Data-Driven Light Model for the MicroBooNE Experiment

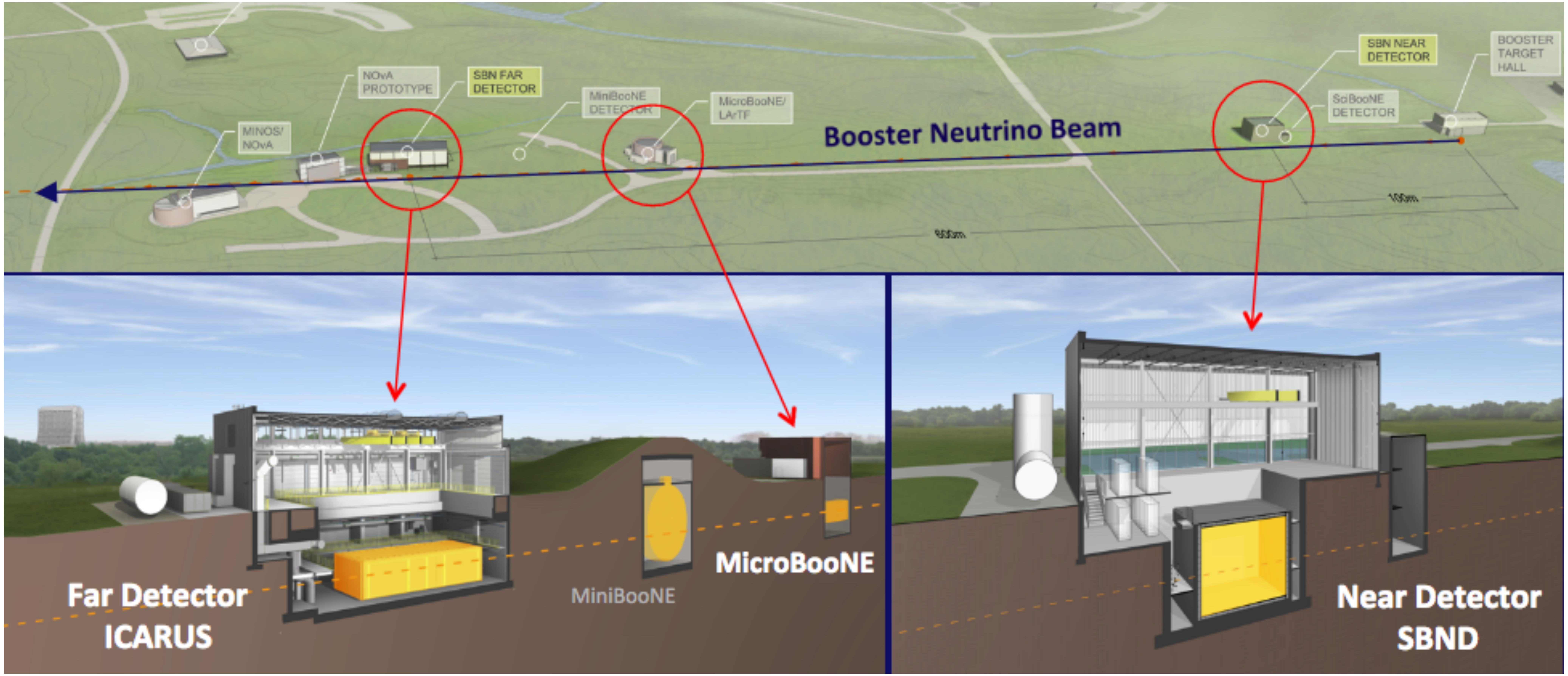
NPML 2024

Polina Abratenko

June 23, 2024

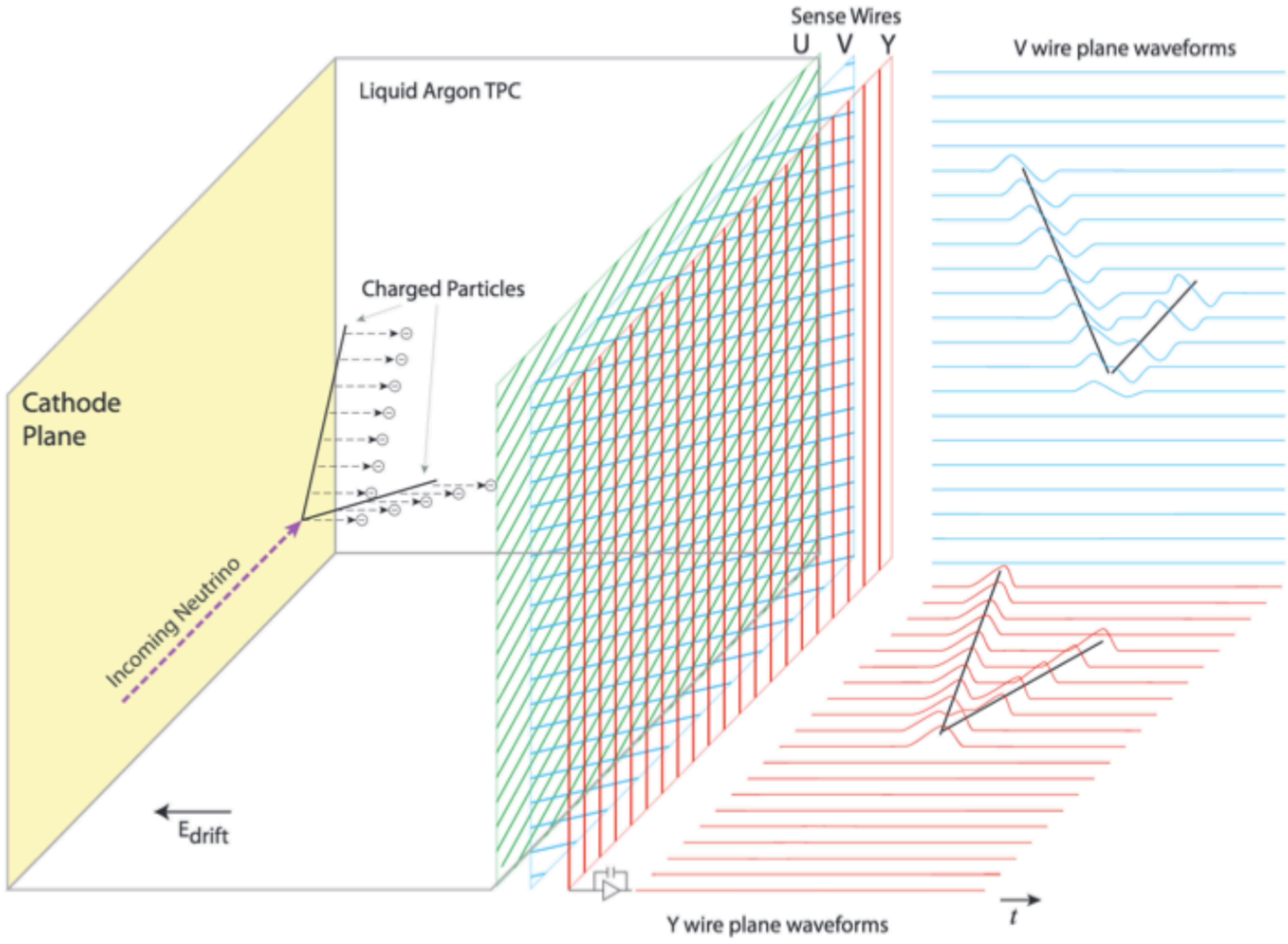
The MicroBooNE Experiment

- Neutrino experiment at Fermilab located along the Booster Neutrino Beamline
  - Currently decommissioned (as of 2021) -> 6 years of data taking
  - Liquid Argon Time Projection Chamber (LArTPC) technology, like ICARUS, SBND, DUNE…

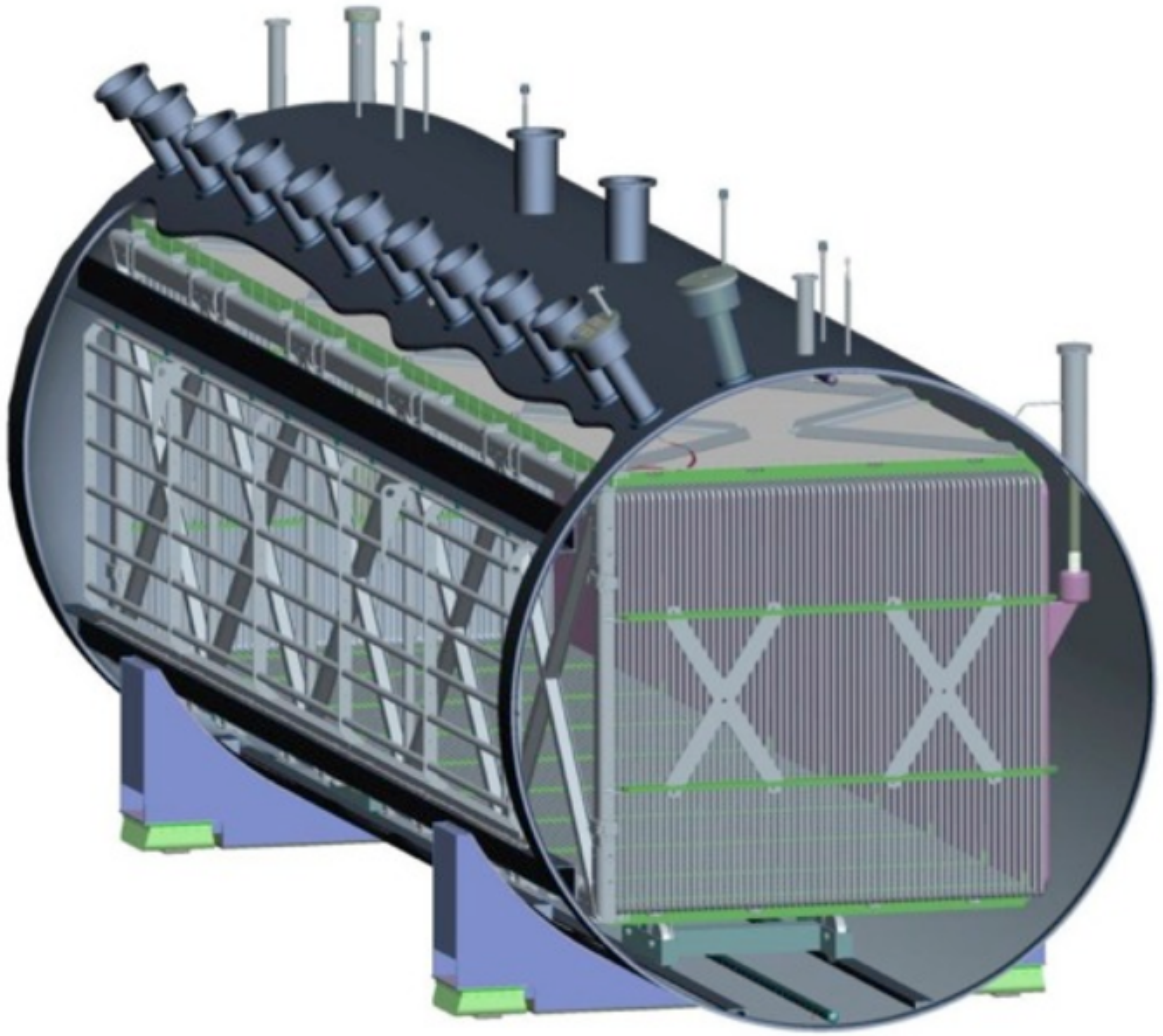


The MicroBooNE Detector: LArTPC

- Particle tracks (charge deposits) reconstructed from wire signals



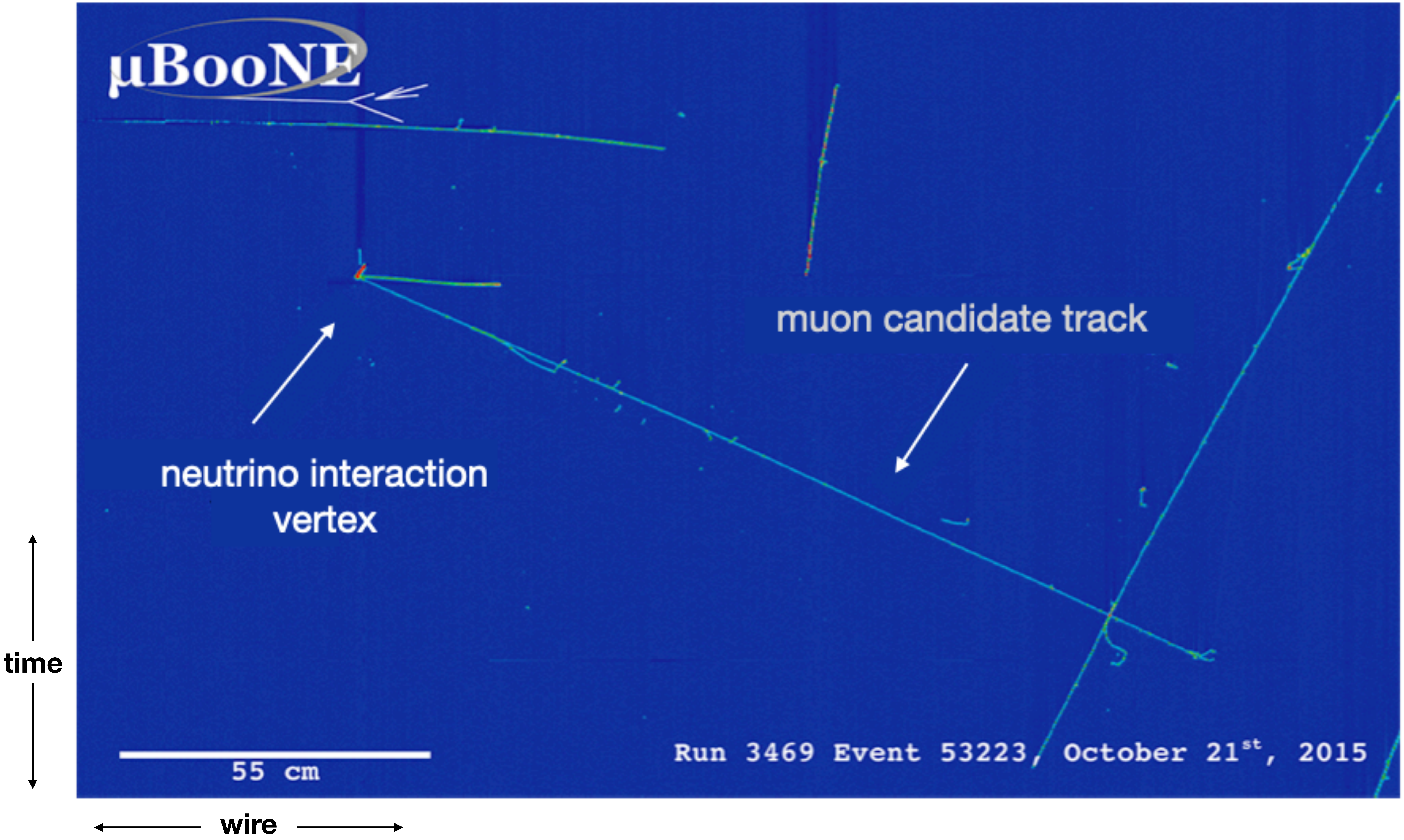
LArTPC schematic. Electrons from ionization of argon drift to the anode plane.



Dimensions: 2.3 m x 2.6 m x 10.4 m (size of a school bus)

had 90 tons active LAr (170 tons in cryostat)

Example MicroBooNE Event Display



3D Visualization of Charge Deposits

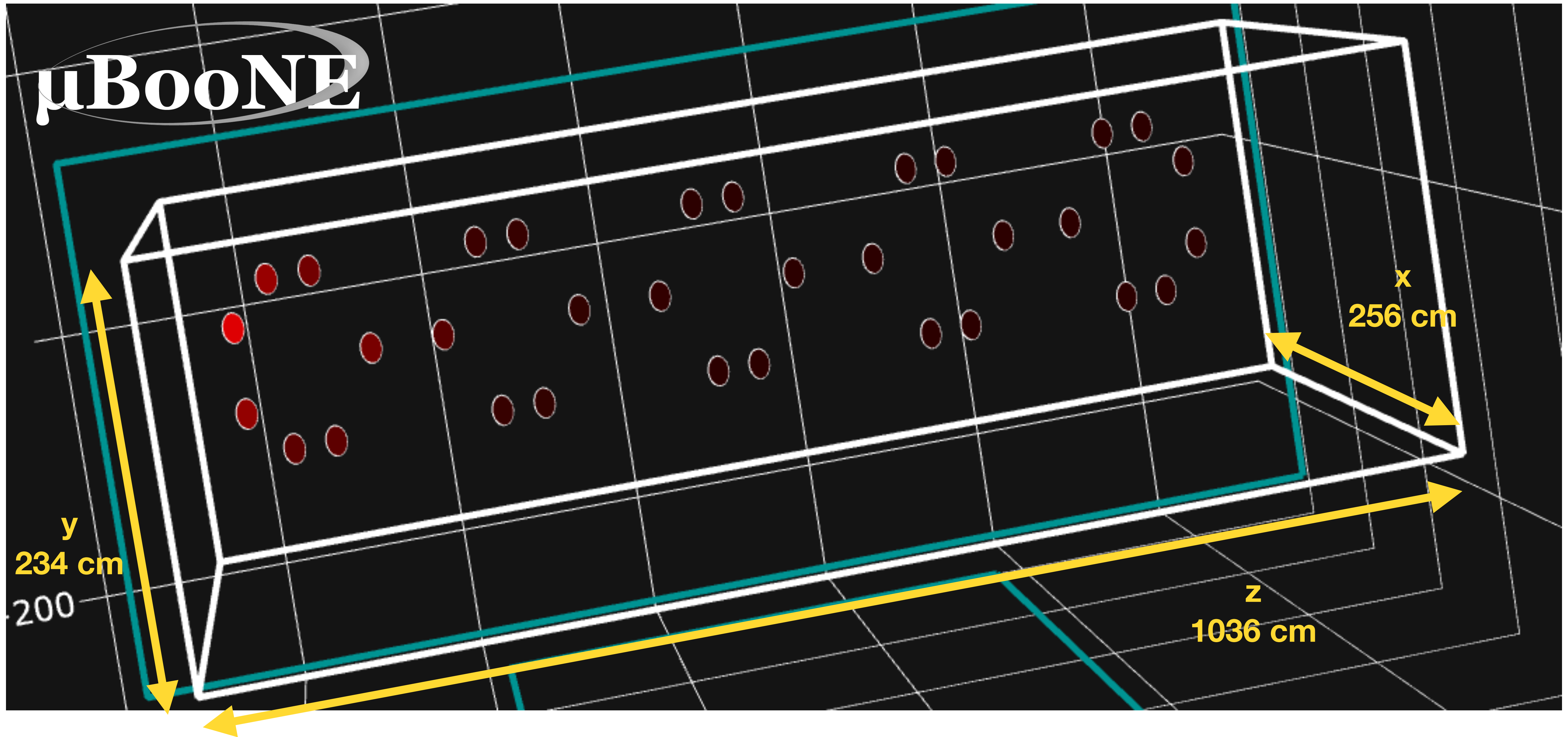
LArMatch
Public Note
MICROBOONE-NOTE-1082-PUB

