Energy Reconstructions with CNNs in IceTop

Frank McNally
for the IceCube Collaboration

Machine Learning Workshop 2022

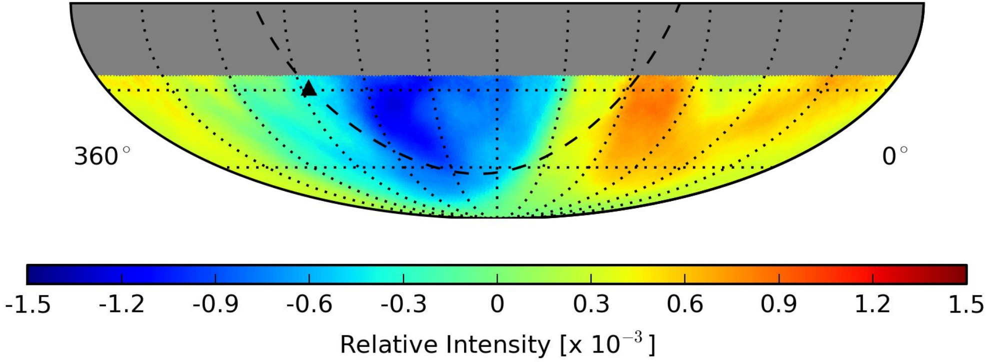
Project Overview

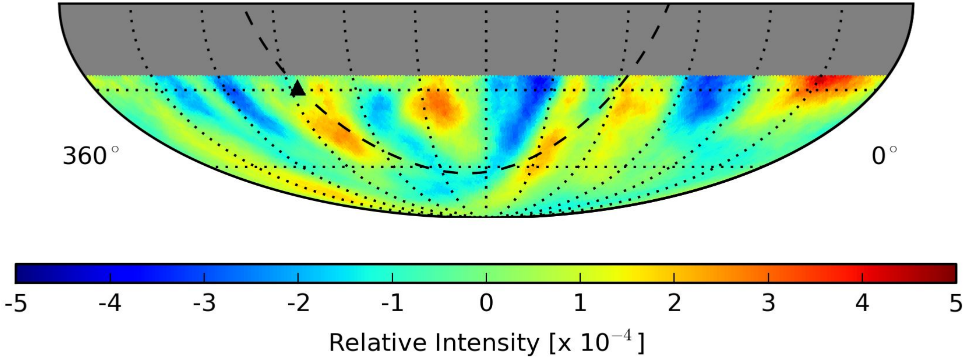
- **Primary Objective:**
  - Wide-FOV reconstruction of composition and energy for use with cosmic ray anisotropy studies

- Secondary Objectives:
  - Proof-of-concept energy reconstruction
  - Undergraduate education

Cosmic Ray Anisotropy

- What is it?
  - Excesses and deficits on the order of one part-per-mille and finer
  - Origins may include regular and turbulent galactic magnetic fields, or nearby sources



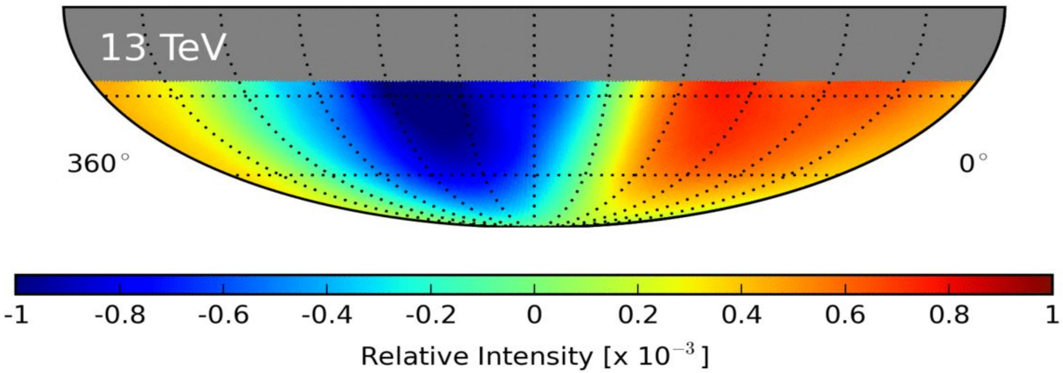


Aartsen et al., “Anisotropy in Cosmic-Ray Arrival Directions in the Southern Hemisphere based on Six Years of Data from the IceCube Detector”, Astrophys.J. **826** (2016) no.2, 220 (arXiv:1603.01227)

Cosmic Ray Anisotropy

- What is it?
  - Excesses and deficits on the order of one part-per-mille and finer
  - Origins may include regular and turbulent galactic magnetic fields, or nearby sources

- Why do we need composition?
  - Need rigidity to properly understand interactions with magnetic fields

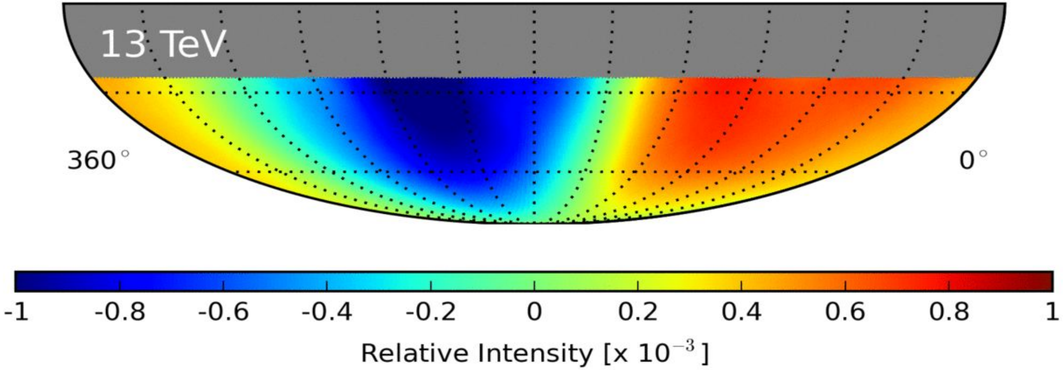


Energy dependence of large-scale anisotropy

(Created from Astrophys.J. **826** (2016) no.2, 220 (arXiv:1603.01227))

