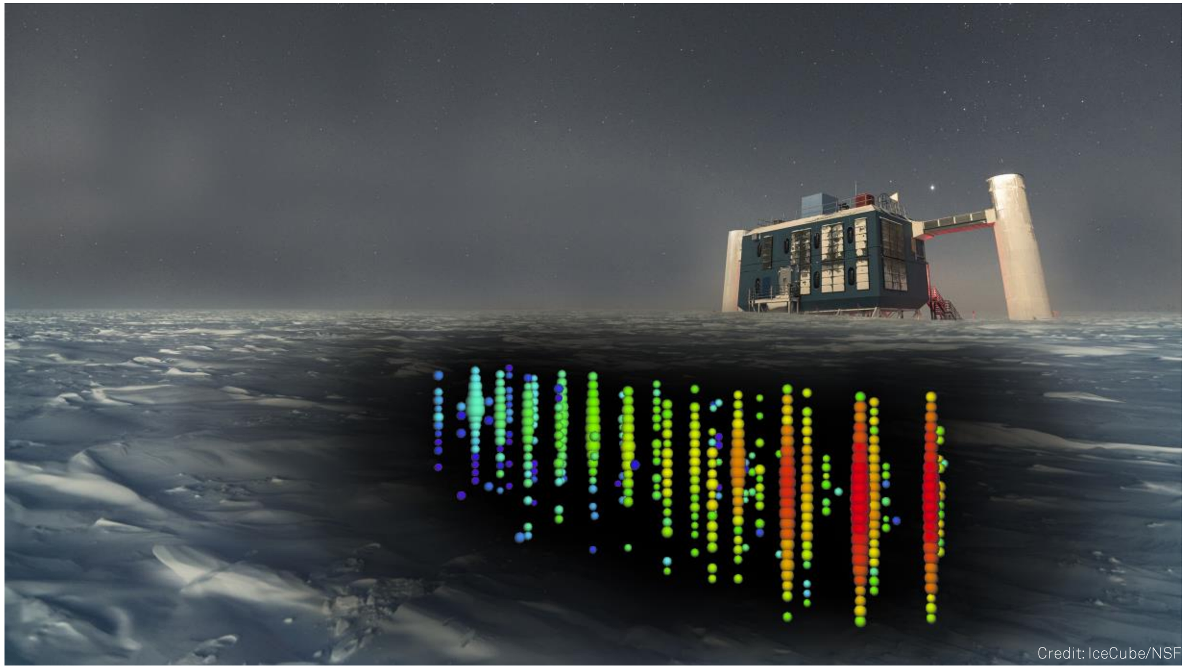
technische universität dortmund

ICECUBE SOUTH POLE NEUTRINO OBSERVATORY

SFB 876 Providing Information by Resource-Constrained Data Analysis

lehrstuhl physik e5

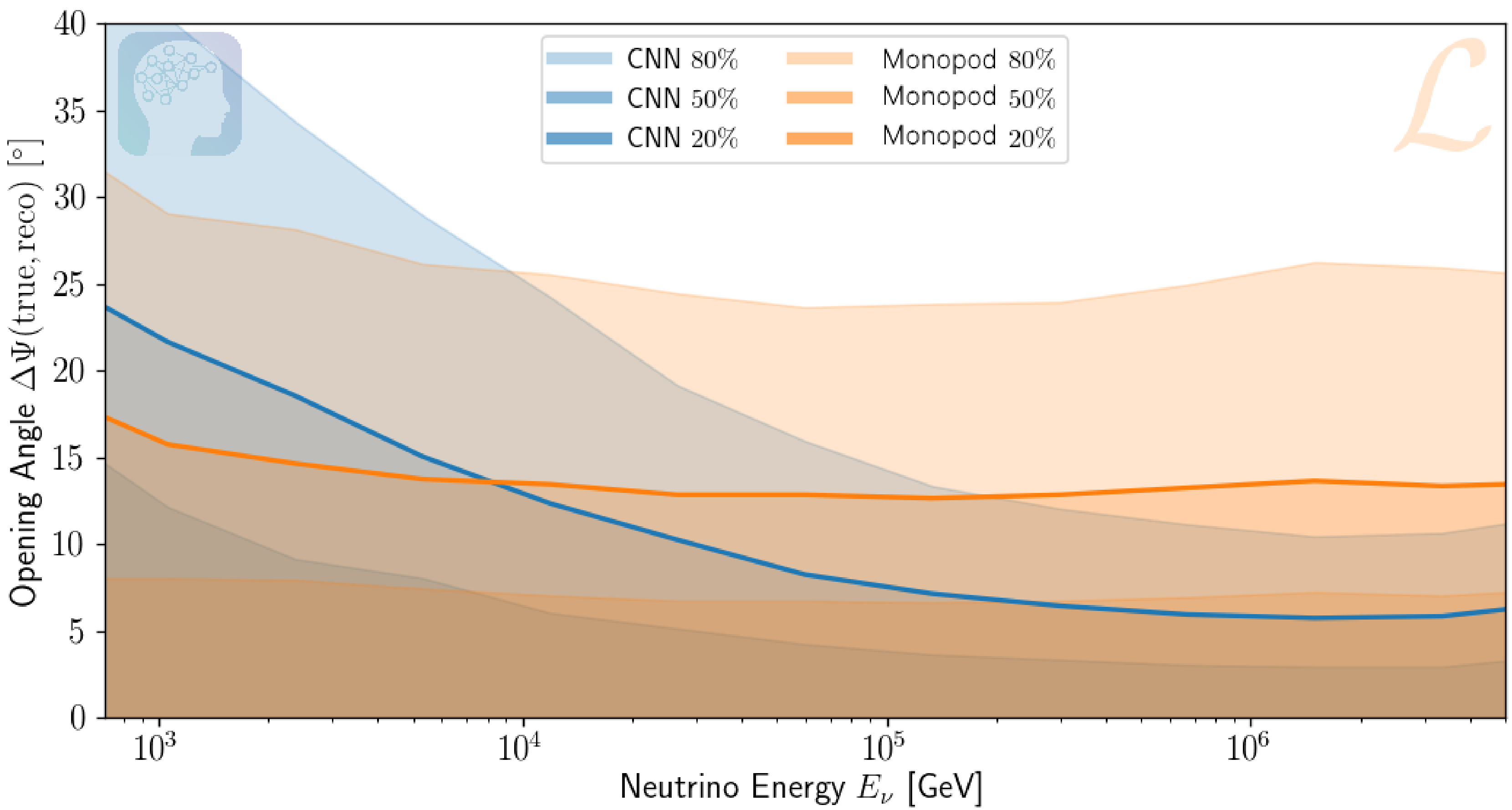


Cascade-Generator: Hackathon Project

Mirco Huennefeld
mirco.huennefeld@tu-dortmund.de

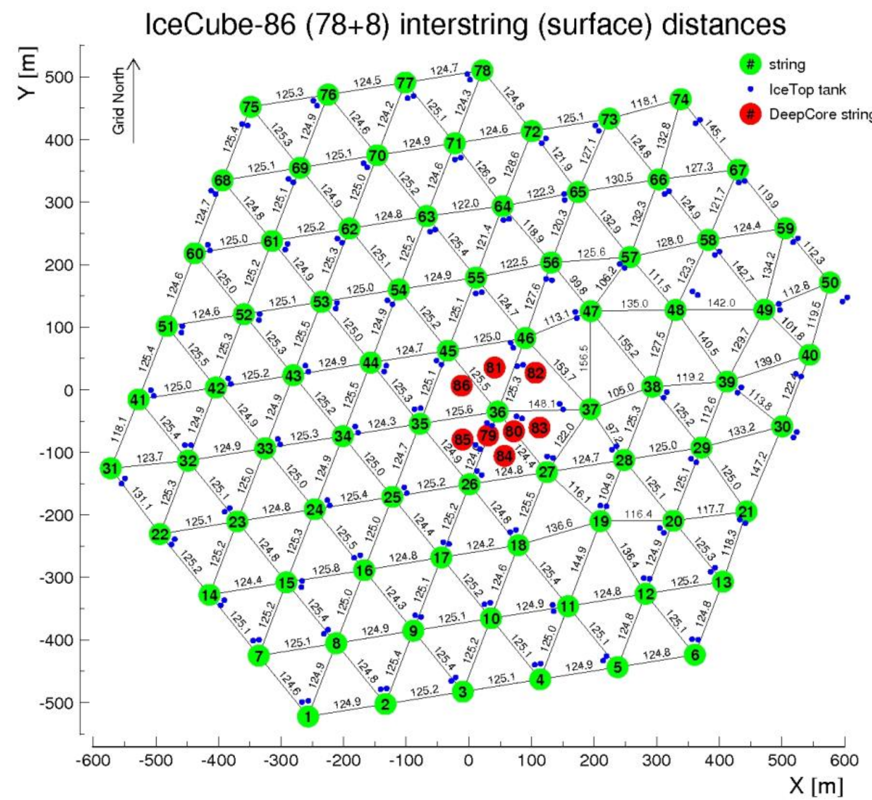
Diffuse Workshop, Tokyo
September 14, 2019

Convolutional Neural Networks – DNN reco



Convolutional Neural Networks – Limitations

- Only translational invariance and locality is used – More information and symmetries available
- Assumptions imposed by CNNs are only approximately met in IceCube
  - Irregularities in detector grid
  - No real translational invariance in observable space
- CNNs "wash out" data:
  - Great for robustness
  - Bad if precise information is needed



Maximum Likelihood Methods

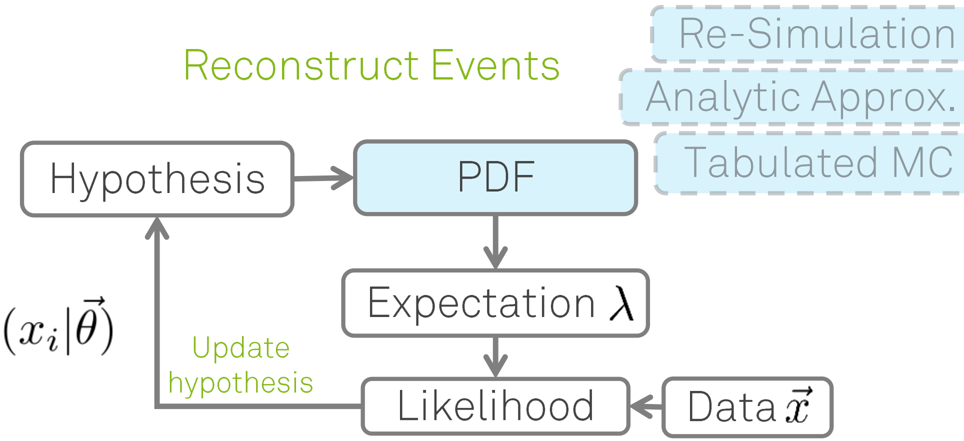
Alter event hypothesis until it matches data
Cascade Hypothesis:

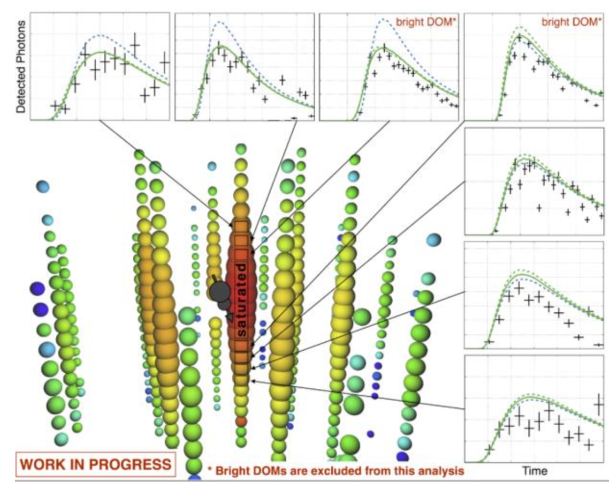
$$\vec{\theta} = (x, y, z, \varphi, \theta, E, t)$$ 7 free parameters

Properties:

$$\mathcal{L}(\vec{x}|\vec{\theta}) = \prod_i p(x_i|\vec{\theta})$$

- ⊕ Physics knowledge is incorporated into the Likelihood and PDF
- ⊕ Exact detector geometry can be used
- ⊕ In theory: optimum of what we can do
- ⊖ Often forced to simplify PDF and Likelihood
- ⊖ Difficult to find global minimum





Credit: HESE team

Cascade Generator – General Idea

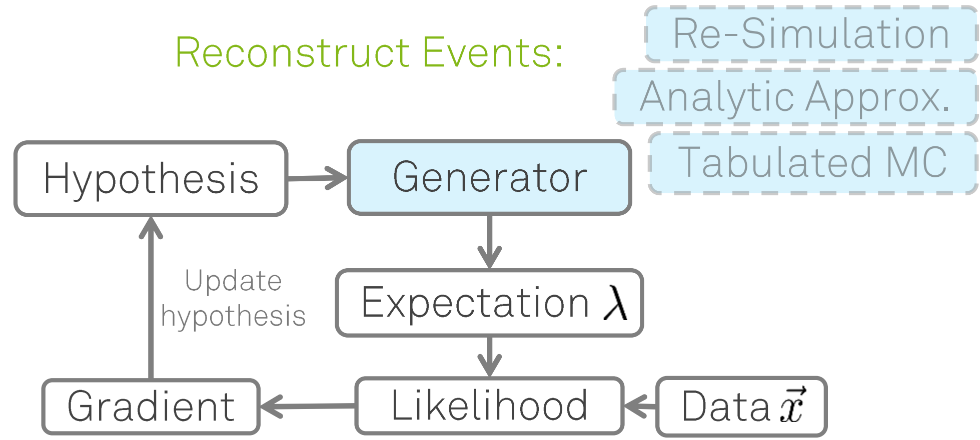
Combine strengths of neural networks and maximum-likelihood methods

Properties of Generator NN:

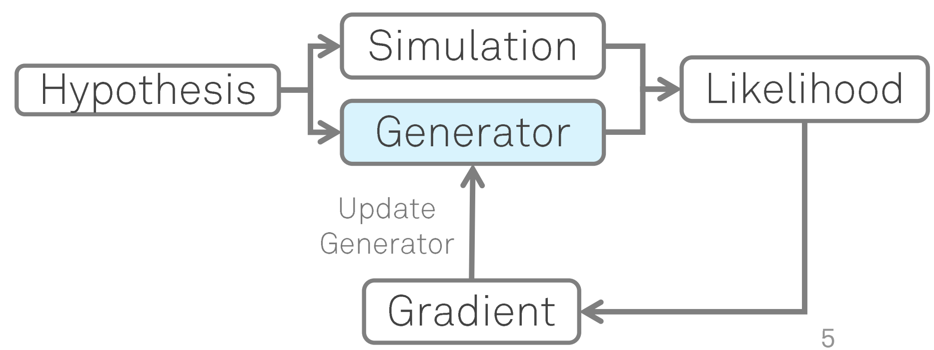
- Fast approximation of MC simulation
- No critic NN or randomness necessary
- Physics knowledge & symmetries can easily be included
- Exact detector geometry can be used
- Use in reverse mode for reconstruction

→ Fully differentiable: Gradient Descent

Reconstruct Events:



Train Generator:



Cascade Generator – Inclusion of Information

- Light yield at each DOM only depends on cascade hypothesis:
  - Weight sharing with 1x1 convolutional kernels
  - Translational invariance included in relative input parameters to DOM

→ Exact detector geometry and translational invariance can be exploited (Well suited for IceCube Upgrade/Gen2 including different DOM types)