- Reconstruction using the LArMatch network developed by the Tufts group

The MicroBooNE Detector: Light Detection System

- Liquid argon is a bright scintillator, emits light when hit by radiation
- Set of 32 Photomultiplier Tubes (PMTs): detect scintillation light

8" diameter PMTs along anode side



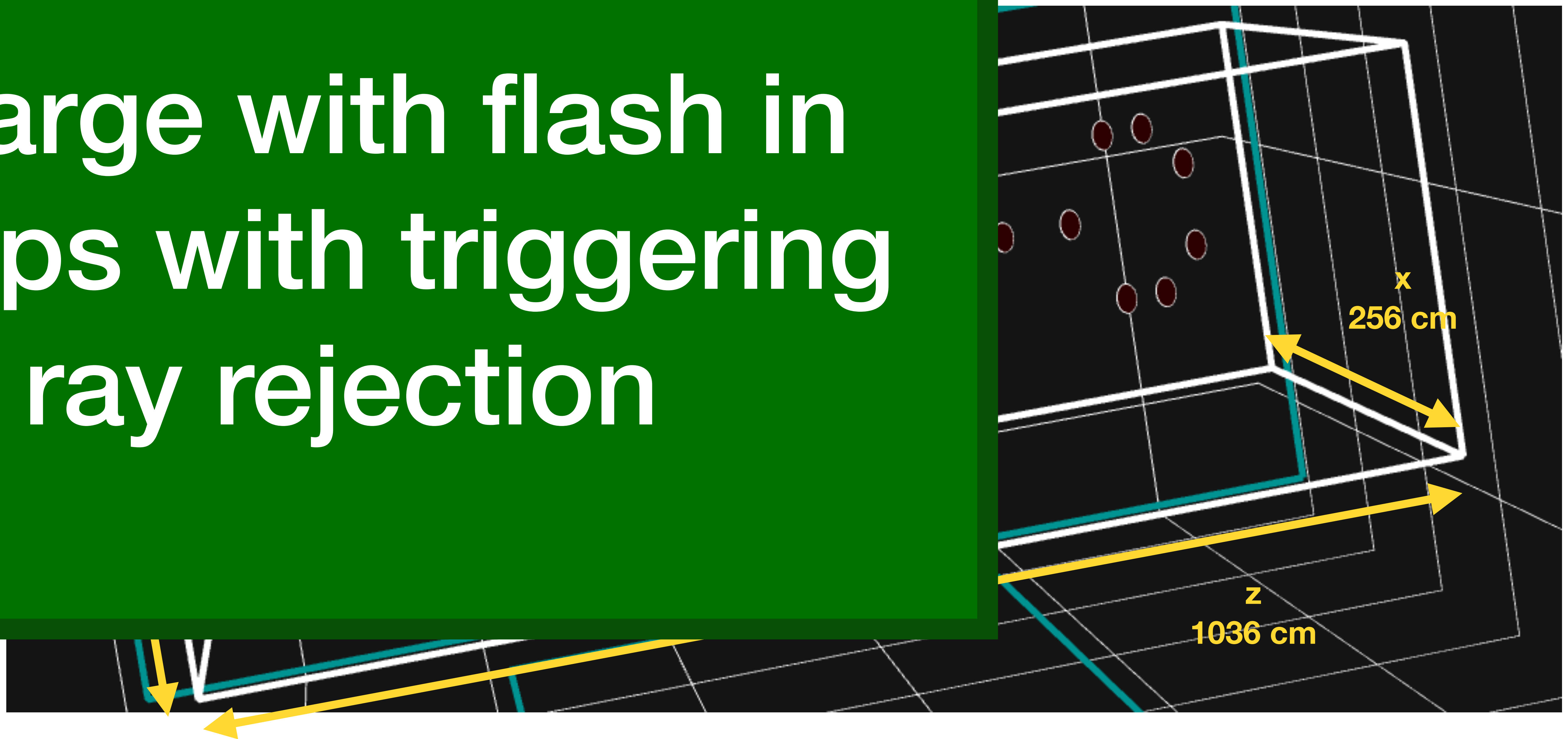
3D visualization: more red means higher photoelectron count

The MicroBooNE Detector: Light Detection System

- Liquid argon is a bright scintillator, emits light when hit by radiation
- Set of 32 Photomultiplier Tubes (PMTs): detect scintillation light

8" diameter PMTs along anode side

Matching of charge with flash in MicroBooNE helps with triggering and cosmic ray rejection



3D visualization: more red means higher photoelectron count

Photon Library in MicroBooNE

- Traditionally, used a lookup table to find the probability of observing a photon produced at a location in the detector
  - Generated simulation of photons emitted isotropically from voxels covering the volume in the detector
  - For all voxel-PMT pairs, calculate visibility: N photons observed / N photons generated
  - Save probability in a library
  - Computationally expensive but done on one go upfront
  - Limitations: Slow to generate, depends on number of voxels, simulation-based, can be inaccurate in certain regions
- It has become clear that the traditional photon library generation approach is not scalable for larger upcoming LArTPC detectors, e.g. in SBN and DUNE

Existing work on improving photon libraries

- Colleagues on MicroBooNE have developed a semi-analytical model, involving an upfront geometric calculation and then simulation-based fitting
  - Is generalized to DUNE and other SBN experiments as well
  - Performs better and faster than the photon lookup library method
  - Relies on simulation for fitting past the analytical calculation
  - Currently used in MicroBooNE and compared to data (public note)
- Light simulation with a 1D generative network (GENN) for protoDUNE/DUNE
  - Lightweight/shallow generative network for running at high speed on CPU
  - Same level of detail and precision as original photon library approach, but faster and more scalable
- SIREN: sinusoidal representation networks for photon propagation
  - Use a MLP with periodic sine function activations with positional information as input
  - Recreates photon library but with fewer parameters than the traditional voxel approach, so is faster, more scalable, differentiable, and potentially tunable to data
  - Is also able to reproduce an acceptance map less sensitive to simulation statistics than the simulated photon library approach

Semi-Analytical Model:
arxiv.org/abs/2010.00324

1D GENN
arxiv.org/abs/2109.07277

SIREN
arxiv.org/abs/2211.01505

Data-Driven Photon Library

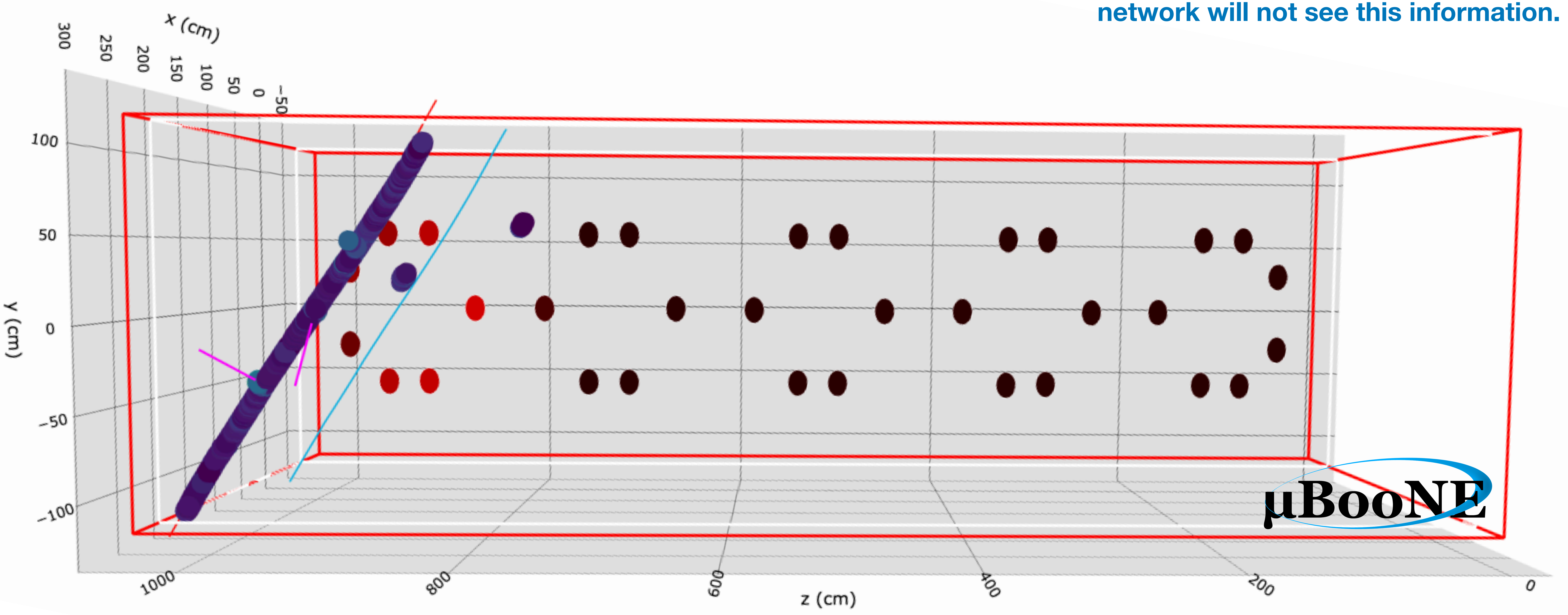
- We are interested in implementing a data-driven photon library in MicroBooNE
- Will allow us to condition on specific runs and detector conditions in MicroBooNE
    - Examples: purity, day; we know the MicroBooNE light yield has declined over time
- May also give us some insight on physics; e.g. behavior of out-TPC light
    - Colleagues have worked on a “point source” Michel selection in data
- My approach is to use custom DL/AI tools developed for MicrobooNE 3D reconstruction
    - Can perform a geometric calculation upfront like the semi-analytical model
    - Combine with neural network output trained on MicroBooNE data
- Have investigated using a baseline network to compare to a CNN
    - We trained a MLP with sinusoidal activations to serve as a baseline
    - The following slides will show results

See talk by Matt R. in this session!

Input Data to the Network

- Match clusters of tracks/showers corresponding to an interaction with associated flash
  - Account for both beam and cosmic events
- Voxelize the charge clusters in 3D

The red and pink lines correspond to truth MCTracks/MCShowers for debugging. The network will not see this information.



Baseline Model for Comparison: SIREN-based

SIRENs for Photon Visibility Mapping

arxiv.org/abs/2211.01505

- Following a path by Patrick Tsang by using a neural network with sinusoidal activations (SIREN)
  - Simple MLP with periodic sine function activations
- Implementation of SIREN from github repo: ‘lucid_rains/siren’
- 7 input variables, represents one voxel
  - (x,y,z) position, each position normalized to 1 by length of detector
  - (dx,dy,dz), distance between the PMT in question and the voxel in the normalized distance units used for (x,y,z)
  - The total distance from the PMT to the voxel, scaled by 1050 cm, roughly the longest dimension of the detector
- 5 hidden layers, 512 features in each hidden layer (to be optimized)
- One output: visibility, a number between [0,1] for the voxel-PMT pair

Predicting OpFlash from the Charge

- Every training example has the amount of charge in a set of voxels as well as the PE for each PMT from the opflash information
- The neural network uses the charge information to calculate PE with:

$$\sum_{i}^{N} q_i * Y * \phi(x_i, y_i, z_i, \Delta x_i, \Delta y_i, \Delta z_i, d)$$

- Here:
  - q is the charge in voxel i
  - Y is the light yield (global charge to PE conversion)
  - ϕ is the visibility function (output of neural network)

Training the Baseline Network

- In training the network, we minimize an objective function with two terms:
  - Norm loss: compare predicted PE sum over the PMTs
  - Shape loss: compare normalized PE in each PMT
- Normalization loss uses the negative Poisson log-likelihood:

λ = predicted PE
x = ground truth PE

$$-\log \mathcal{L}(\lambda|x) = \lambda - x\log\lambda + \log x!$$

- Shape loss uses the Earth Mover's Distance, also known as the transport plan:
  - C is the "cost" between PMT i and j, chosen as location distance
  - Π is the fraction of probability mass from predicted PE to true PE for PMT i to j
  - Must be solved for every (x, x') pair such that it minimizes d

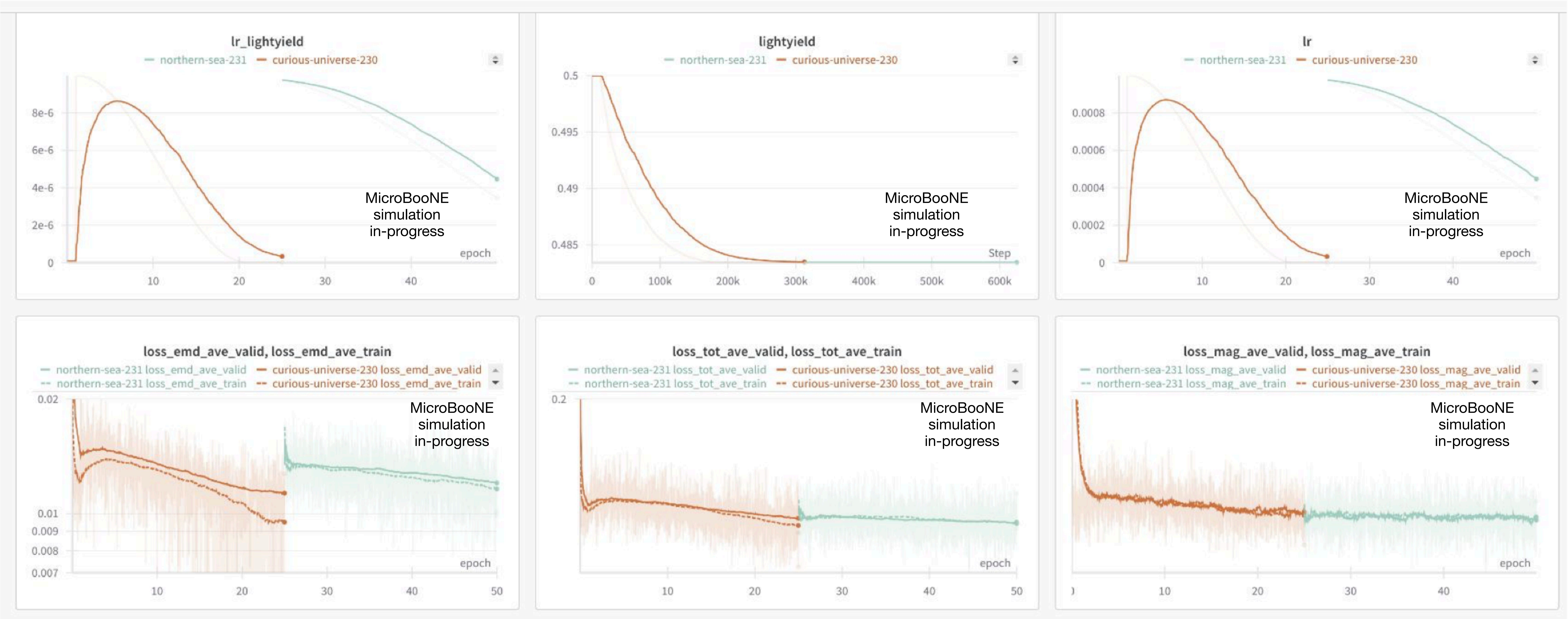
$$\min_{\Pi(x,x')} d_{x,x'} = \sum_{i<j} C(x_i, x'_j)\Pi(x,x')_{ij}$$

Training Methods

- Two stages in training:
    1. Allowed light yield parameter to flow
    2. Hold light yield fixed and using data augmentation techniques
- Apply data augmentation:
    - Upped weight by a random scale factor between [1,5] for examples with: total PE below some threshold (<1) in our normalized units and charge-averaged distance from the anode above 175 cm
    - Apply Mixup: Draw two random training examples. Draw two scale factors from uniform distribution between [0.5, 2]. Add the charge voxels of each example using the scale factors as weights. Add the ground truth PE vectors to each other using the same scale factor weights
    - We apply both, e.g. if small charge cluster drawn for mixup example, it can be scaled up

