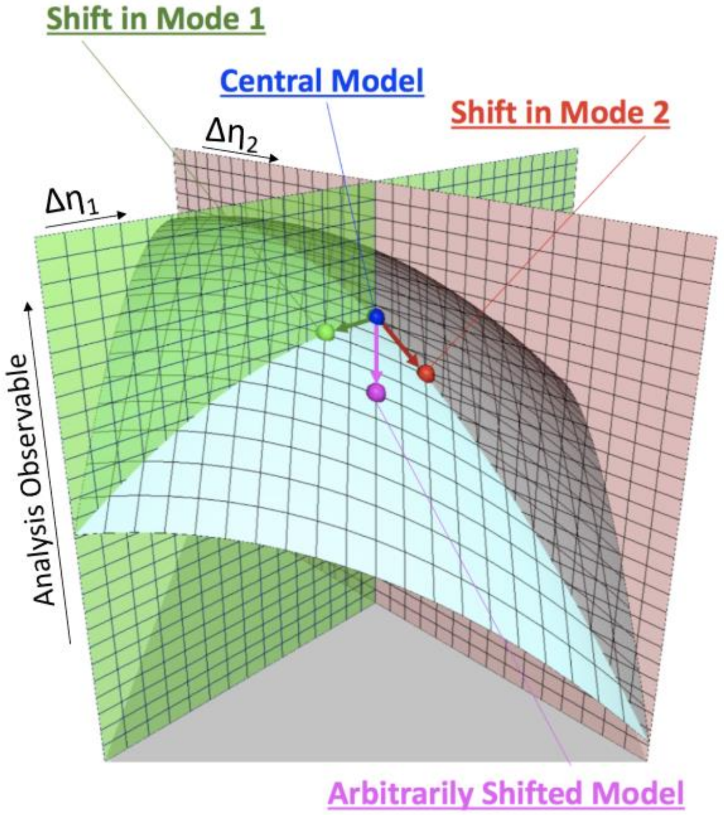
SnowStorm Aplication – Gradient Method

- Use the Snowstorm MC set to extract the gradient in analysis space (chapter 2 in the paper)

- Assumptions:
  1. “effects of systematic uncertainties on analysis variables are sufficiently small that they can be treated perturbatively”
  2. “statistical uncertainty on Monte Carlo event counts is very small compared to that on the data”

- “If the effects of the nuisance $\vec{\eta}$ parameters are perturbative, this implies that the distribution function at any $\vec{\eta}$ can be written as a Taylor expansion around the central distribution:”

$$\psi_{\vec{\rho},\vec{\eta}} = \Psi_{\vec{\rho}} + \vec{\eta}.\vec{\nabla}_{\eta}\left[\psi_{\vec{\rho},\vec{\eta}}\right]_{\vec{\eta}=\vec{0}} + \mathcal{O}(\eta^2),$$