Project Requirements

- High statistics
- Detailed event information
- Wide field of view
- Composition-sensitive parameter(s)
- Undergraduate-friendly



Energy dependence of large-scale anisotropy

(Created from Astrophys.J. **826** (2016) no.2, 220 (arXiv:1603.01227))

Strategy

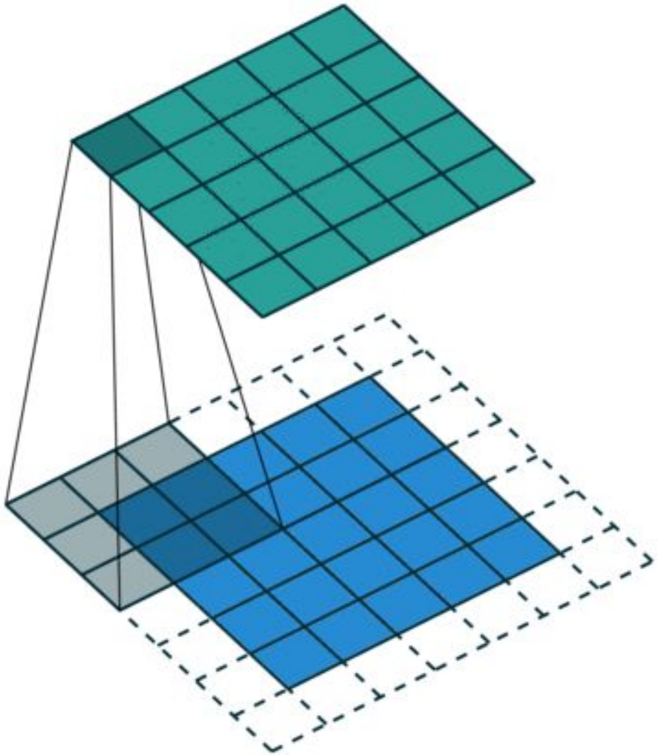


Image credit: MathWorks

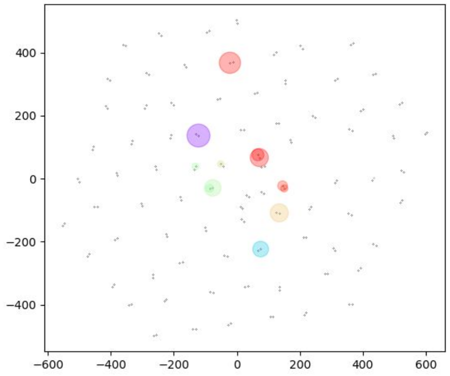
- Use as much base-level information as possible
  - Convolutional neural network (CNN)
  - Electromagnetic component of shower in IceTop
- Start with an energy reconstruction
  - Help understand the data and method
  - Compare to existing reconstructions

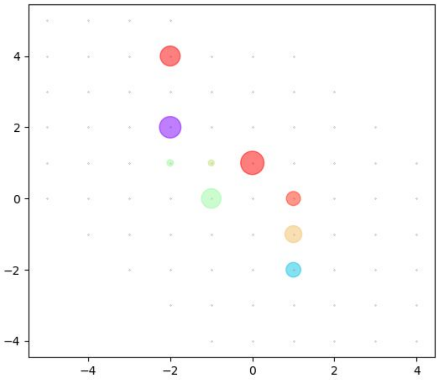
Data

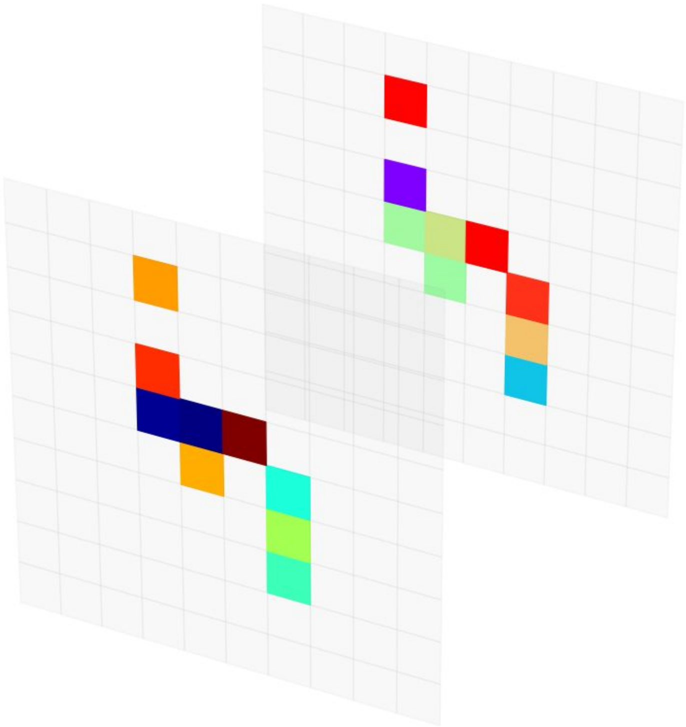
- 81 stations
  - 2 tanks/station
  - 2 DOMs/tank (high/low gain)