Training Plots

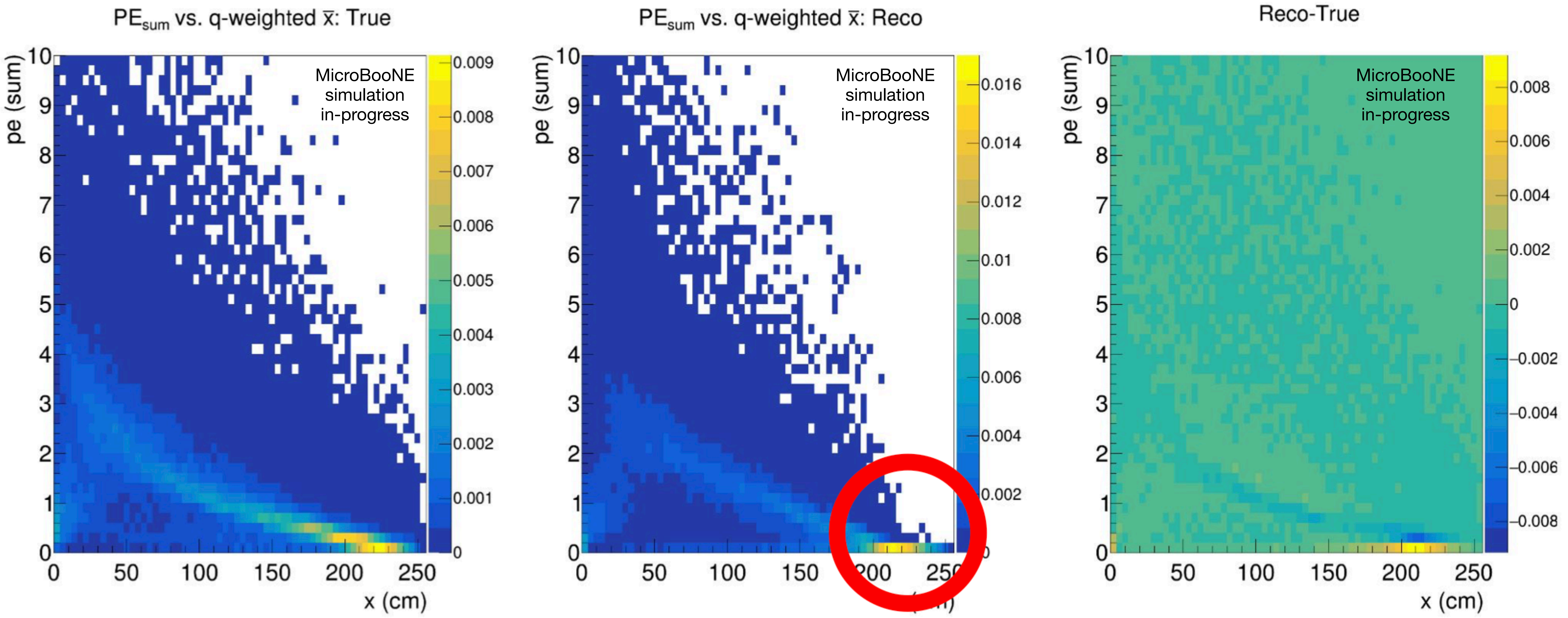
- Used cosine annealing for both stages



Results after Stage 1

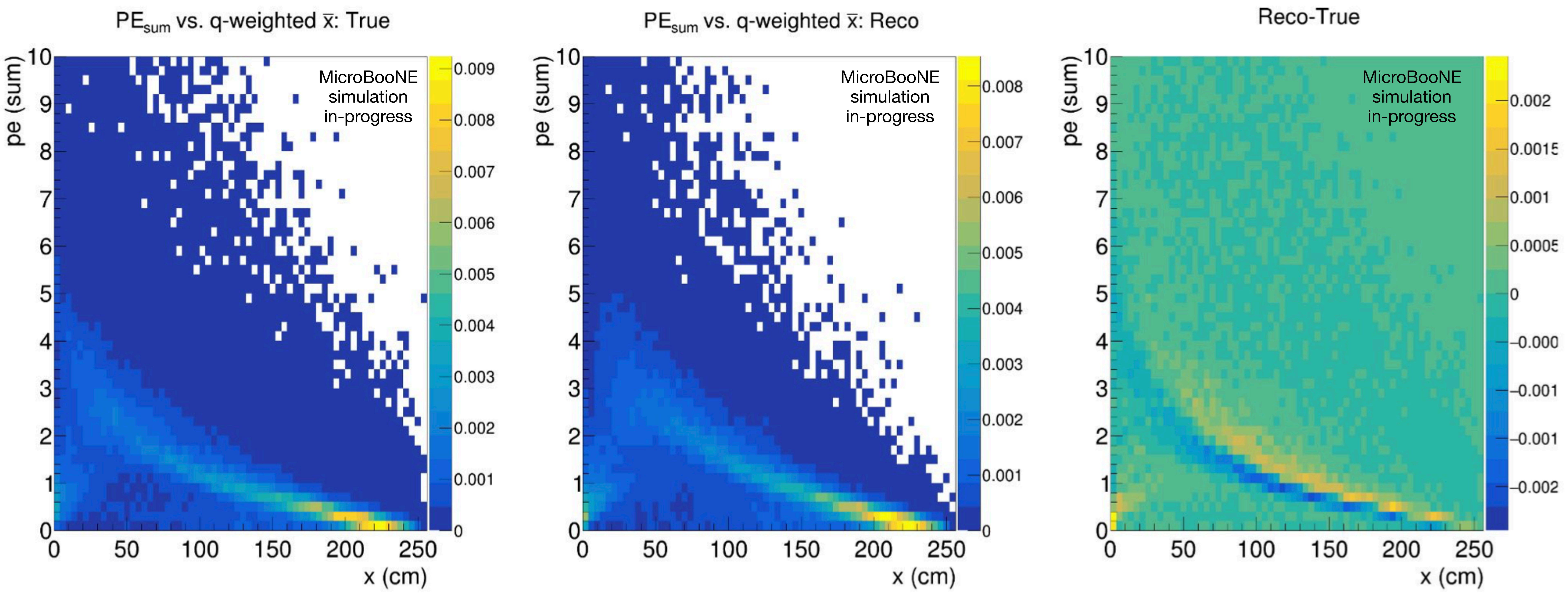
- Plots show the PMT-averaged, voxel-averaged visibility function versus the distance from the anode
- Captures x-dependence, including bimodal distribution near the anode

  follows the trend if you are in front of a pmt cluster, but the visibility drops quickly if you are between the clusters
- Problem for examples near the cathode (circled in red), network seems to be predicting zero
  - The light from here is usually low, so the model is ignoring it
- Note that the reason that the data distribution ends before the cathode is that the position of the charge deposits are not corrected for space charge at this time



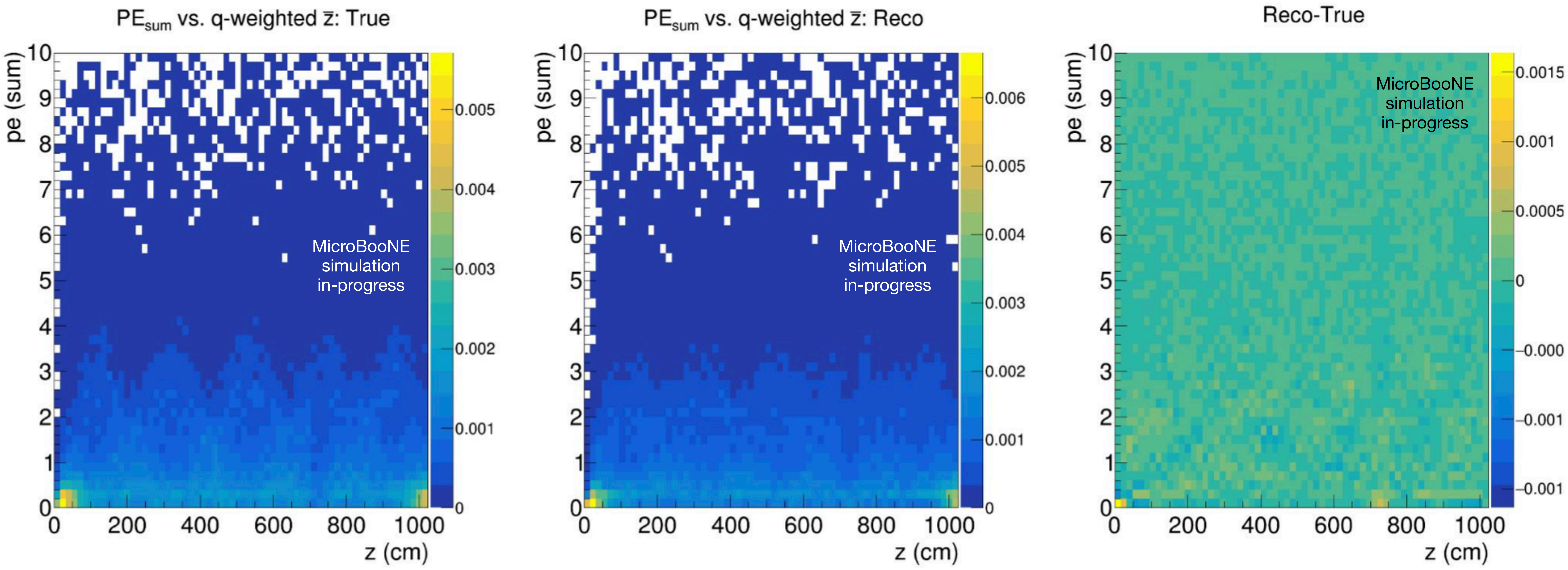
Results after Stage 2

- Data augmentation helps to avoid zero prediction for charge near the cathode
- Also captures the x-dependence overall
- However, is systematically high
  - Possibly from need to provide PE from unobserved charge (outside TPC?)



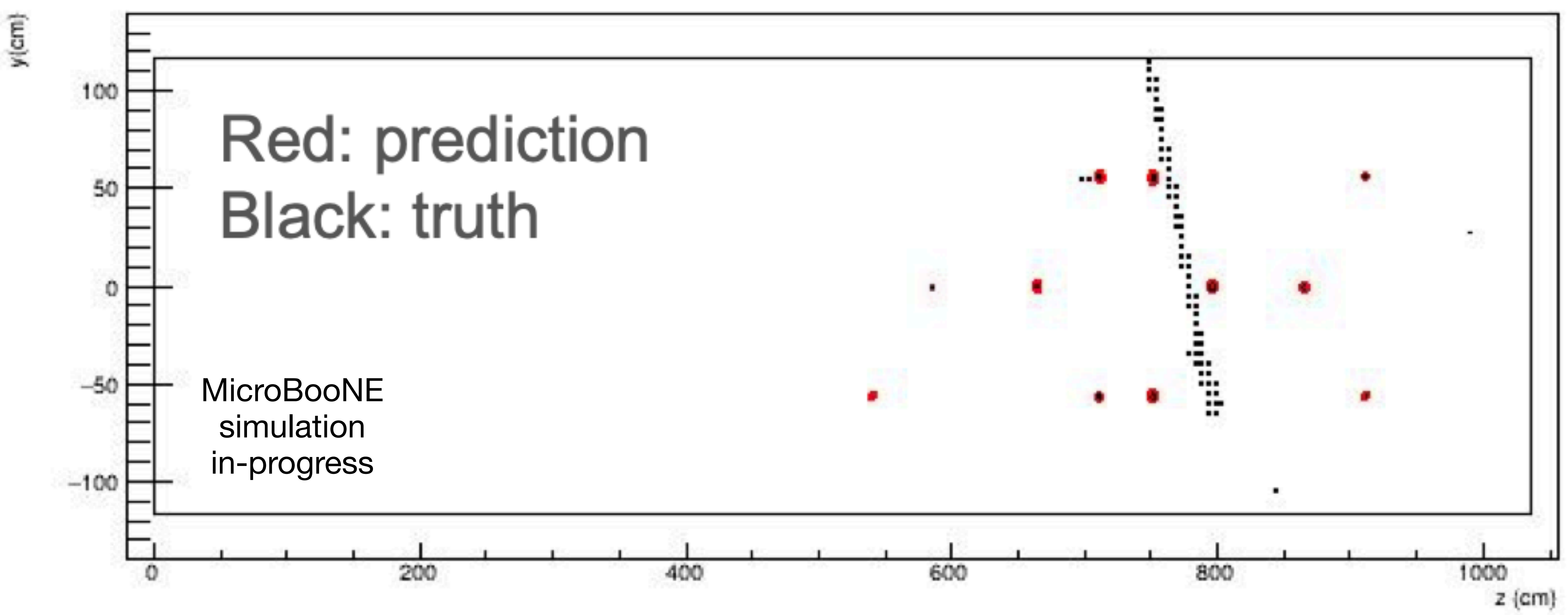
Results after Stage 2

- A different view, along the z-axis
- Visibility drop between groupings of PMTs captures somewhat, but the effect is more smeared out than in the ground truth

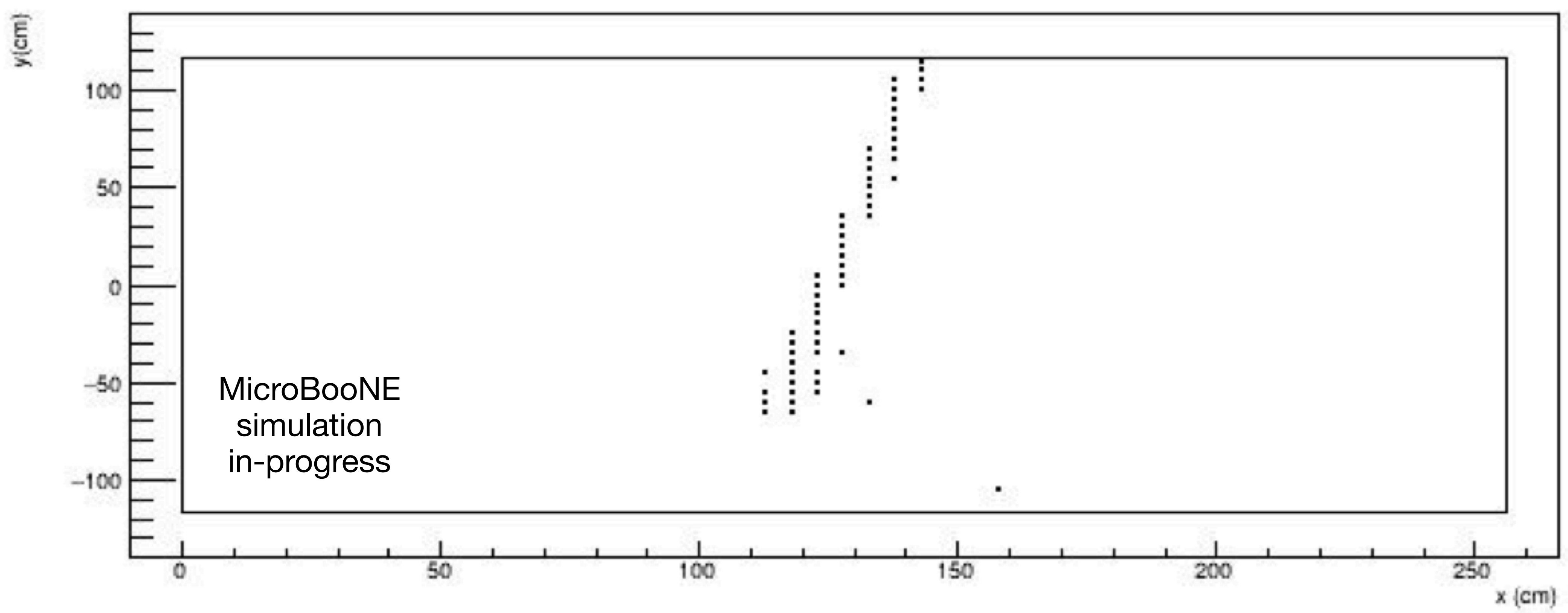


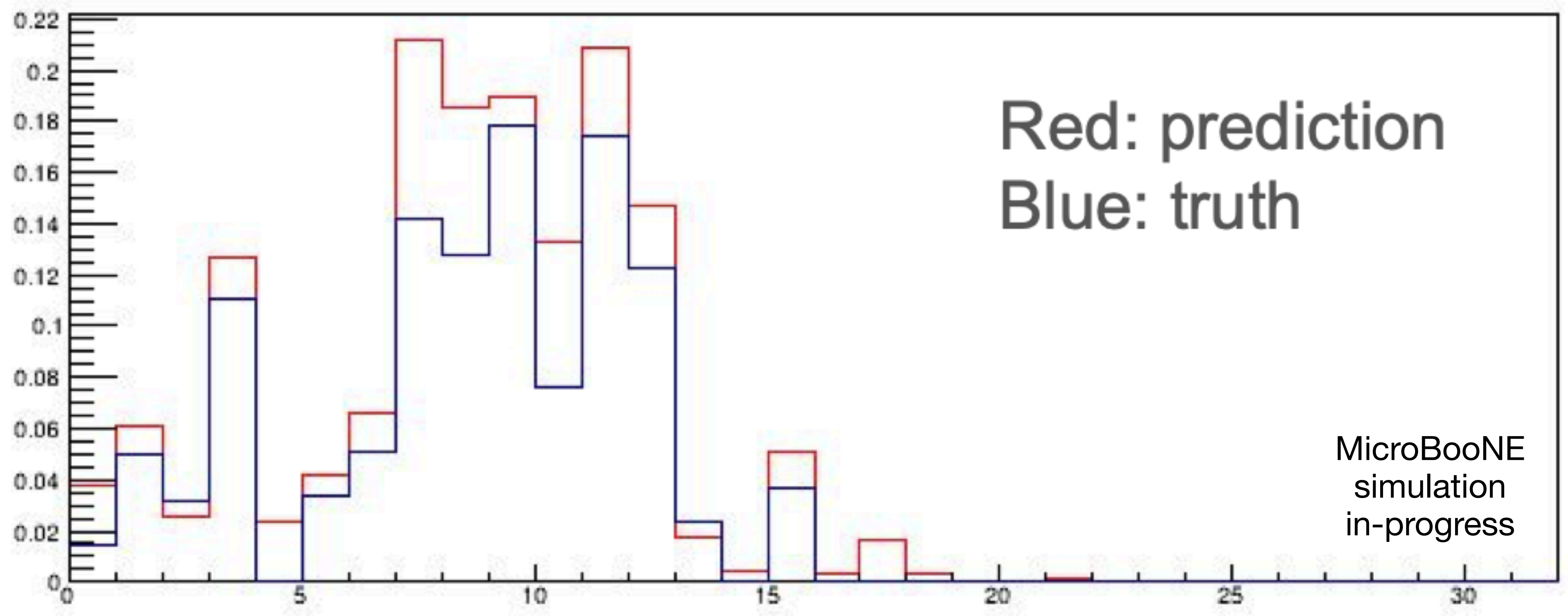
Individual Examples (after stage 2)