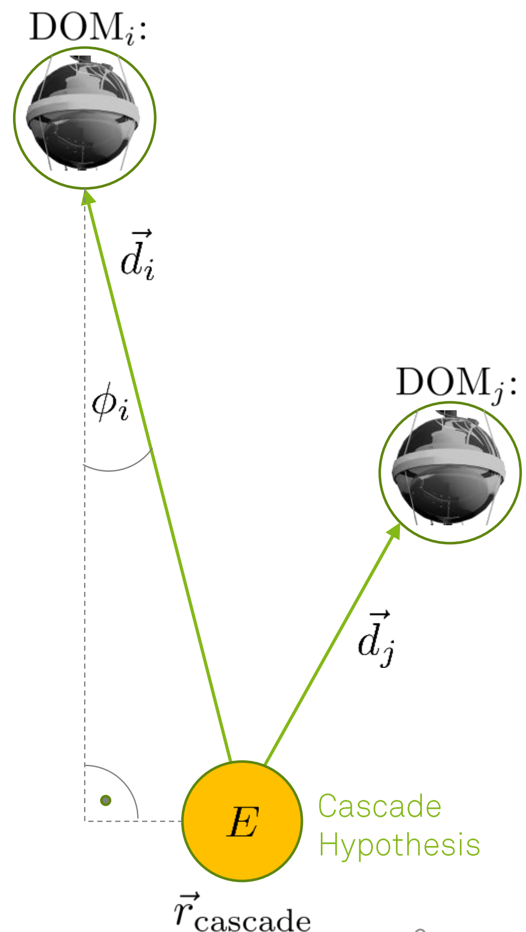
Input per DOM:

$\vec{d}, \phi, \ldots$ } Relative parameters: include geometry and translational invariance

φ_{cascade}
θ_{cascade}
$\vec{r}_{\text{cascade}}$
$E, \ldots$

Weight sharing over DOMs:
Exploit invariances in physics variable space

Additional independent weights per DOM:
Account for asymmetries and inhomogeneities

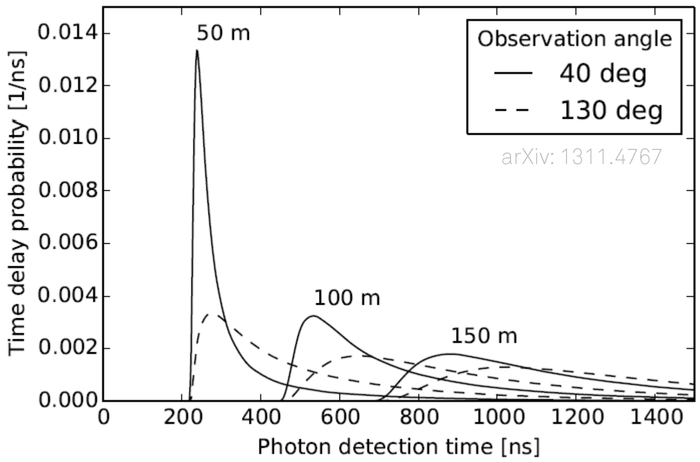


Cascade Generator – Inclusion of Information

- Examples how specific information can be included:
  - Energy scaling:

    Light yield scales linearly with cascade energy

    Apply this scaling before adding noise and detector response
  - Parameterization of waveforms at each DOM
- Can decouple physics and detector effects

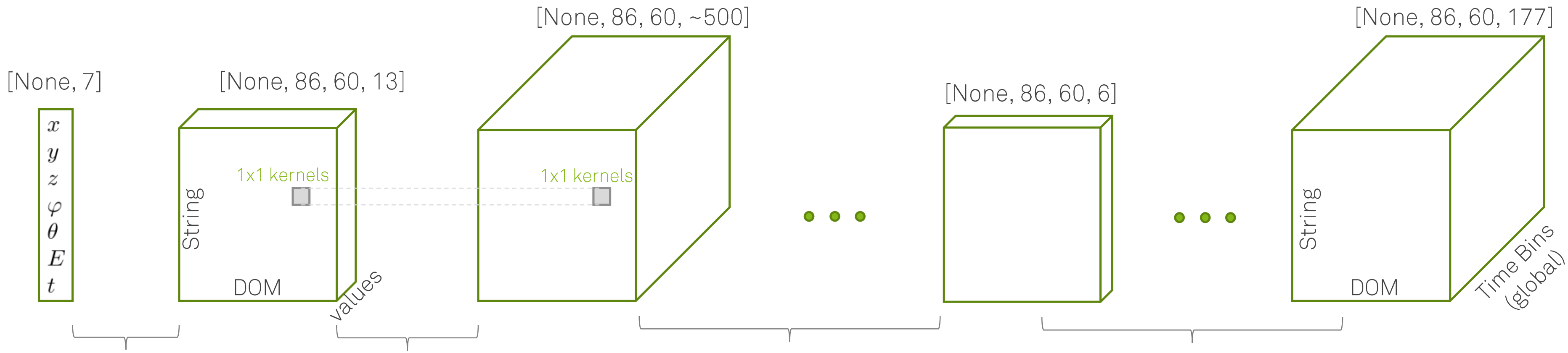


arXiv: 1311.4767

→ Easier to include information in forward direction when not convolved with detector response yet

→ We know how to do this (we simulate the data!)

Cascade Generator – Architecture



Calculate relative input values for each DOM

Locally connected 2D convolution (1x1 kernels)

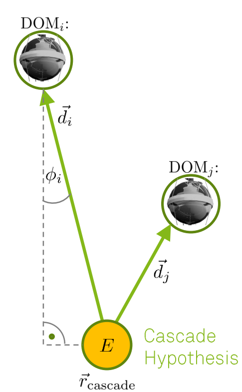
2D convolutions with weight sharing (1x1 kernels)

“Implicit Autoencoder”: Bottleneck + 2D convolutions with weight sharing (1x1 kernels)

Input values per DOM:

$\vec{d}, \phi, \ldots$ } Relative parameters: include geometry and translational invariance

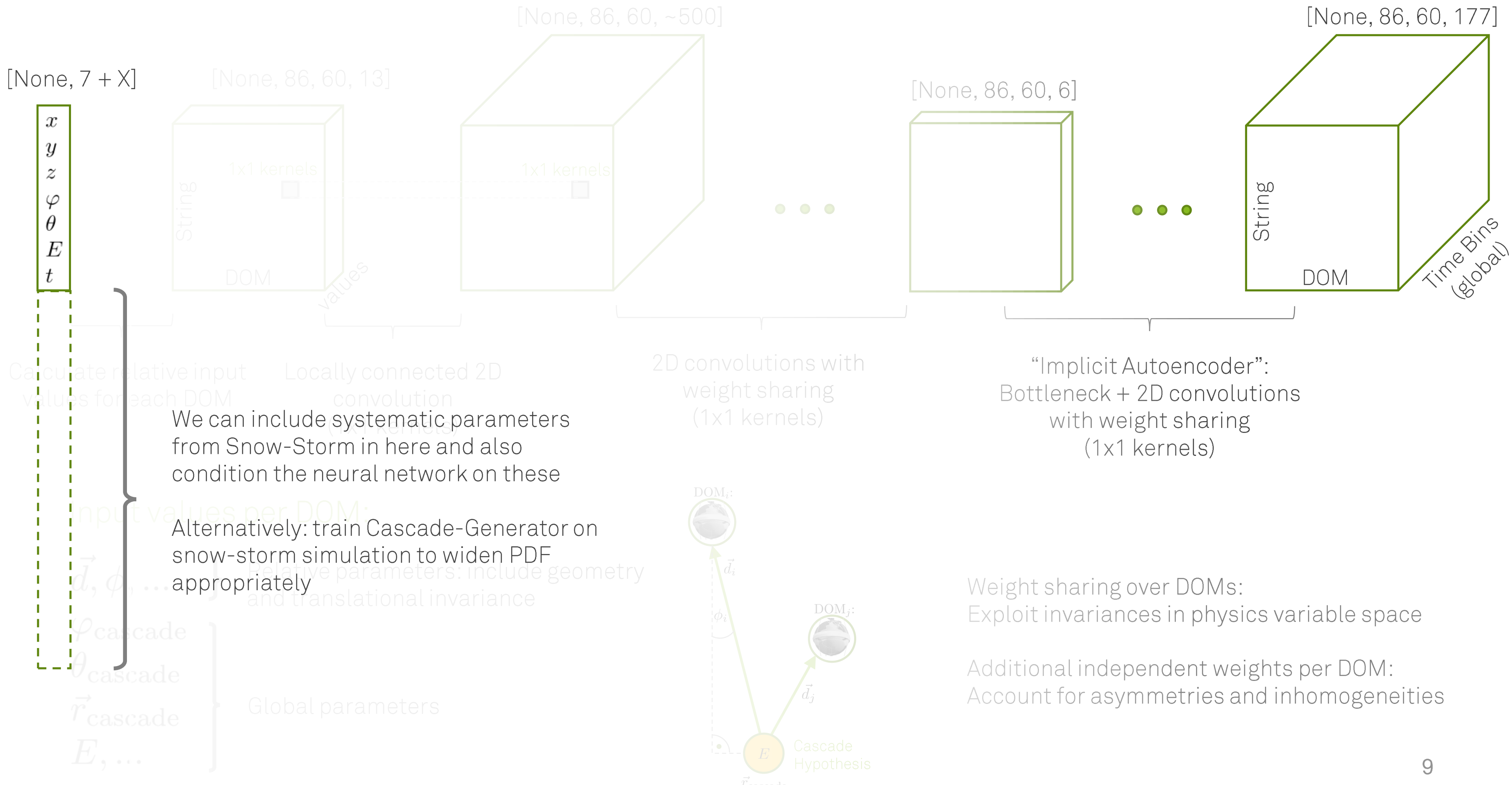
φ_{cascade}, θ_{cascade}, $\vec{r}_{\text{cascade}}$, $E, \ldots$ } Global parameters



Weight sharing over DOMs:
Exploit invariances in physics variable space

Additional independent weights per DOM:
Account for asymmetries and inhomogeneities

Cascade Generator – Architecture



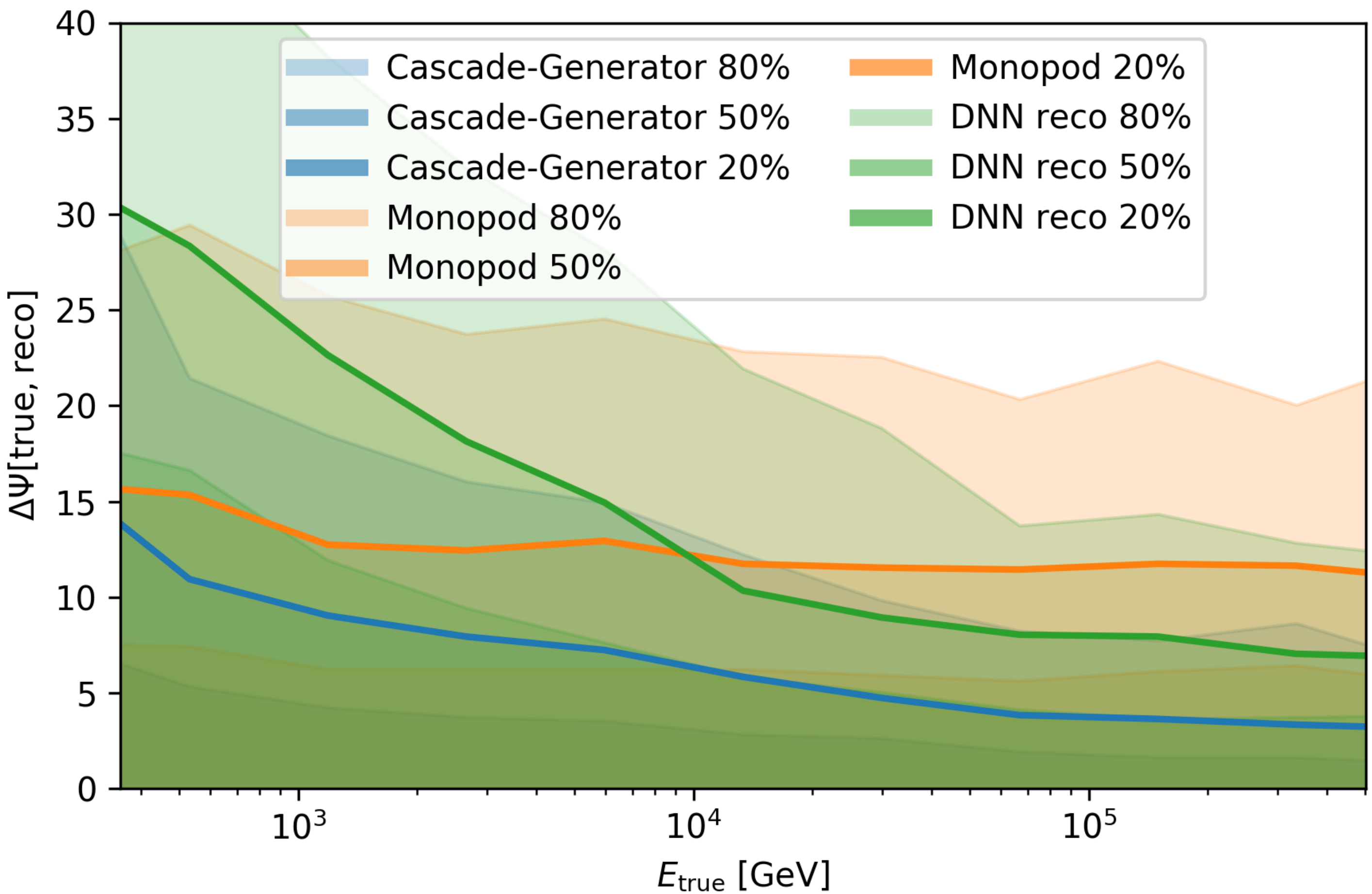
We can include systematic parameters from Snow-Storm in here and also condition the neural network on these

Alternatively: train Cascade-Generator on snow-storm simulation to widen PDF appropriately

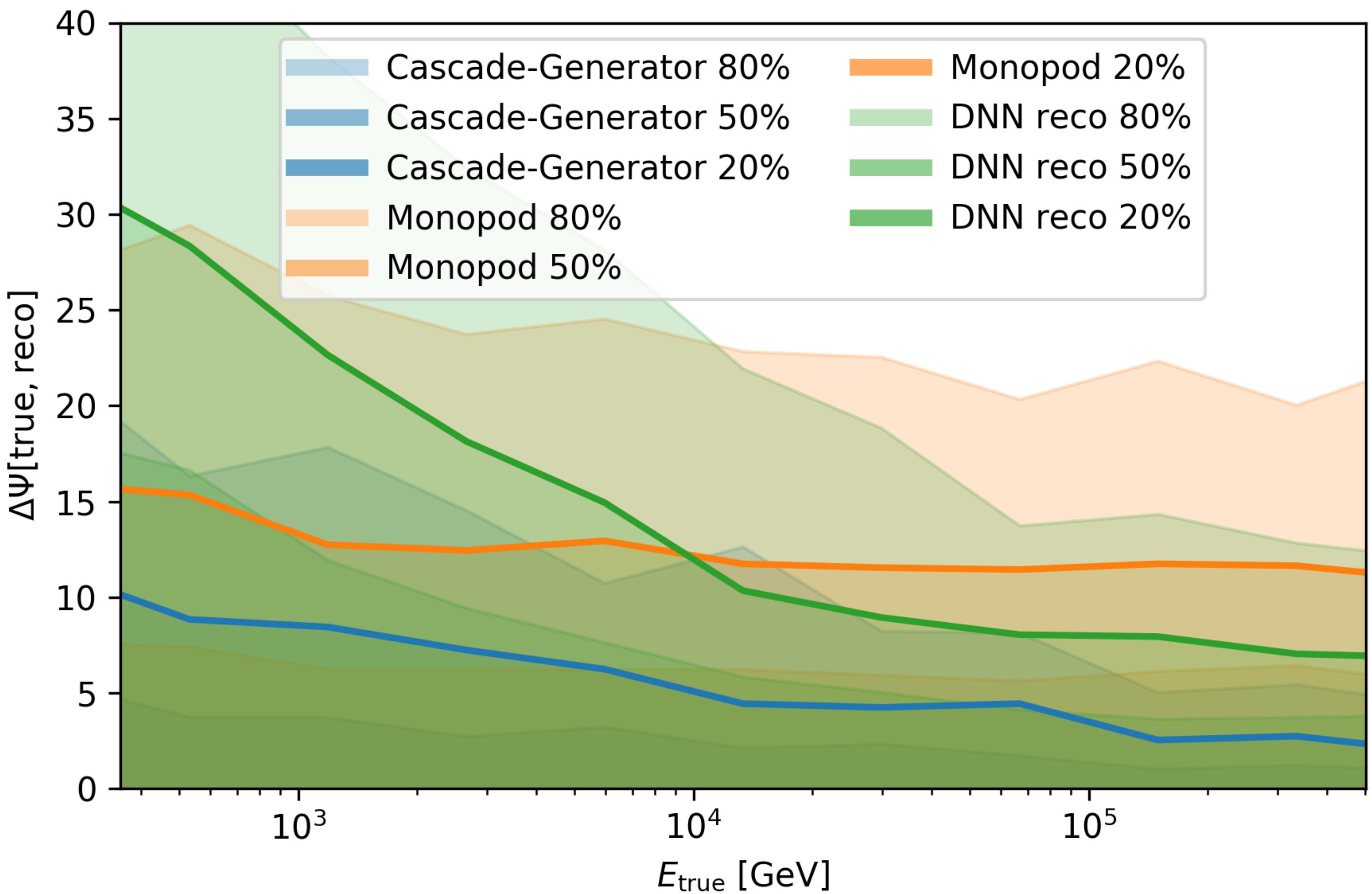
Weight sharing over DOMs:
Exploit invariances in physics variable space

Additional independent weights per DOM:
Account for asymmetries and inhomogeneities

Cascade Generator – Results (Deep Core included)



Cascade Generator – Results (Deep Core included | MC Seed)



Cascade Generator – Summary & Outlook

- Cascade-Generator as a combination of MLE and DL
- Solves issues of standard DNNs and is applicable to IceCube Upgrade/Gen2
- Significant improvement in comparison to Monopod and DNN reco: Where is theoretical limit?
- Current Limitations:
  - Time resolution is currently limited to 20ns
  - Saturation and calibration errata is not handled yet
- Posterior Sampling
  - MCMC to sample posterior and to obtain uncertainty estimates
- Create photonics-service as drop-in replacement for spline tables
  - Use in Monopod/Millipede/...

