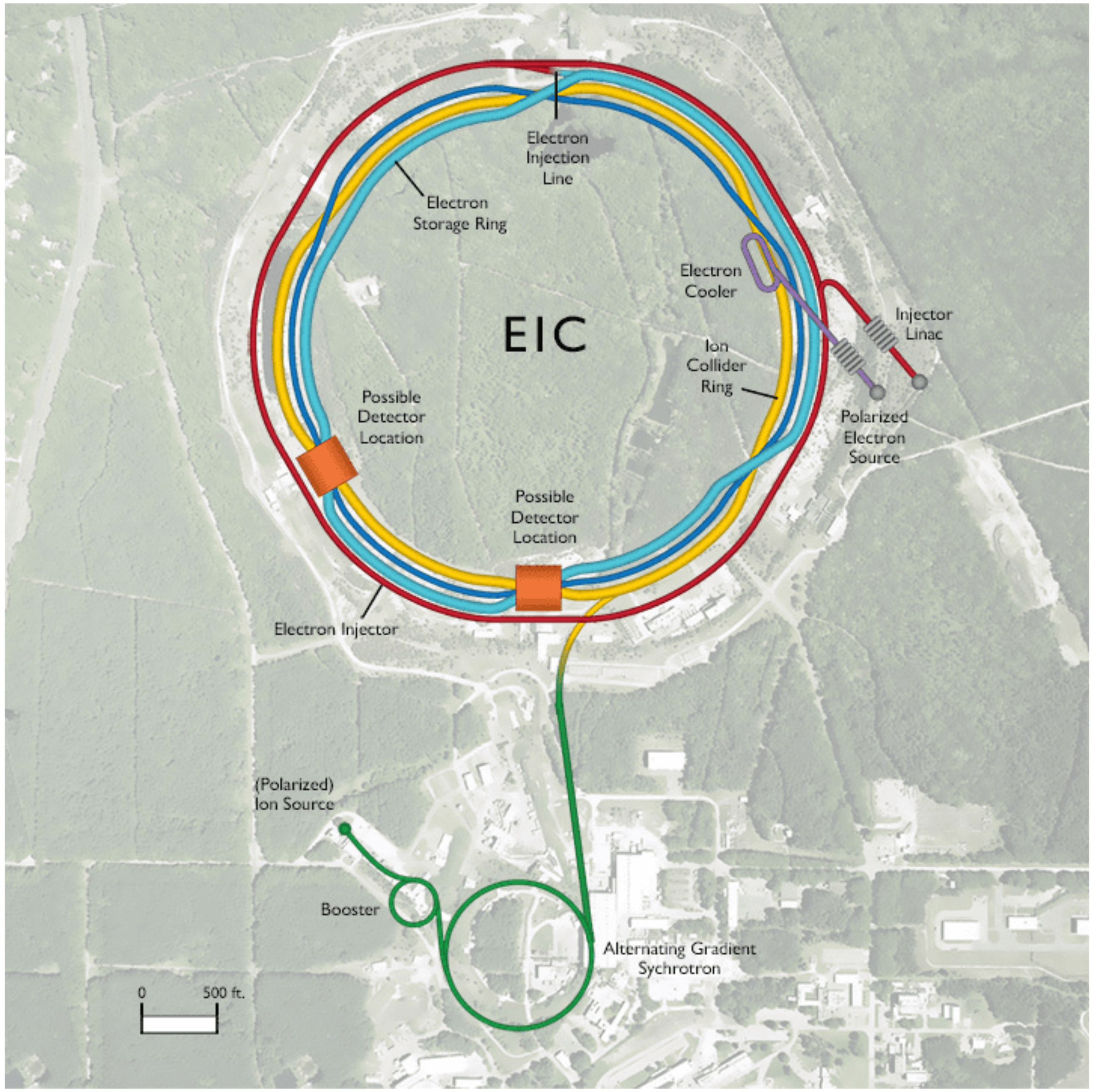
ML for Detector Optimization & Simulation

