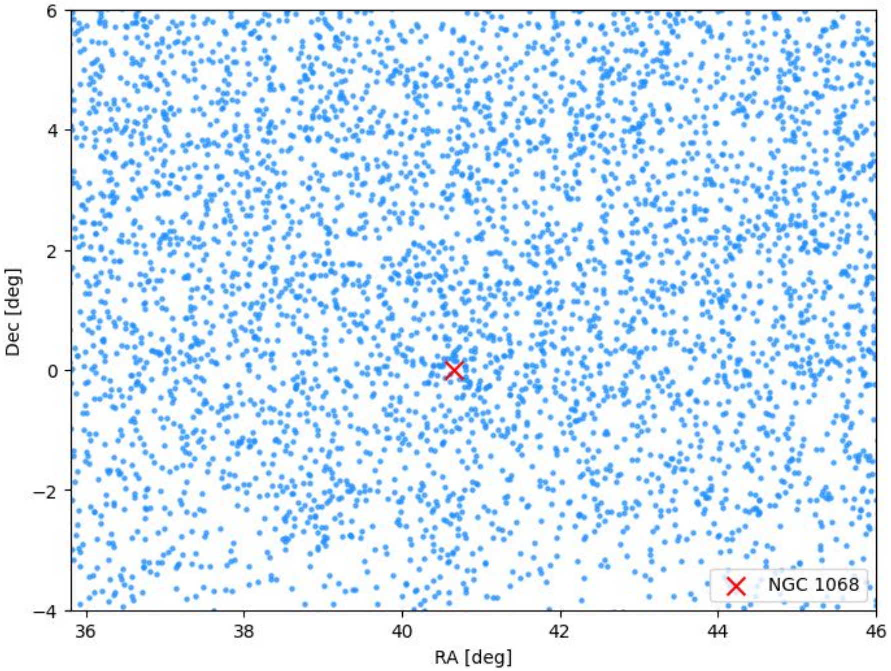
Physics informed NN Reconstruction for Muon Neutrino Tracks

Rishi Babu and Hans Niederhausen

For the IceCube Collaboration

Michigan State University

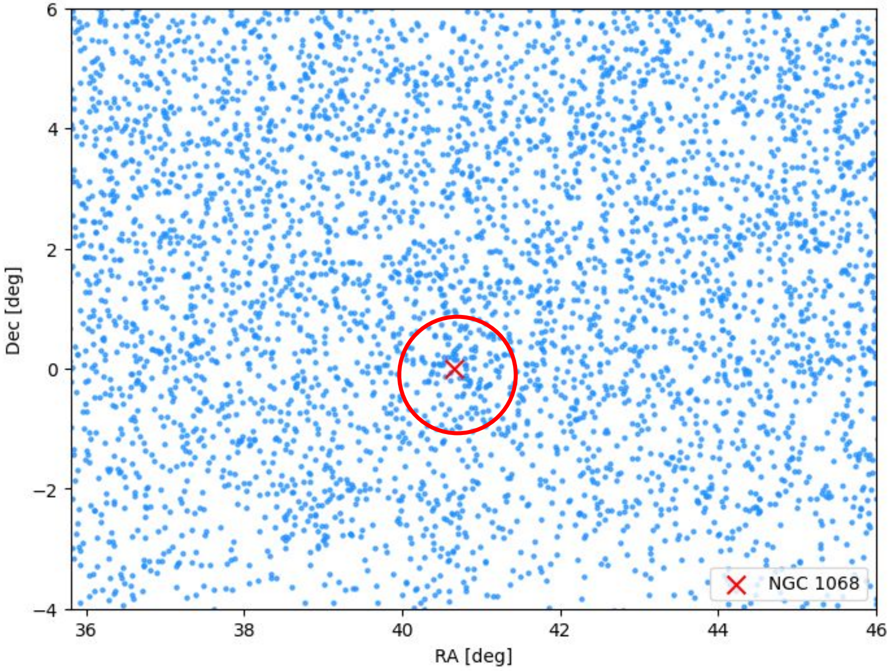
Detecting Neutrinos from the Universe



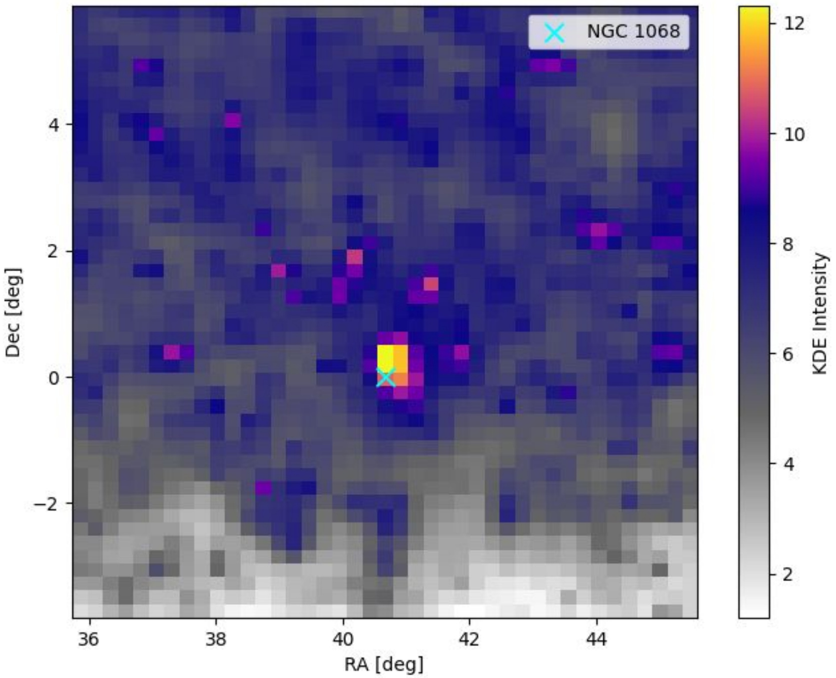
Counts from IceCube NGC 1068 Public data
DOI:[10.1126/science.abg3395](https://doi.org/10.1126/science.abg3395)

- Detecting neutrino is a challenging problem due to their very small interaction cross-section
- As a result the number of neutrinos detected from astrophysical objects are far less.
- ~10 astrophysical neutrinos per year (> 100 TeV)
- I'm primarily a theoretical astrophysicist
  - Working on MW modeling of gamma-rays and neutrinos
- What I look for is if a few of them cluster near an astrophysical source!

Detecting Neutrinos from the Universe

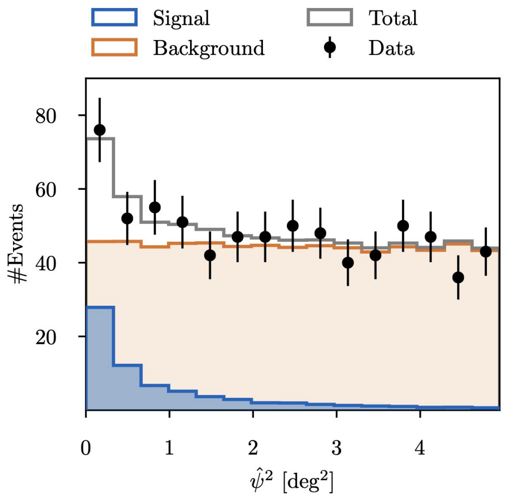


Counts from IceCube NGC 1068 Public data
DOI:10.1126/science.abg3395

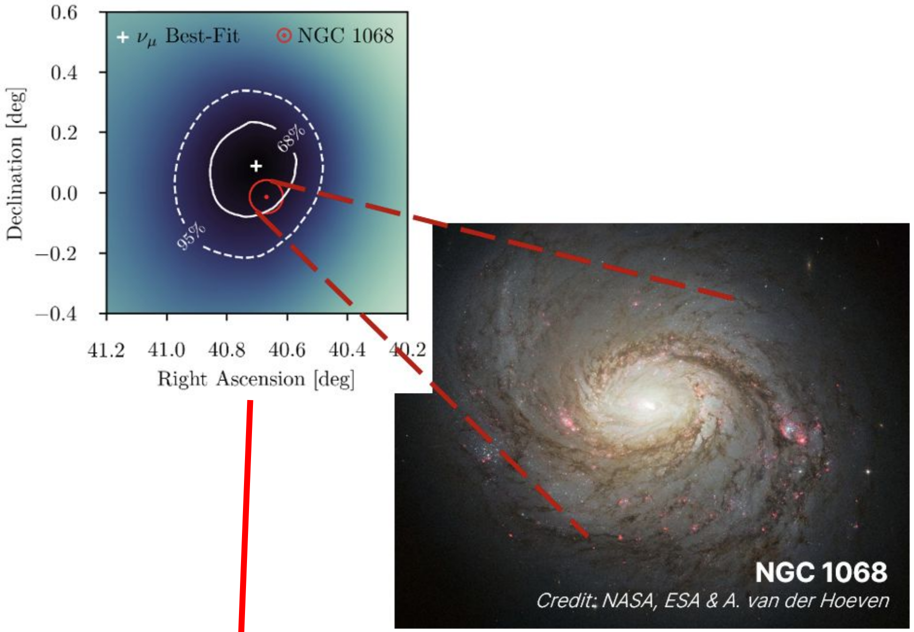


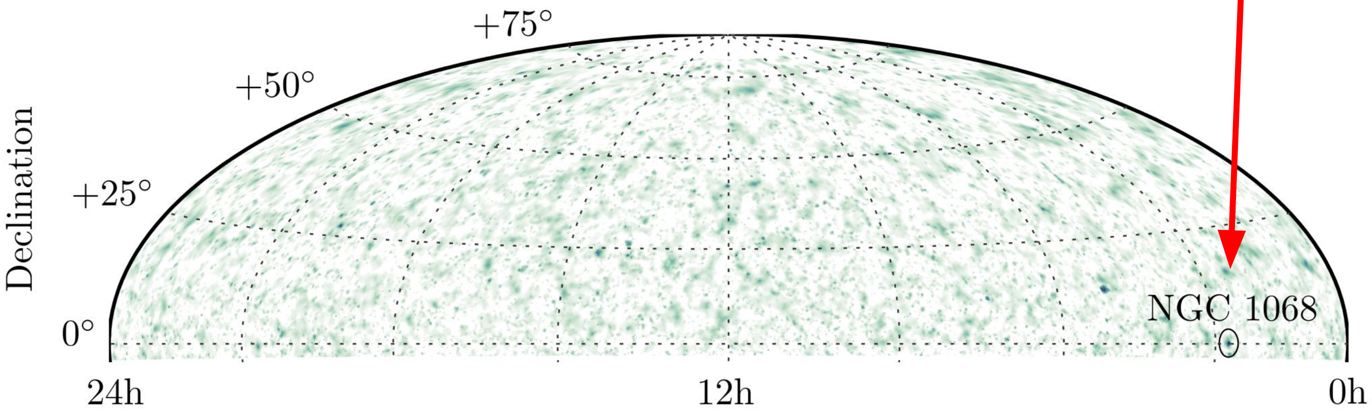
Event Density Visualization from
IceCube NGC 1068 Public data

Neutrino Emission from NGC 1068



- Astrophysical neutrino events: 79
- Evidence for neutrino emission from NGC 1068 with a significance of 4.2σ using neutrino tracks



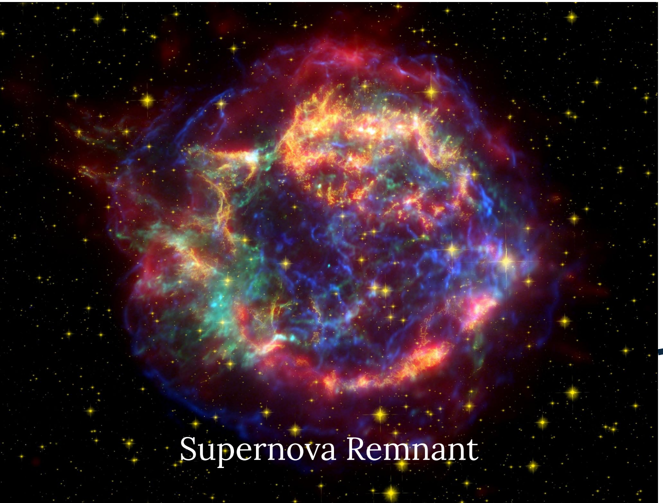


Ref:

IceCube Collaboration, "Evidence for neutrino emission from the nearby active galaxy NGC 1068", Science, vol. 378, no. 6619, pp. 538–543, 2022.