  } Focus on this for the Hackathon
- Train for minimal ionizing muons
  - All event hypotheses as linear combinations of these + cascades

Cascade Generator – Hackathon: Motivation

Spline Tables:

- Have considerate limitations
- candidate for the discrepancy between Monopod's and Cascade-Generator's resolution
- Affect a lot of reconstruction methods in IceCube (not just cascades!)

Cascade-Generator

- Some of current limitations (saturation and calibration errata) are solved in millipede/monopod framework
- Can output PDFs and gradients, e.g. we can build a photonics-service out of it
- Cascade-Generator-based photonics-service has potential to impact and improve **all** reconstruction methods that rely on spline tables (Monopod/Millipede/…)

Crosscheck for Millipede/Monopod Framework

- Can be used as crosscheck, since we know what resolution we can achieve with the PDFs

Cascade Generator – Hackathon: Tasks

Hackathon Project could be organized in the following steps:

1) Build dummy python-based photonics-service and setup test framework
2) Modify dummy version to output Cascade-Generator PDF (without gradients)
3) Crosschecks and tests:
   - How do PDFs compare to spline tables?
   - Does everything look sane?
   - Can we reconstruct cascades with Monopod + Cascade-Generator backend?
4) Add Gradients to the mix
5) Crosschecks and tests:
   - Implementation and handling of gradients is extremely error prone
   - Verify implementation with tensorflow-based minimization

We will not be able to cover all steps, but a starting point is very valuable!

Backup

Cascade Generator – Training and Evaluation

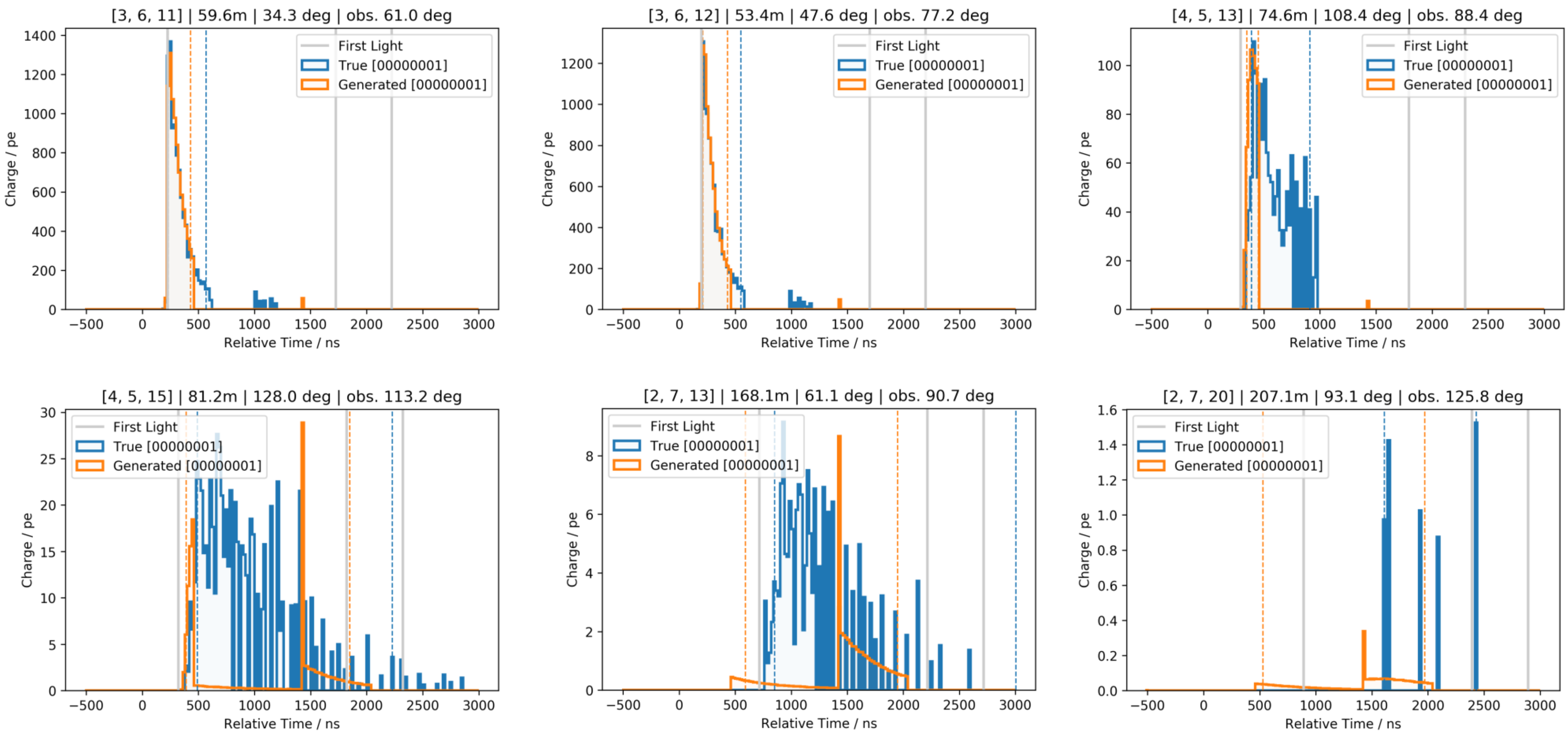
Generator Training

- Generator is currently trained on Cascades simulated with the following settings:
  - Flat (logarithmic) energy distribution between 100 GeV and 1 PeV
  - Over-simulated cascades (n=1000, 100, 10 depending on energy range)
  - Evenly sampled within convex hull around IceCube Detector + 60m
  - Ice model: SpiceLea
  - Trained on full simulation (**not** on MC pulses): includes detector simulation

Generator Evaluation

- Trained Generator is evaluated on MESE NuE sample
- Seeded with Monopod

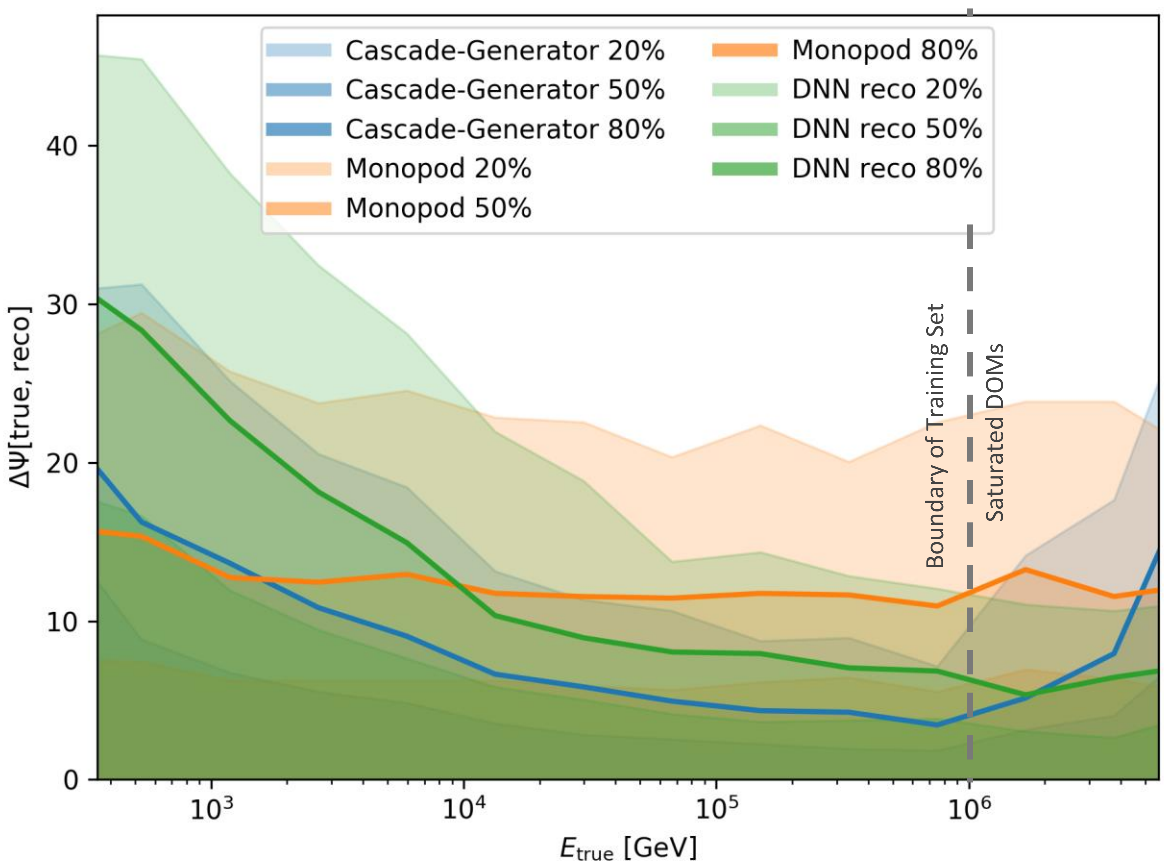
Cascade Generator – Generated Waveforms



Cascade Generator – Current Limitations

- Current Limitations
  - Saturation and calibration errata is not handled yet
  - Time bins are limited to a width of 20 ns due to global time binning
- Local DOM reference time
  - Necessity for local reference time:
    - Can focus on relevant time window for DOM and increase time resolution
    - Challenges: how to shift bins in time in a way that it is differentiable?
    - Currently: map data onto generated bins via a time offset; use interpolation between bins to obtain a meaningful gradient for translational shift in time
    - Possibly use a spline representation? B-splines? Wavelets?

Cascade Generator – Results without DeepCore

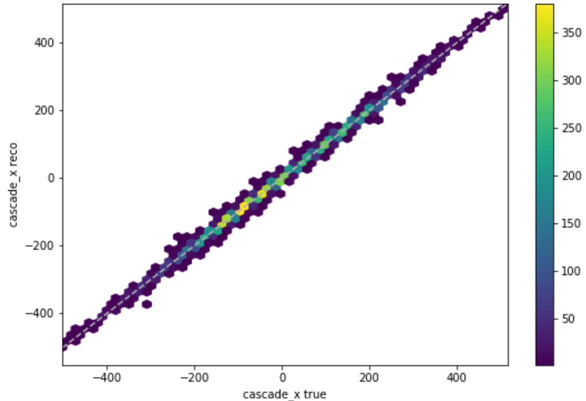


- Training set only goes up to 1 PeV
- Linear light scaling is baked into generator architecture:
  - Should in principle be able to extrapolate
  - However: the model does not know anything about saturation
  - Currently: saturated DOMs are not excluded

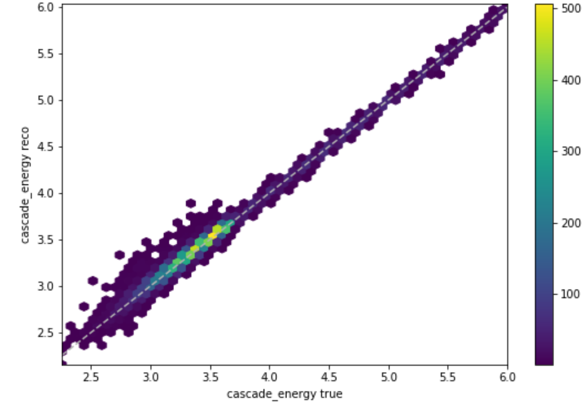
→ Solvable Problem

Cascade Generator – Results without DeepCore (Energy < 1PeV)