Y vs. Z

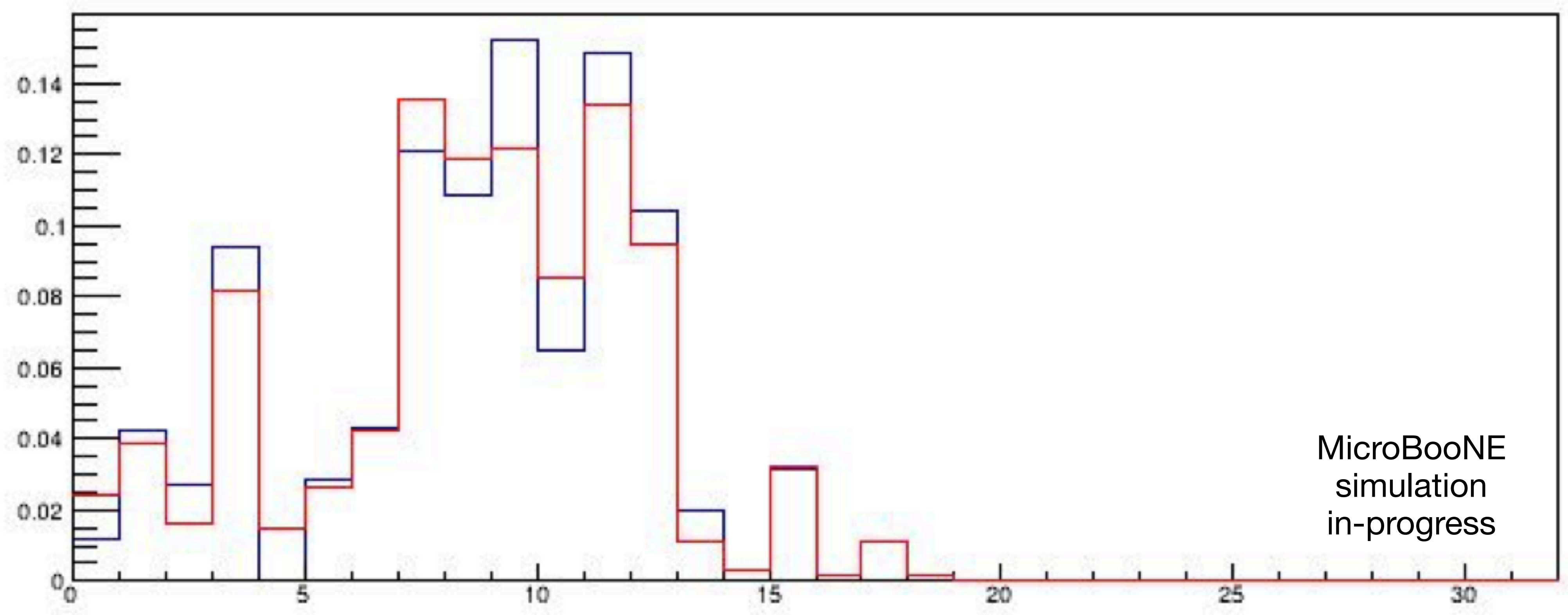


Y vs. X



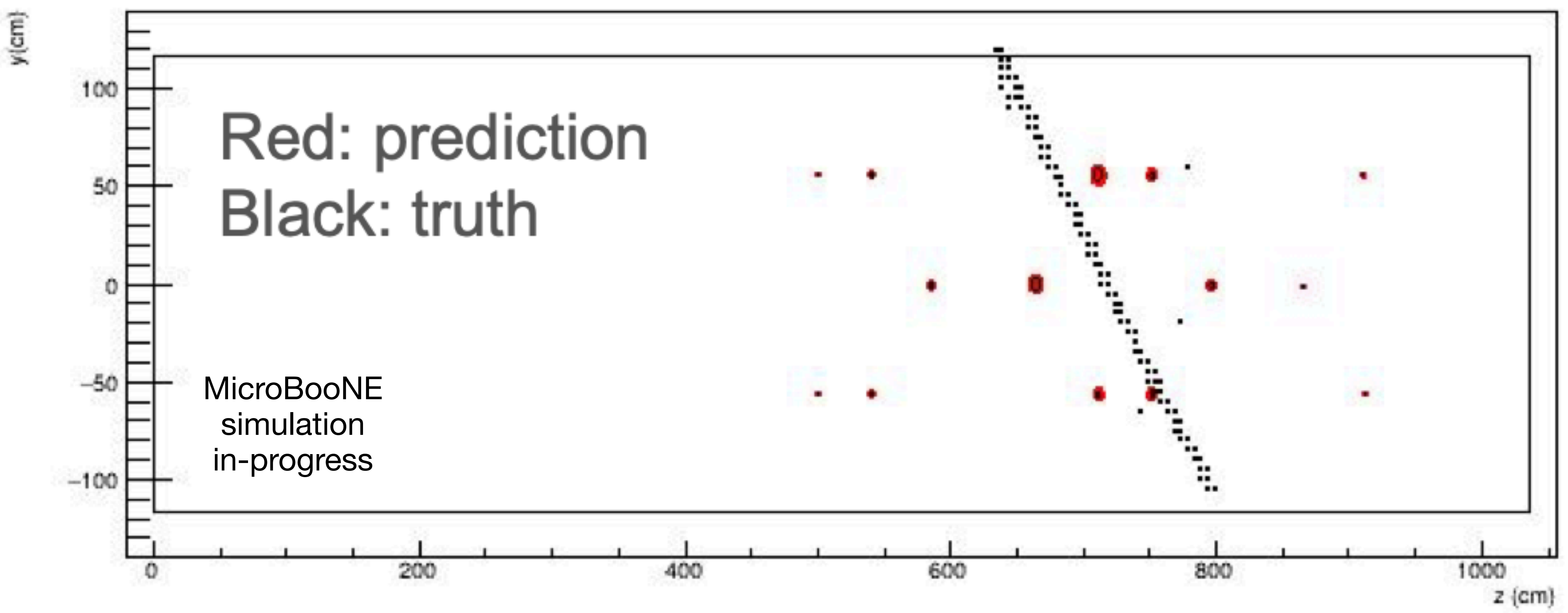


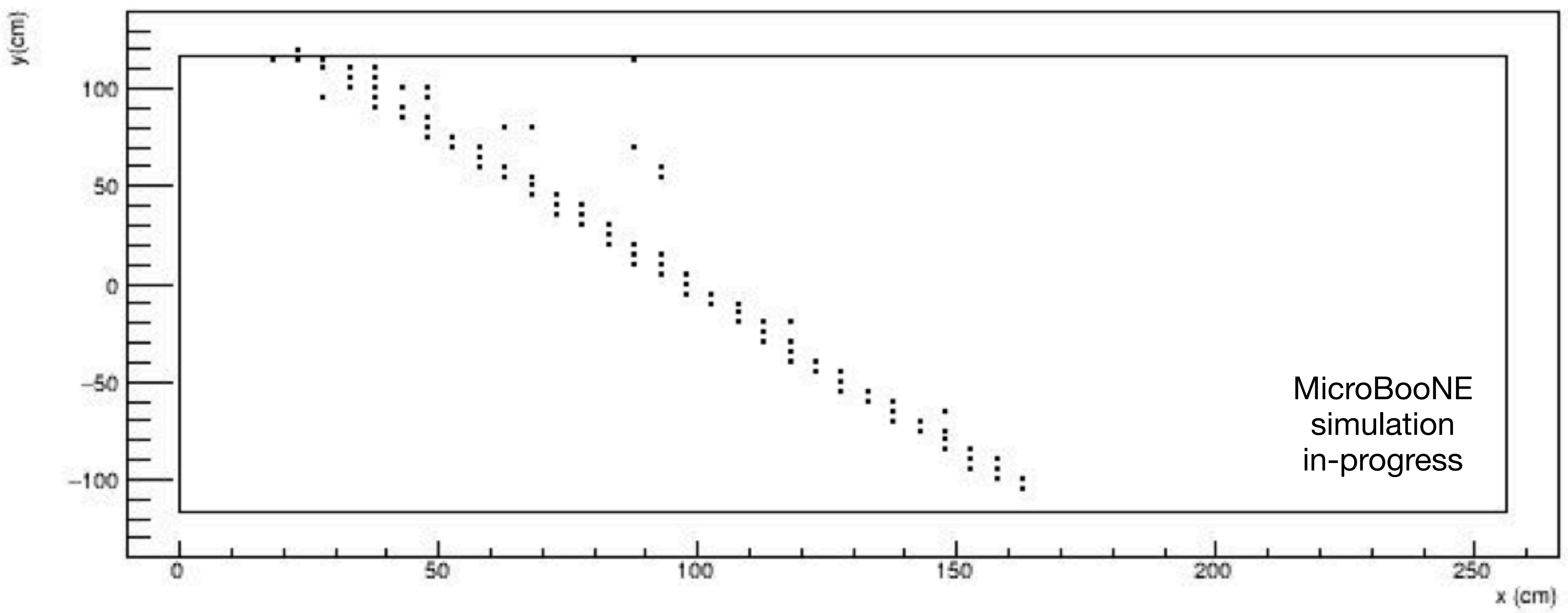
PE vs. PMT Number

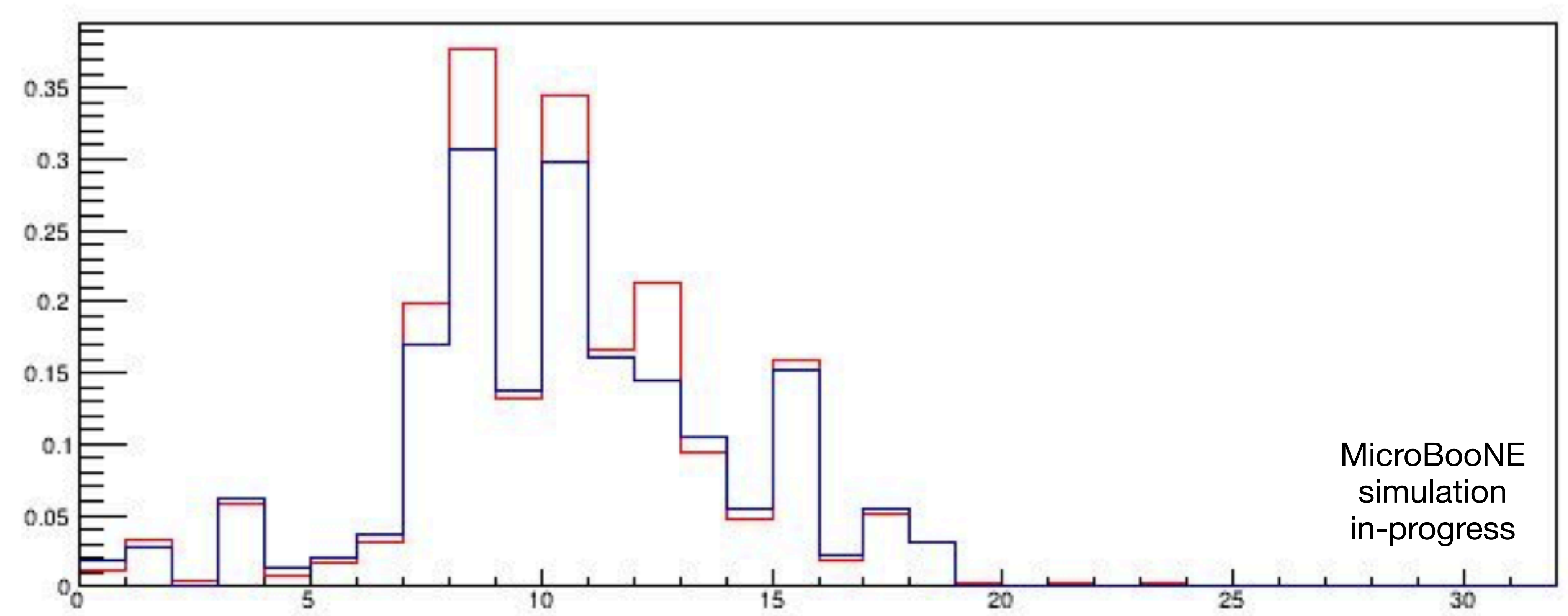


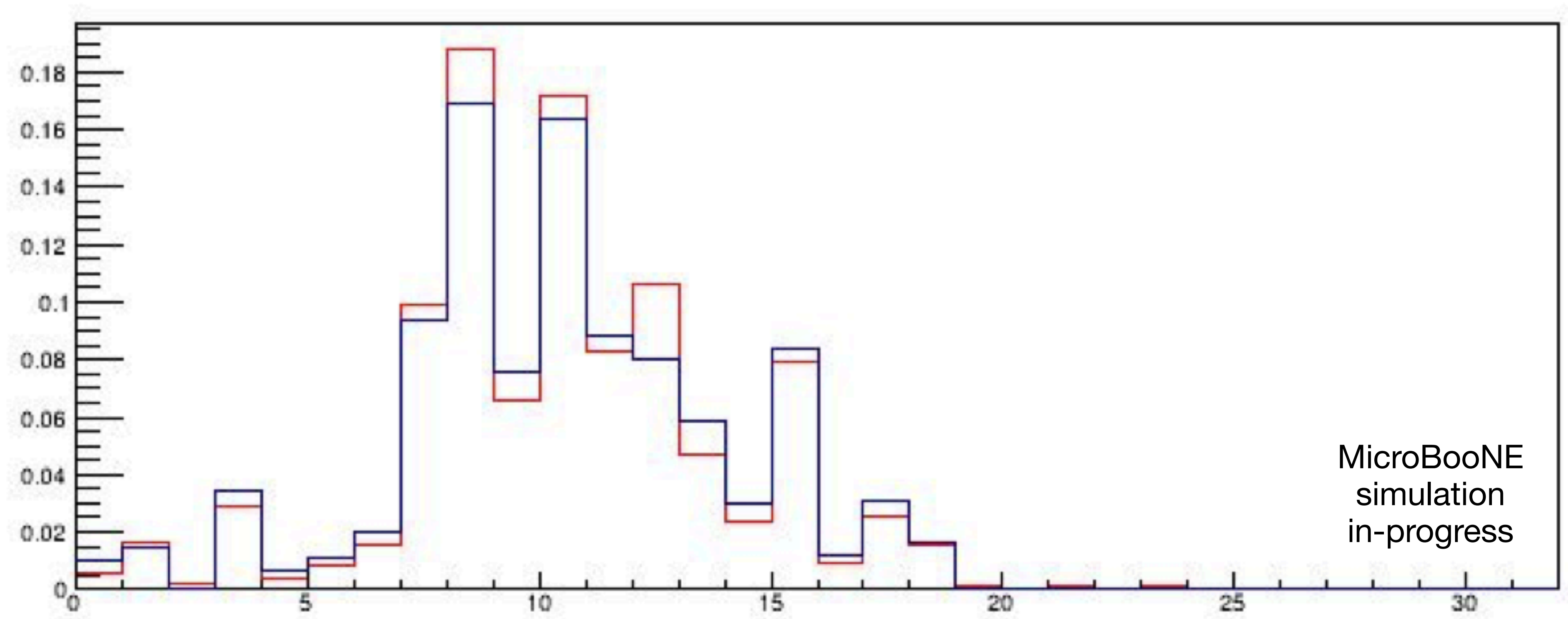
Probability Mass vs. PMT Number
(Comparing Shape)

Individual Examples (after stage 2)