- Information per tank
  - Charge
  - Arrival time*

    *waveform available but complicated by reflective tanks

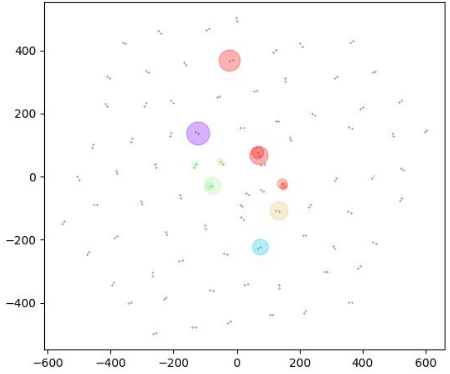


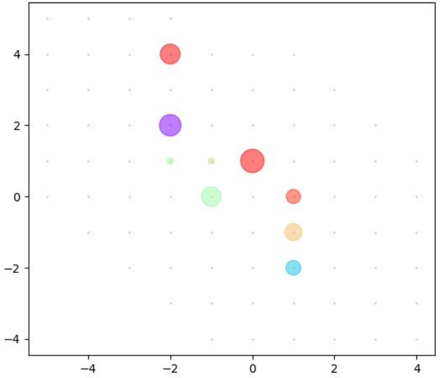


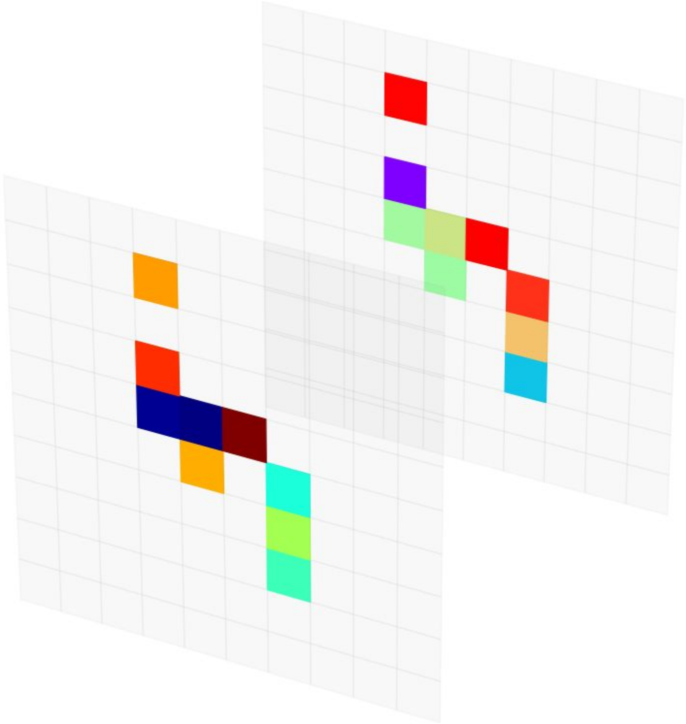


Data

- Array shape: 10 x 10 x N_{layers}
- Possible layers:
  - q_1, q_2, t_1, t_2
  - Q_{1+2}, T_{1+2}







Additional Details

- Simulation
  - Zenith range: 0-65°
  - Energy range: 5-8 in $\log_{10}(E/\mathrm{GeV})$
  - Spectrum: E^{-1}
  - Composition: Proton and Iron
  - Interaction model: Sybill 2.1
- Pre-processing
  - Hex-to-square lattice transformation
  - Charges treated as separate layers
  - Time removed
  - Infill tanks removed
- Partition
  - 90% training
    - 85% training
    - 15% validation
  - 10% assessment
- Quality cut
  - Max charge > 6 VEM
  - Max charge not on boundary
  - Zenith ≤ 40°
  - Keeps ~28% of events
- Resources
  - Models trained on local supercomputer built by Dr. Anthony Choi (Mercer Engineering)

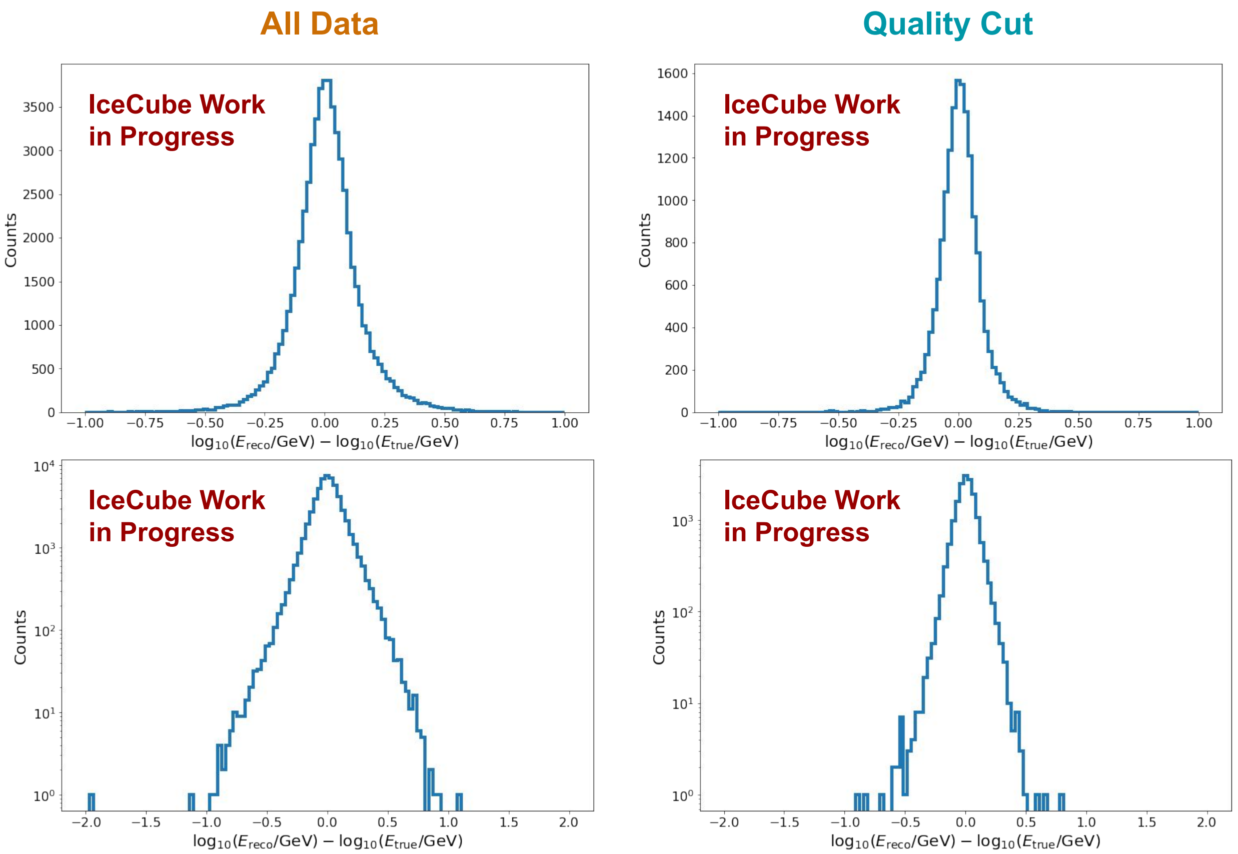
Baseline Model

- Array shape: 10 x 10 x 2
  - Time omitted
- 3 convolutional layers
  - 64 → 32 → 16 filters
- Flatten
  - Concatenate zenith
- 3 dense layers
  - 256 units each
- Output energy