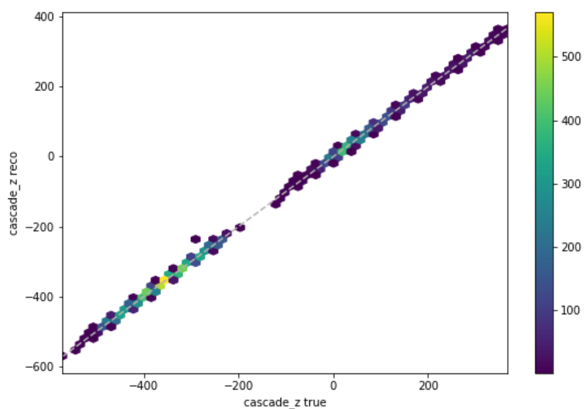
Vertex X



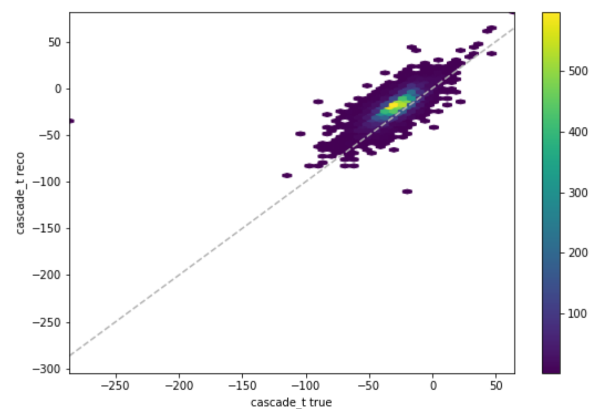
Energy



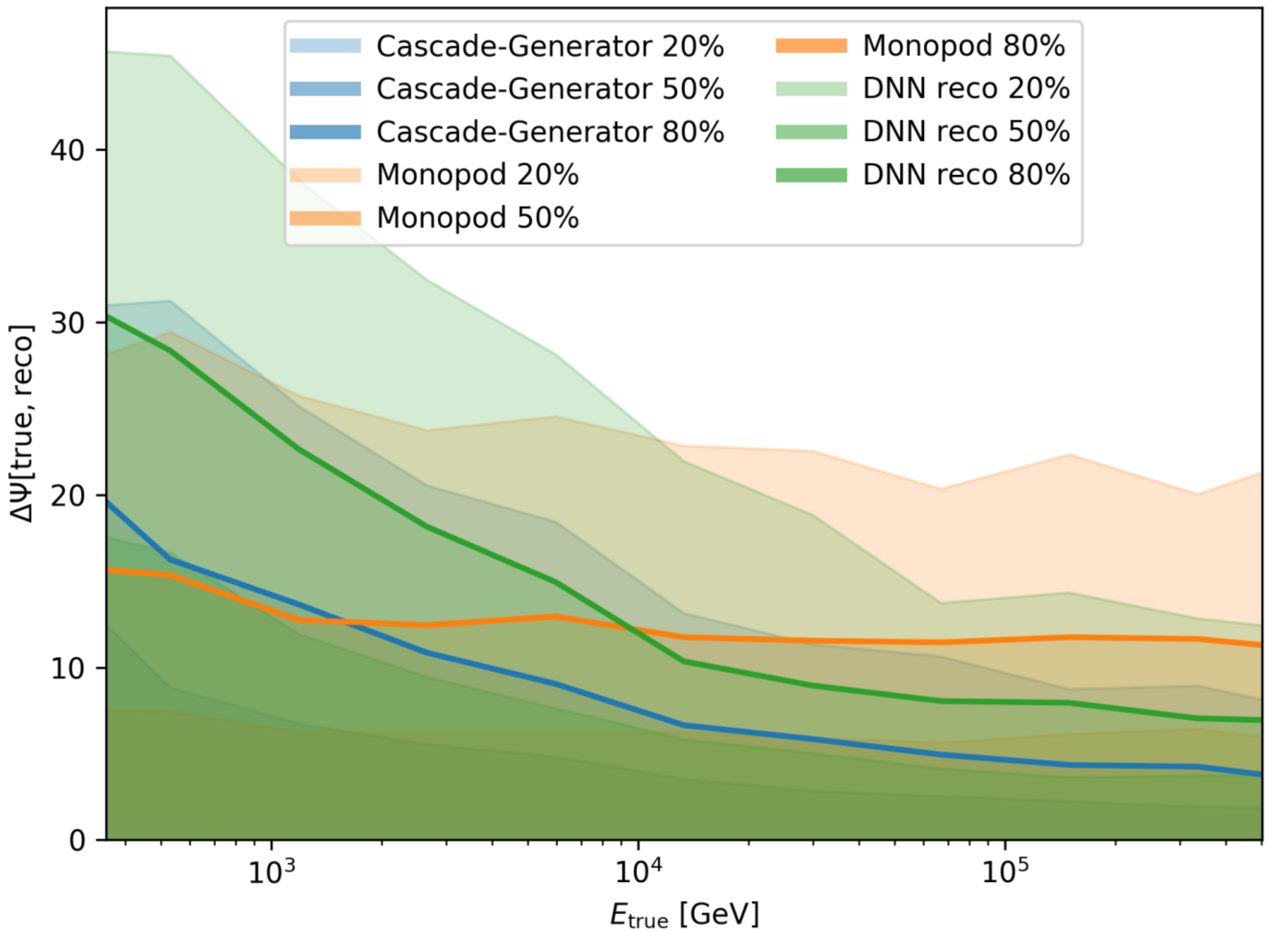
Vertex Z



Vertex time

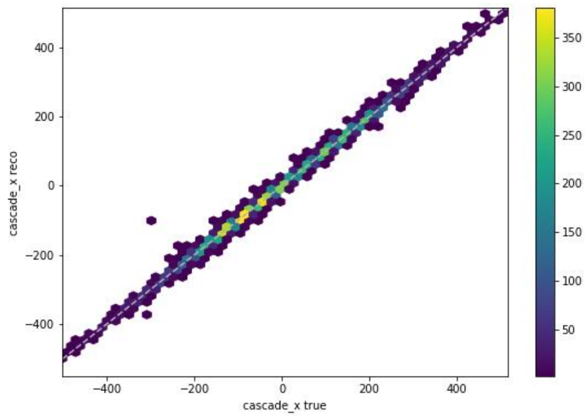


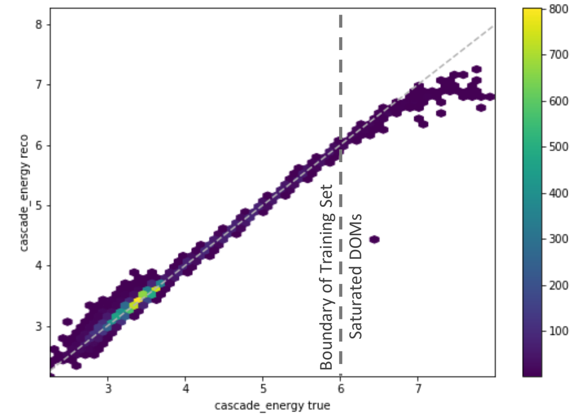
Cascade Generator – Results without DeepCore (Energy < 1PeV)



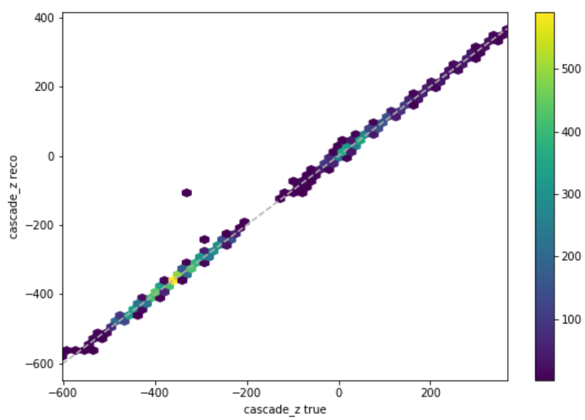
Cascade Generator – Results without DeepCore

Vertex X

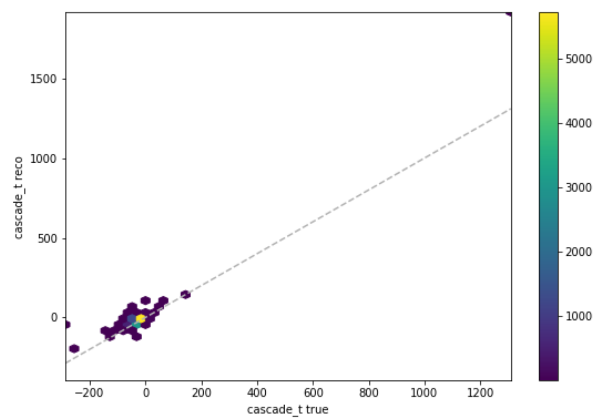




Vertex Z

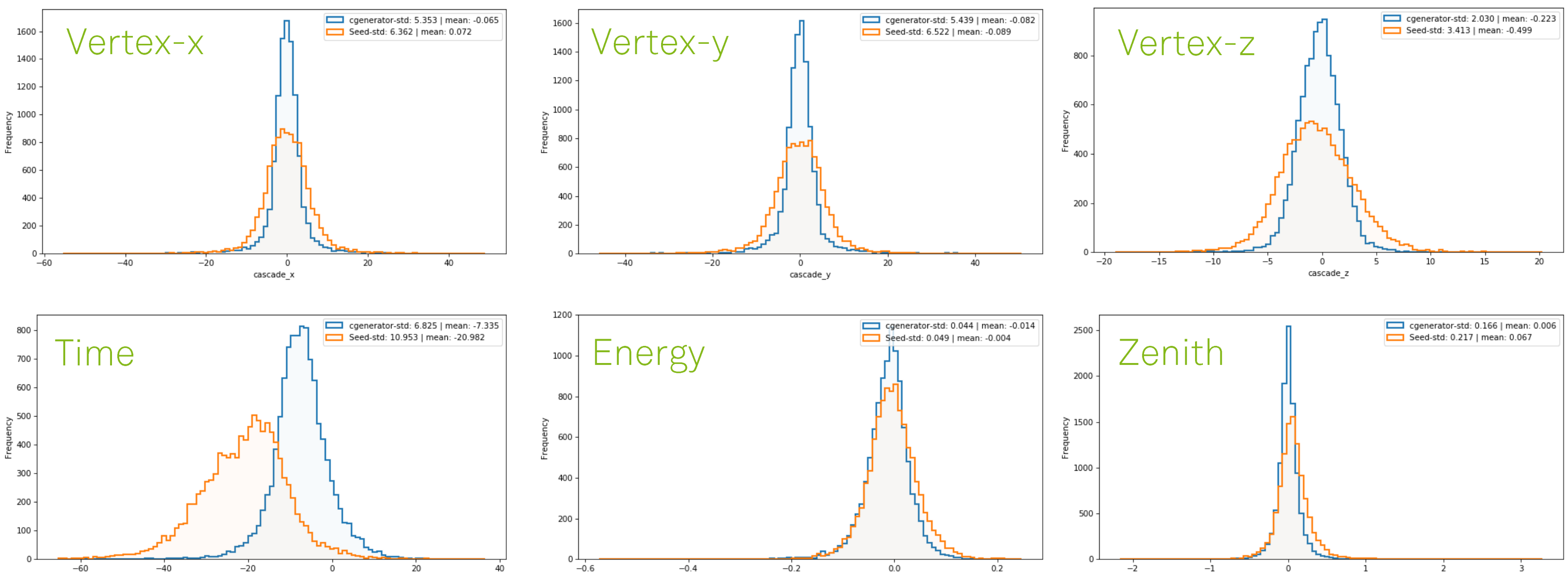


Vertex time

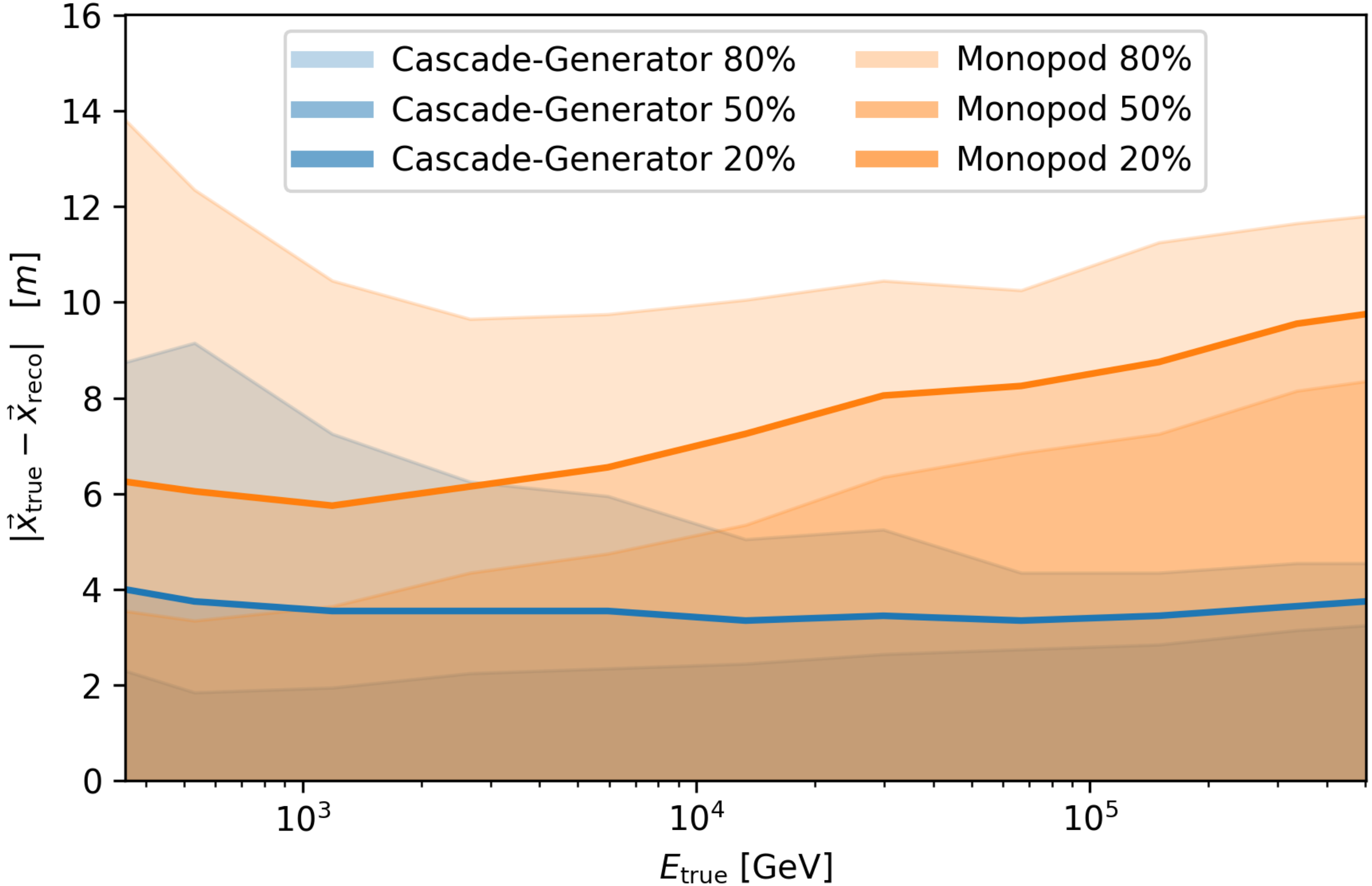


Cascade Generator – Including DeepCore

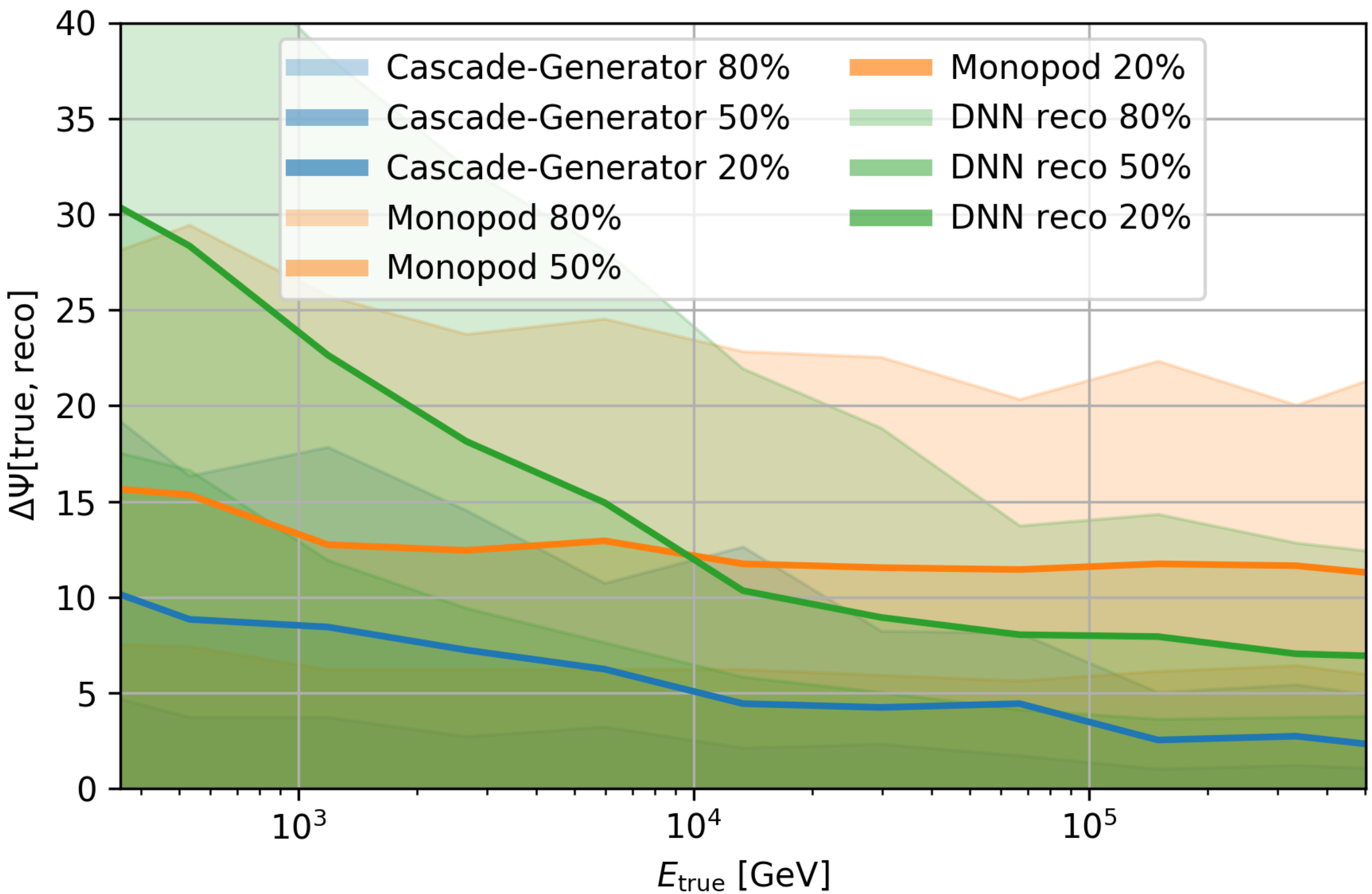
- Next step: include DeepCore DOMs
  - Straight forward to do due to nature of cascade-generator
  - Can handle arbitrary geometry
  - Well suited for IceCube Upgrade/Gen2 including different DOM types (mDOMs, ...)