IceCube Observatory and a Neutrino Track Event

- Cubic-kilometer Cherenkov Detector in Ice at the South Pole
- Detects neutrinos and atmospheric muons
- Over 5k PMTs stored in DOMs (digital optical modules)
- Active and taking data since 2011



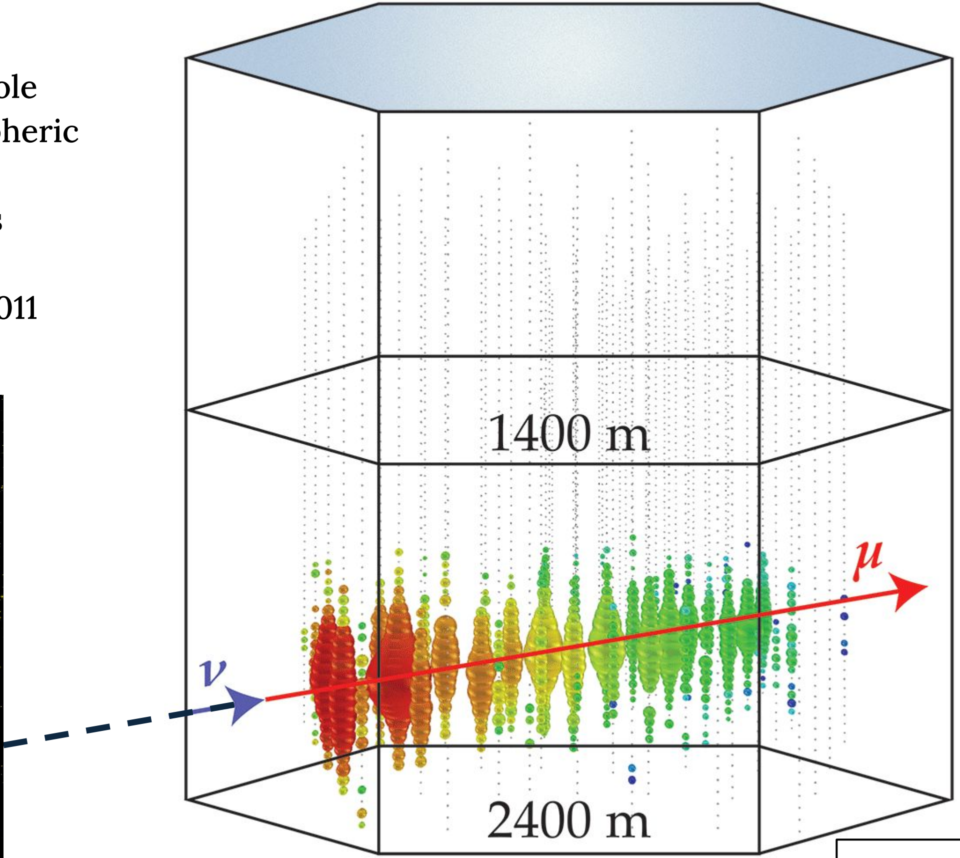


Image Courtesy: APS/Joan Tycko.

IceCube Neutrino Tracks

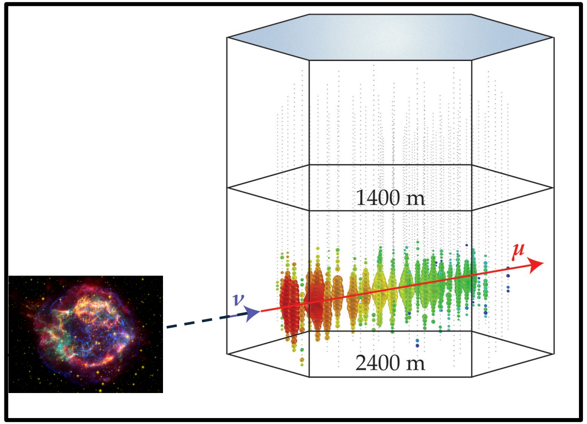
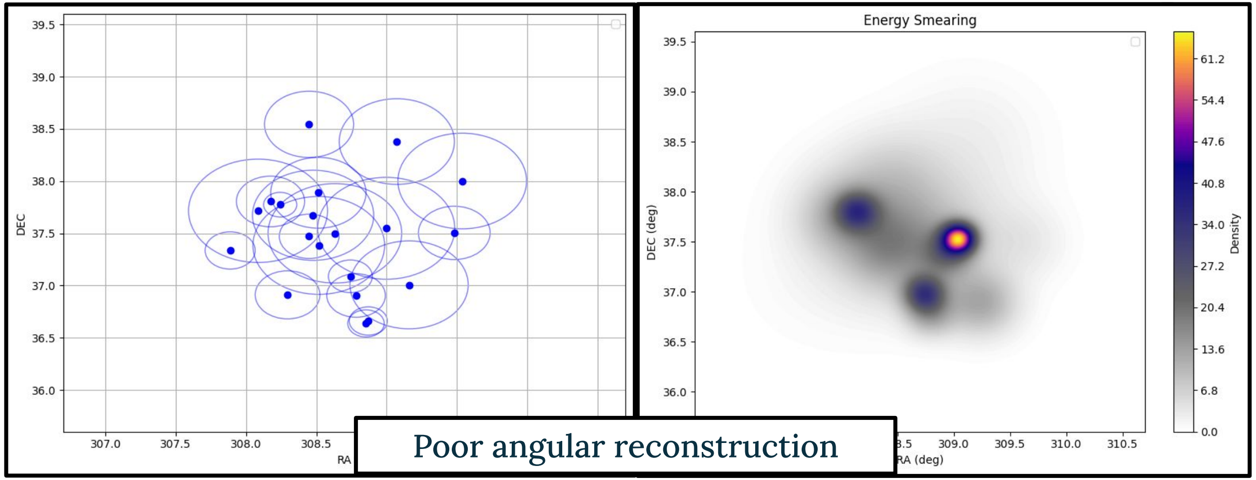
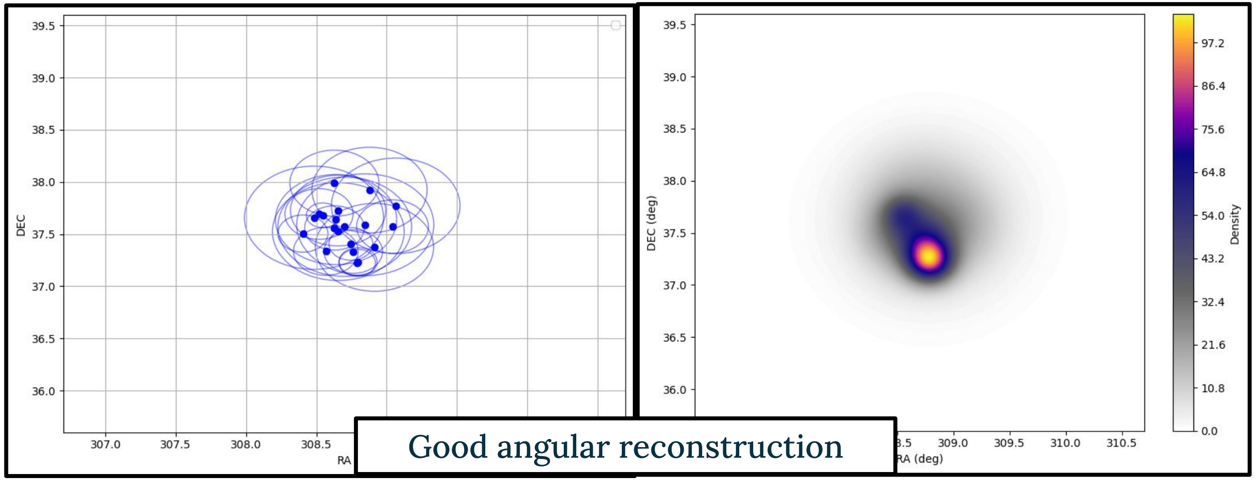


Image Courtesy: **APS/Joan Tycko.**

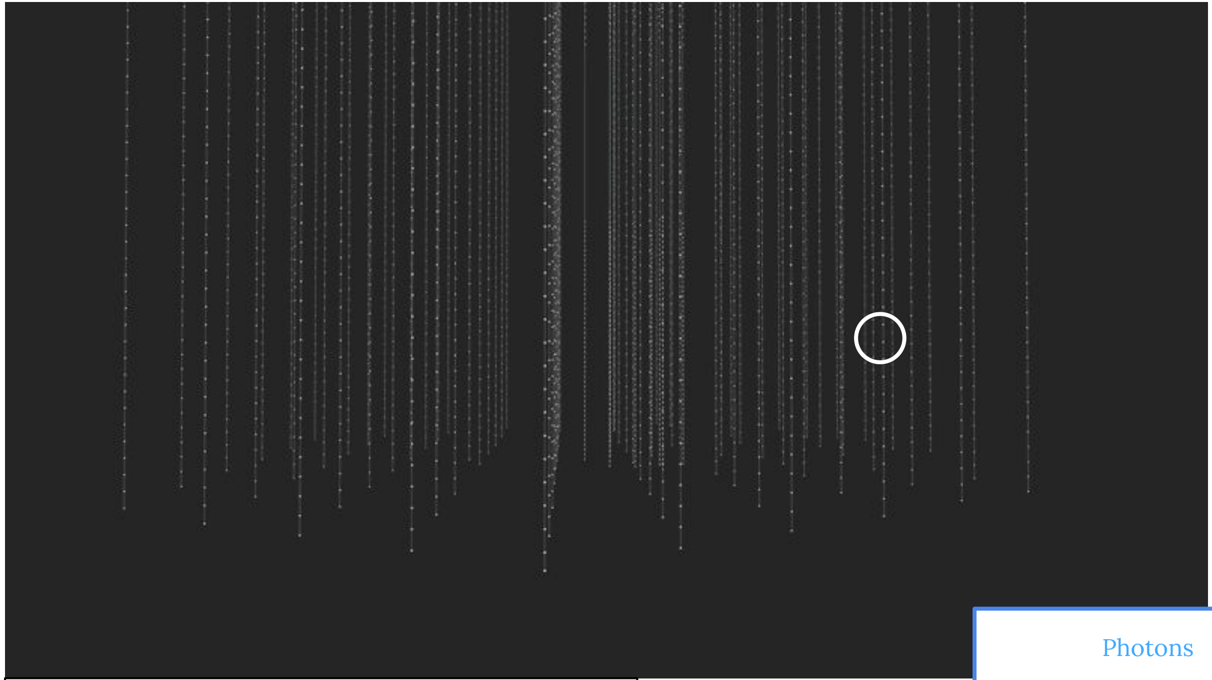


Poor angular reconstruction



Good angular reconstruction

Photon Propagation through Ice



Ref: Chirkin, D. and IceCube Collaboration, "**Photon tracking with GPUs in IceCube**", Nuclear Instruments and Methods in Physics Research A, vol. 725, Elsevier, pp. 141–143, 2013.