Goal:
Best experimental design suited for the Best detector reconstruction

Fernando Torales Acosta, Benjamin Nachman, Miguel Arratia, Kenneth Barish, Bishnu Karki, Ryan Milton, Piyush Karande, and Aaron Angerami

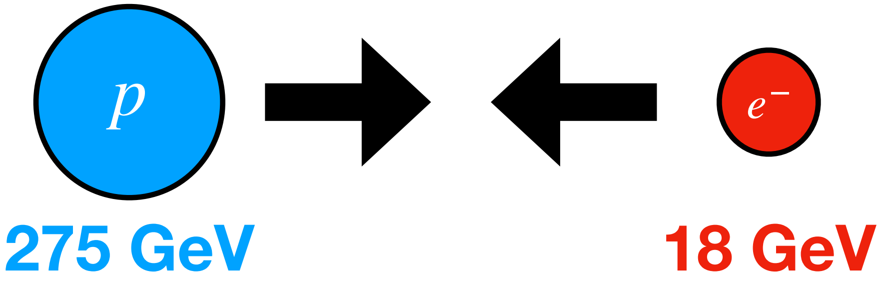
Electron Ion Collider

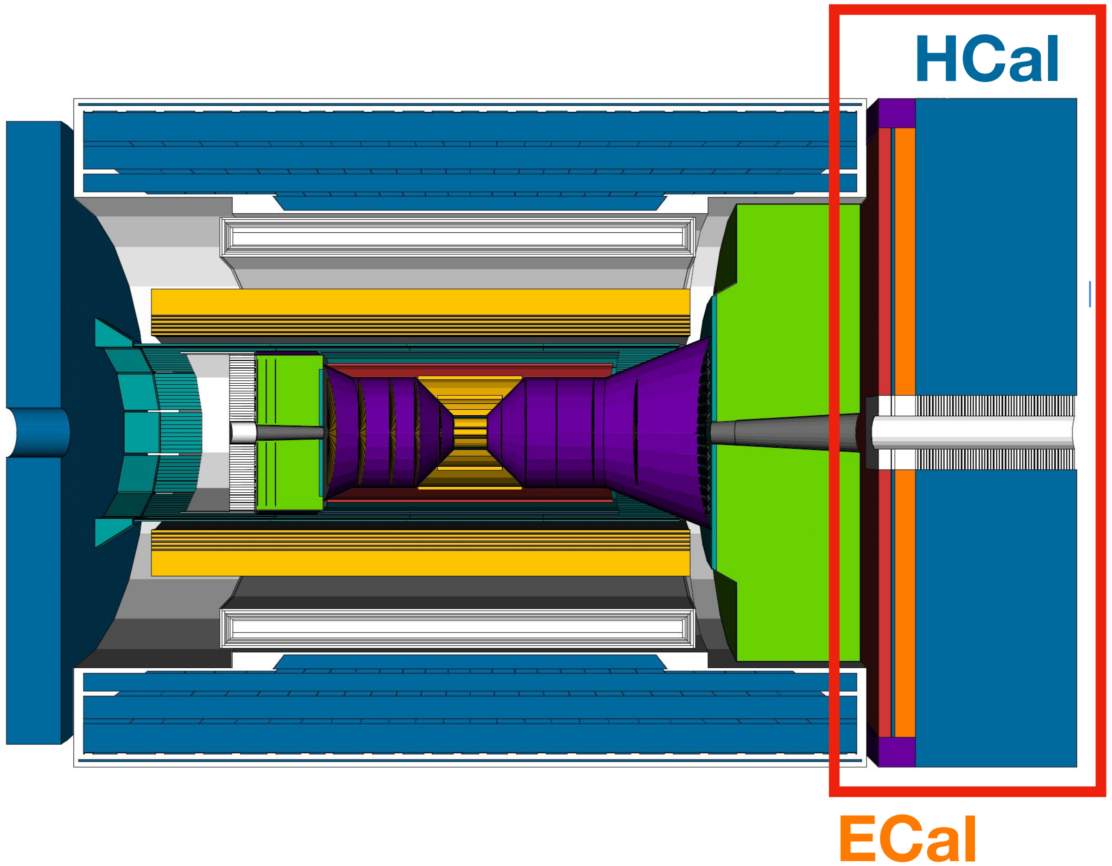


- Collide Electron and Protons + Ions
  - 18 GeV Electrons
  - 275 GeV Protons/Ions
  - $\sqrt{s} = 89$ GeV
- To be built an Brookhaven national lab, Long Island
- Provide access to regions in the nucleon/nuclei where their structure is dominated by gluons