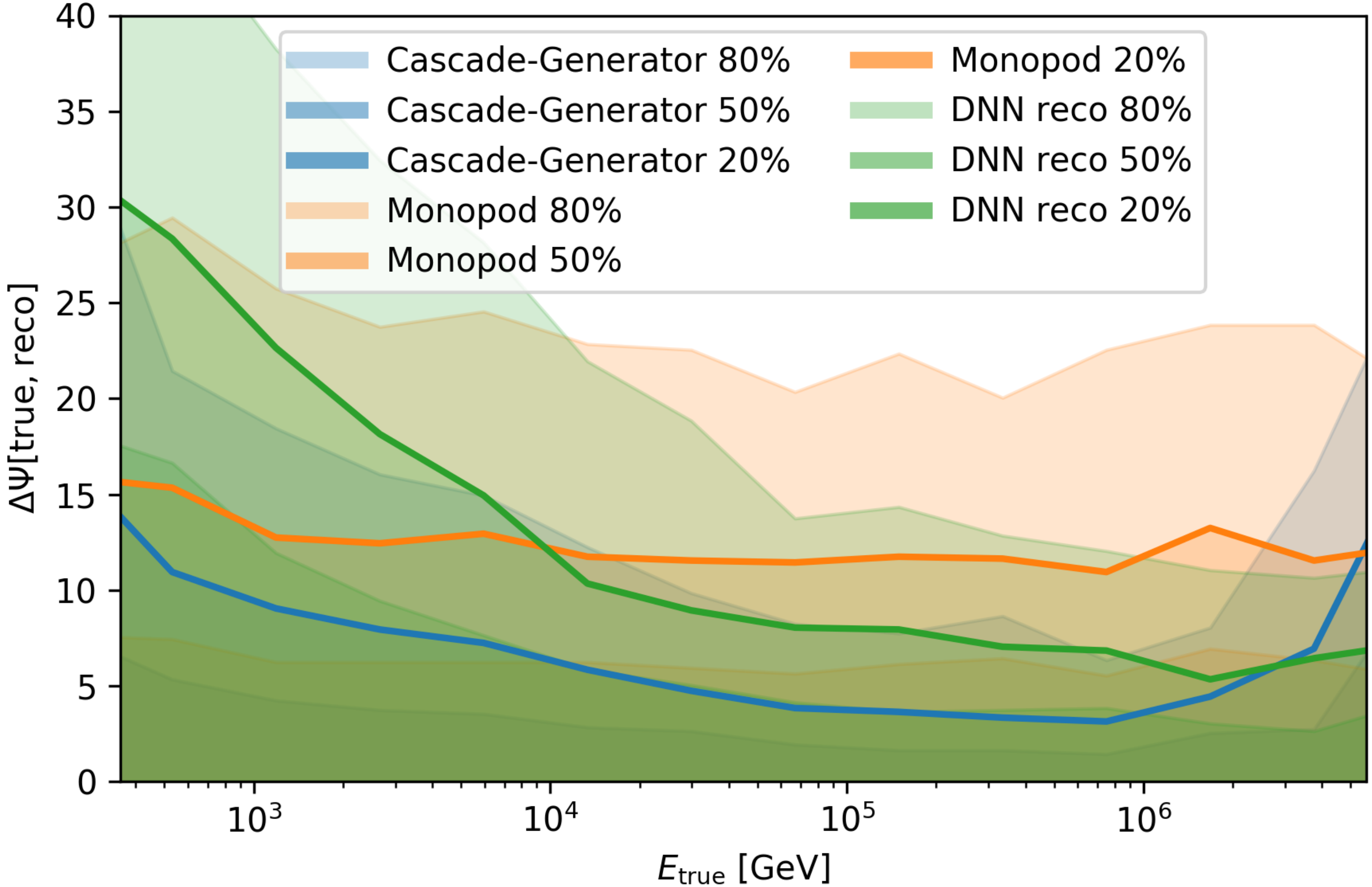
Cascade Generator – Results (Deep Core included)



Cascade Generator – Results (Deep Core included | MC Seed)



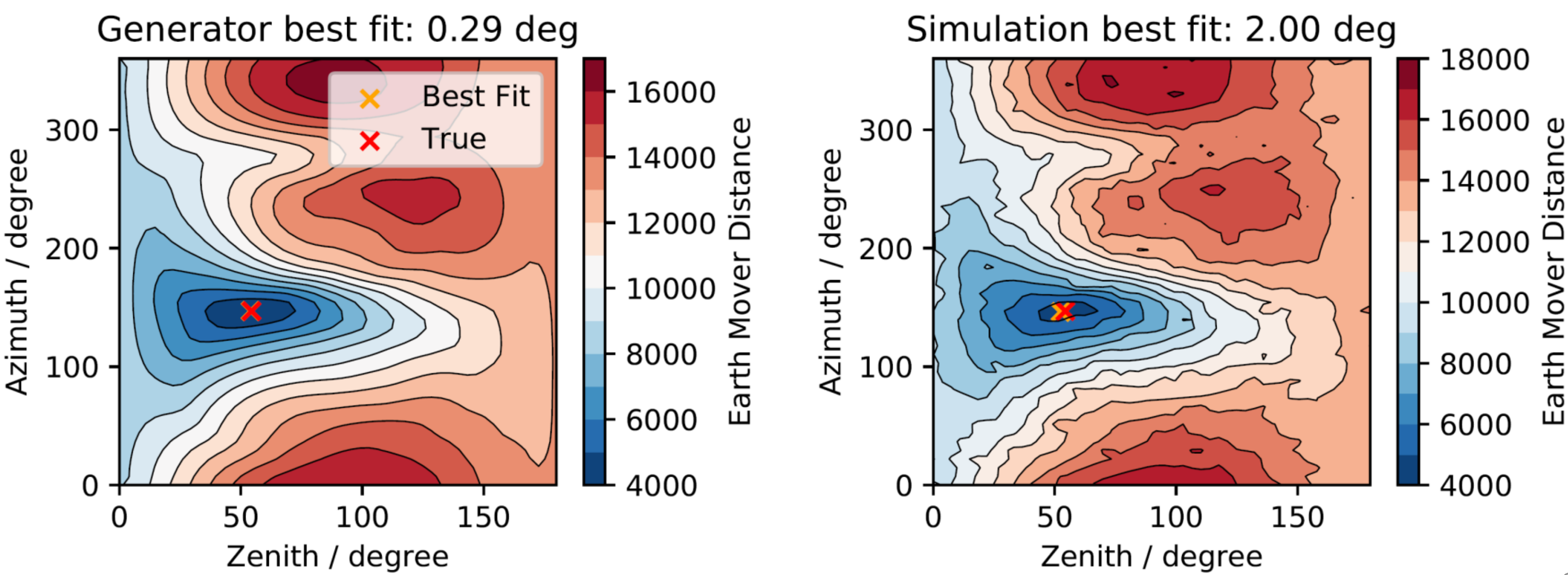
Cascade Generator – Results (Deep Core included)



Cascade Generator vs. Direct Re-Simulation

- Neural Network is extremely good at interpolating and smoothing
- Landscape very well behaved, smooth gradients available
- Much faster per-event runtime

Potential to surpass direct re-simulation



$\mathcal{O}(s)$ vs $\mathcal{O}(10-100\text{h})$

Cascade Generator – Main Changes since Madison

- Use Poisson likelihood as loss
- Move back to generating charge binned in time as output instead of latent vars of AE:
  - "Implicit" AE now built into the architecture: enables training of ~200 * 5160 bins
  - Reconstruction of single pulses better defined with Poisson likelihood and binned PDF as opposed to generated latent vars
- Training on EM showers instead of hadronic showers:
  - Light yield smearing in hadronic showers unnecessarily complicates training
- Slight fixes in generator architecture:
  - Make Poisson likelihood more stable

Deep Learning in Physics: Necessity for dedicated Methods

Special circumstances in Physics:

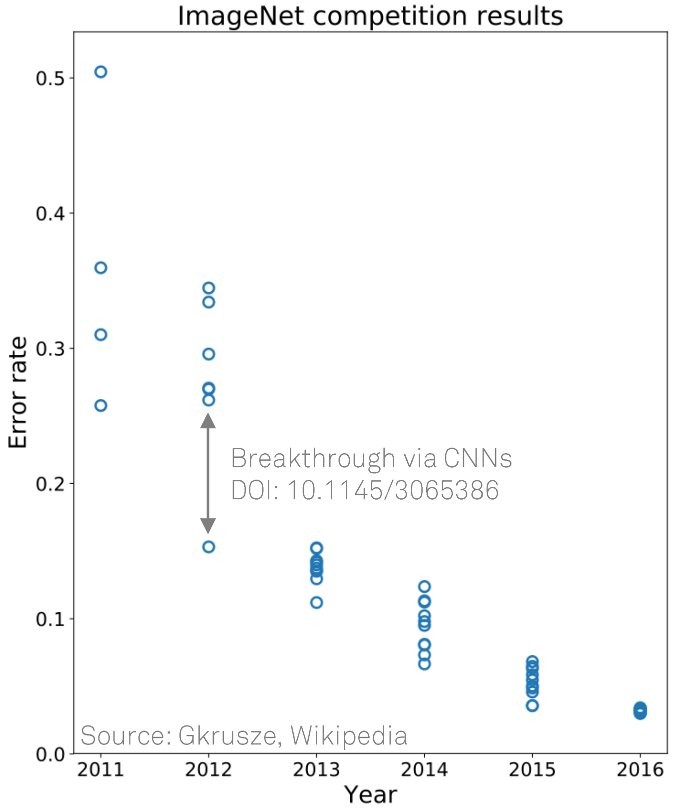
- Lots of well labeled MC simulation
- Symmetries & physical laws
- Data generation process is extremely well known
- ...

We are just beginning to exploit these circumstances!

Exploiting a priori knowledge is crucial

- ImageNet Breakthrough in 2012 as an Example:
  - Convolutional Neural Network to exploit translational invariance & locality

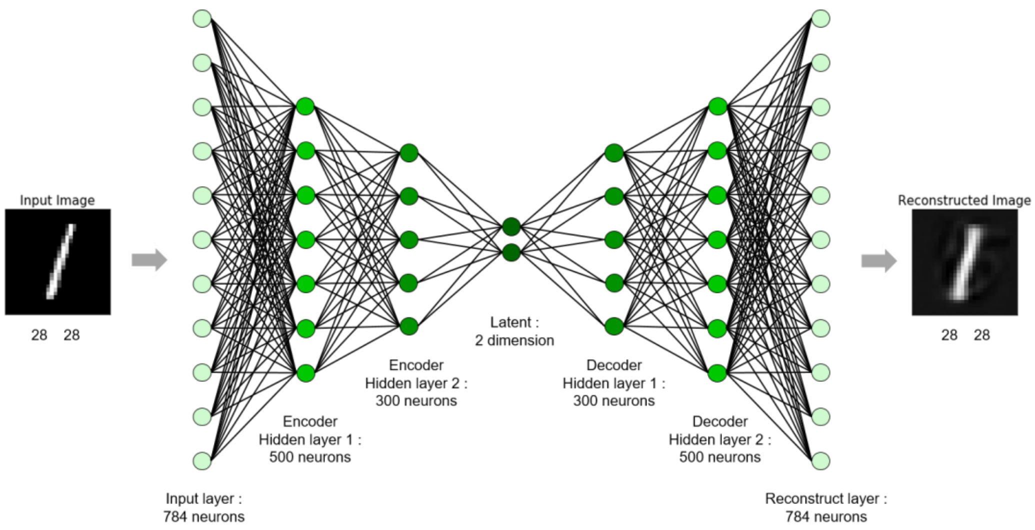
→ Need to develop dedicated Methods!

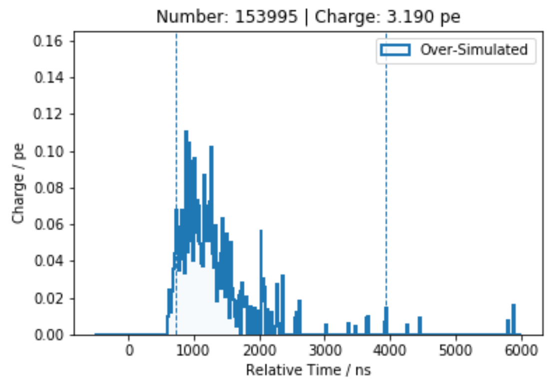


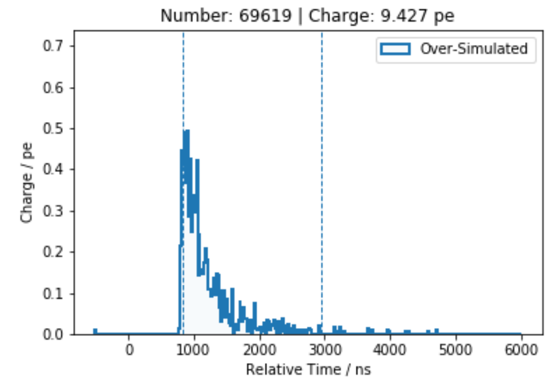
Cascade Generator – Time Parameterization

How can the time dimension be included?

- Bin charge in time bins: too many parameters!
- Calculate quantiles of cumulated charge
- Find a parameterization for the expected arrival times
- Train an autoencoder to reduce the dimensionality



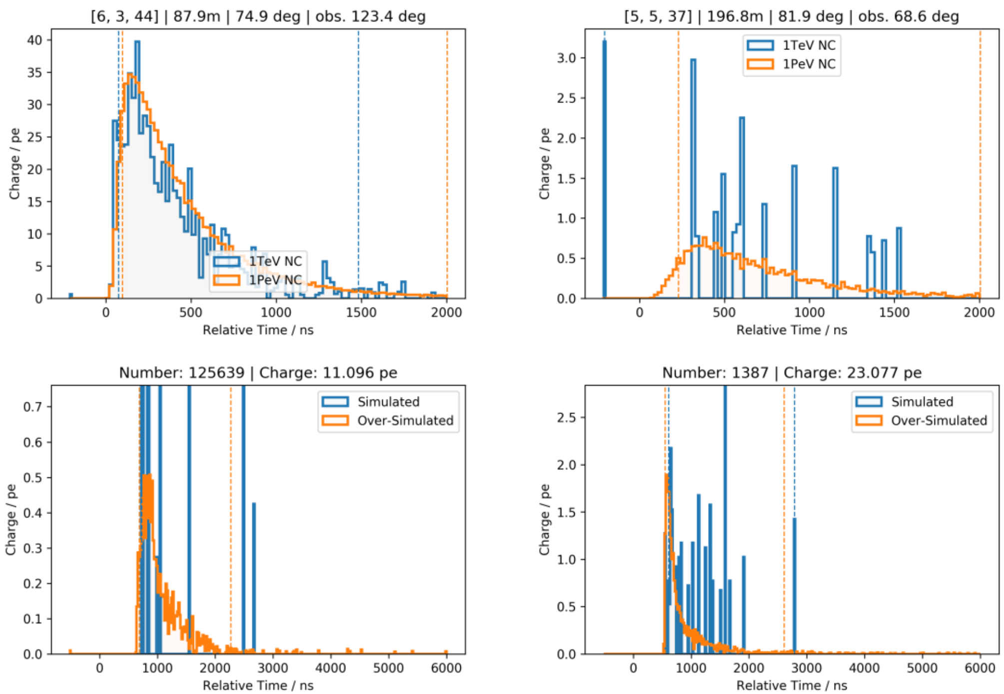




Cascade Generator – Autoencoder

How to define pulse arrival time pdf when only a few hits are present?