- **Goal**: Reconstruct which direction a muon/neutrino came from in-ice
- Reconstruction requires knowing the expected light arrival time at every sensor
  - Expensive to compute
- For a given position along the track
  - Assume that the Cherenkov photons produced have a cylindrical geometry
  - Sample the arrival time distribution of photons using simulations
  - Stored in photon tables
  - Takes ~100s of GBs storage

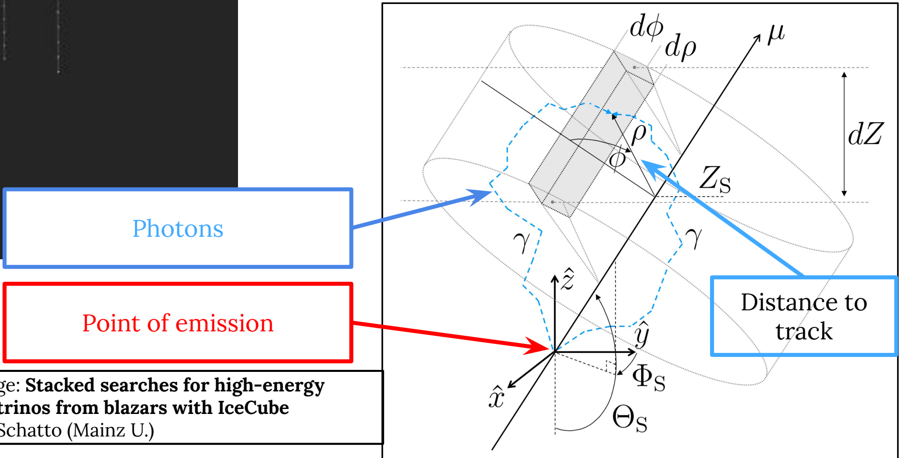


Image: **Stacked searches for high-energy neutrinos from blazars with IceCube**
Kai Schatto (Mainz U.)

Photon Propagation through Ice: Photon Tables

- **Goal**: Reconstruct which direction a muon came from
- Reconstruction requires knowing the expected light arrival time at every sensor
  - Expensive to compute
- For a given position along the track
  - Assume that the Cherenkov photons produced have a cylindrical geometry
  - Sample the arrival time distribution of photons using simulations
  - Stored as photon tables
  - Takes ~100s of GBs storage

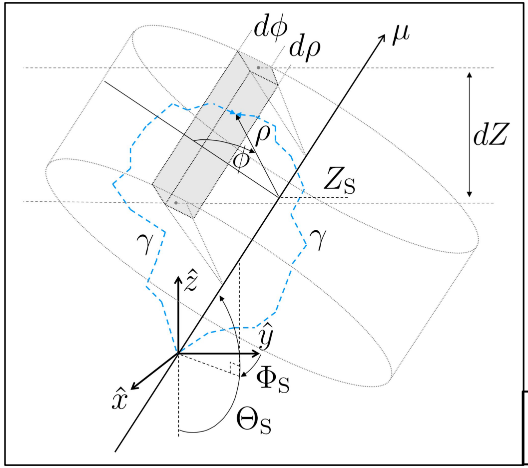


Image: **Stacked searches for high-energy neutrinos from blazars with IceCube**
Kai Schatto (Mainz U.)

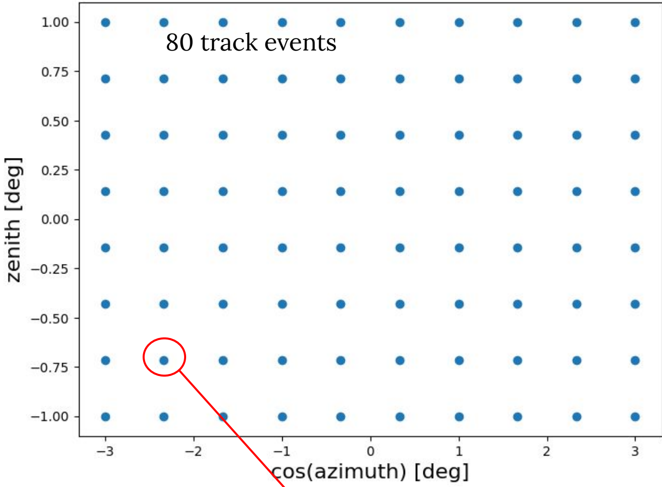
Parameters :
For each position on the track
- 5 spatial parameters
  - Point in space (3D)
  - Track (zenith, azimuth)
- Binned time of arrivals histograms, residual times
- Typically 100,000 bins per track

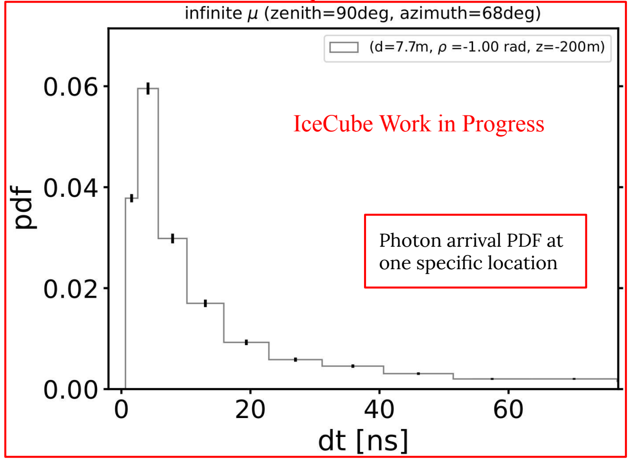
Say if we have 80 tracks

80*100,000 time-bins ~ 8×10^6 histograms

~ 400 GBs of storage

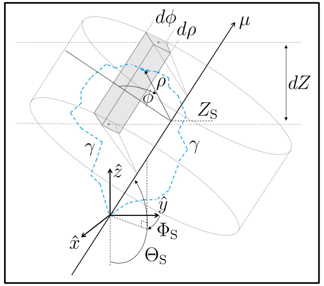
Reconstruction using lookup tables: Computationally expensive to interpolate for all track inputs





Photon Propagation through Ice: NN

- **Goal**: Reconstruct which direction a muon came from
- Reconstruction requires knowing the expected light arrival time at every sensor
  - Expensive to compute
- For a given position along the track
  - Assume that the Cherenkov photons produced have a cylindrical geometry
  - Sample the arrival time distribution of photons using simulations
  - Stored as photon tables
  - Takes ~100s of GBs storage



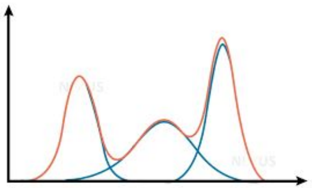
This work:
Use a neural network to predict the track parameters for tracks with different types of geometries

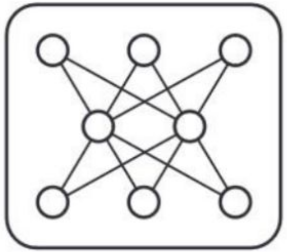
We iterated over multiple names depending on the network combination

We currently call it the **Gupta Network**

Conditional Probabilities

Loss Function

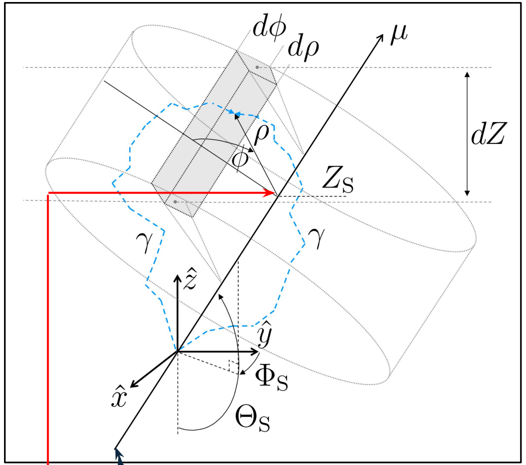




Fully connected Feedforward NN

Photon Tables

Network Architecture: Inputs



Data

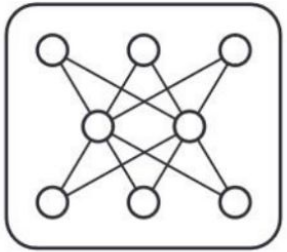
- TFRecords shard batching
- Split: 80/10/10
- Single GPU
- Model: JAX/Equinox
- TensorFlow used only for data ingestion, not the model

7 Input Parameters
(Photon Tables)

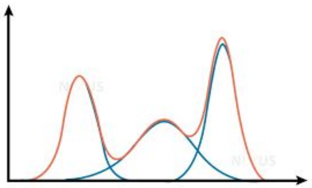
Track zenith
Track azimuth
Track depth, z
Distance to track, d
Angle around track, rho

Cylindrical coordinates converted to cartesian

Photon Tables



Fully connected Feedforward NN

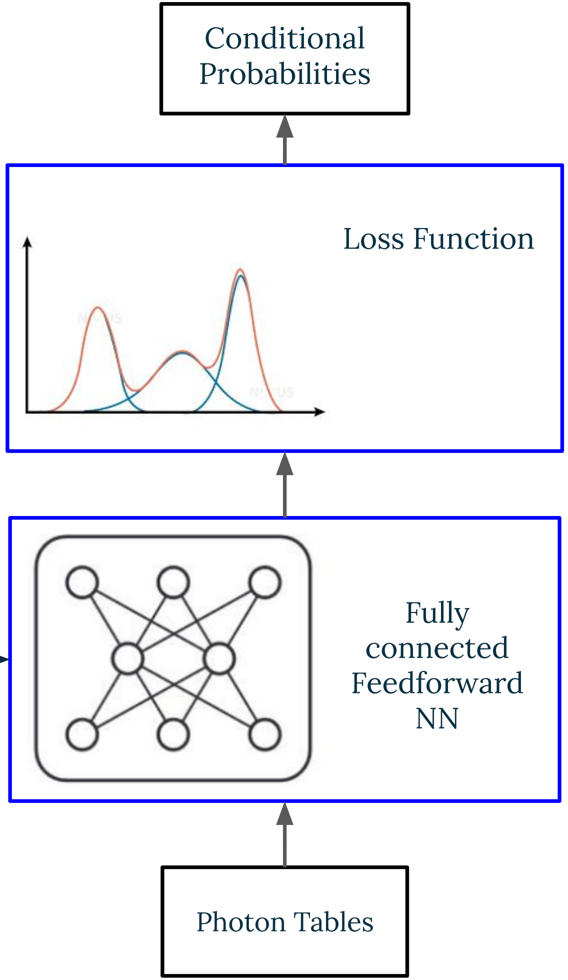


Loss Function

Conditional Probabilities

Network Architecture: Network

- ResMLP
- Layers = 7
- hidden size = 96
- Input layer
- 5 Residual blocks
- Output later
- Batch = 1024
- Optimizer: Yogi
- Cosine decay learning rate scheduler