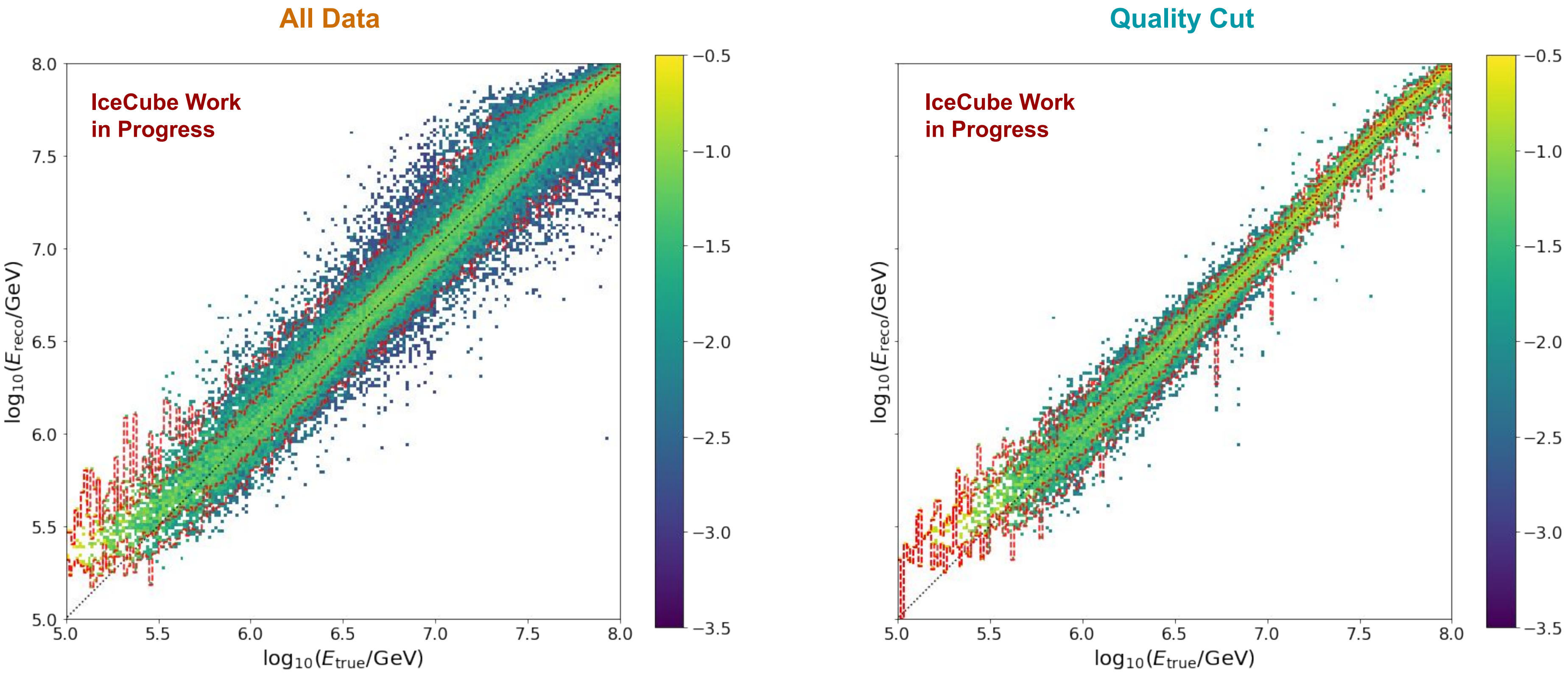
```python
# Create model using functional API for multiple inputs
charge_input=keras.Input(shape=(10,10,2,))

conv1_layer = layers.Conv2D(64,kernel_size=3,padding='same',activation='relu')(charge_input)

conv2_layer = layers.Conv2D(32,kernel_size=3,padding='same',activation='relu')(conv1_layer)

conv3_layer = layers.Conv2D(16, kernel_size=3, padding='same',activation="relu")(conv2_layer)

flat_layer = layers.Flatten()(conv3_layer)
zenith_input=keras.Input(shape=(1,))
concat_layer = layers.Concatenate()([flat_layer,zenith_input])

dense1_layer = layers.Dense(256,activation='relu')(concat_layer)

dense2_layer = layers.Dense(256,activation='relu')(dense1_layer)

dense3_layer = layers.Dense(256,activation="relu")(dense2_layer)

output = layers.Dense(1)(dense3_layer)

model = models.Model(inputs=[charge_input,zenith_input],outputs=output,name=name)

model.compile(loss='mean_squared_error', optimizer='adam', metrics=['mae','mse'])
```

Energy Resolution

- Energy resolution
  - Median ± 34%
- All data:
  - $\Delta E = 0.0 \pm 0.1$
- Quality cut:
  - $\Delta E = 0.00 \pm 0.07$