- Shape of arrival time pdf not well defined if only a few hits are there
- Need to over-simulate events: simulate same event n times and average hits



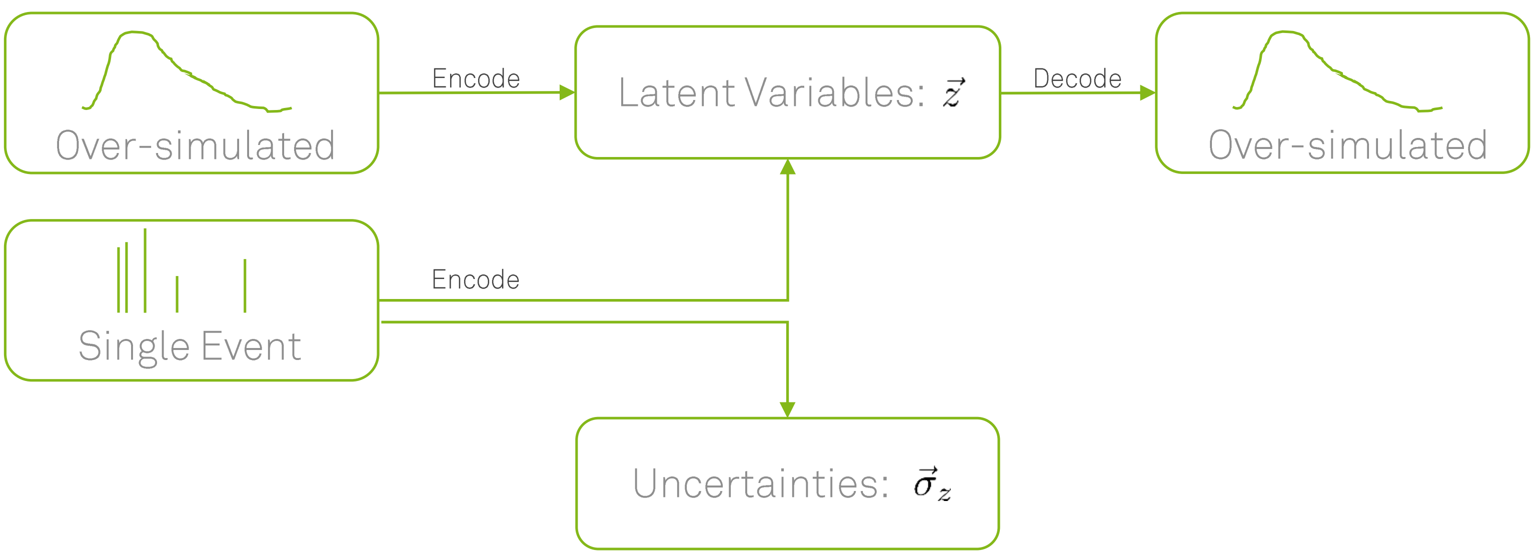
Increase cascade energy:
Light yield is proportional to energy

Over-simulate events:
Hits averaged over 100 simulations

Cascade Generator – Autoencoder

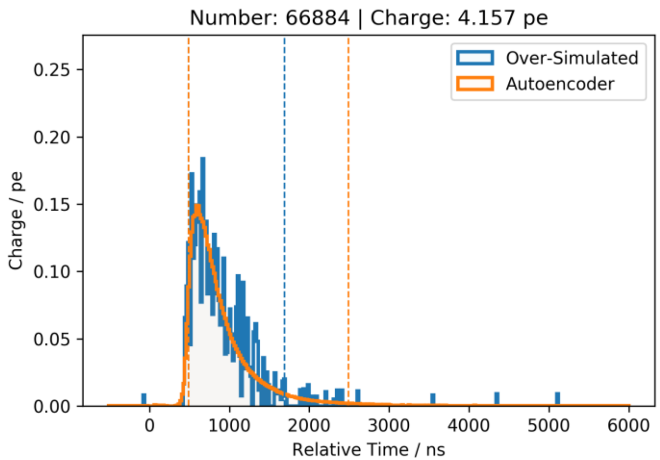
How to define pulse arrival time pdf when only a few hits are present?

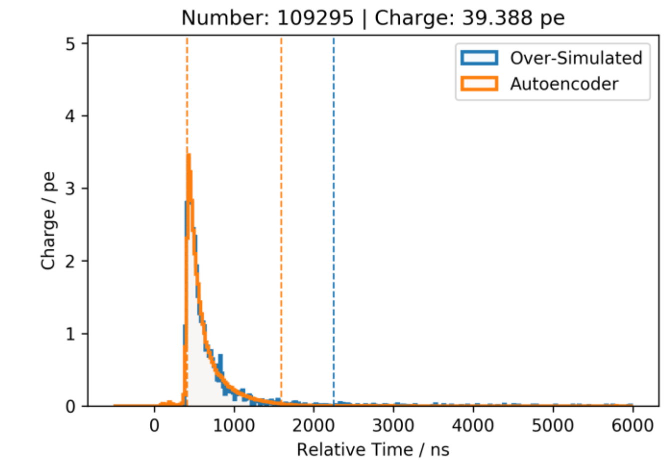
1. Train autoencoder on over-simulated events
2. Train second encoder on single events
    - Also learn to estimate uncertainty on latent variables