Many detectors are still at the design stage

Forward Hadronic Calorimeter

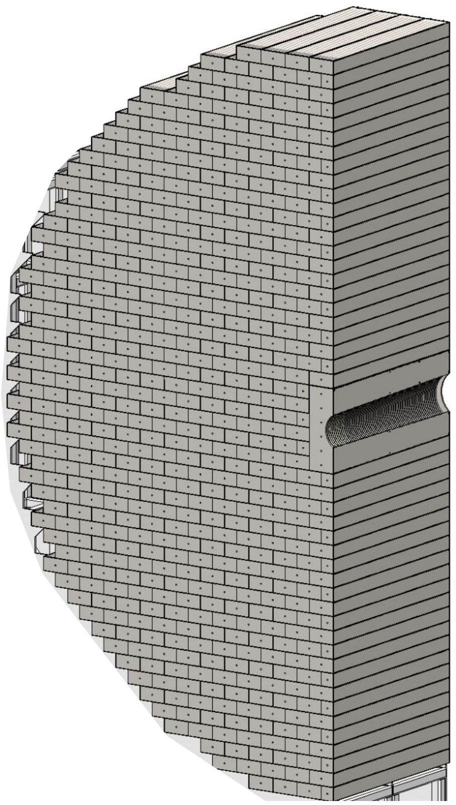




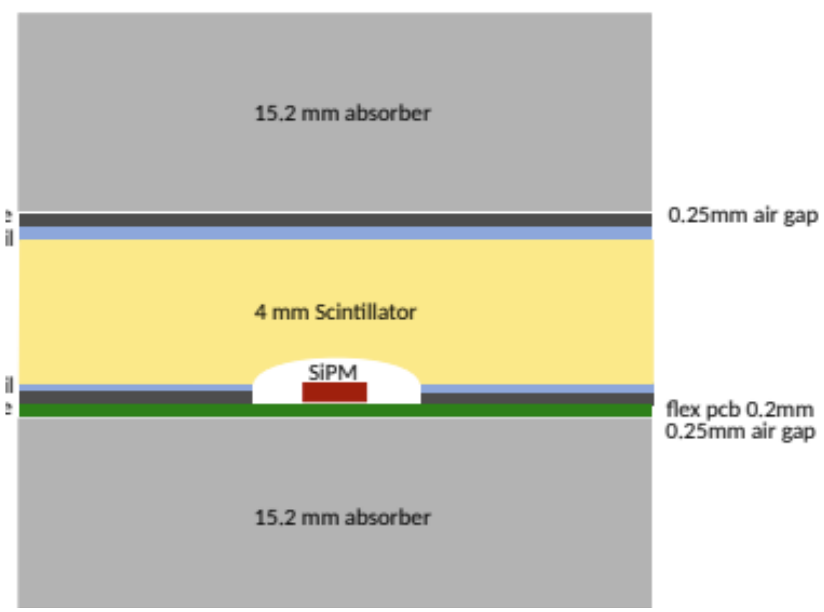
- The incoming proton/ion has a significantly larger kinetic energy than the incoming electron.
- If we want to measure *jets*, we need a granular, forward calorimeter
  - Forward region, $1.2 < \eta < 3.5$
- DeepSets and GNNs for pion *energy regression*
- Software Compensation (*energy scale*)
- G4 geometry modeled *approximately* after *ePIC*