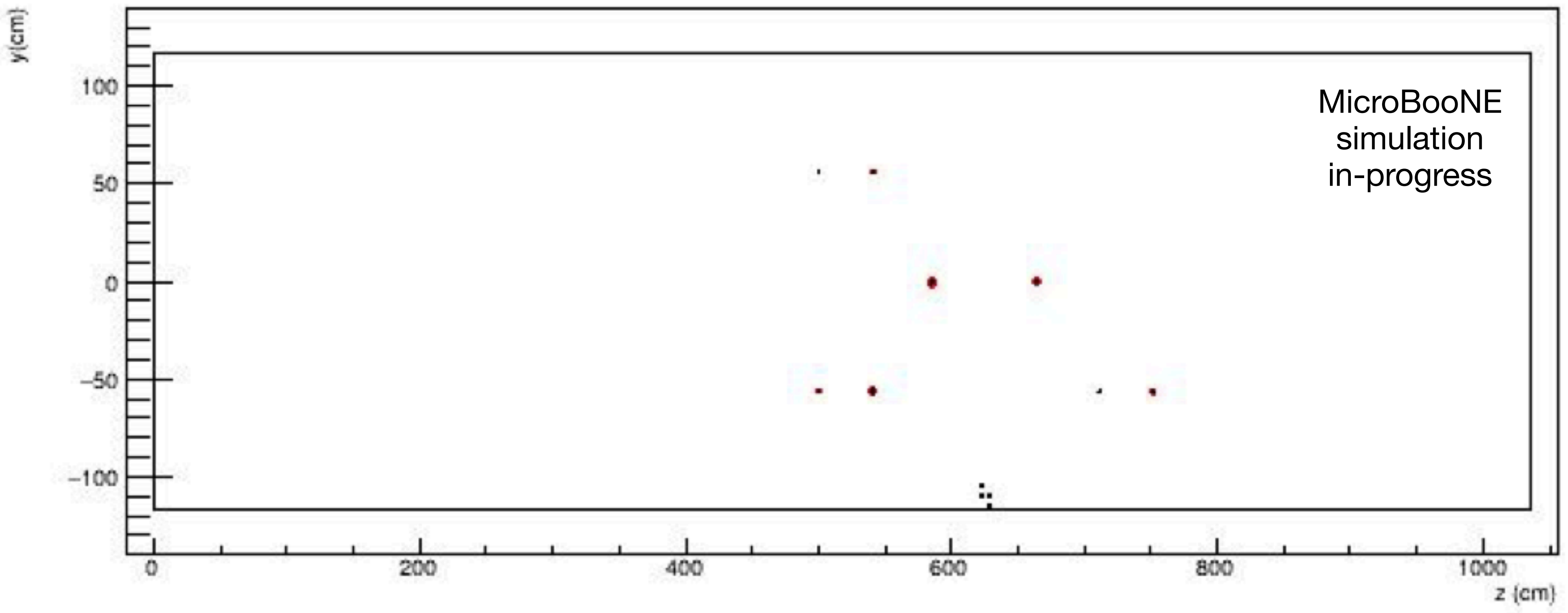


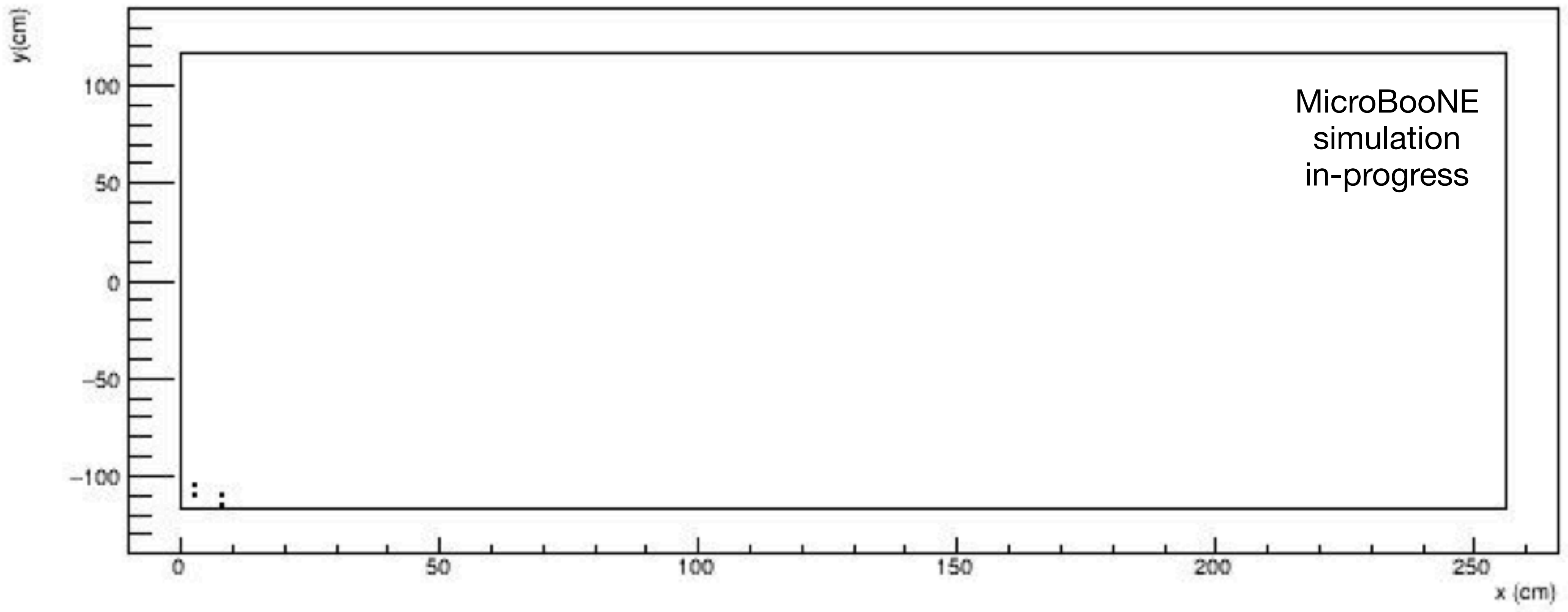


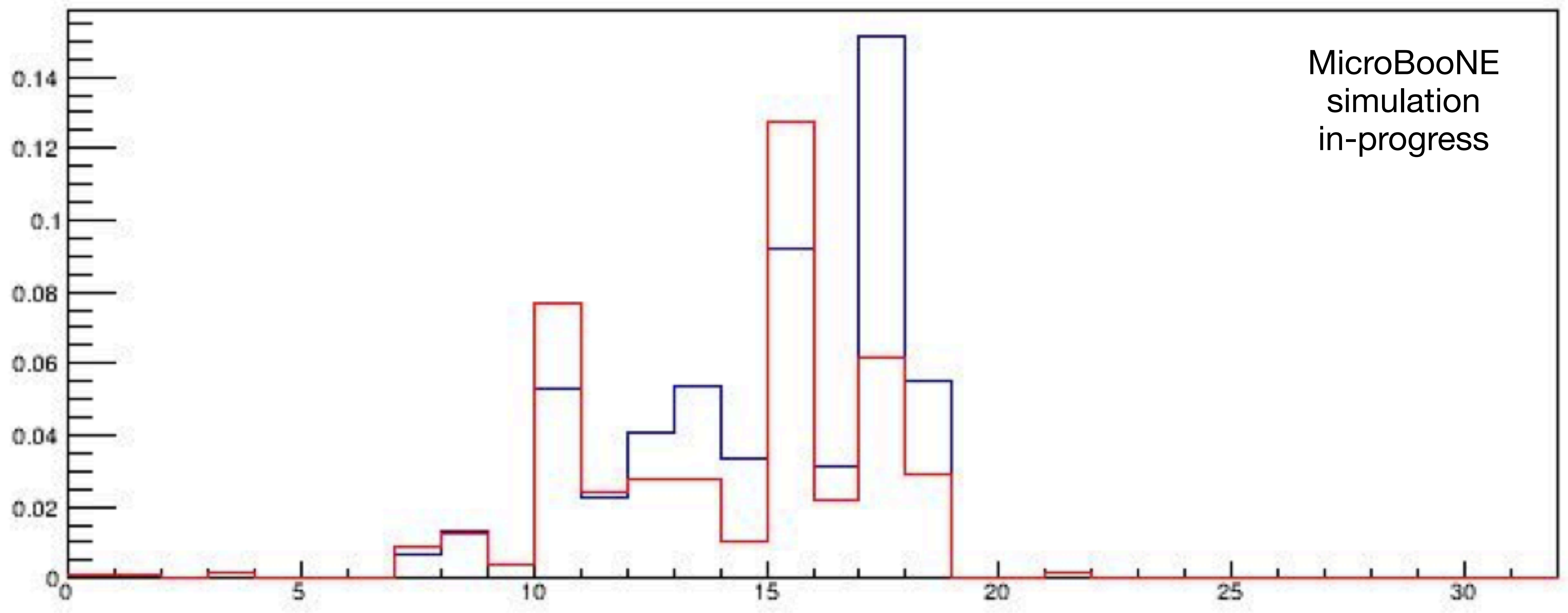


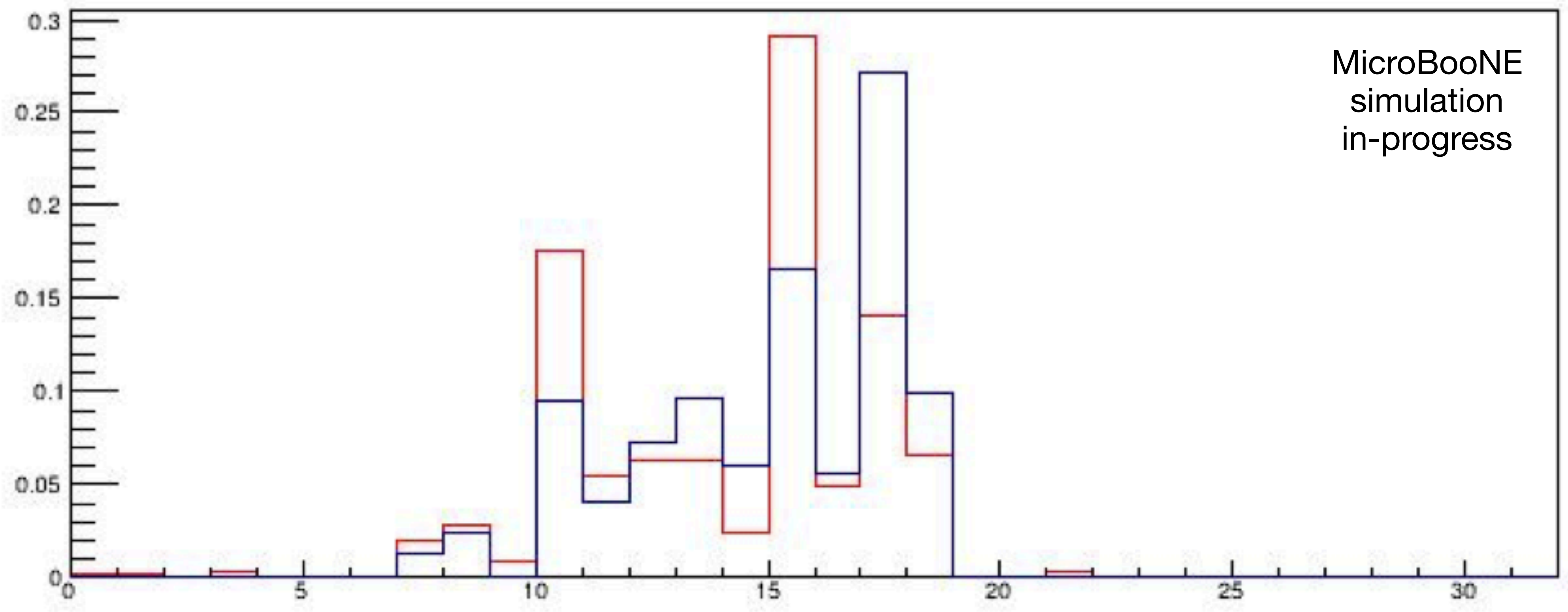


Individual Examples (after stage 2)



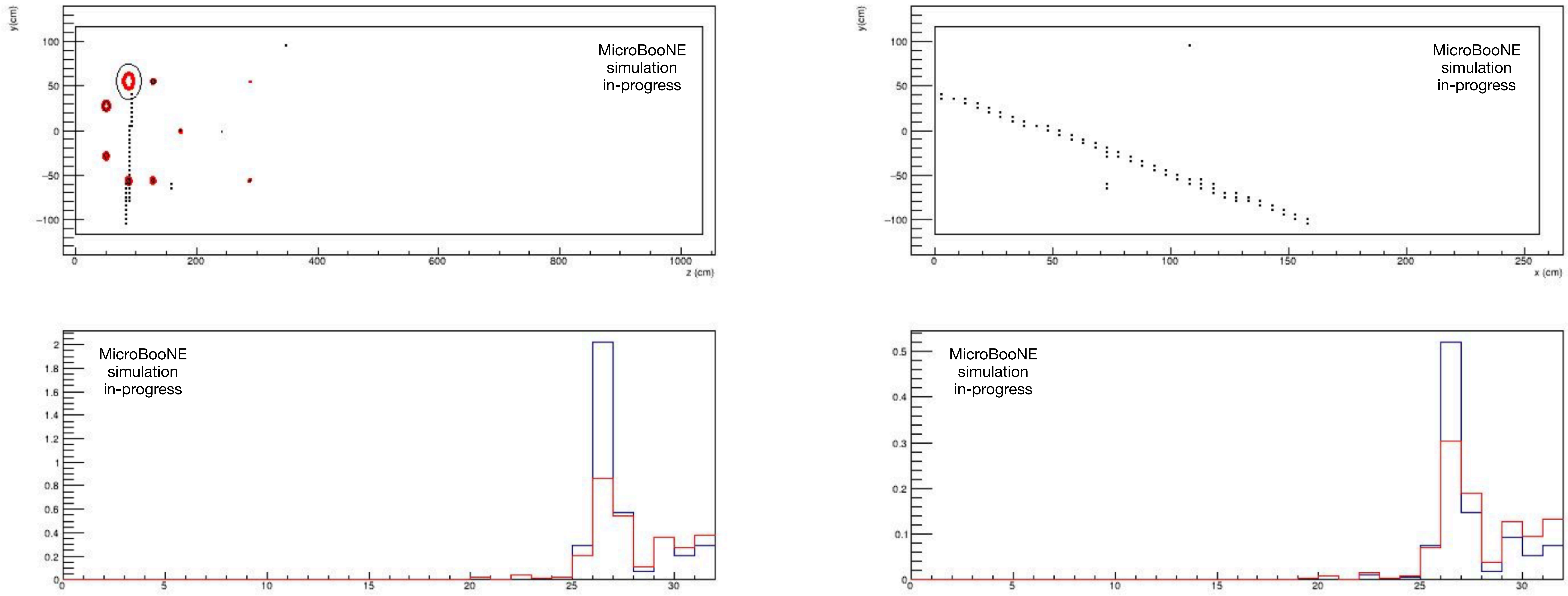




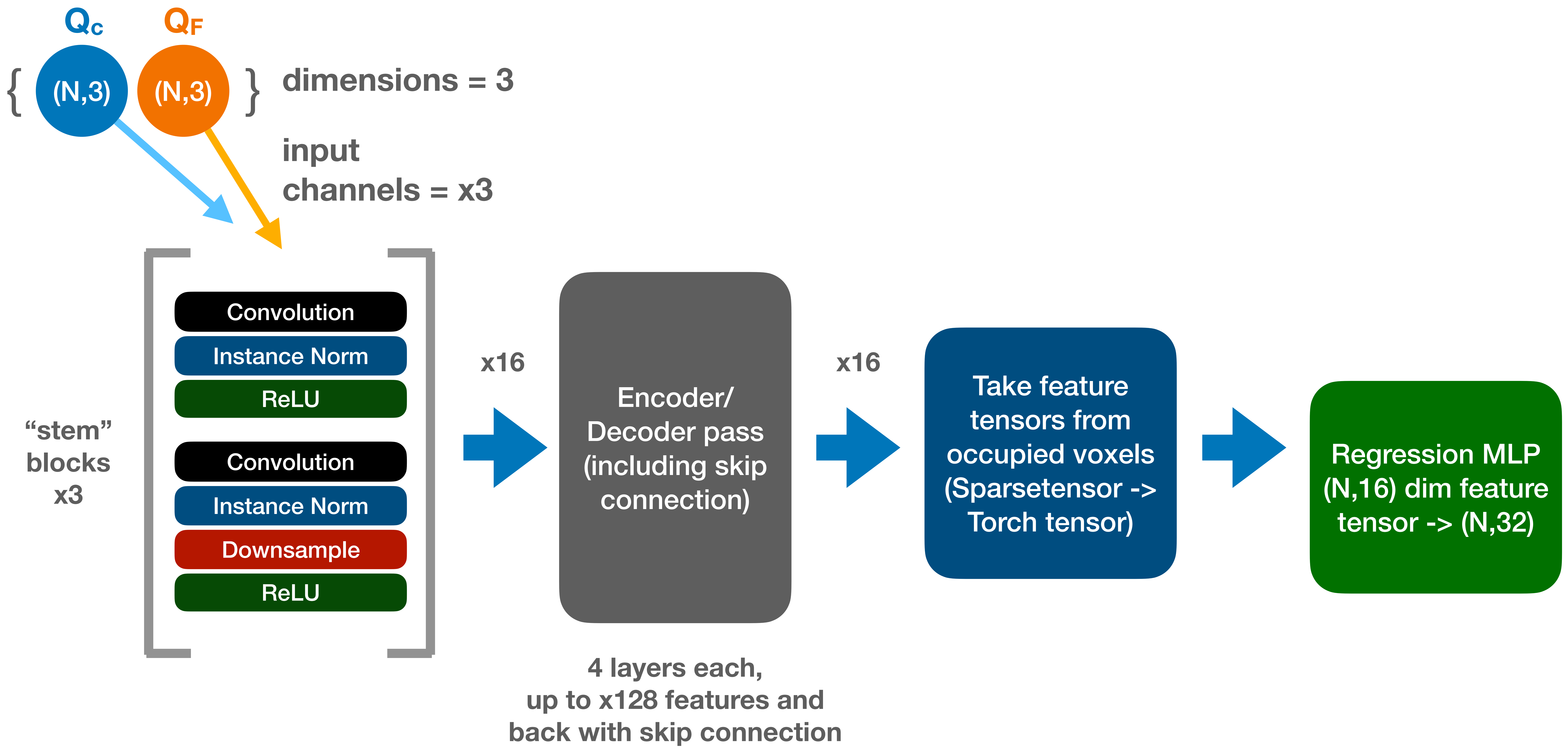


Individual Examples (after stage 2)

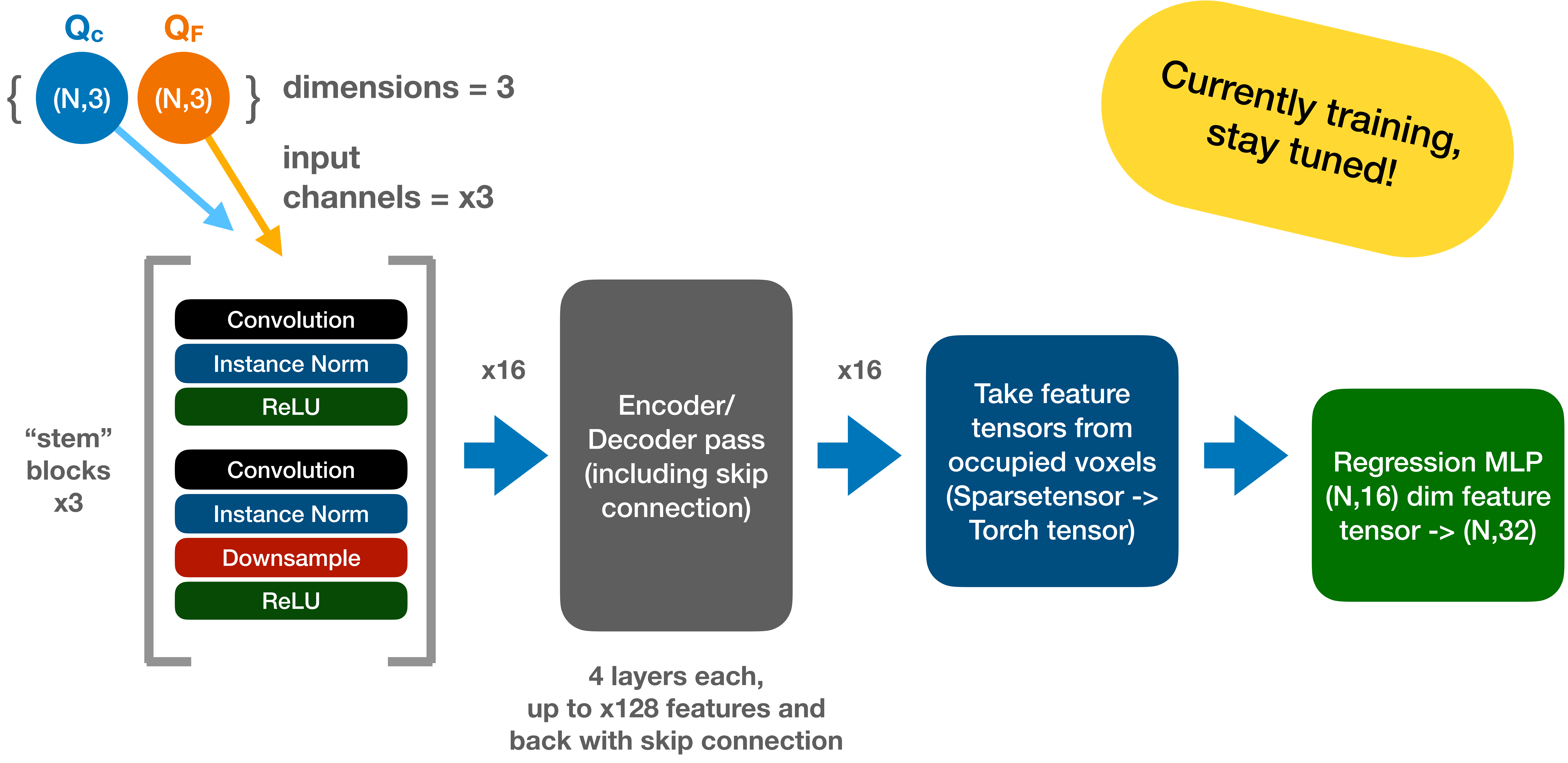
- This is an example where the cosmic track passes near the PMT. The network as it is currently set up cannot account for this: motivates applying CNN on voxels to provide adjustments



CNN Architecture: LArMatch-based UResNet



CNN Architecture: LArMatch-based UResNet



Next Steps

- Results from baseline network show that learning the visibility function is possible with non-point sources
- Can begin to try with data:
  - Collect cosmic muon examples from EXTBNB
  - Will need to use anode/cathode crossing or CRT information for timing
  - Can also use current MC model to bootstrap a dataset by finding events with flashmatch solution that we can assign a high confidence level to for e.g. events with a low number of tracks
- Continue working on CNN
  - Can help with out-of-TPC charge estimate
    - Can use voxel patterns to determine revelant path length outside of TPC
  - Can this address the systematically higher PE prediction?