Network Architecture: Loss Function

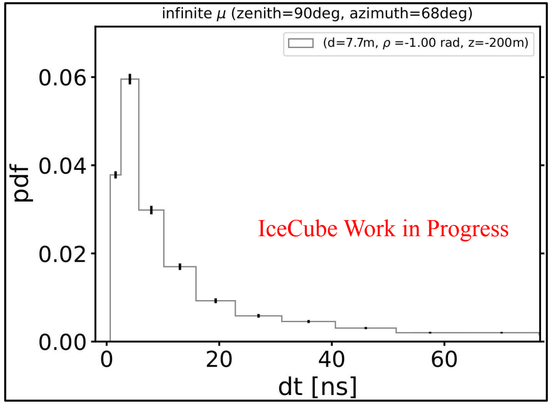
Gupta Function

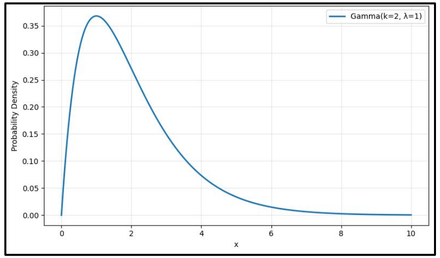
$$P(t_{\text{res}}|\theta) = \left(1 - e^{-\lambda t_{\text{res}}}\right)^{\alpha}, \qquad \text{(CDF)}$$

$$p(t_{\text{res}}|\theta) = P'(t_{\text{res}}|\theta)$$

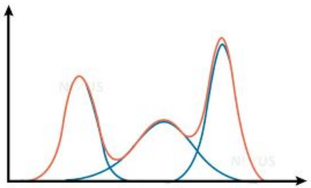
$$= \alpha\lambda\left(1 - e^{-\lambda t_{\text{res}}}\right)^{\alpha-1} e^{-\lambda t_{\text{res}}}, \qquad \text{(PDF)}$$

Gupta Function (Exponentiated exponential) : function behaves similar to the gamma function with a scale and shape parameter





x 4

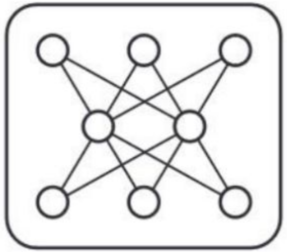


Loss: Mixture Density Function

- Propagation and diffusion of laser light in ice were studied at the Baikal neutrino telescope
- Use gamma functions to represent the PDF for photon arrival times

- Fit 4 Gupta functions to the photon arrival time pdf
- Loss: Minimize the difference between the sum of 4 Gupta function fit and the time histogram

Conditional Probabilities



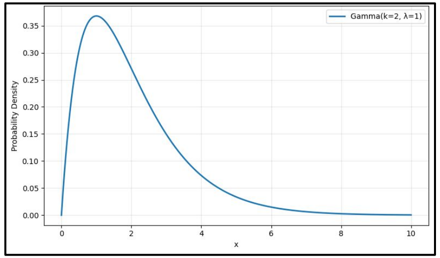
Fully connected Feedforward NN

Photon Tables

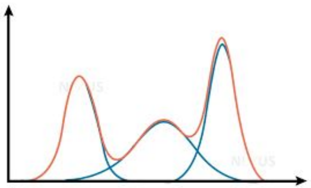
Network Architecture: Outputs

- For the [7] input → [12] **Network Outputs**,
  with the 12 decomposing as
  [4, 4, 4] for the mixture weights

- Weight/Gupta component Mode weights x 4
- Gupta component rates x 4
- Gupta component shape x 4

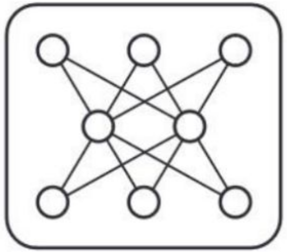


x 4



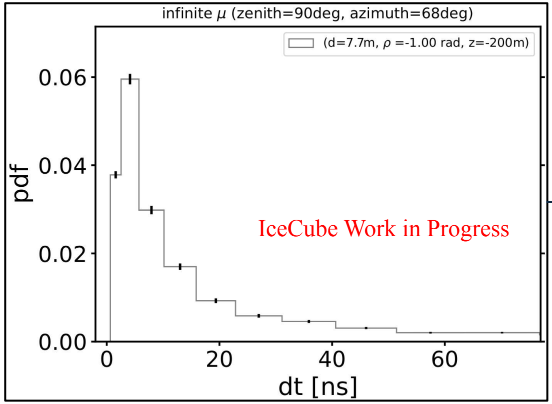
Loss:
Mixture
Density
Function

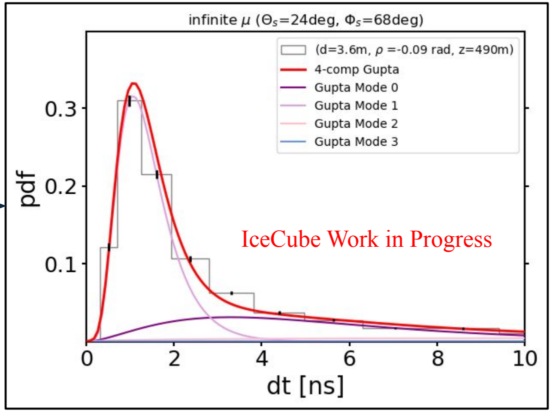
Conditional Probabilities



Fully connected Feedforward NN

Photon Tables





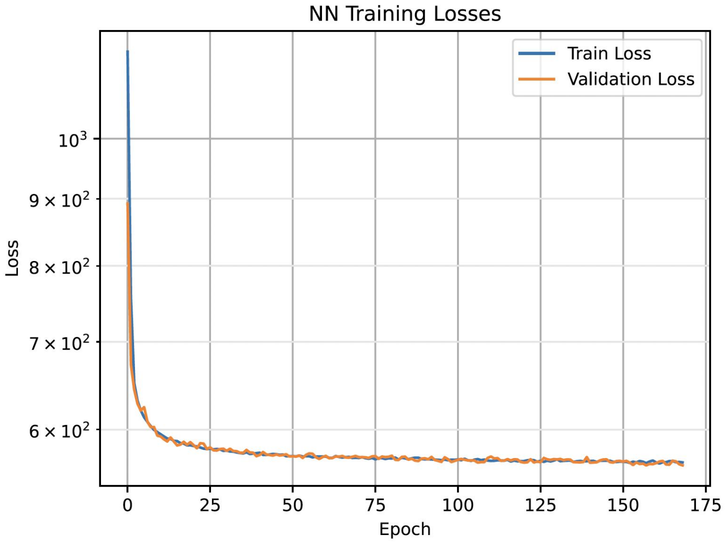
Training

- The entire workflow is written using TensorFlow + JAX integration
  => Input photon tables through **tensorflow**
  => Network Initialized - **JAX**
  => All computations for training vectorized - **JAX**
  => Training and gradient propagations - **JAX**

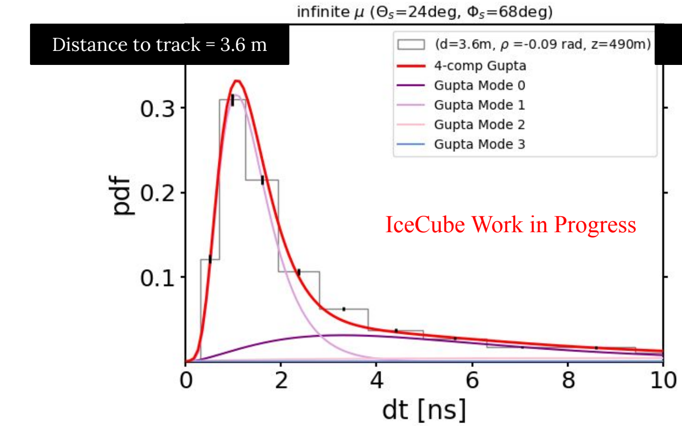
- **What we gain**: Massive gpu parallelism and gradient tracking
- **NVIDIA RTX 4090**: Training the network parameters per batch ~ **100 ms**
- **NVIDIA H200**: Training the network parameters per batch ~ **1 ms**

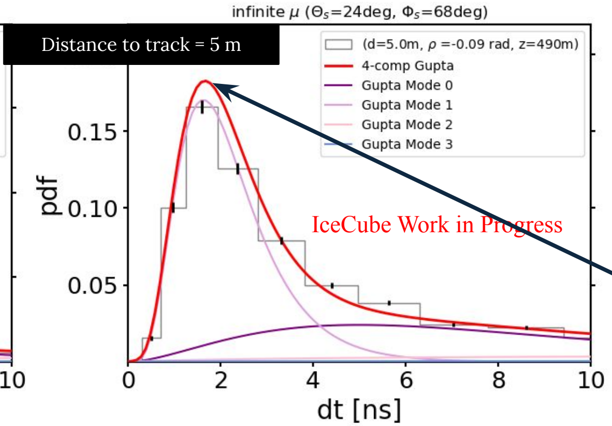
In the following slides, showing early results of training with photon simulations with the latest (SpiceFTRv3) icemodel

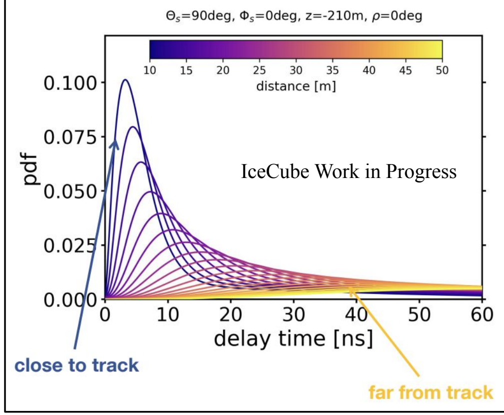
- Preliminary results
- 255 epochs
  - Single H200 GPU
  - Wall time: 6 hrs



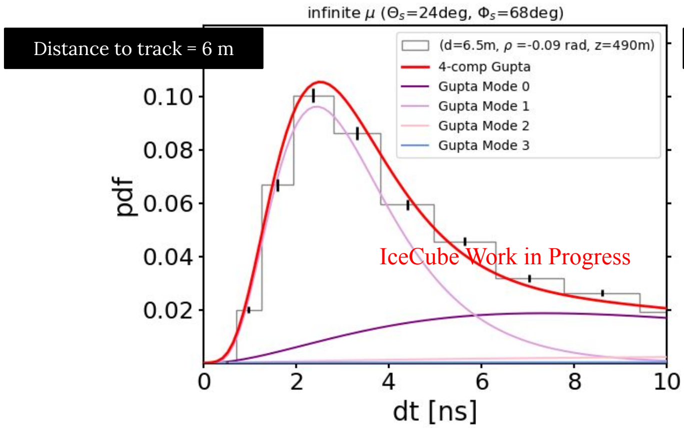
Gupta Mixture Model Network Fits to PDFs