Reconstructed vs. True Energy

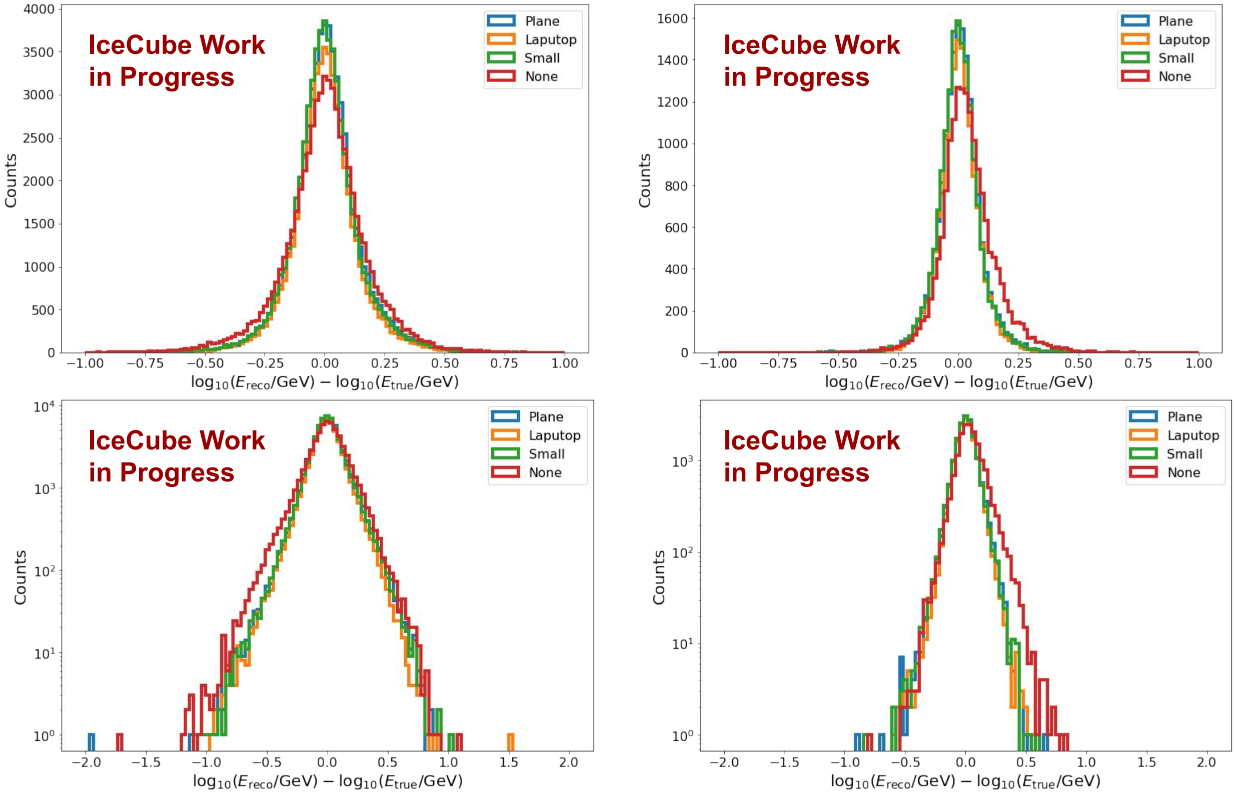


Recent Work: Testing Zenith

Energy Resolution: $\Delta E \pm \sigma/2$

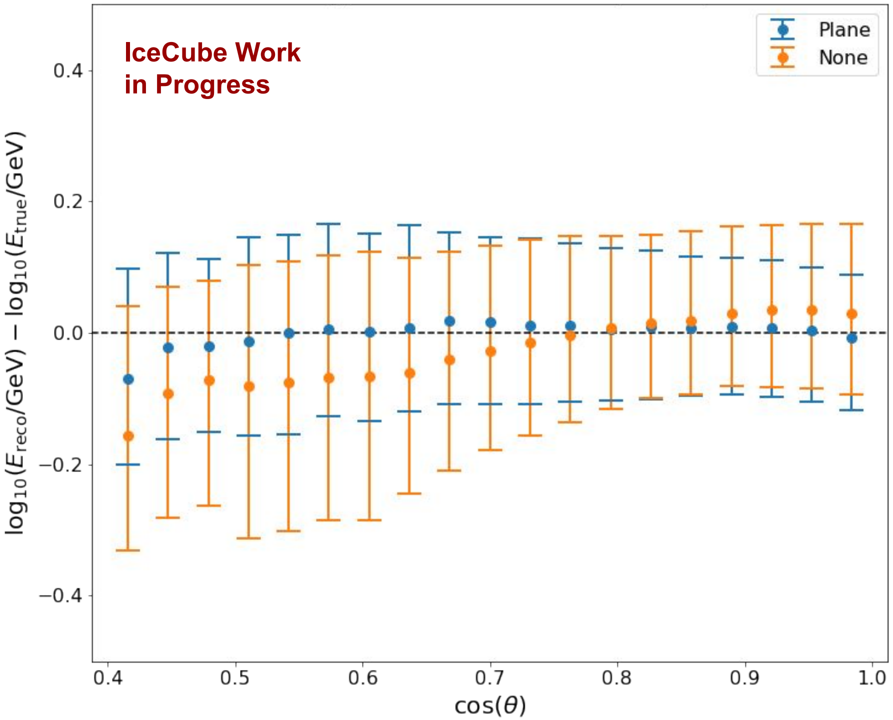
	All Data	Quality Cut
Plane	0.0 ± 0.1	0.00 ± 0.07
Laputop	0.0 ± 0.1	0.00 ± 0.07
Small	0.0 ± 0.1	0.00 ± 0.07
None	0.0 ± 0.1	0.0 ± 0.1

All Data **Quality Cut**

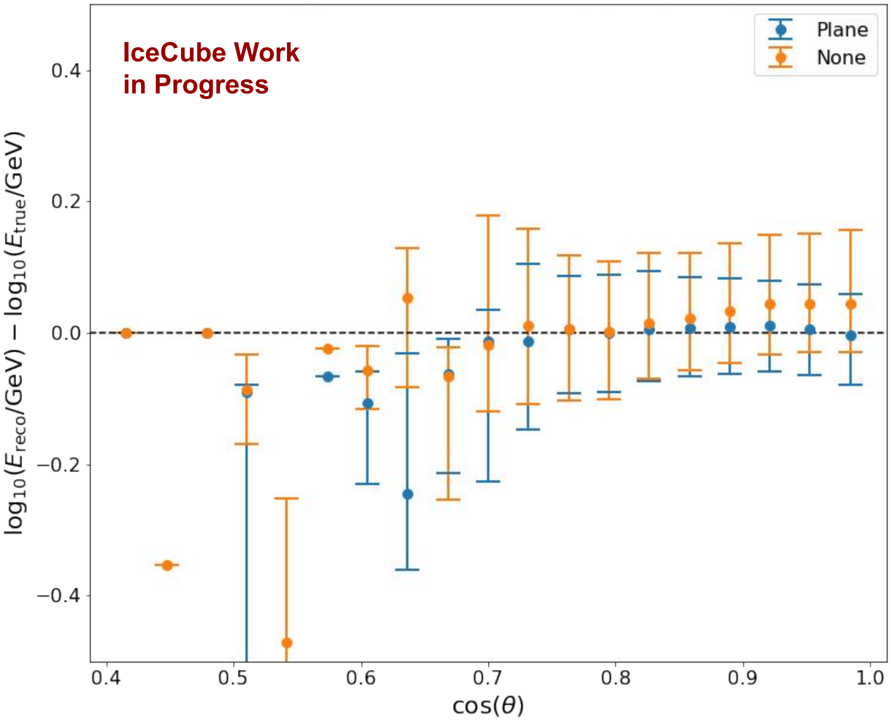


Recent Work: Testing Zenith

All Data



Quality Cut



Summary

Results

- Early energy reconstructions performing well
  - 68% of events within ~10% of true energy with quality cuts
- Incorporating zenith improves reconstruction
  - All reconstructions comparable