Backup

Semi-Analytical Model

Semi-Analytical Model:
arxiv.org/abs/2010.00324

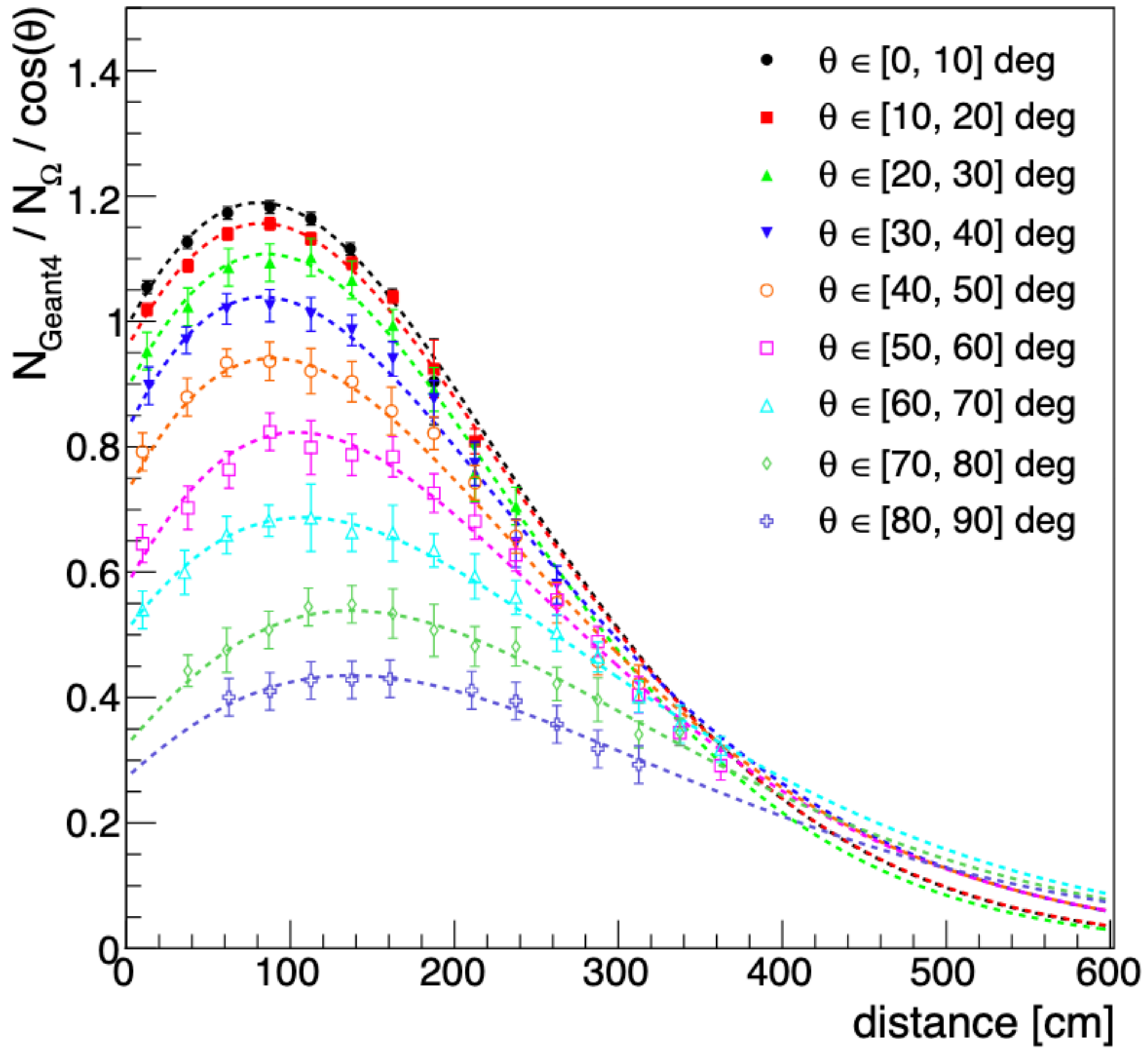
- Steps:

  1. Geometric calculation for the number of photons seen by a photodetector
     - Need to calculate the solid angle subtended by e.g. PMT in infinite detector

$$N_\Omega = e^{-\frac{d}{\lambda_{abs}}} \Delta E \cdot S_\gamma(\mathscr{E}) \frac{\Omega}{4\pi},$$

  2. Corrections based on Rayleigh scattering
     - Compute ratio of geometric calculation and with simulated hits
     - Can describe distribution with Gaisser-Hillas functions

  3. Correction for border effects



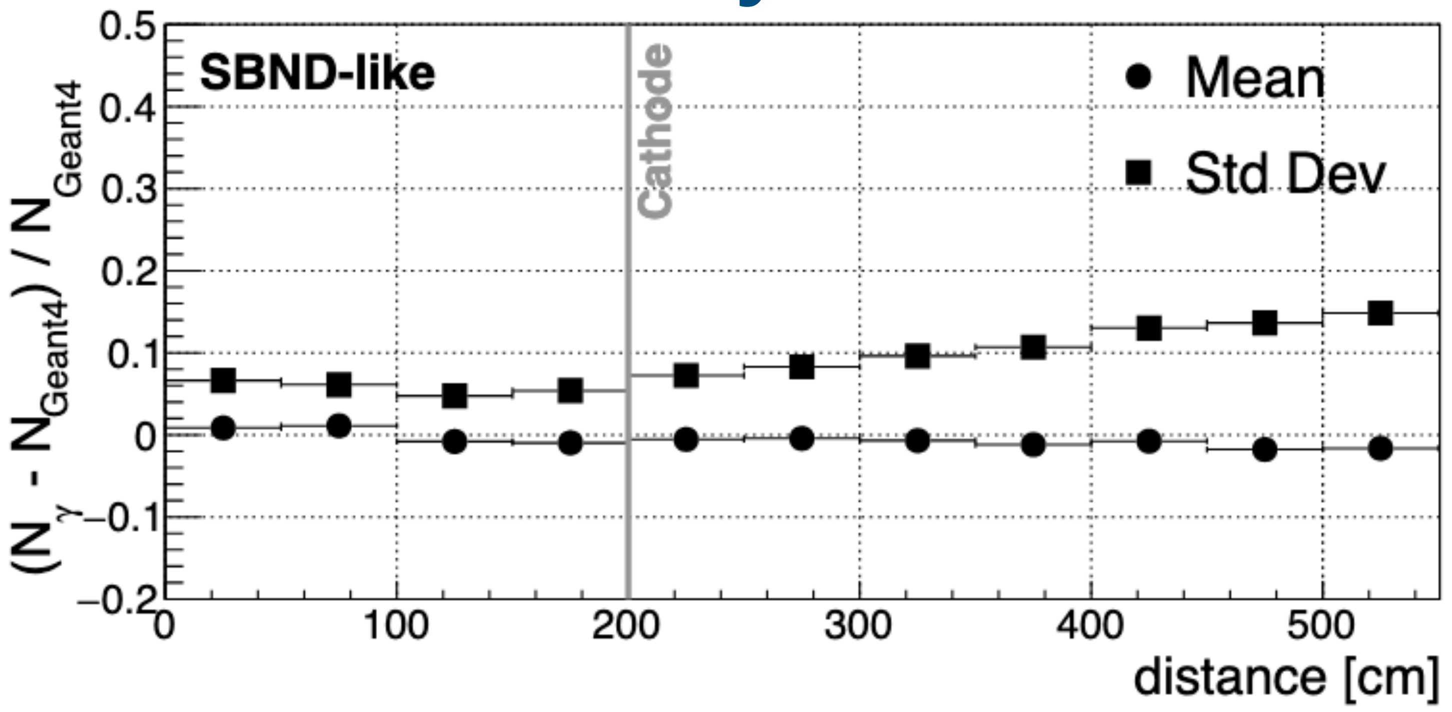
Semi-Analytical Model Performance

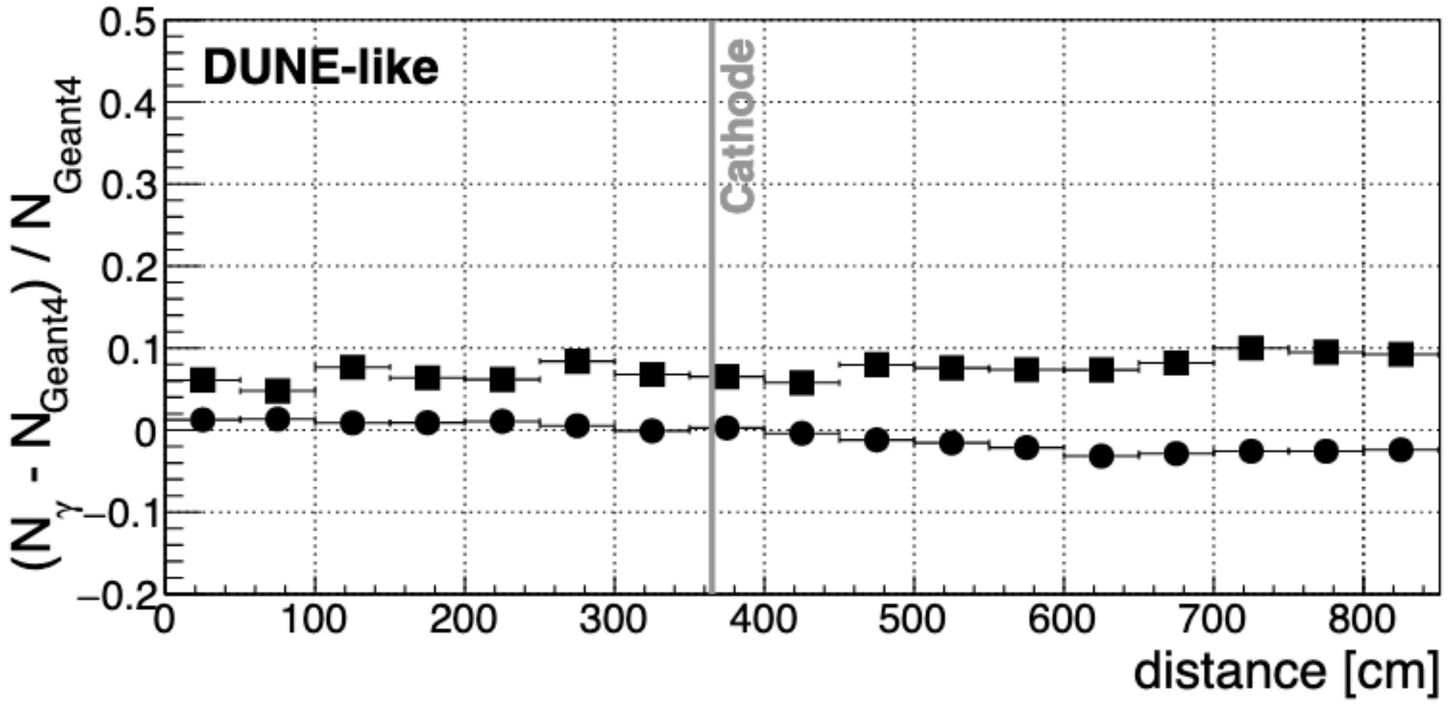
Library	Total Phot.	Phot. per Voxel	Voxel Size [cm^3]	Size[6] [MB]
SBND-like	61.4×10^9	192×10^3	5×5×5	390
DUNE-like	353.5×10^9	158×10^3	5×5×11	826
SBND-like Hi-Res	159.9×10^9	500×10^3	5×5×5	499

- Plots of bias & resolution for both geometries for VUV light
- Generated lookup library with same number of photons + a "high res" ver. for SBND, uniform distribution