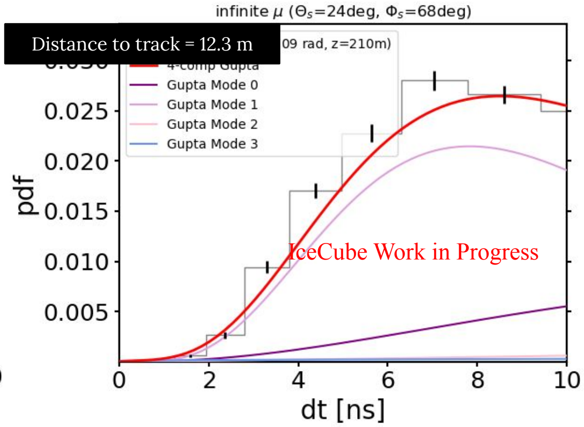


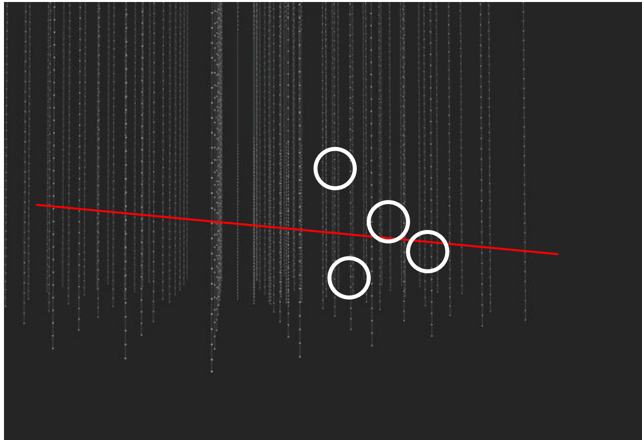




- 4 Gupta function fits capture the PDF
- Gupta component Mode 1 captures the leading edge well
    - Leading edge determines the vertex position of the track

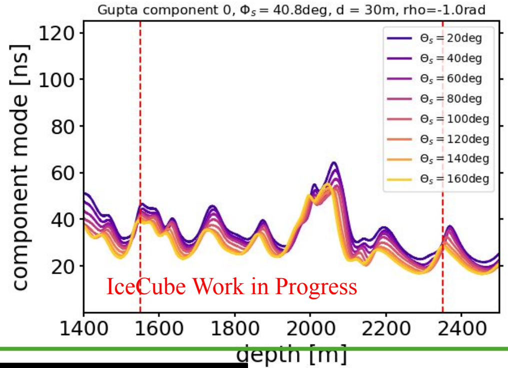




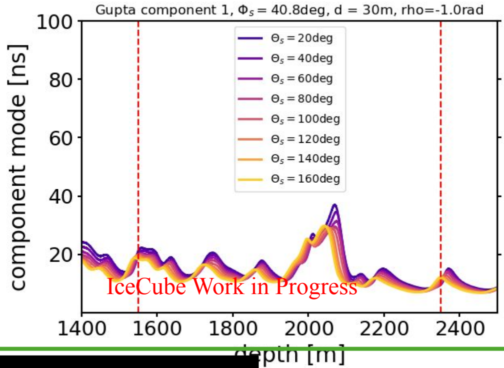


Gupta Component Modes - Depth Variations

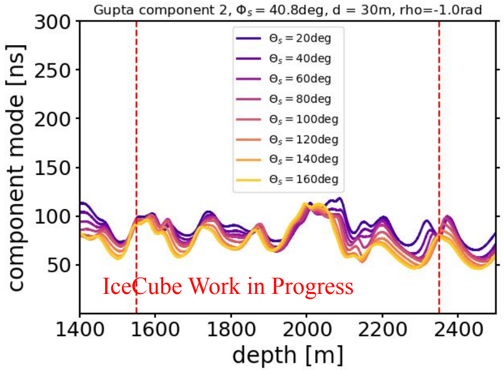
Mode0



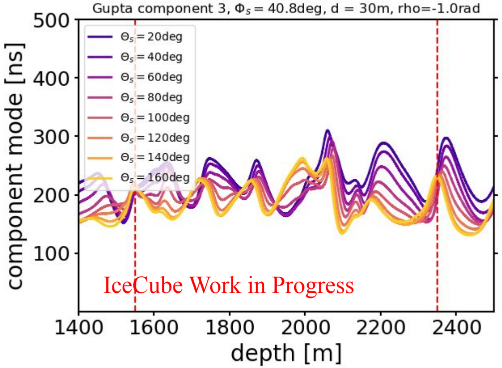
Mode1



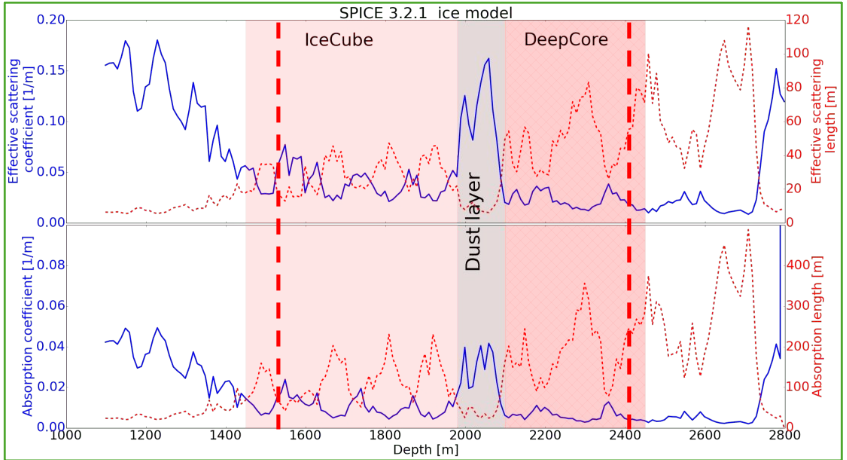
Mode2



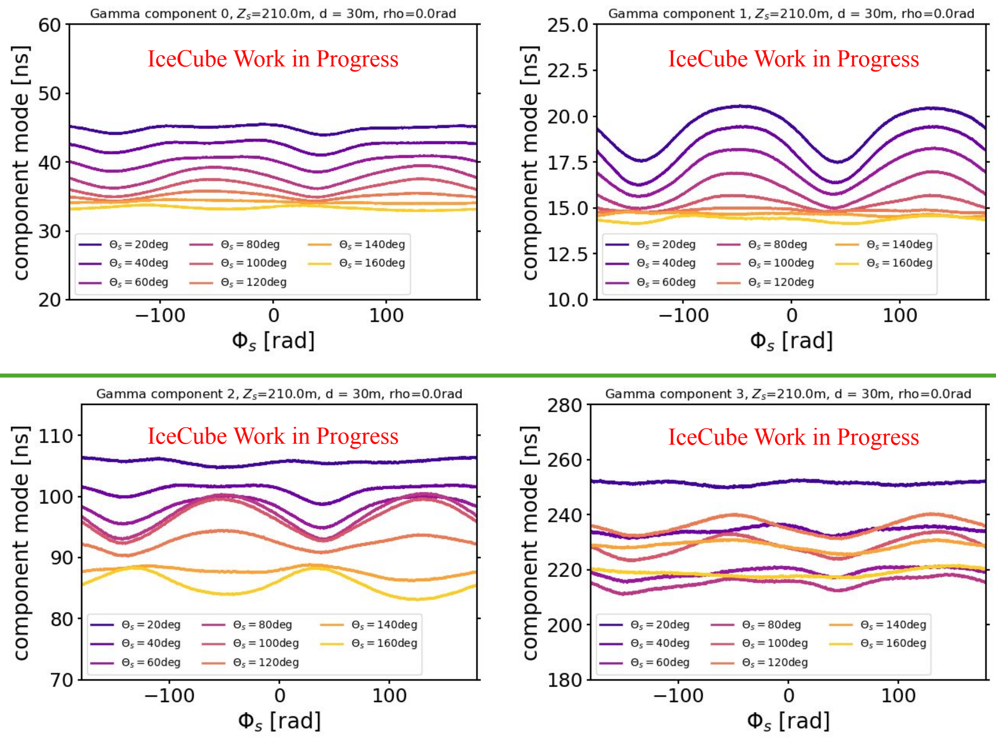
Mode3



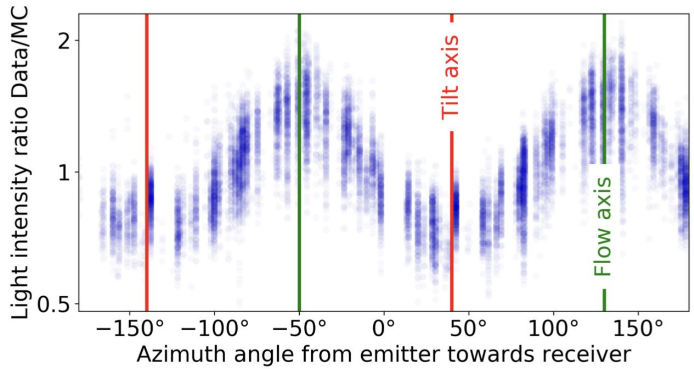
- Distribution of each weight components of the 4 Gupta Mixture model vs depth
- Mode 0 and 1, capture the leading edge of the PDF and the maximum amplitude of the PDF
  - Learn the distribution of dust layers in ice faster than the trailing modes
- Network even sees the small variations in the dust present in the icelayers at South pole



Gupta Component modes - Azimuthal variations



- Distribution of each components of the 4 Gupta Mixture model vs azimuth
- Mode 0 and Mode 1, capture the ice optical anisotropy
  - We see the birefringent property of the ice
- Modes/Weights capture the light propagation properties in ice for both depth and azimuth
- **Network learns the underlying physics of photon propagation**

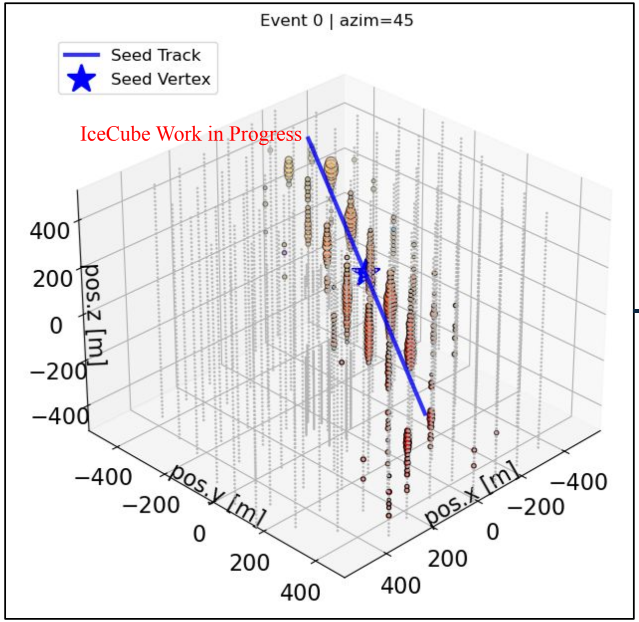


Neutrino Track Reconstruction

- Perform reconstruction using the 4 Gupta network parameters
- Use the photon PDF to compute the likelihood of the track direction
- **Network/Gradients**: fp32 precision (speed)
- **Likelihood minimization:** fp64 precision (accuracy)

- Using **Optax**,
  - Gradient calculation and optimization library for JAX for likelihood profiling
- BFGS minimizer

- Reconstruction performed in tensorflow batches
- SplineMPE - current most accurate reconstruction method in IceCube
  - Uses splines, piecewise polynomial functions, sampled from the photon tables to fit the photon PDF
- **Results from the Gupta Network is compared to SplineMPE in the following slides**