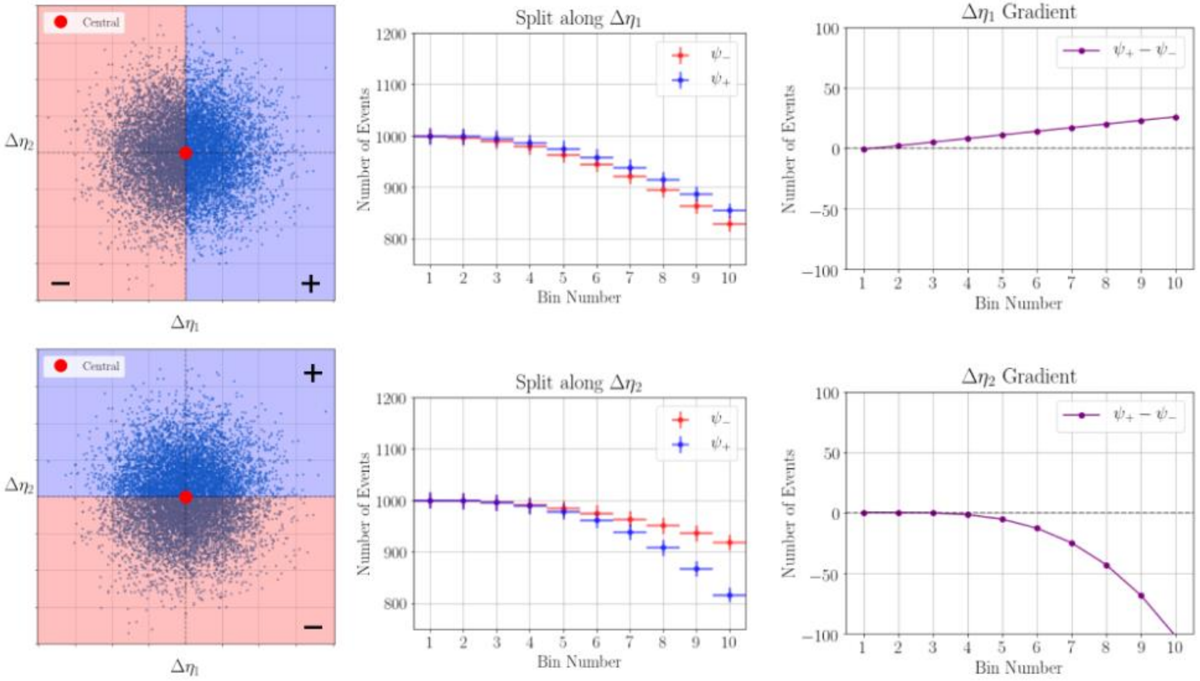


SnowStorm Aplication – Gradient Method

- If “the prediction of the SnowStorm ensemble and the central model are identical $\mathcal{O}(\eta^2)$ effects” (and “comparison show[s] they are equivalent within the available statistical uncertainty”), this expression reduces to

$$\psi_{\vec{\rho},\vec{\eta}} = \Psi_{\vec{\rho}} + \vec{\eta}.\vec{G}_{\vec{\rho}}, \qquad \vec{G}_{\vec{\rho}} \equiv \vec{\nabla}_{\eta} \left[\psi_{\vec{\rho},\vec{\eta}}\right]_{\vec{\eta}=\vec{0}} .$$

- The “magic” part is to extract the gradient from the SnowStorm Ensemble:
  - Cutting the ensemble in half
  - By weighting: “consider constructing a prediction where each event in the SnowStorm ensemble is weighted by a factor of η_i”



SnowStorm Application – Reweighting Method

- Re-weighting of the SnowStorm parameters within the MC set to some (arbitrary) distribution on analysis level (e.g. normal dist.)
  - Directly yields a MC prediction for a specific choice of nuisance parameters, i.e. the current hypothesis, in the fit

- Can easily be achieved by adding an additional weight to each MC event:

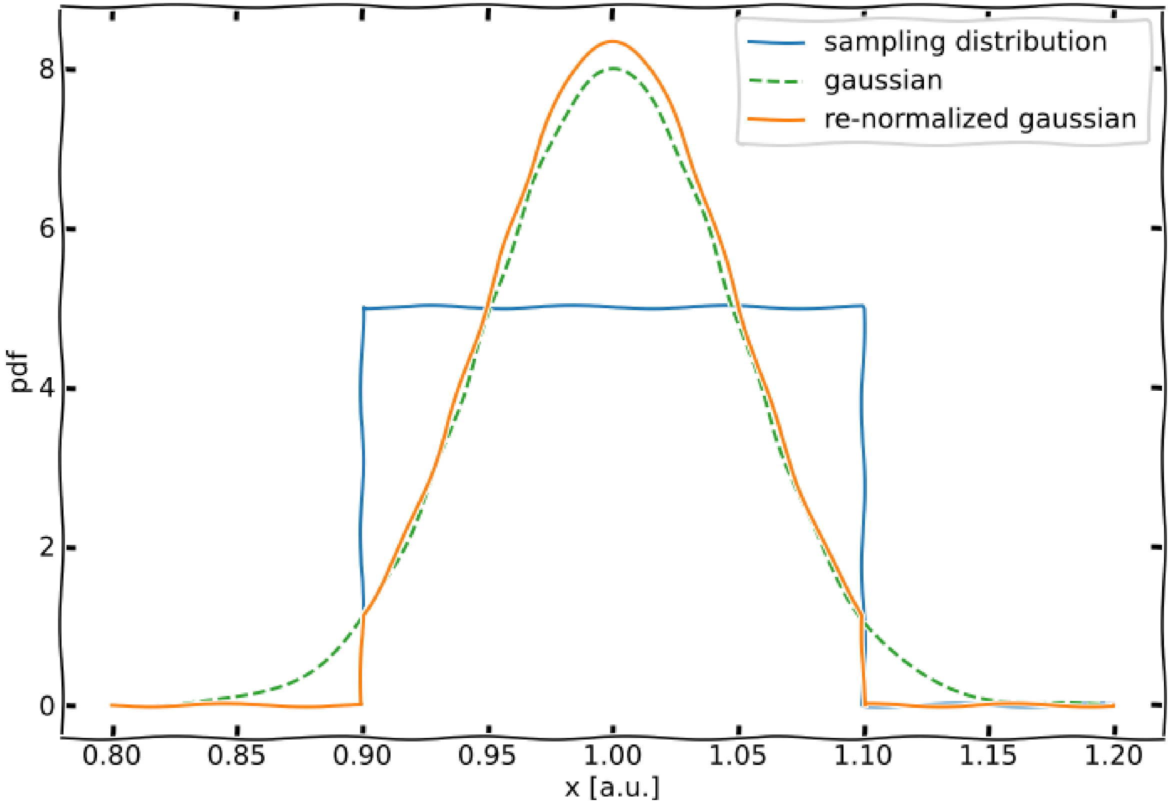
$$w_i = \frac{p_{sys}\,(sys_i, \xi_i)}{p_{sys}^{sim}(sys_i)} \cdot \ldots$$

p_{sys} : reweighting distribution

p_{sys}^{sim} : sampling distribution

sys_i : event′s systematic value

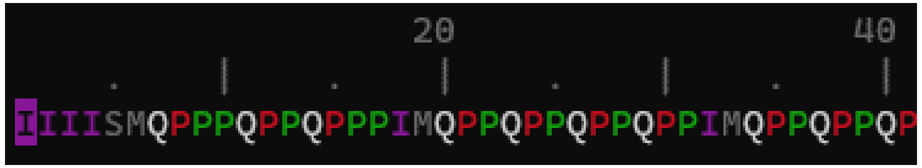
ξ_i : nuisance parameter



SnowStorm Simulation Sets – How To use them

- All SnowStorm parameters, ice-model settings, etc. are stored in M and S frames
  - All SnowStorm simulations can be processed/used in the same way as previous/other NuGen simulations
  - ➢ Make sure to not drop M+S frames during further processing of the files

- You can use any recent combo/icetray version for processing that can deal with custom frame types
  - combo/V01-00-02 was used for production which is available in /cvmfs/icecube.opensciencegrid.org/…

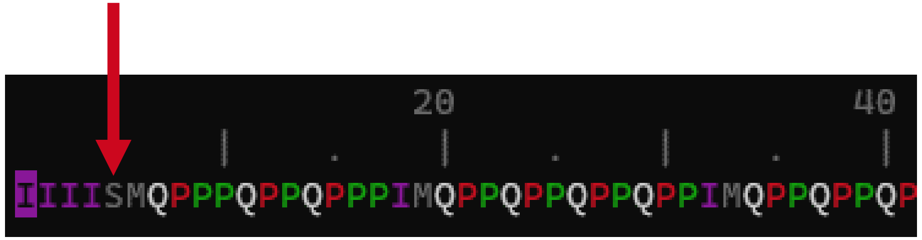


```
                  20                  40
    .    |    .    |    .    |    .    |
IIIISMQPPPQPPQPPPIMQPPQPPQPPQPPIMQPPQPPQP
```

SnowStorm Simulation Sets – How To use them

- All SnowStorm parameters, ice-model settings, etc. are stored in (new) M and S frames
  - All SnowStorm simulations can be processed/used in the same way as previous/other NuGen simulations
  - ➢ Make sure to not drop M+S frames during further processing of the files