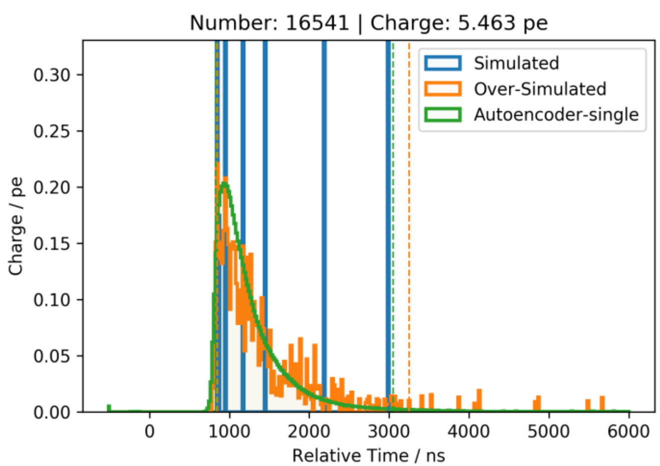
Cascade Generator – Autoencoder

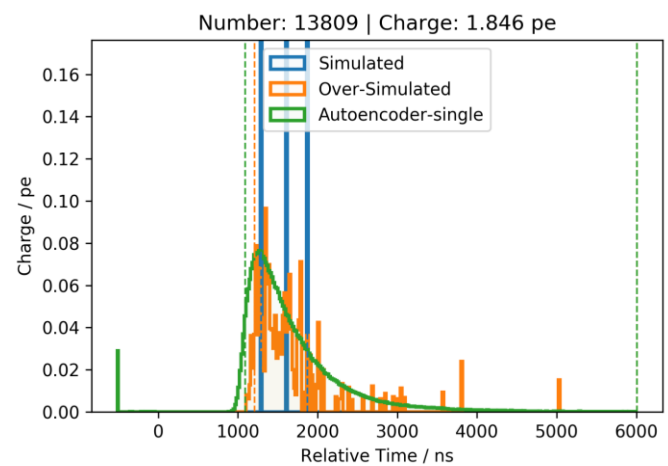
N = 100



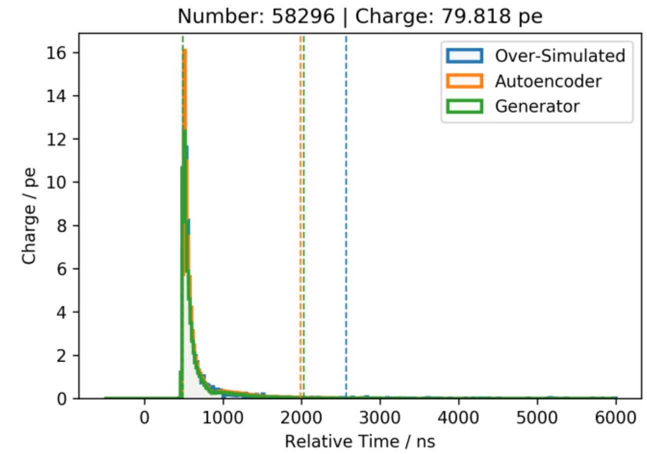


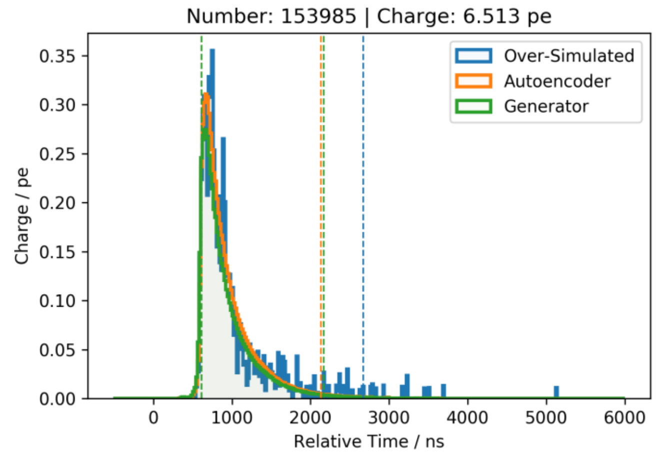
N = 1

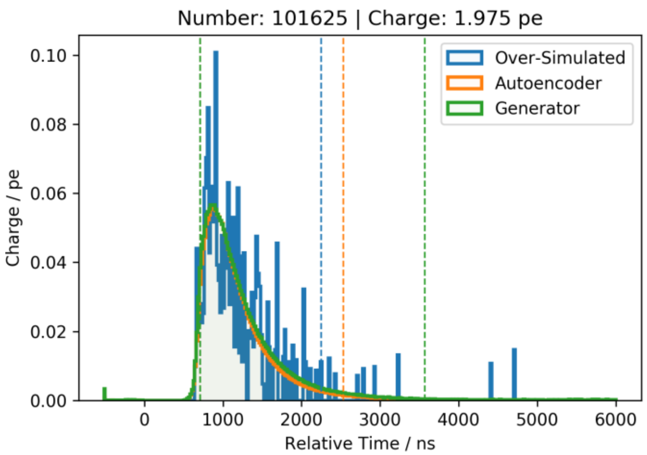


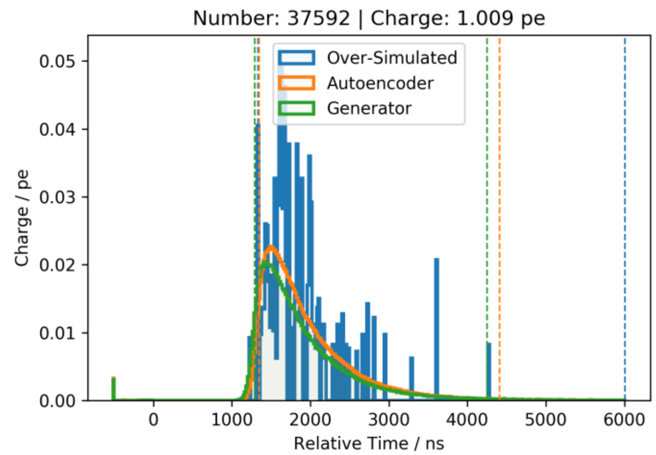


Cascade Generator – Generated Pulse Arrival Time PDFs

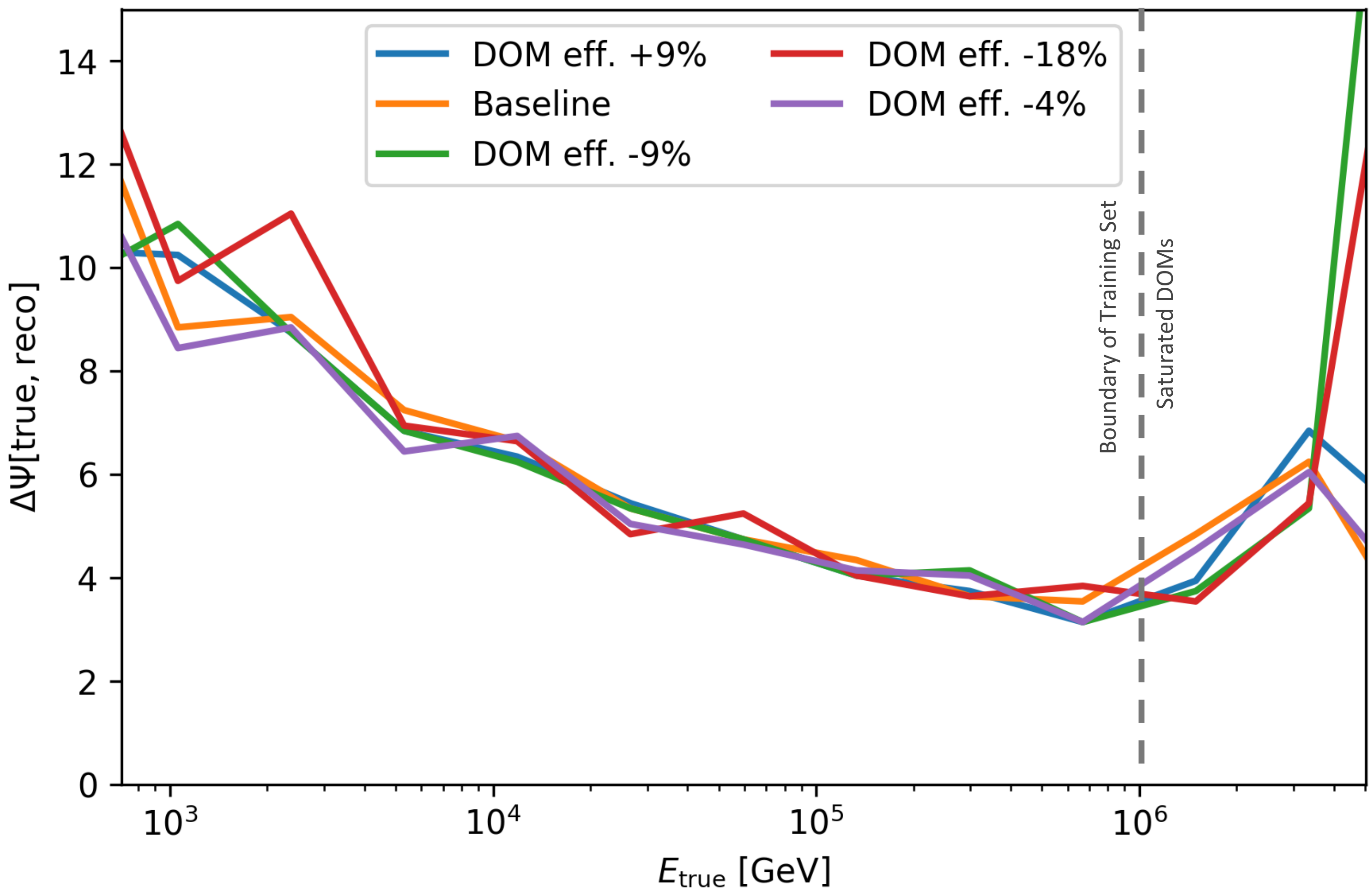




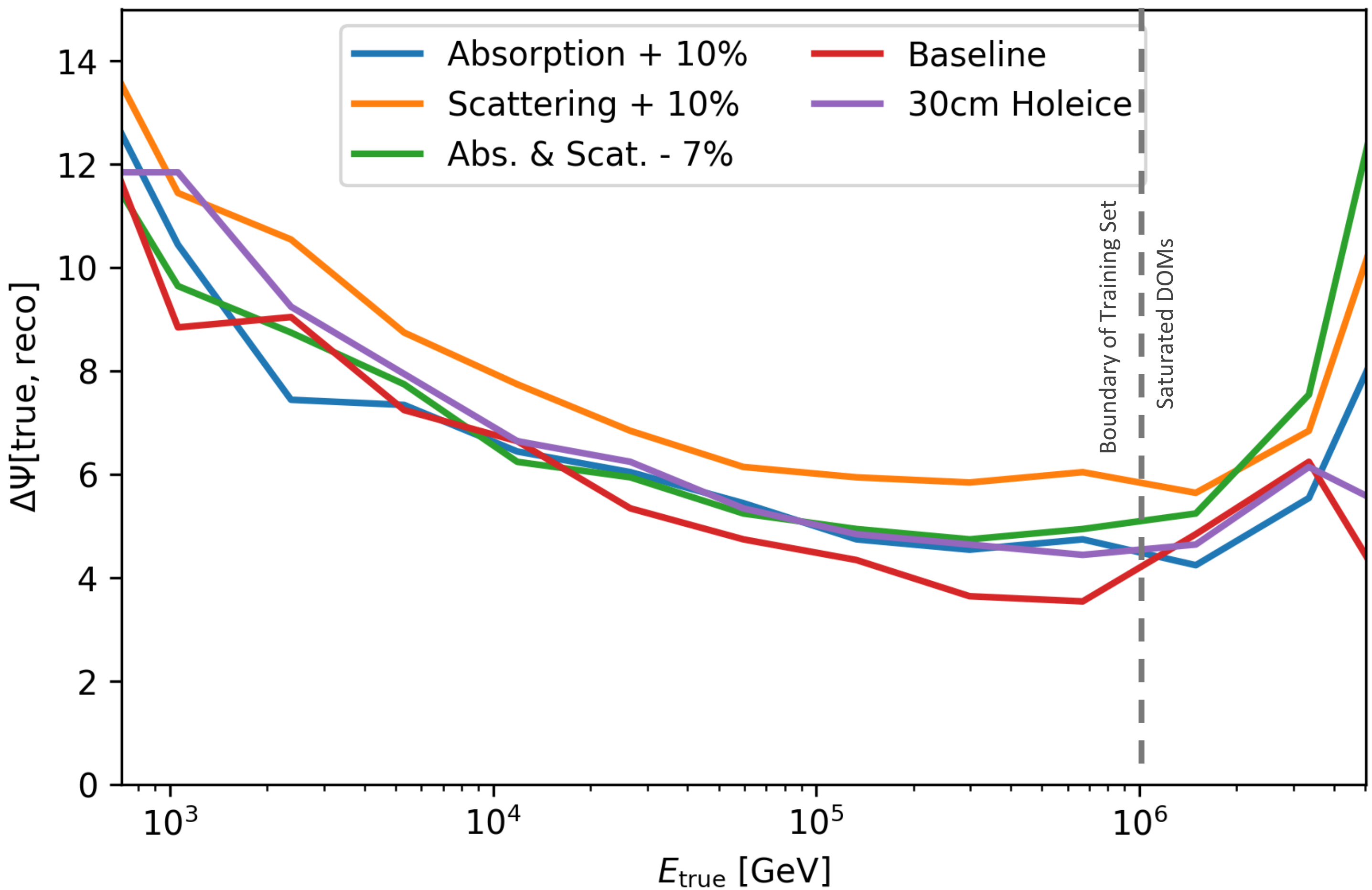




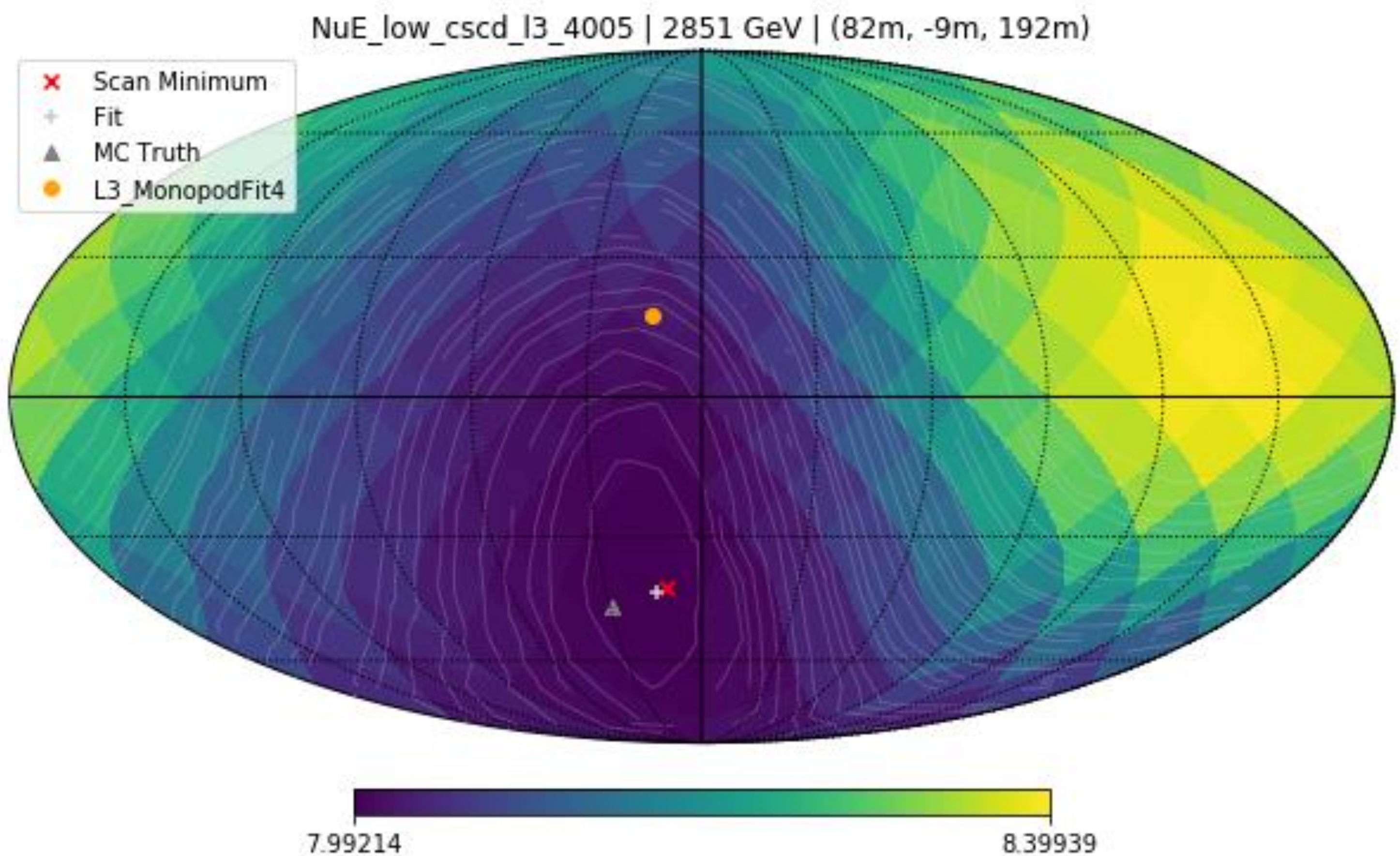
Cascade Generator – Effect of Systematics: DOM Efficiency



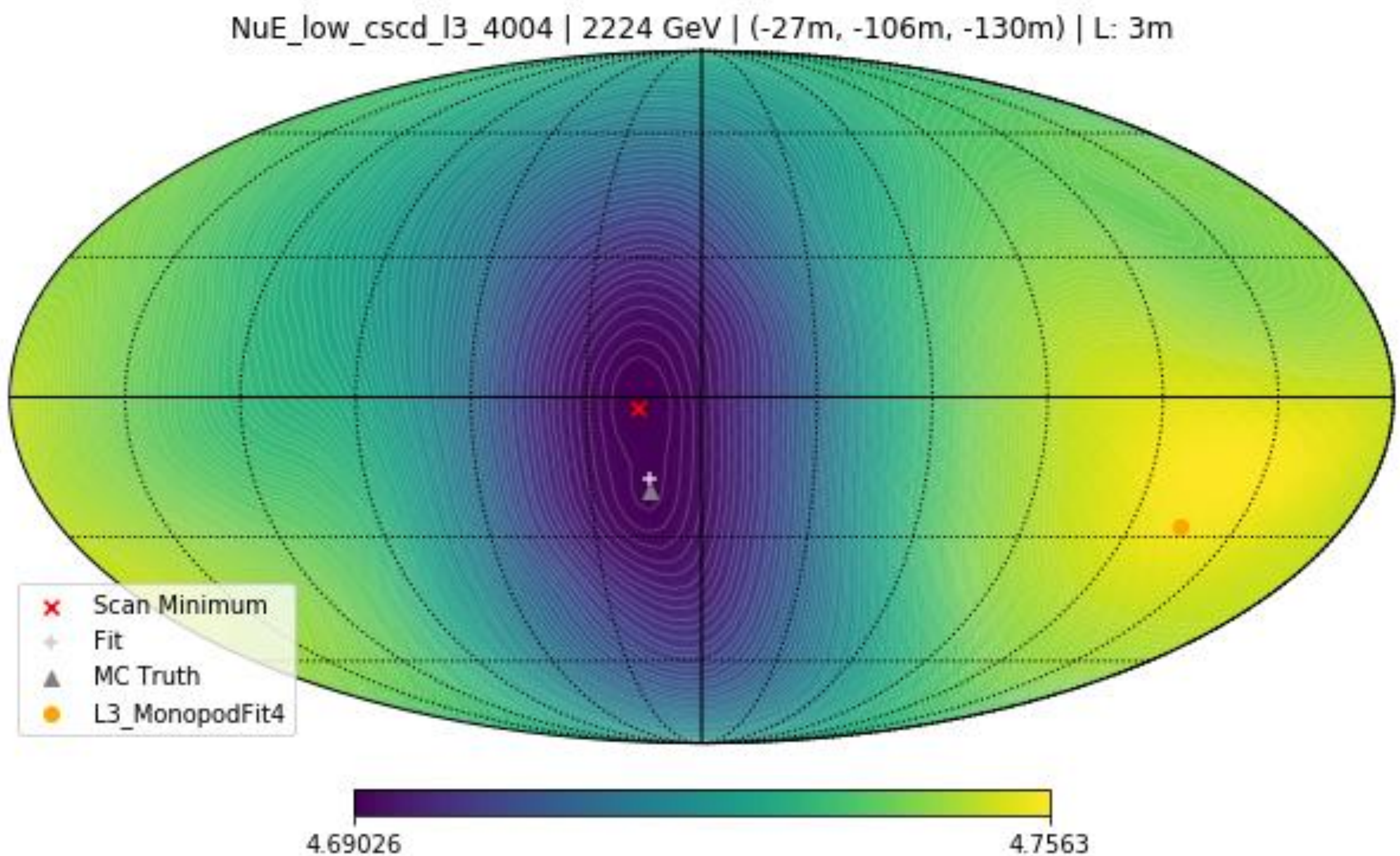
Cascade Generator – Effect of Systematics: Scat. & Absorption