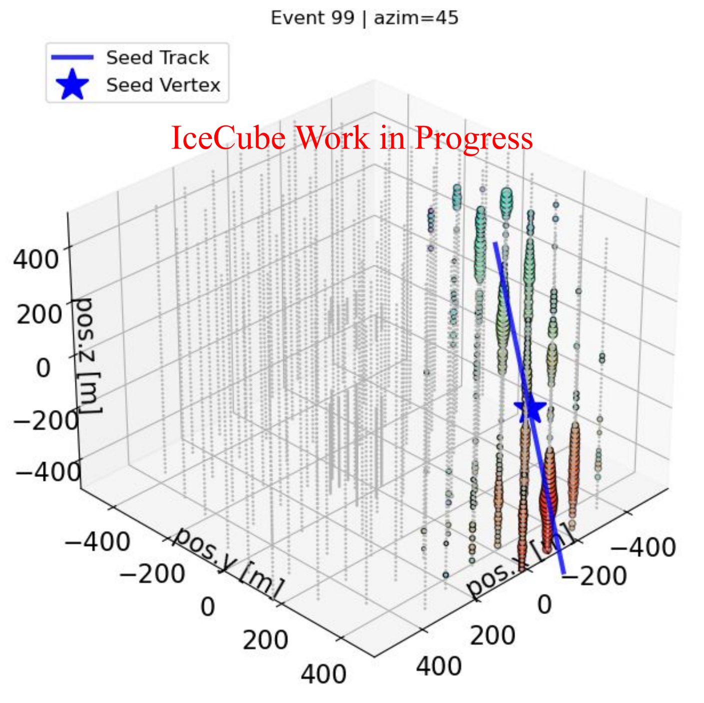


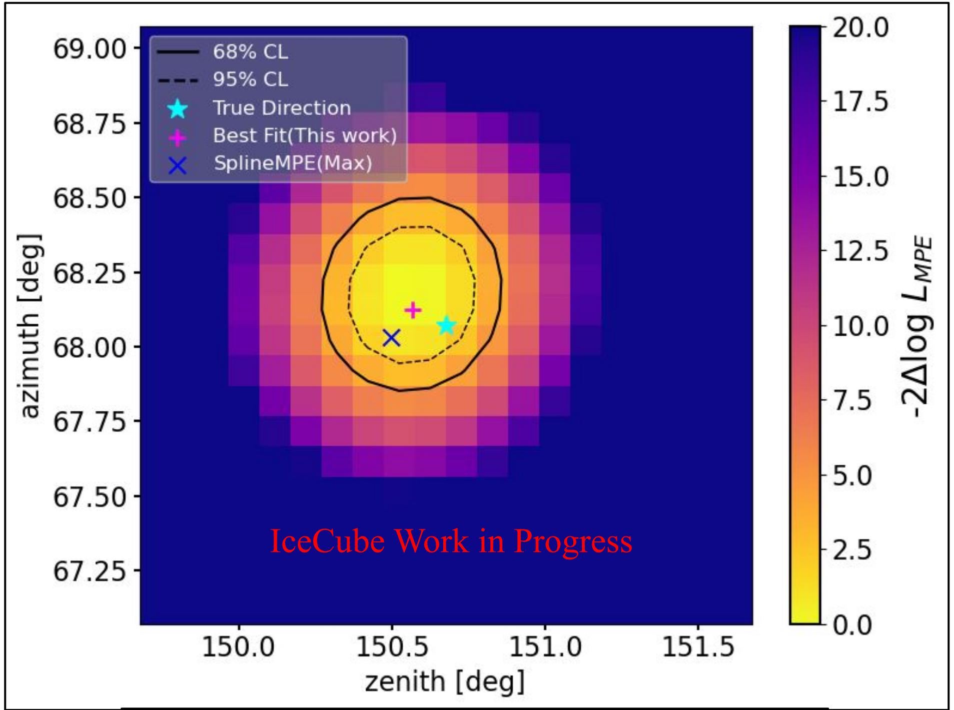
Gupta Network PDF Weights

Reconstruction Block

Predict Track Parameters

Event Alpha: High Energy, Large Number of PMT Hit

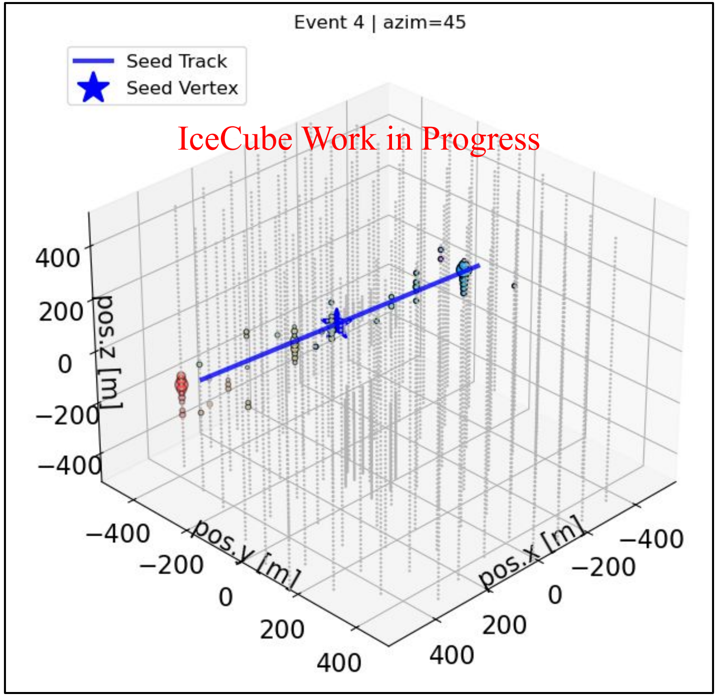


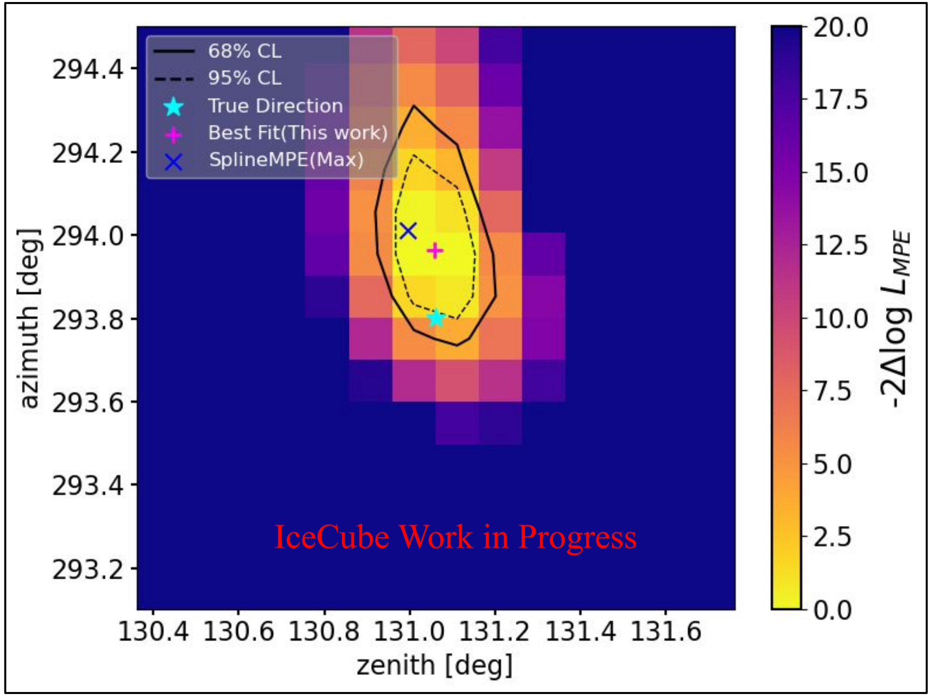


True Neutrino energy : 4805.7 TeV
Distance from Gupta Network pred to Truth: 0.11 deg
Distance from SplineMPE to Truth : 0.18 deg

Likelihood profile scans

Event Beta: High Energy, Small Number of PMT Hit

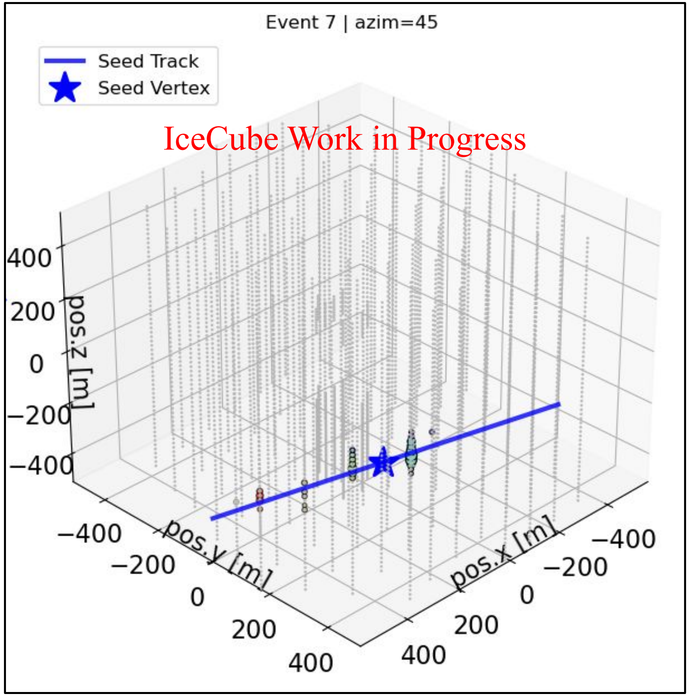


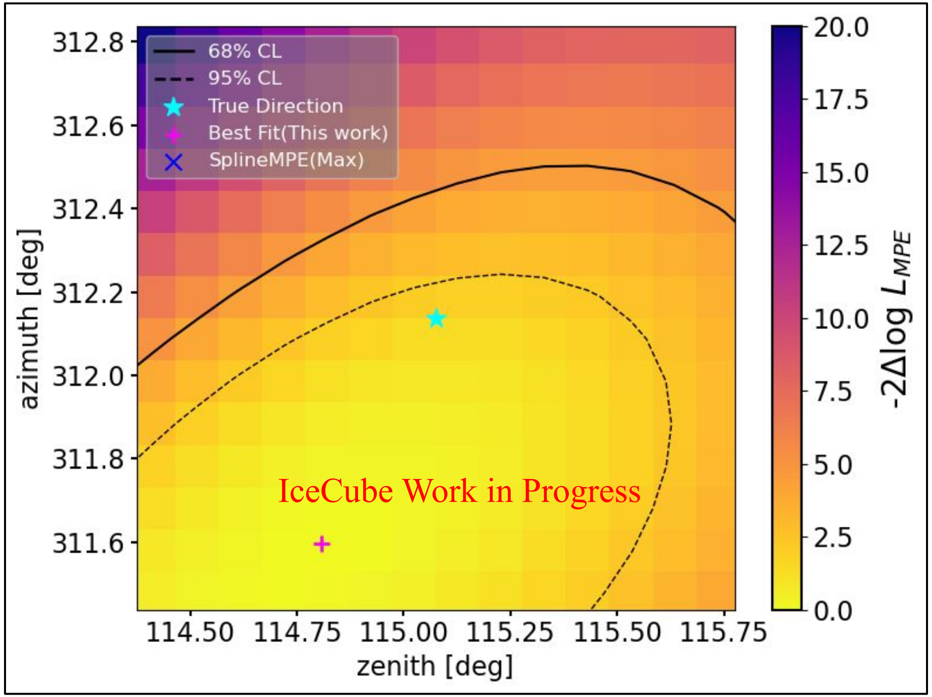


True Neutrino energy : 4911.1 TeV
Distance from Gupta Network pred to Truth: 0.12 deg
Distance from SplineMPE to Truth : 0.17 deg