Additional Tests

- Training on quality cut data
  - Worsens reconstruction
- Merging charge layers
  - All reconstructions comparable
- More complex network architecture
  - Reconstructions comparable

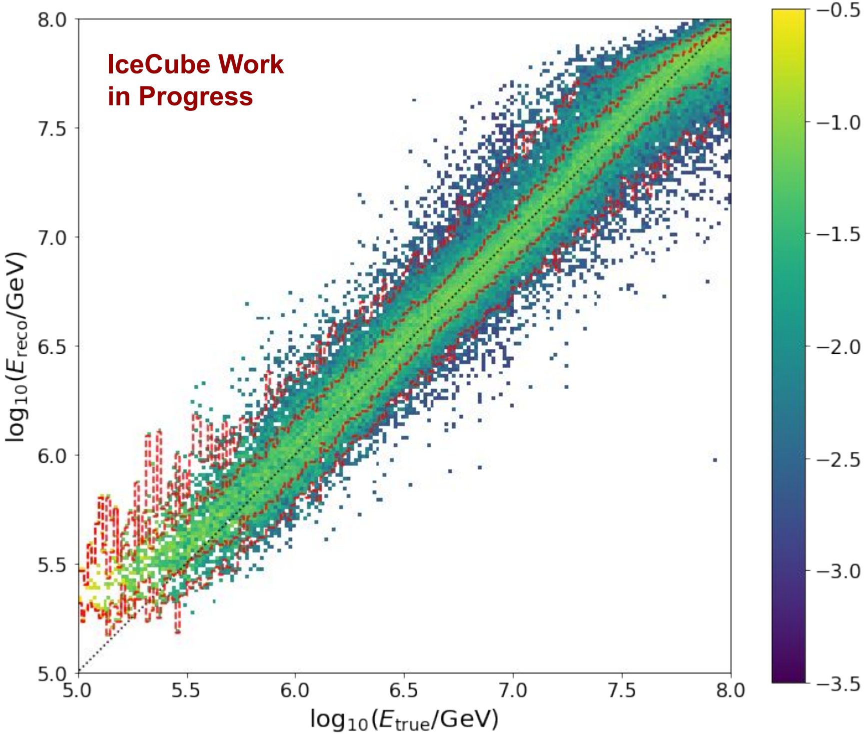
Suggested Tests

- Creative network architectures
- Hyperparameter optimization with Talos
- Including time
  - Time to peak, time to 50%, etc.
- Including more summary parameters
  - S125, CoG, etc.
- Treat events as rotationally symmetric
- Graphical Neural Network

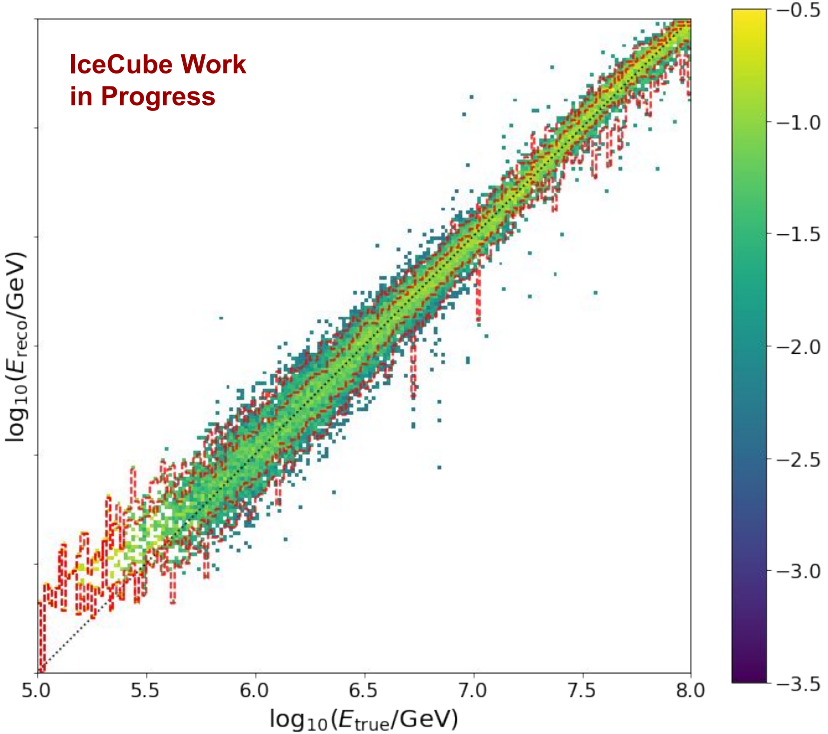
Backup Slides

Reconstructed vs. True Energy (Plane)