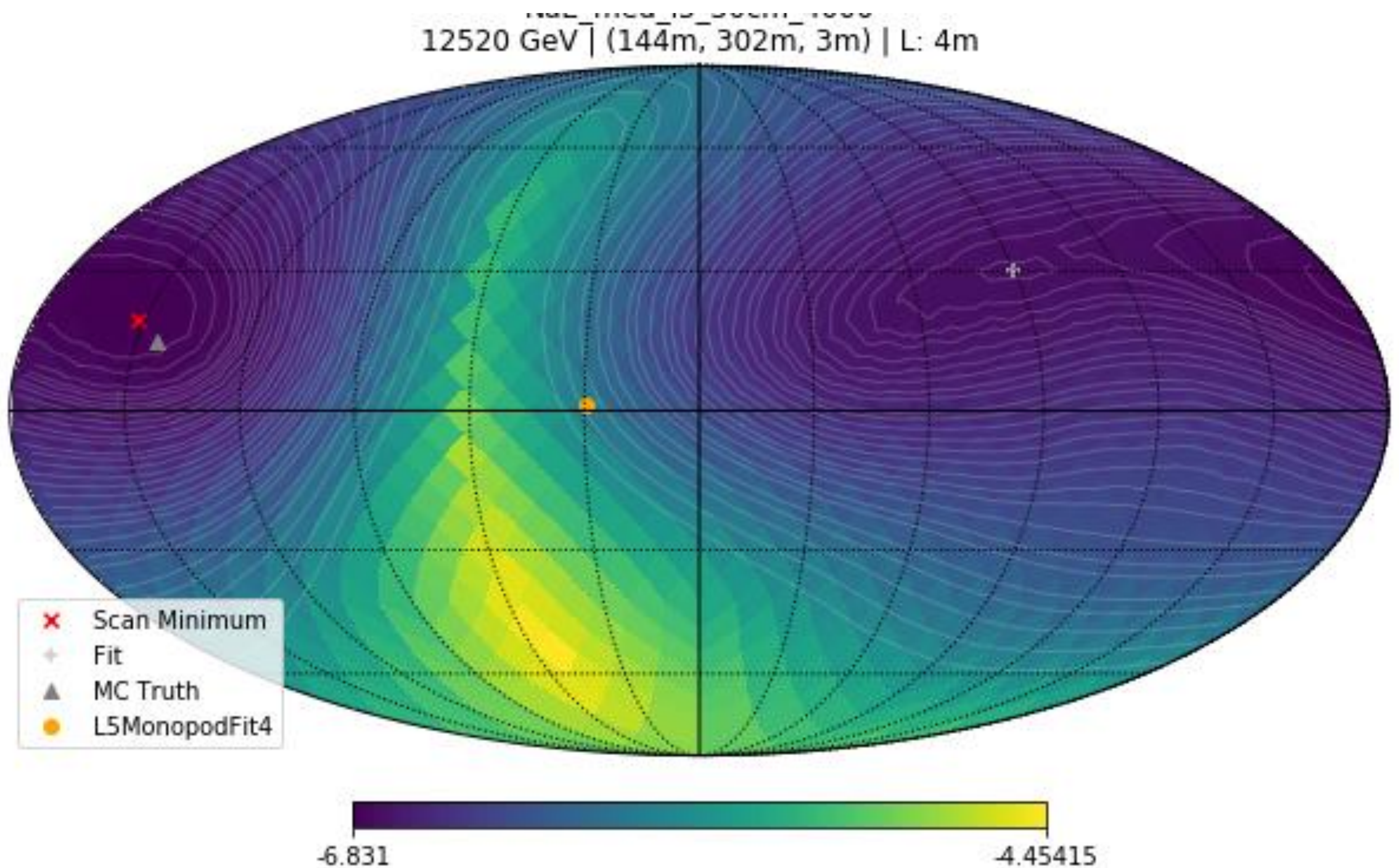
Cascade Generator – Likelihood Scan



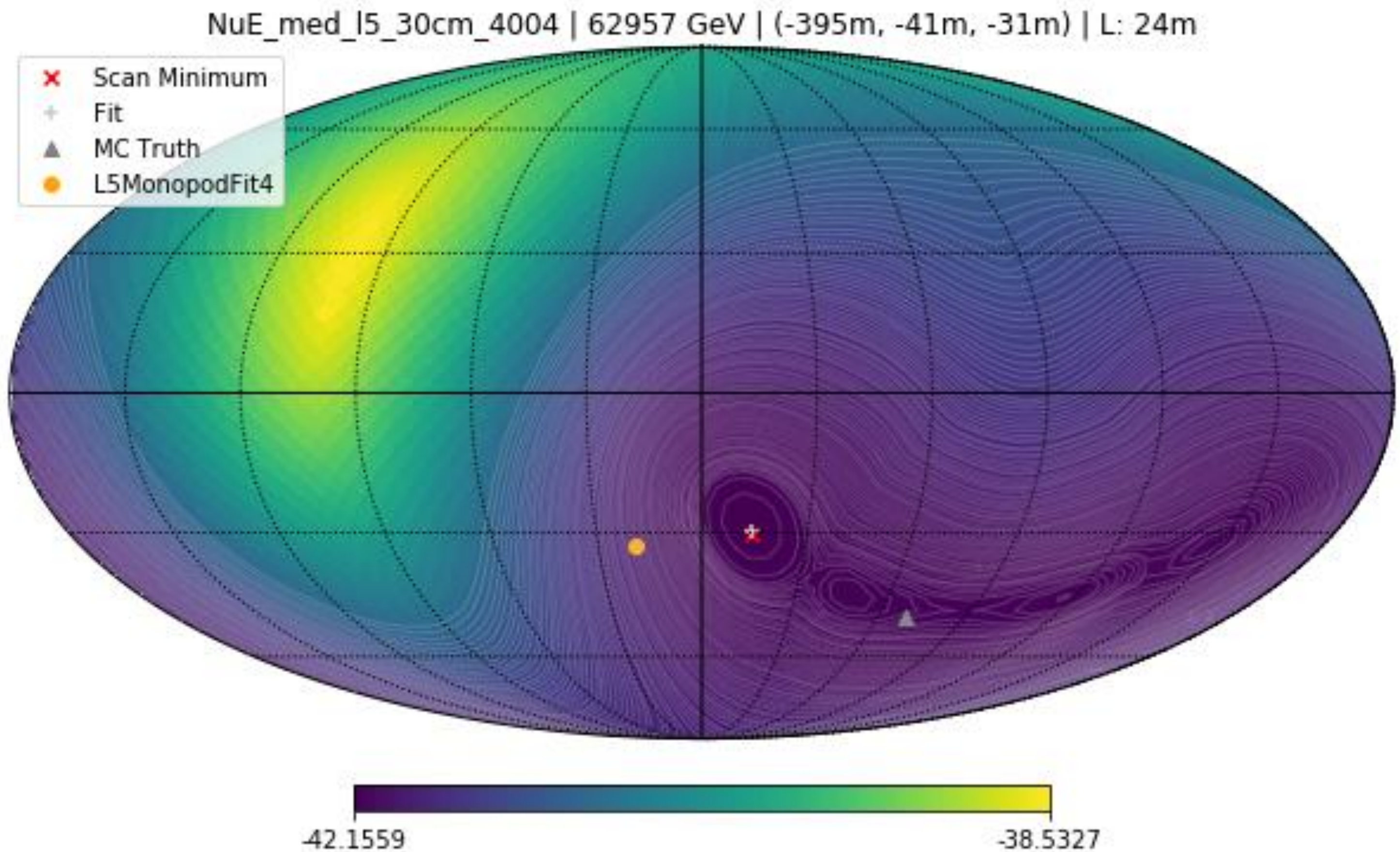
Cascade Generator – Likelihood Scan



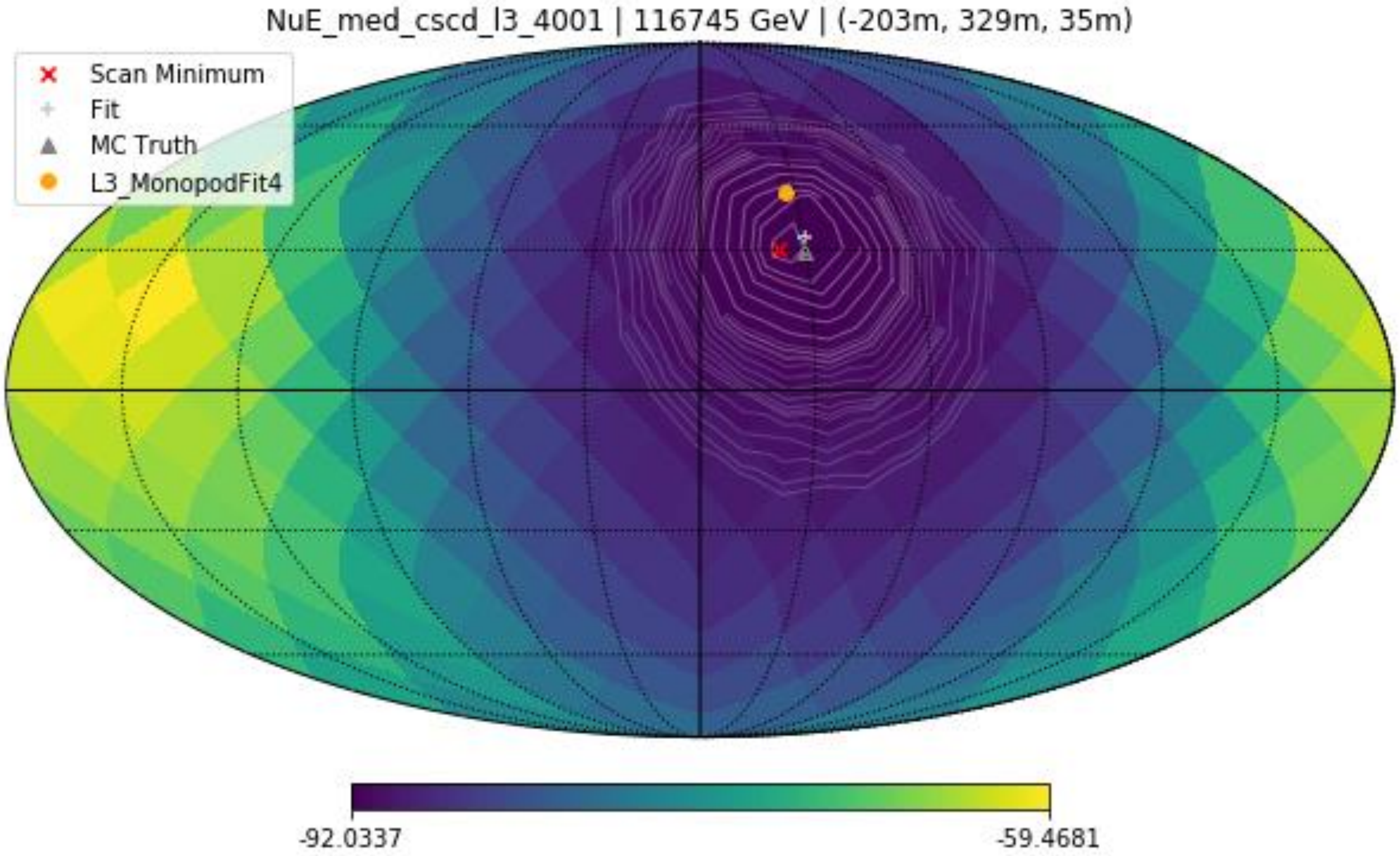
Cascade Generator – Likelihood Scan



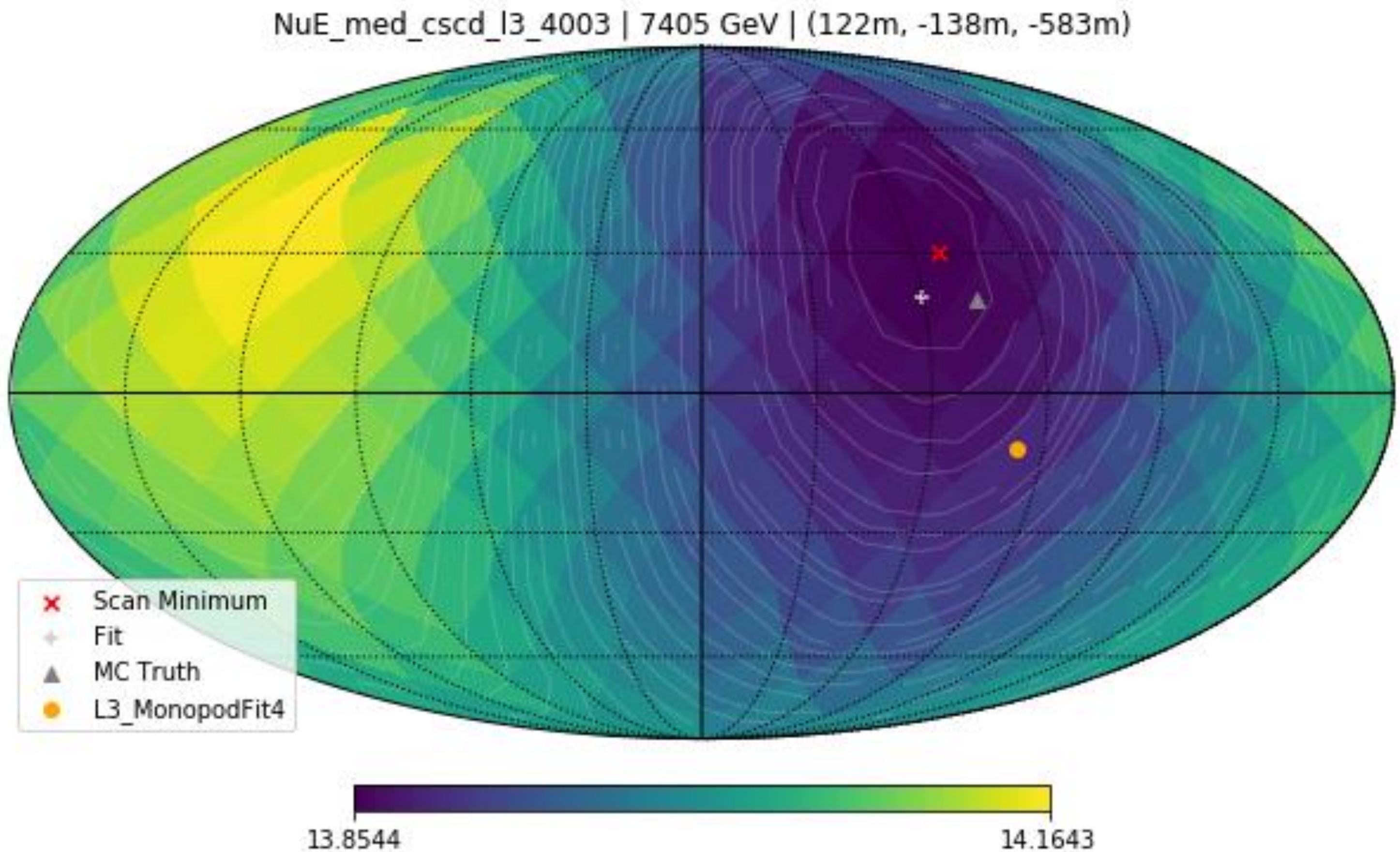
Cascade Generator – Likelihood Scan



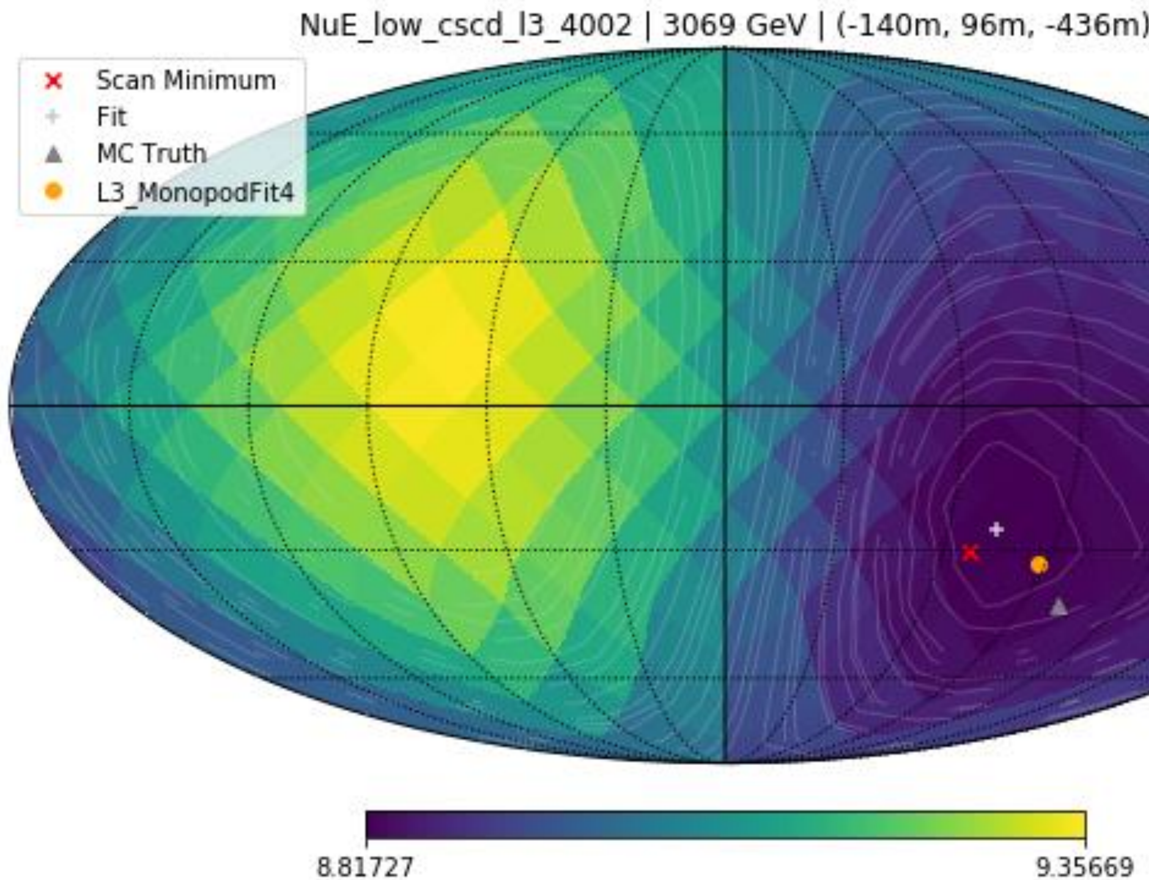
Cascade Generator – Likelihood Scan



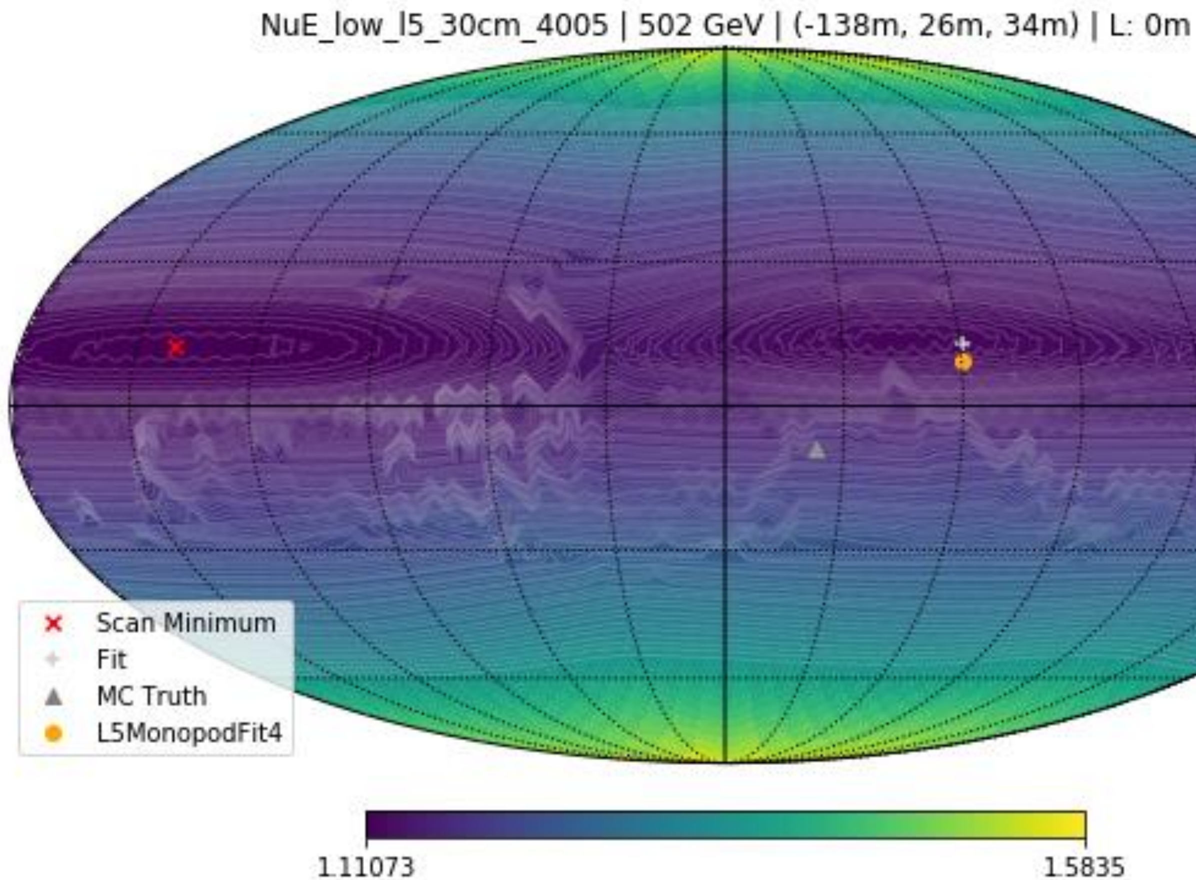
Cascade Generator – Likelihood Scan



Cascade Generator – Likelihood Scan



Cascade Generator – Likelihood Scan



Cascade Generator – Likelihood Scan