Figure Courtesy ePIC

Forward HCal



- High-granularity iron-scintillator calorimeter
- Forward region, 1.2 < η < 3.5
- 55x55x55 cells
- Sampling calorimeter comprised of 0.3 cm scintillator tiles sandwiched between 2.0 cm steel plates



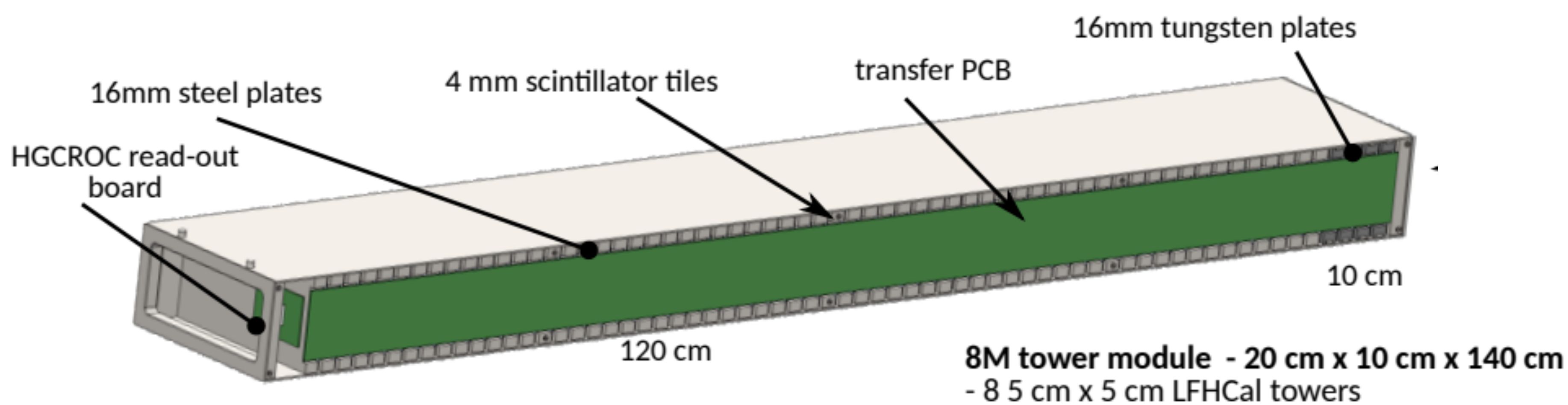
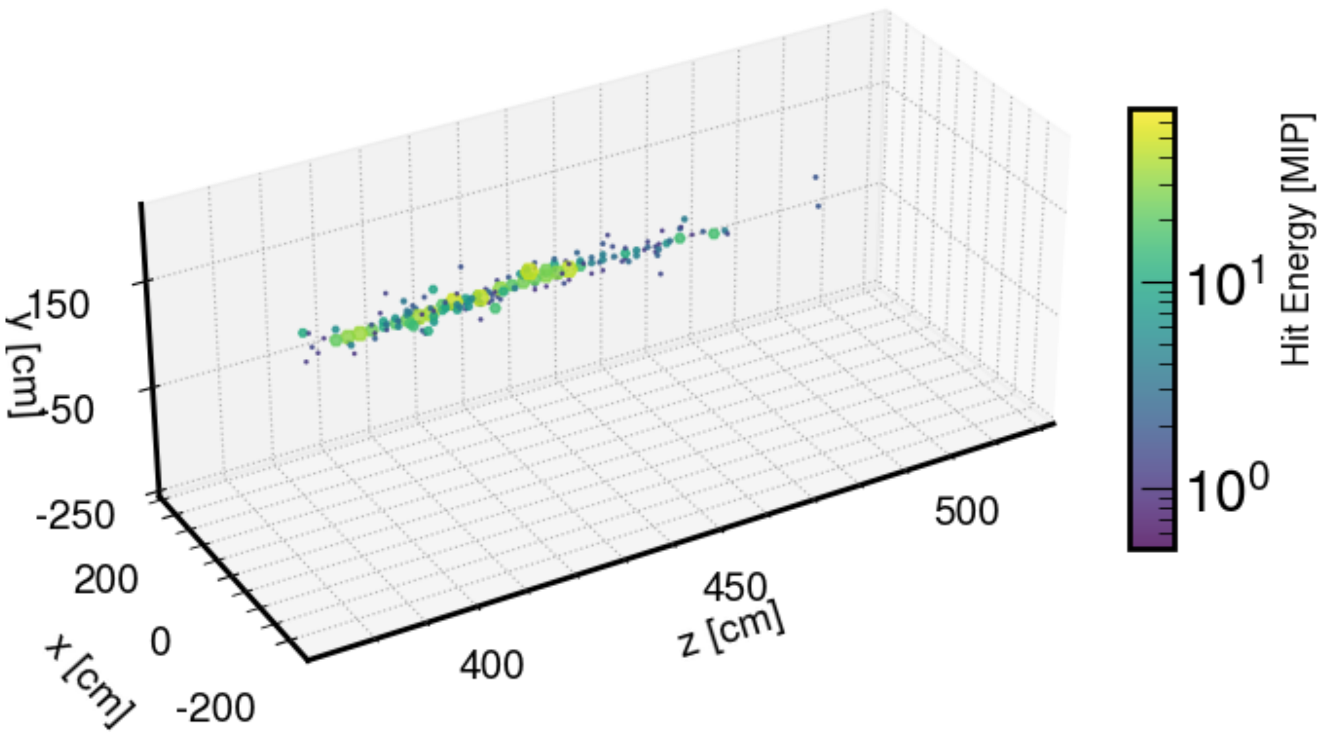
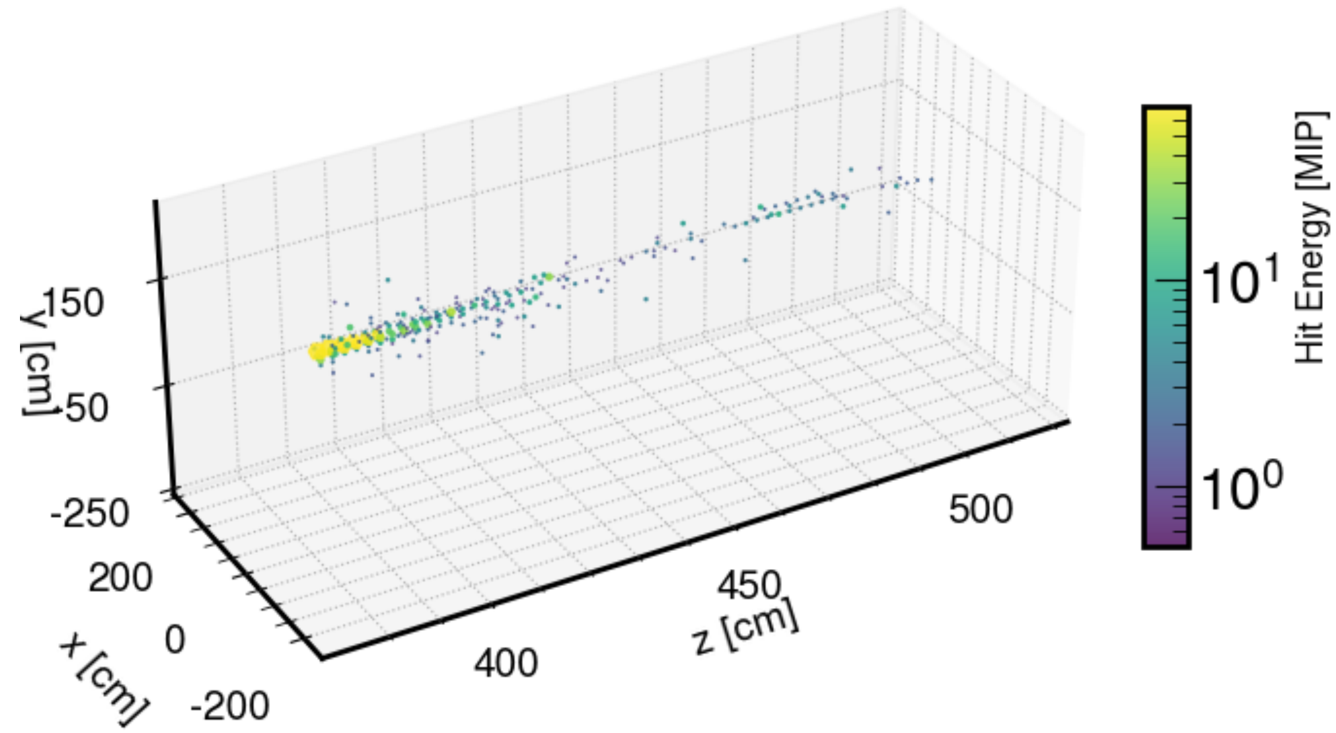