Likelihood profile scans

Event Gamma: Low Energy, Small Number of PMT Hit

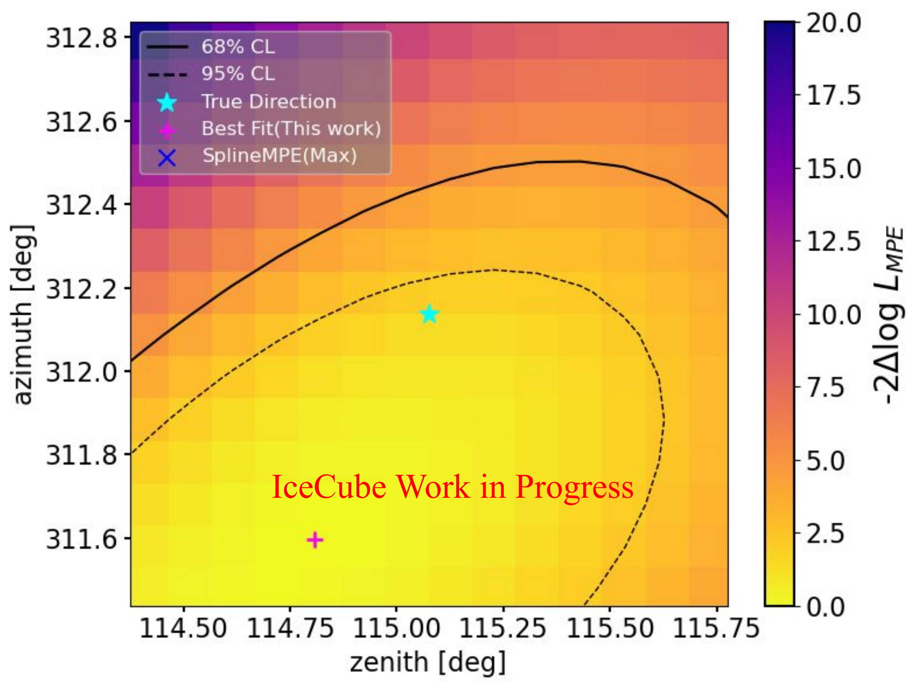


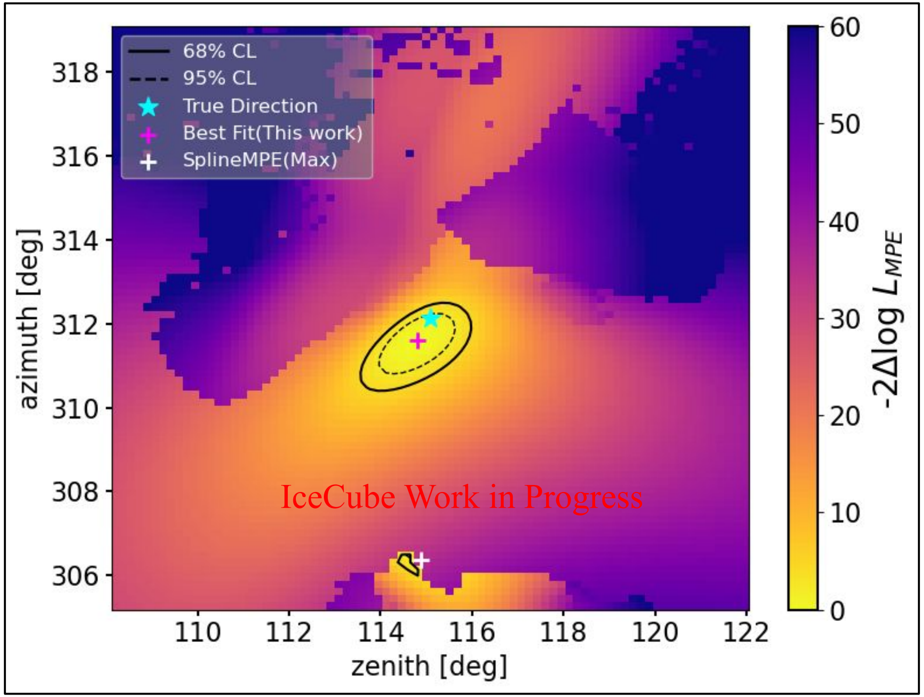


True Neutrino energy	: 405.8 TeV
Distance from Gupta Network pred to Truth	: 0.56 deg
Distance from SplineMPE to Truth	: 5.25 deg

Likelihood profile scans

Event Gamma improvements over SplineMPE





Distance from Gupta Network pred to Truth : 0.56 deg
Distance from SplineMPE to Truth : 5.25 deg

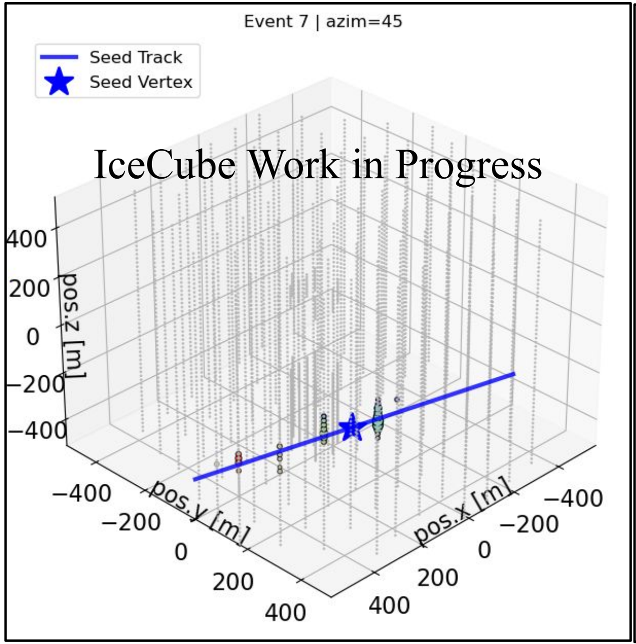
For the improvement:

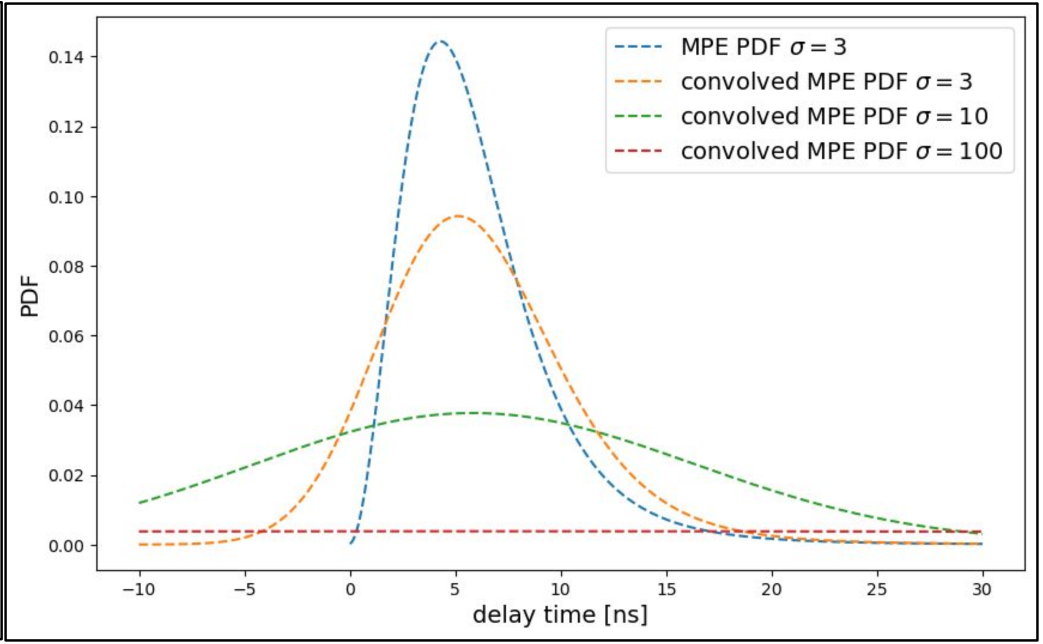
- Need to account for uncertainties in timing resolution from PMT Jitter
- Help the minimizer get unstuck at local minima

Gaussian Convoluted Network PDFs

We account for the uncertainties in timing resolution, PMT Jitter
For the improvement:

- Compute the time smearing of the PDFs using gaussian convolution
    - Convolve individual PDFs with a gaussian



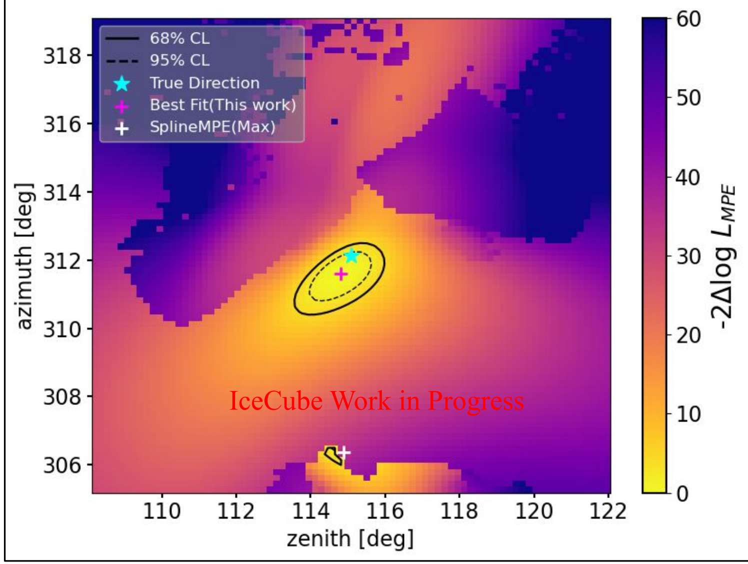


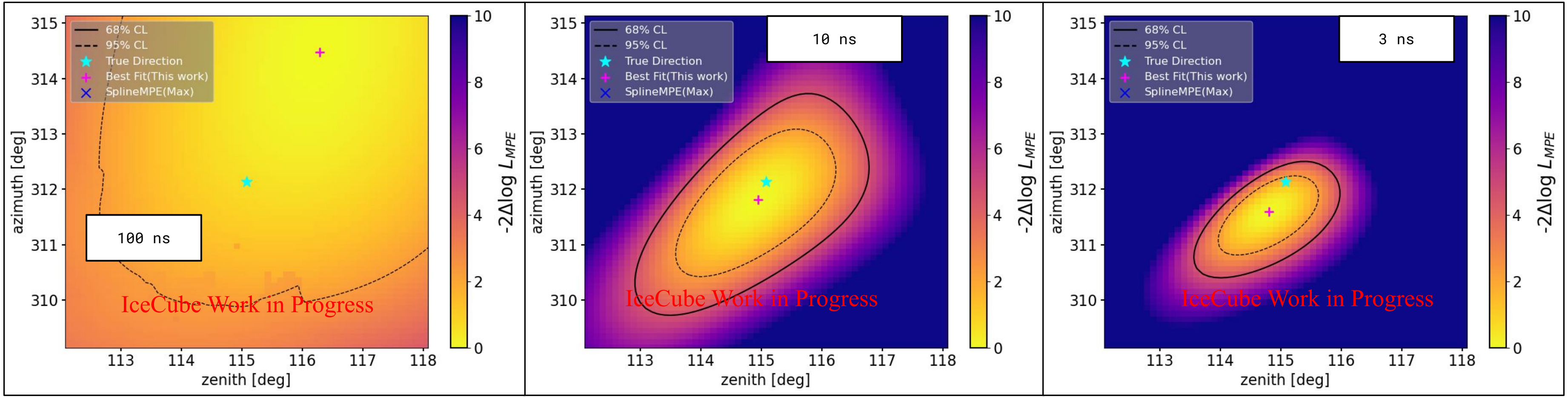
Event Gamma: Time Smearing

We account for the uncertainties in timing resolution, PMT Jitter
For the improvement:

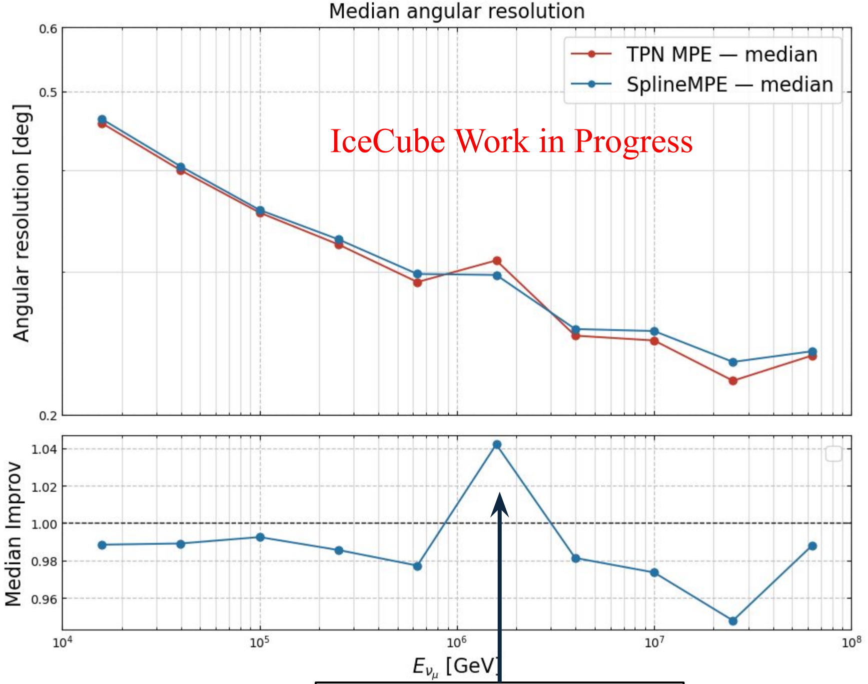
- Compute the time smearing of the PDFs using gaussian convolution
  - Convolve individual PDFs with a gaussian

- Method: Perform 3 likelihood fits with different gaussian widths
(100ns, 10ns, 3ns)
  - Completed in 2s with JAX





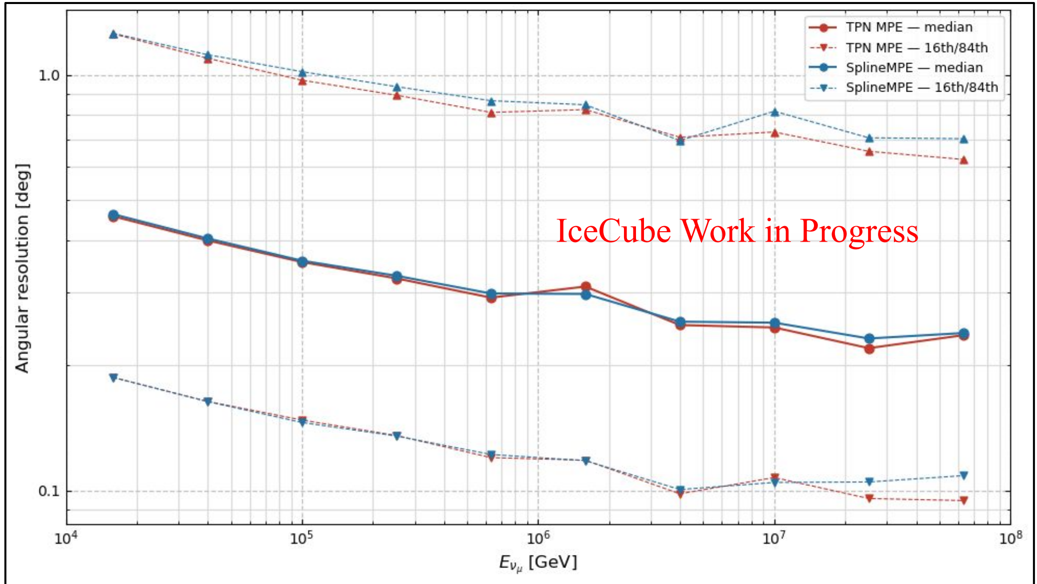
Preliminary Performance compared to Current Reconstruction



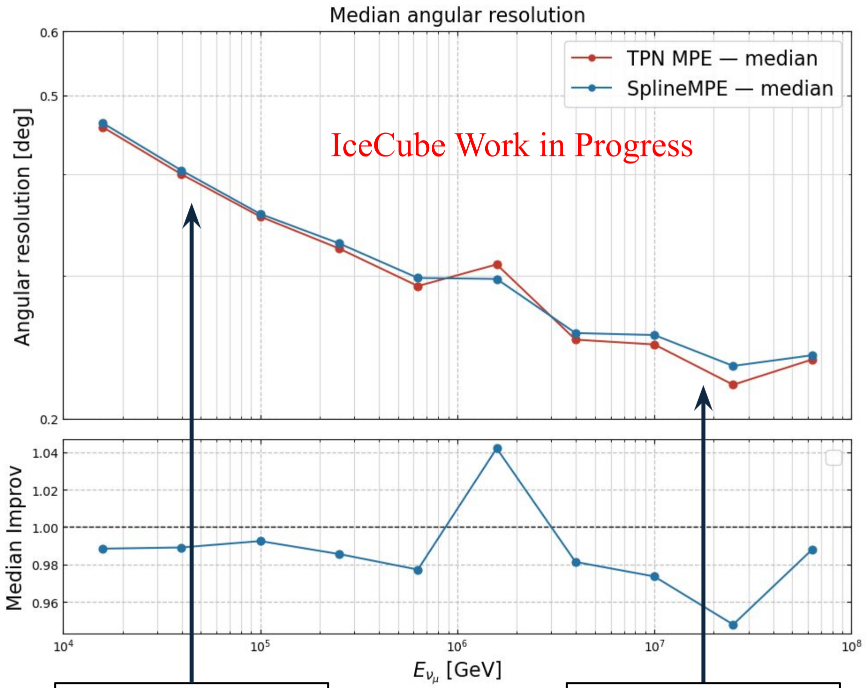
Dataset transition spike due to very low number of events

Total number of events: 118898
(Reconstructed in cumulative total wall time ~ 10 hrs)

- Minimizer seems to be stable
- Caveats: Performing 3 likelihood fits
- Slight improvement over SplineMPE over the full energy range



On-going Improvements



Single PE regime

(Noise Dominated)

High Multi PE

regime

(Stochastics

Dominated)

Final Gupta

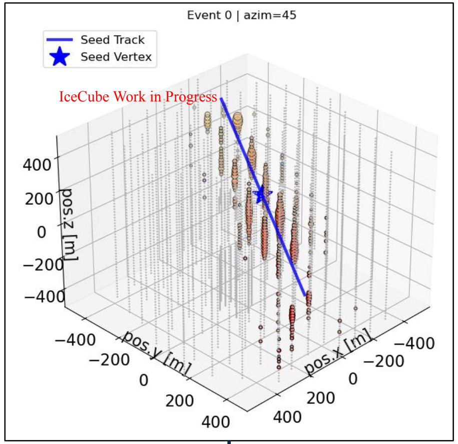
Reconstruction

3 Networks

Gupta Network
PDF

Noise Network
(Model Noise)

Stochastics
Network
(Model
Stochastic Energy
Losses)



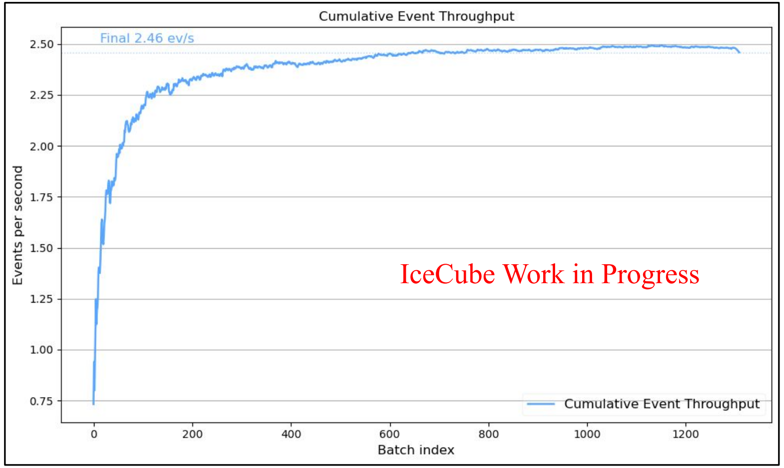
Reconstruction
Block

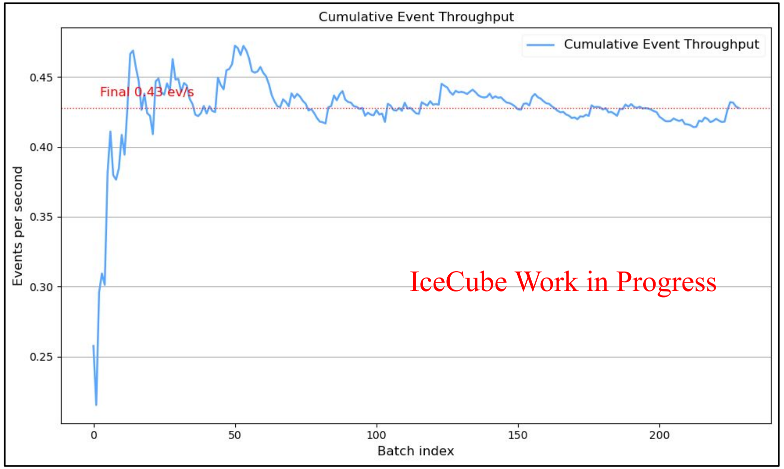
Predict Track
Parameters

Computational Performance: Single H200 GPU

- Simulation event energies: **10 TeV to 1 PeV**
  - **Low DOM hits per event**
- Reconstruction in batches
  - 115k events completed in 6 hrs
    (parallel computation on HPC cluster)
- Mean runtime per event reconstruction: **400ms**

- Simulation event energies: **1 PeV to 100 PeV**
  - **Higher DOM hits per event**
- Reconstruction in batches
  - 5k events completed in 2.5hrs
    (parallel computation on HPC cluster)
- Mean runtime per event reconstruction: **2.3 s**





Outlook:

- This study shows improvement of Gupta Network over the current implementation of track reconstruction SplineMPE
- Additional Improvements we are working on are
  - Addition of noise models
  - Implementing a new neural network to model the stochastic energy losses for very high energy neutrinos
- First full implementation of a **blazingly fast JAX implementation** of track reconstructions based on the Gupta PDFs
- Addition of JAX to this project improved the computational performance by an order of 15 times compared to its conception

Rishi, Hans, Brandon, Mehr
Development of the entire reconstruction scripts and the stochastic loss network

Matti
Working on MPE PDFs and theoretical limit of the MPE Reconstruction Resolution

Chang, Minji
Development of noise models and optimization schemes for the training network