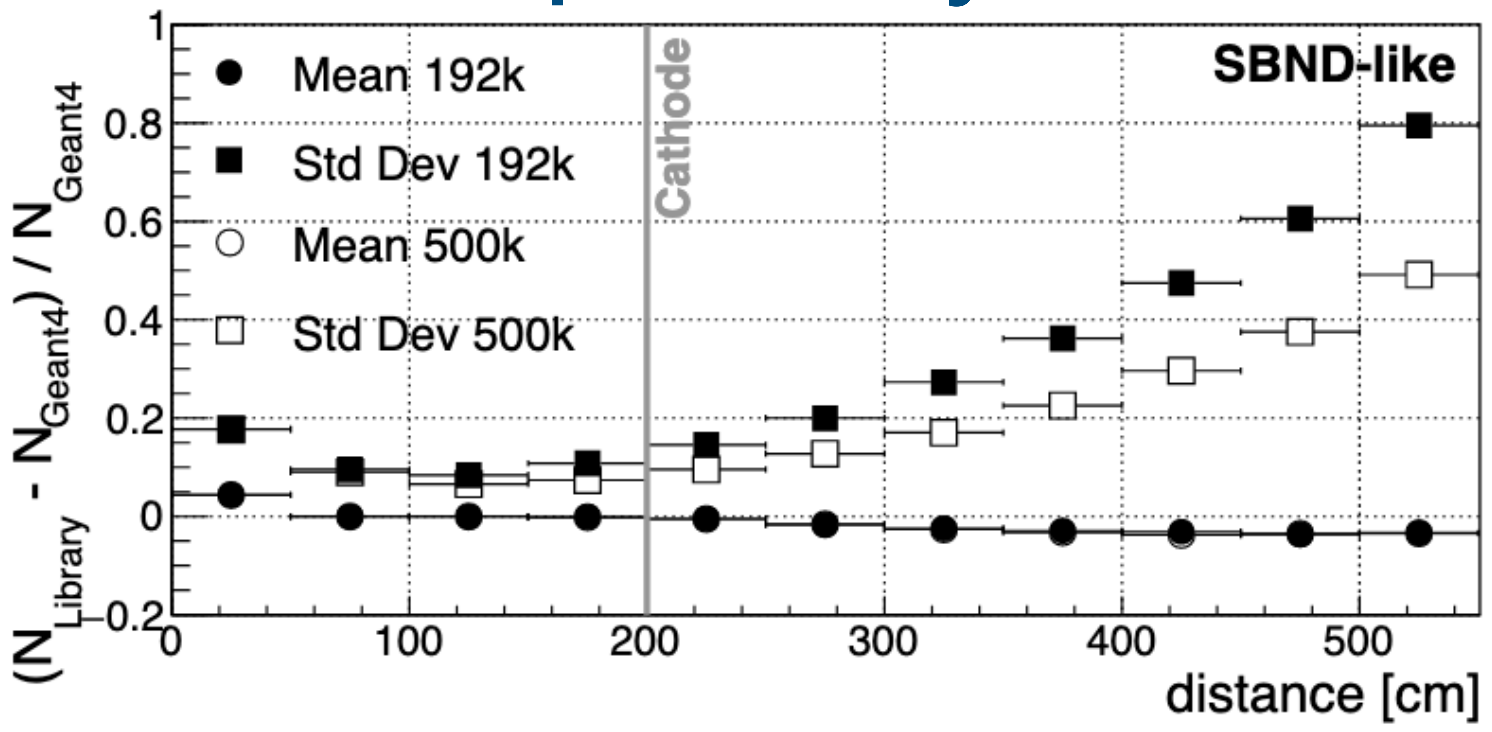
Semi-analytical model

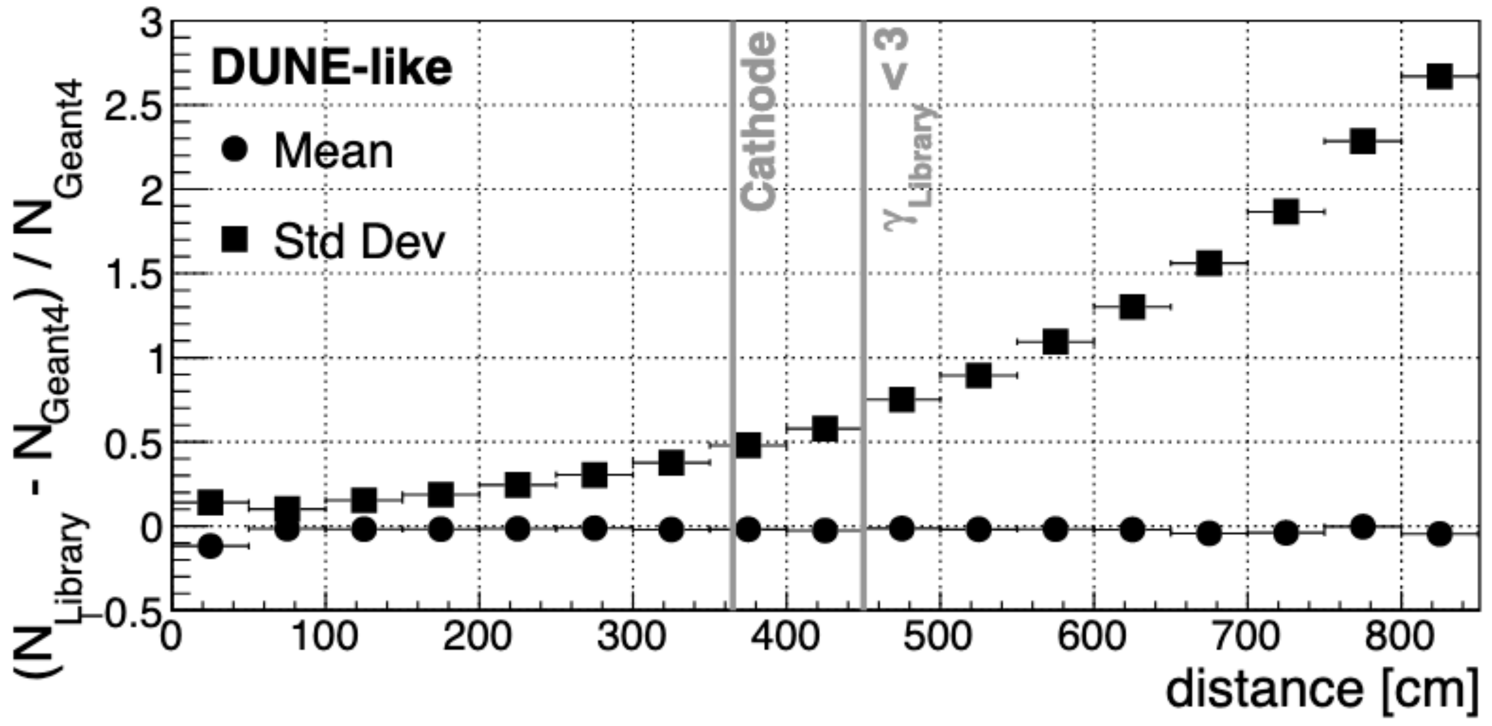
distance between scintillation and the PD, resolution goes up to 15% for farther away





Lookup library method





worse performance at larger distances due to undersampling

worse performance at very low distances due to voxel size (discrete jumps close to PD)

at distances larger than 450 cm, based on samples of less than 3 photons per voxel-PD pair

Semi-Analytical Model: arxiv.org/abs/2010.00324

Implementing Analytical Calculation of Photons on PMT

- Recall this is the first part of the semi-analytical model

$$N_{\Omega} = e^{-\frac{d}{\lambda_{abs}}} \Delta E \cdot S_{\gamma}(\mathscr{E}) \frac{\Omega}{4\pi},$$

solid angle subtended by photodetector (disk in uboone) in an infinite detector

idealized case with no reflections and not considering Rayleigh scattering

loss due to absorption effects

a given energy deposition

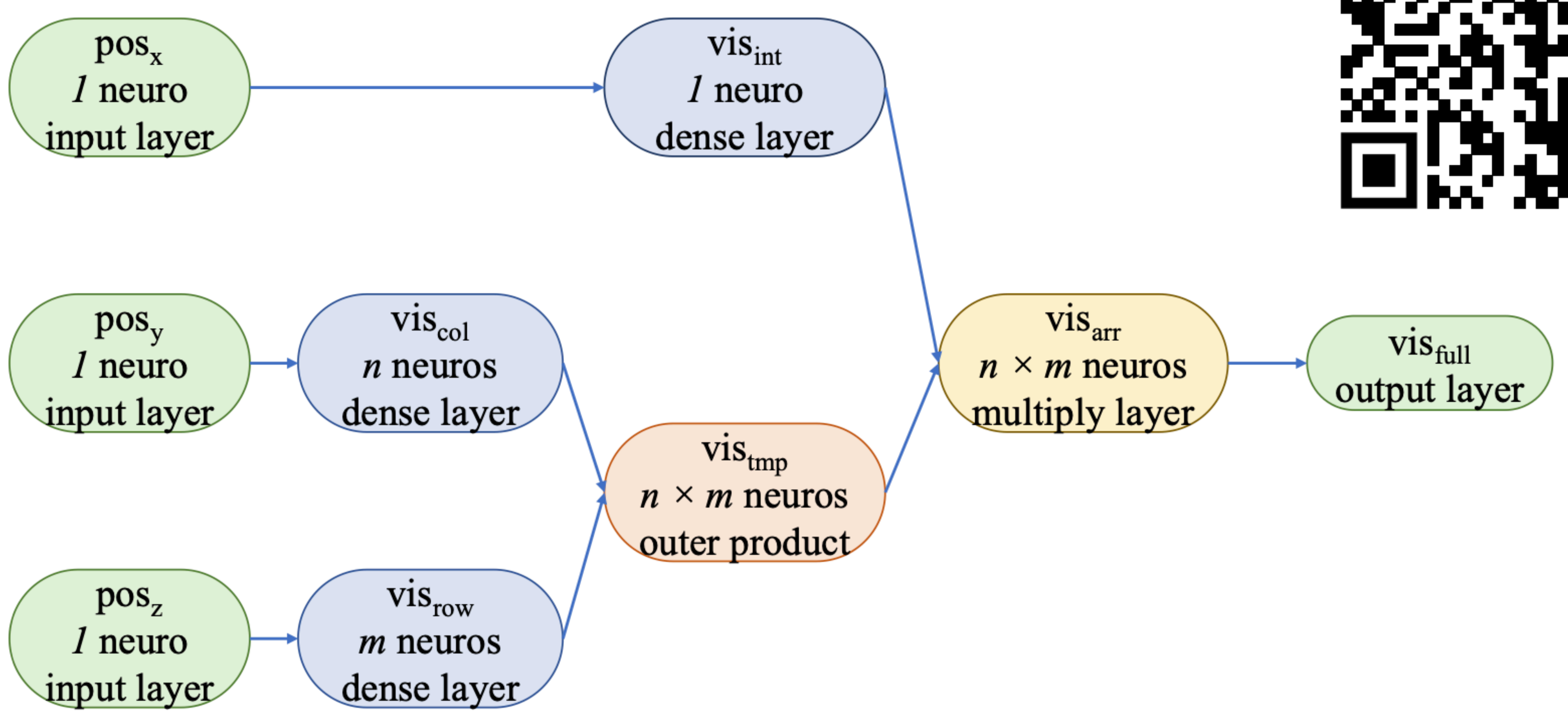
scintillation yield of LAr for a given electric field

- To calculate solid angle, needed to compute elliptical integrals for each (voxel, pmt) pair
  - I used scipy and mpmath in python, which don't have Pytorch equivalents for running on GPU
  - Explored implementation in Cuda for running on GPU
- Decided to calculate upfront for voxelized detector, takes ~23 min

1D GENN

1D GENN
arxiv.org/abs/2109.07277

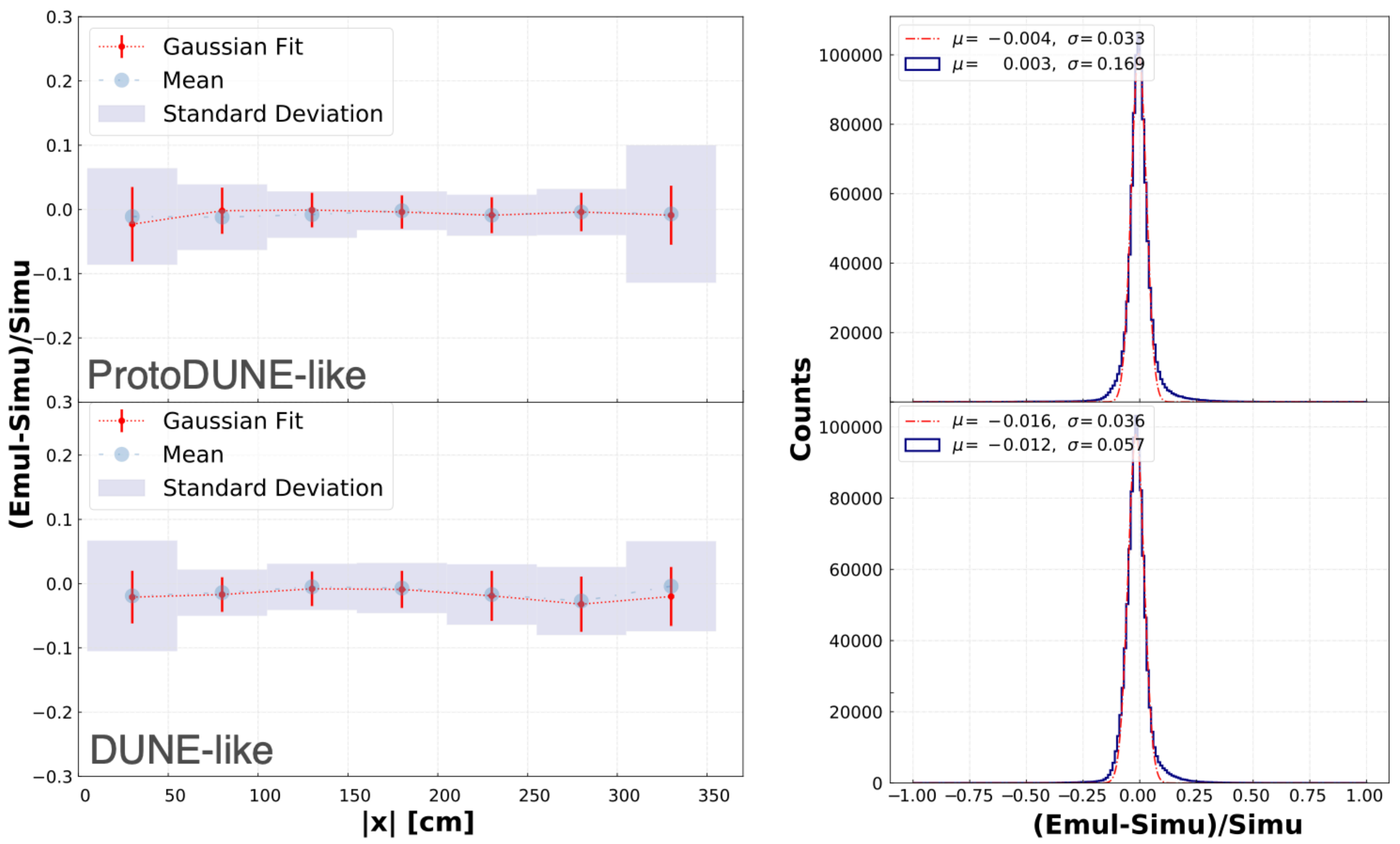
- One goal is to predict photon distribution probabilities at high speed using a CPU
- For this reason, use a lightweight generative architecture
  - Use an OuterProduct layer rather than transpose convolutional (Conv2DTranspose) or upsampling (UpSampling2D) layers



- 1D vector is represents the "image" of hit pattern on a PD obtained for each scintillation vertex
  - Used as truth from photon library, and output of GENN
- Does not compared results within the GAN framework, but rather uses the following loss function:

$$D_{\mathrm{vKL}}(P||Q) = \left| \sum_x \Big(P(x) - Q(x)\Big) \log \frac{P(x)}{Q(x)} \right|,$$

P(x) is 1D vector from GENN,
Q(x) is "true" 1D vector from simulation



SIREN

Voxel-wise loss for training on photon library:

$$\mathcal{L}_2 = \sum_{\mathbf{x}_i \in \mathcal{D}} \sum_{j=1}^{N_d} w_{ij} [\tilde{\Psi}_j(\mathbf{x}_i) - \tilde{v}_{ij}]^2,$$

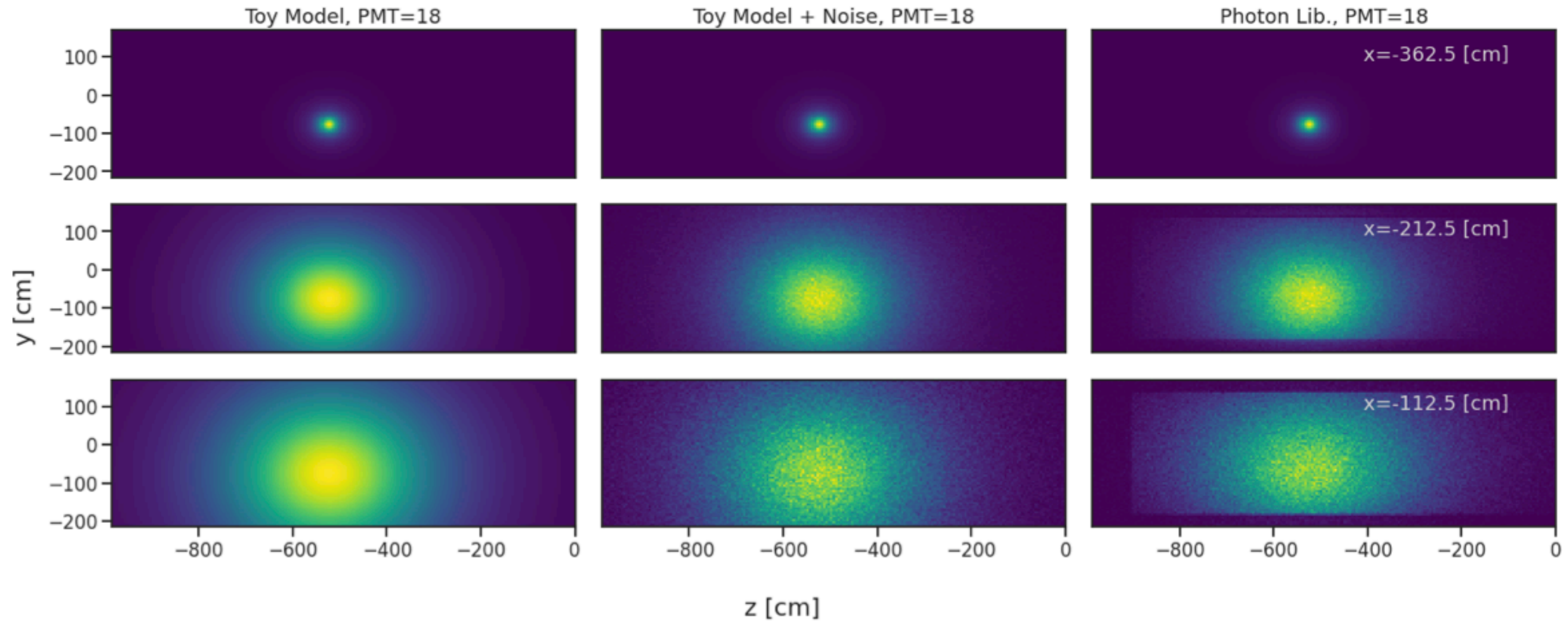
Visibility for voxel-PMT pair

SIREN
arxiv.org/abs/2211.01505

- SIREN: implicit neural representation with periodic sinusoidal activation functions
  - Uses simple MLP with periodic sine function activations
- Parametrizes signals (XYZ coordinates) as continuous functions via neural networks, train to map to average photon yield at a PD
  - Reproduces an acceptance map with higher accuracy than simulated photon library approach
  - More scalable (time and computationally) than original photon library, also differentiable and able to be calibrated
- Performs voxel-wise training on original photon library visibility
- Can be used for flash-matching and calibrated to data with track-wise loss function: minimize negative log Poisson likelihood:

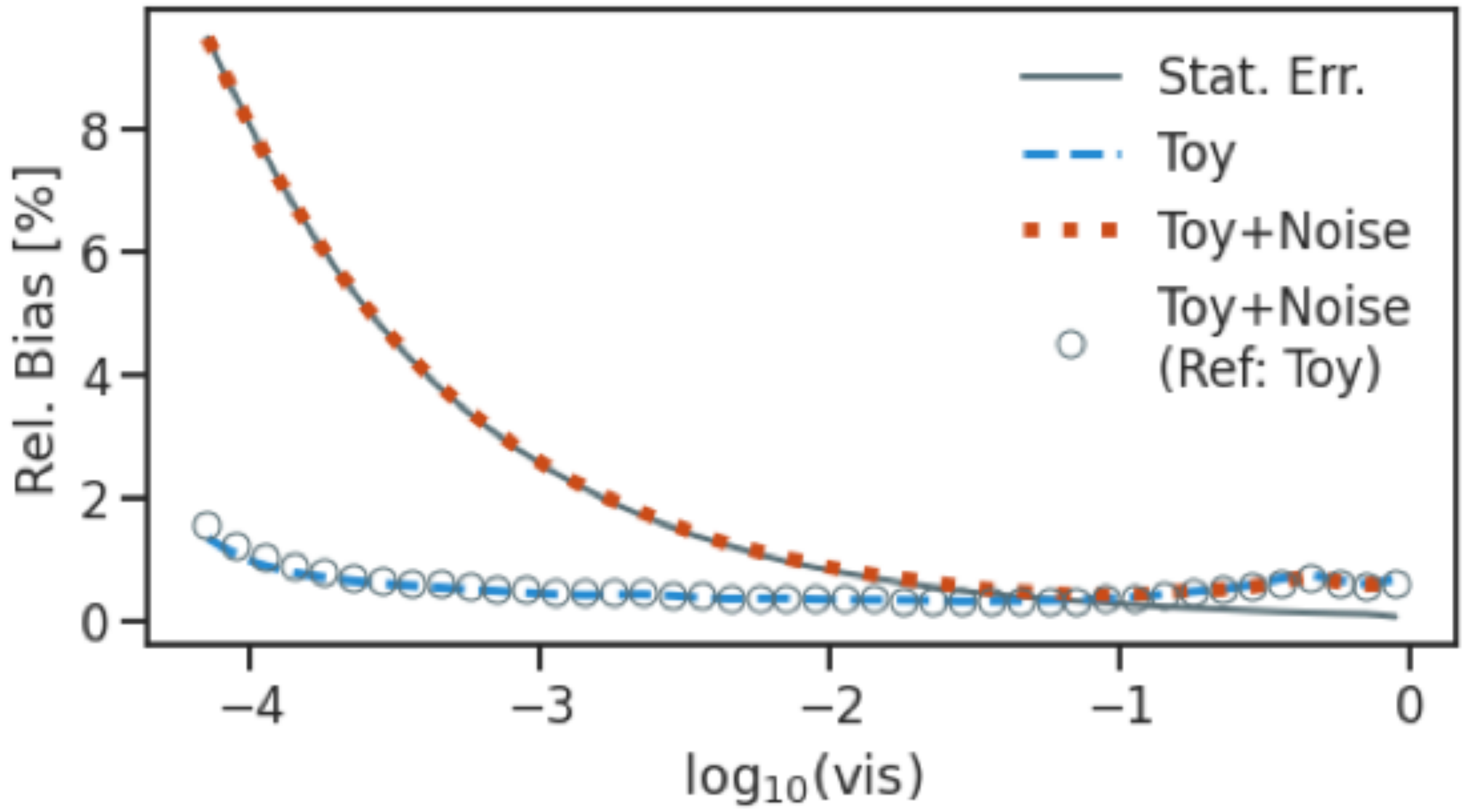
$$\mathcal{L}_{\text{track}} = \prod_{j=1}^{N_d} \text{Pois}(n_j | \lambda_j),$$

Add noise to approximate voxels in original photon library



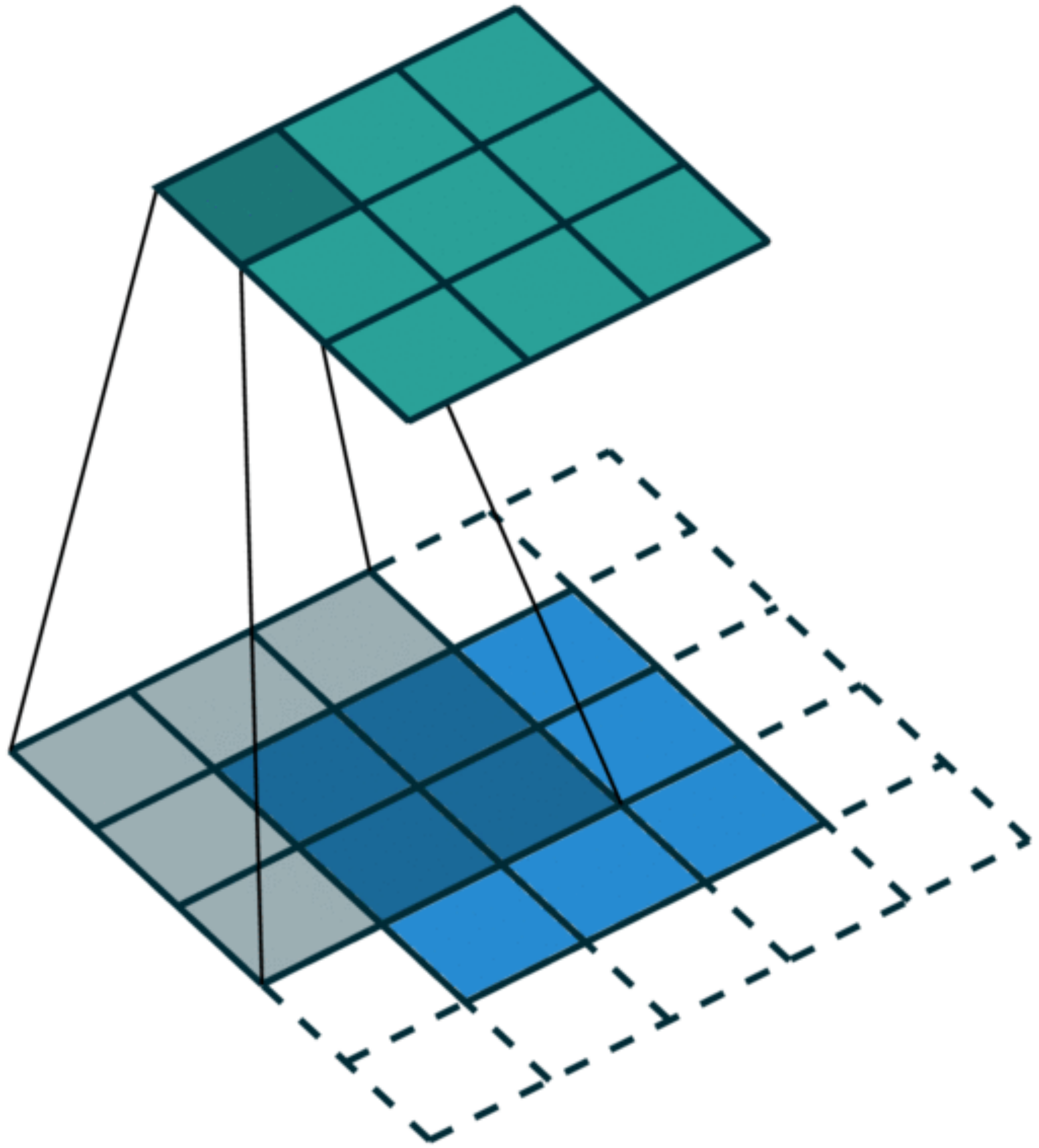
Analytic "toy" representation of photon library:

$$\mathbf{\Psi}^{\text{toy}}(\mathbf{x}) = \max(ae^{-k\mathbf{r}(\mathbf{x})}/\mathbf{r}(\mathbf{x})^2, 1),$$

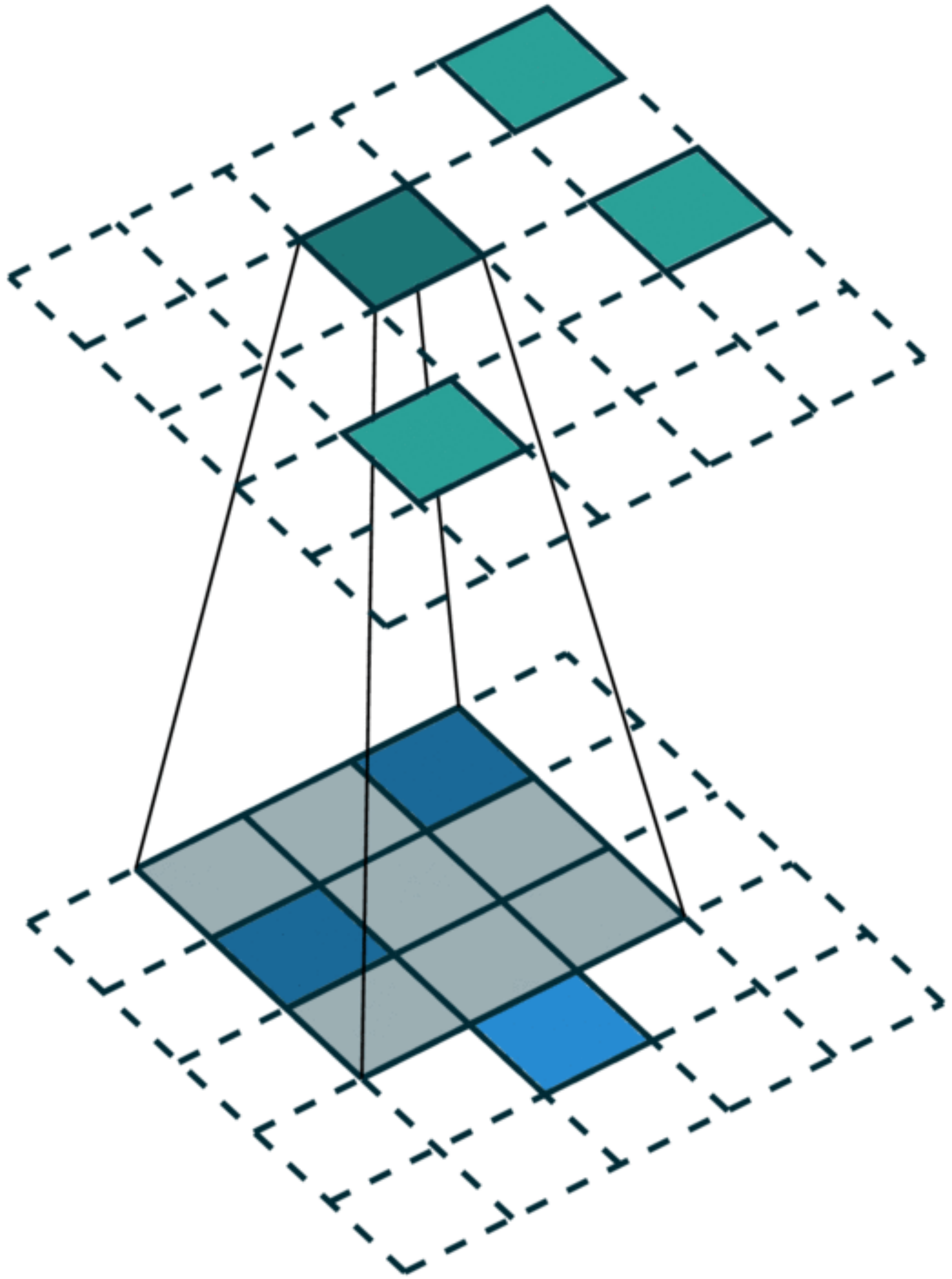


Sparse Tensor Networks

Order of a convolution on sparse tensor is not sequential



Dense Tensor



Sparse Tensor

Note*: We have sparse submanifold convolutions

Submanifold Convolutions

- Convolution output is counted when kernel center covers an input site
  - Better suited for irregular sparse data
- Submanifold convolutions help take care of the "submanifold expansion problem"

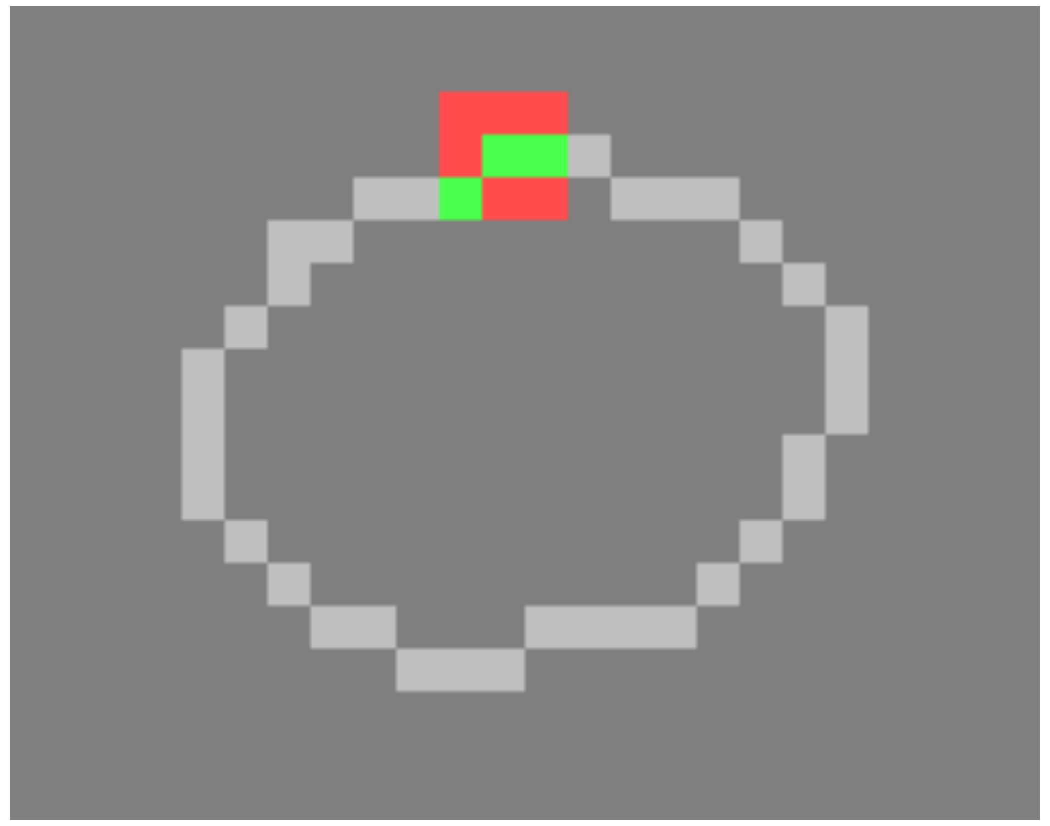
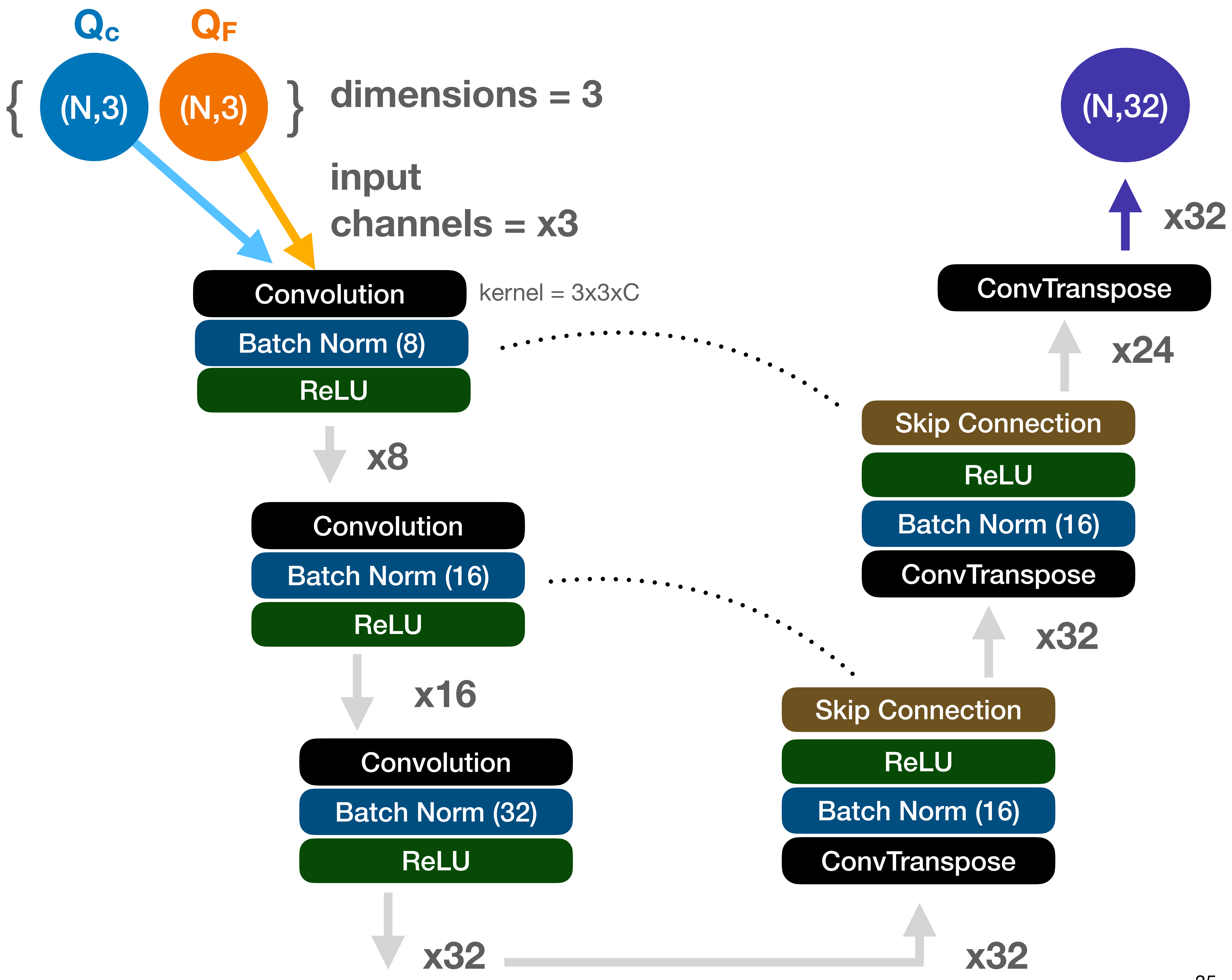


Figure 1: Example of "submanifold" dilation. **Left:** Original curve. **Middle:** Result of applying a regular 3×3 convolution with weights $1/9$. **Right:** Result of applying the same convolution again. The example shows that regular convolutions substantially reduce the sparsity of the feature maps.

Figure 1: Submanifold expansion [Source: https://arxiv.org/abs/1706.01307]

CNN Network Architecture: U-Net

- Started with NVIDIA MinkowskiEngine's default U-Net
  - Library for sparse tensors
- U-Net: a CNN with an encoding and decoding portion
- Input and output are "same size"
  - Here, we have N voxels as input with 3 features (ADC per wire plane)
  - Output is N voxels with 32 features at each voxel
    - One "fudge factor" calculated per PMT
- Skip connections via concatenation
  - Concatenate sparse tensors along feature dimension; this uses info from previous feature maps to e.g. preserve spatial info



CNN Network Architecture: LArMatch UResNet

- Also a U-Net that uses MinkowskiEngine, but has residual layers