- You can use any recent combo/icetray version for processing that can deal with custom frame types
  - combo/V01-00-02 was used for production which is available in /cvmfs/icecube.opensciencegrid.org/…



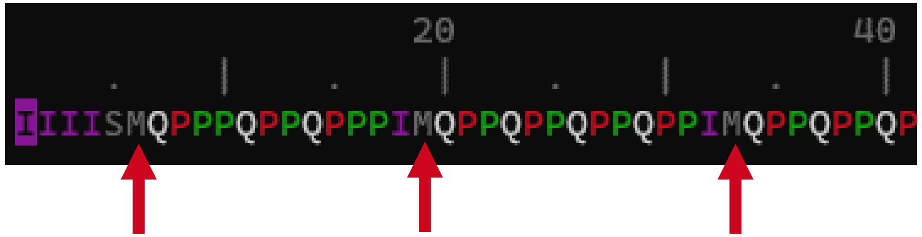
```
                   20                  40
          |    .    |    .    |    .    |
IIIISMQPPPQPPQPPPPIMQPPQPPQPPQPPIMQPPQPPQP
```

- S-Frame:
  - Only one per file
  - Simulation frame for bookkeeping:
    - initialized SnowStorm parametrizations (parameters)
    - SnowStorm sampling distributions

SnowStorm Simulation Sets – How To use them

- All SnowStorm parameters, ice-model settings, etc. are stored in (new) M and S frames
  - All SnowStorm simulations can be processed/used in the same way as previous/other NuGen simulations
  - ➢ Make sure to <u>not drop M+S frames </u>during further processing of the files

- You can use any recent combo/icetray version for processing that can deal with custom frame types
  - combo/V01-00-02 was used for production which is available in /cvmfs/icecube.opensciencegrid.org/…



```
                  20                  40
    .    |    .    |    .    |    .    |
IIIISMQPPPQPPQPPPIMQPPQPPQPPQPPIMQPPQPPQP
```

- S-Frame:
  - Only one per file
  - Simulation frame for bookkeeping:
    - initialized SnowStorm parametrizations (parameters)
    - SnowStorm sampling distributions

- M-Frame:
  - One per every “bunch” of events
  - Sampled SnowStorm parameters
    - It also tells you how many events/frames the ice-model was applied to during simulation (before triggering)

SnowStorm – Reading the Frame Objects

✓ A „snowstorm parameter wrapper“ exists taking care of all the different frame objects

✓ It creates a “SnowstormParameterDict” with a simple mapping of parameter name → value in the M-Frame

- Like the I3MCWeightDict from NuGen
- Single frame object with all important information for the user
- Works with I3HDFWriter

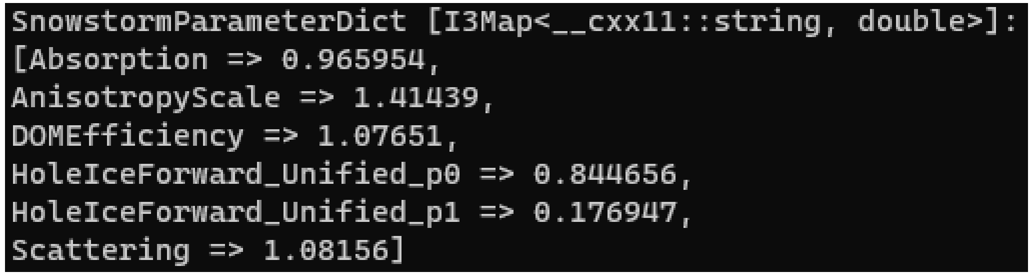
➢ This was not applied for Level2 during initial production but on later analysis level only…

➢ Will add this for the second round of SnowStorm MC production

M(odel)-Frame

- SnowstormEventsPerModel
  - Number of events this ice model was originally being applied to (on generation level, i.e. before triggering)

- SnowstormParameterDict



```
SnowstormParameterDict [I3Map<__cxx11::string, double>]:
[Absorption => 0.965954,
AnisotropyScale => 1.41439,
DOMEfficiency => 1.07651,
HoleIceForward_Unified_p0 => 0.844656,
HoleIceForward_Unified_p1 => 0.176947,
Scattering => 1.08156]
```