All Data

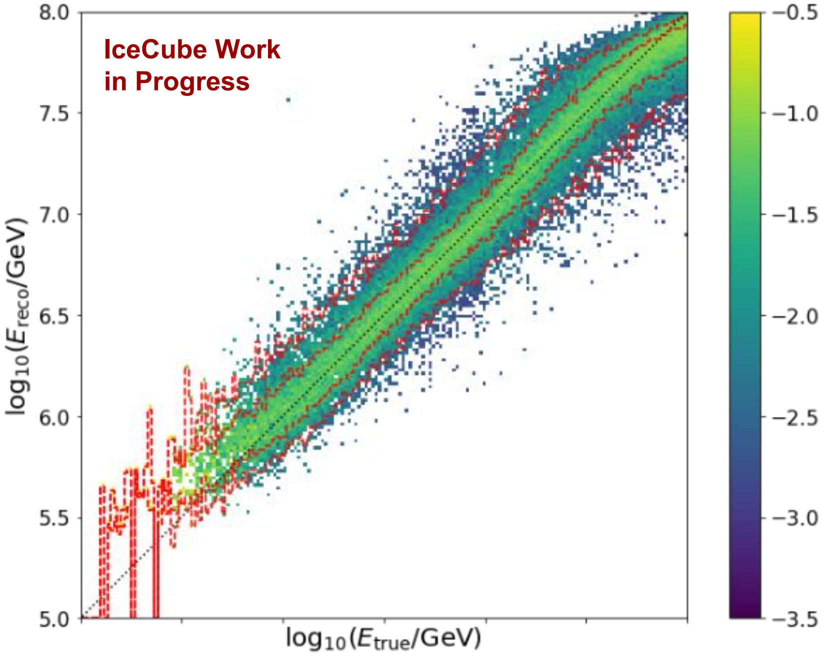


Quality Cut

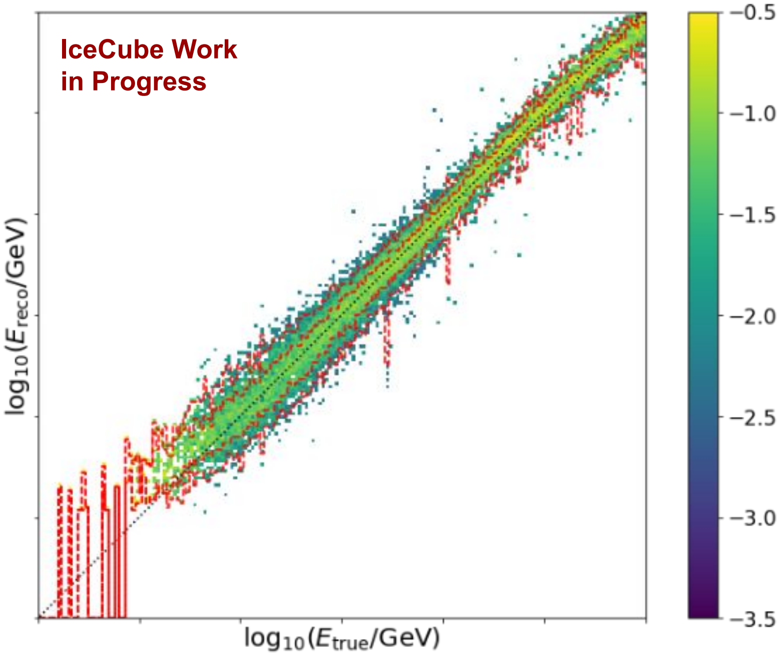


Reconstructed vs. True Energy (Laputop)

All Data

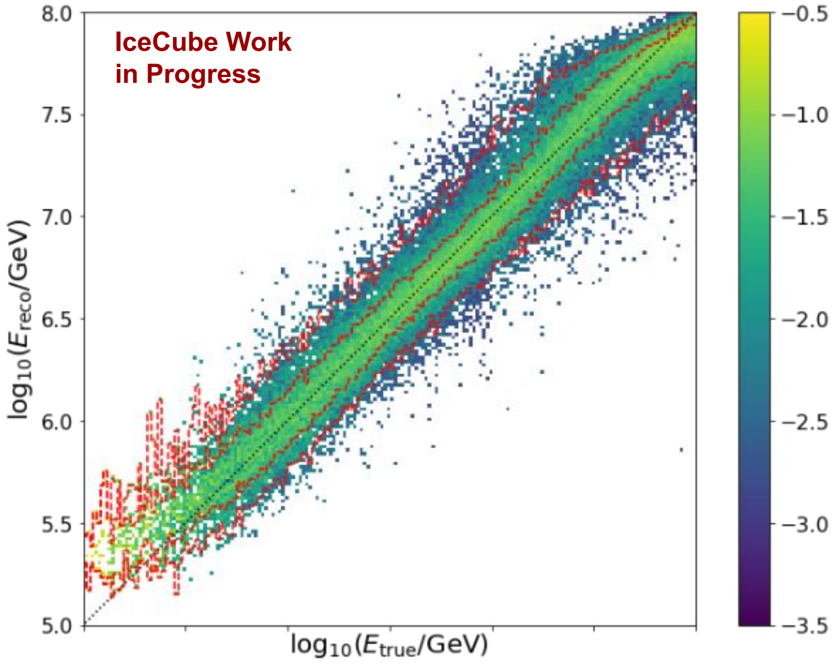


Quality Cut

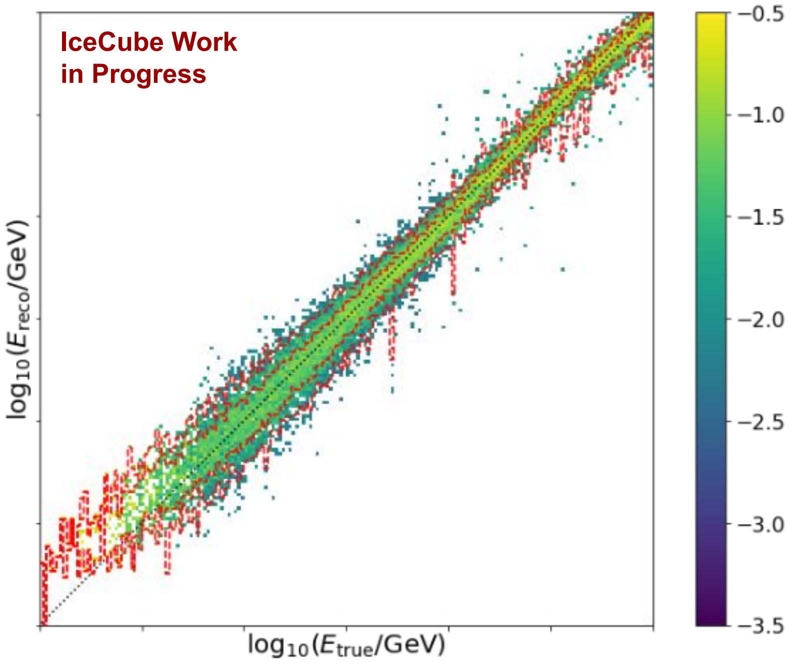


Reconstructed vs. True Energy (Small)

All Data

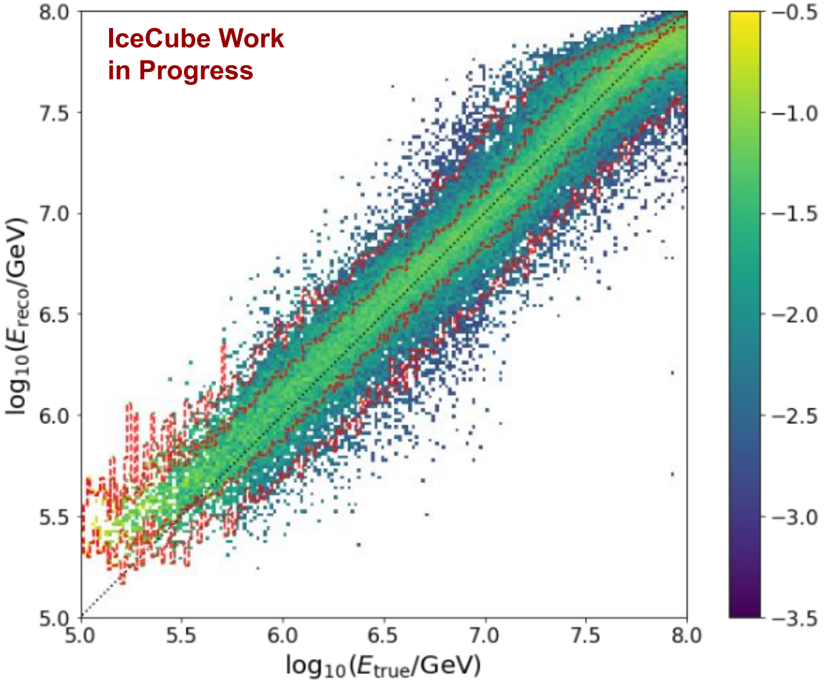


Quality Cut

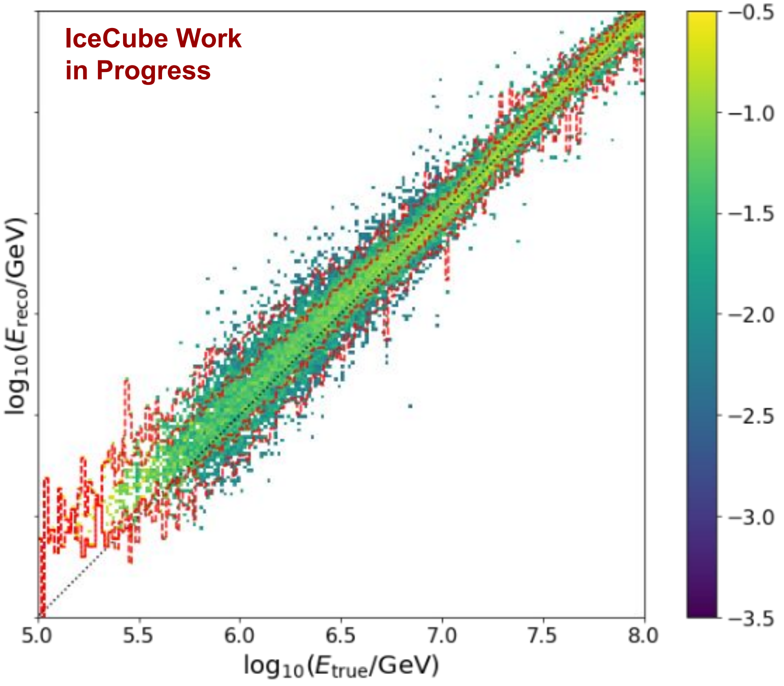


Reconstructed vs. True Energy (No Zenith)

All Data



Quality Cut



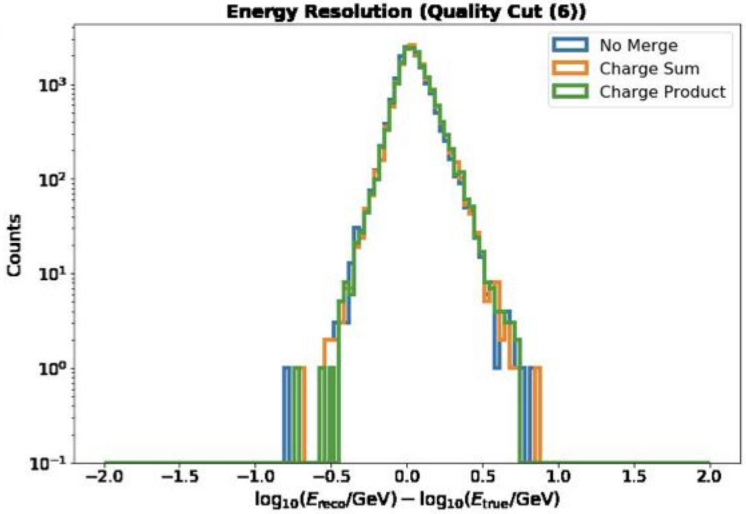
Energy Resolution of Merge Methods

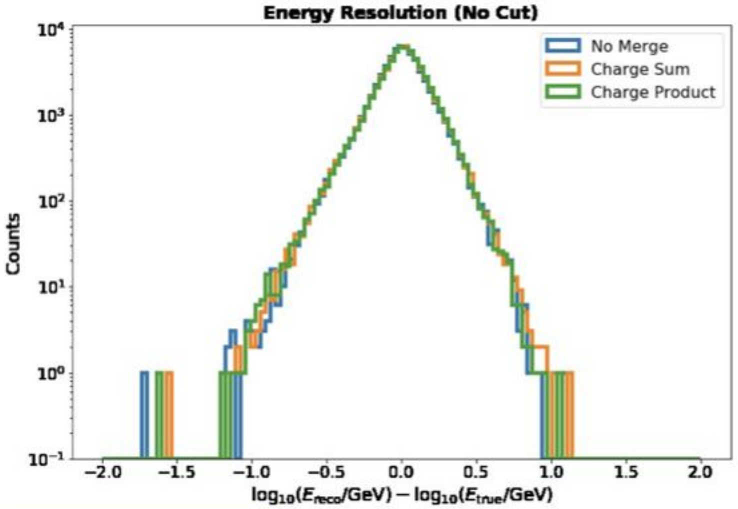
These plots compare reconstructed energy and true energy. In it, there are 3 models each with different methods of merging charge layers.

- No Merging
- Sum
- Product

The results show that there is no perceivable advantage or disadvantage of merging.

Brandon Pascali
Mercer University





Model Architecture

- Model based on the architecture used by Winter et al., 2018