Figure Courtesy

Detector Simulation and Reconstruction



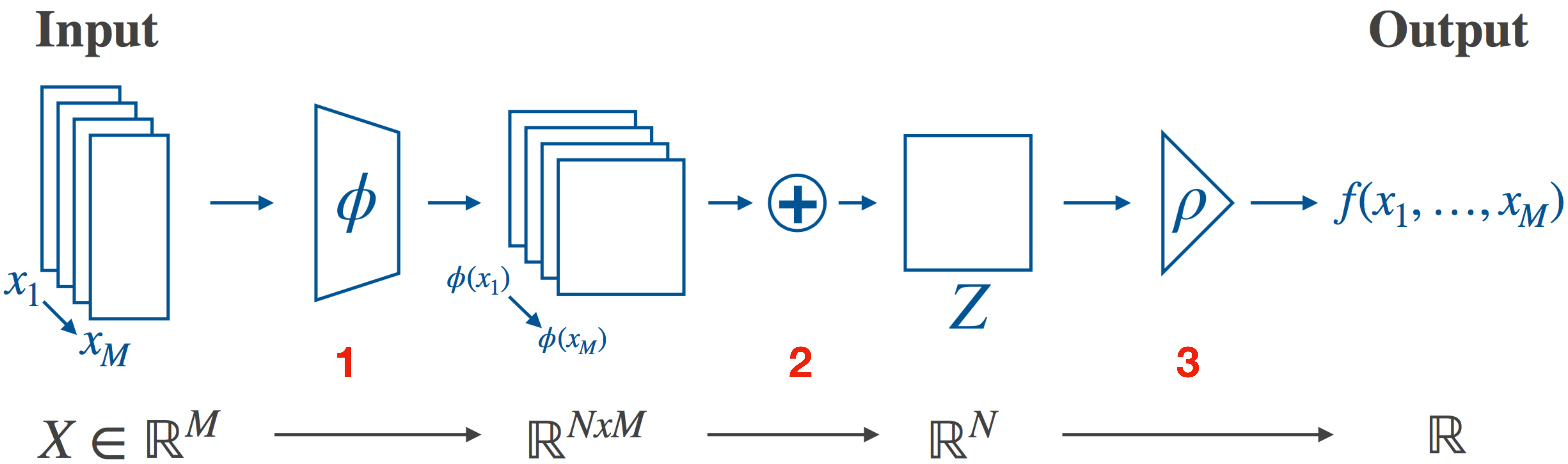


- Geant4 Simulation of single π^+ showers$1 < P_{\mathrm{Gen.}} < 125\ \mathrm{GeV}/c$
- $\mathcal{O}100 - 1000$ Cell Hits per shower, ***point clouds***
- Establish a model to predict $P_{\mathrm{Gen.}}$ given cell information
- ML for *Software Compensation* and *Energy Reconstruction*
- **Optimal reconstruction scheme is non-trivial in a complex system**

AI Codesign

- Obtain dependable, highly performant energy reconstruction scheme
- “co” design: surrogate models provide the optimal reconstruction of the high-dimensional calorimeter dataset
- Fast Simulation using generative models
- **Optimal detector design informed by the optimal detector reconstruction scheme**

Deep Sets



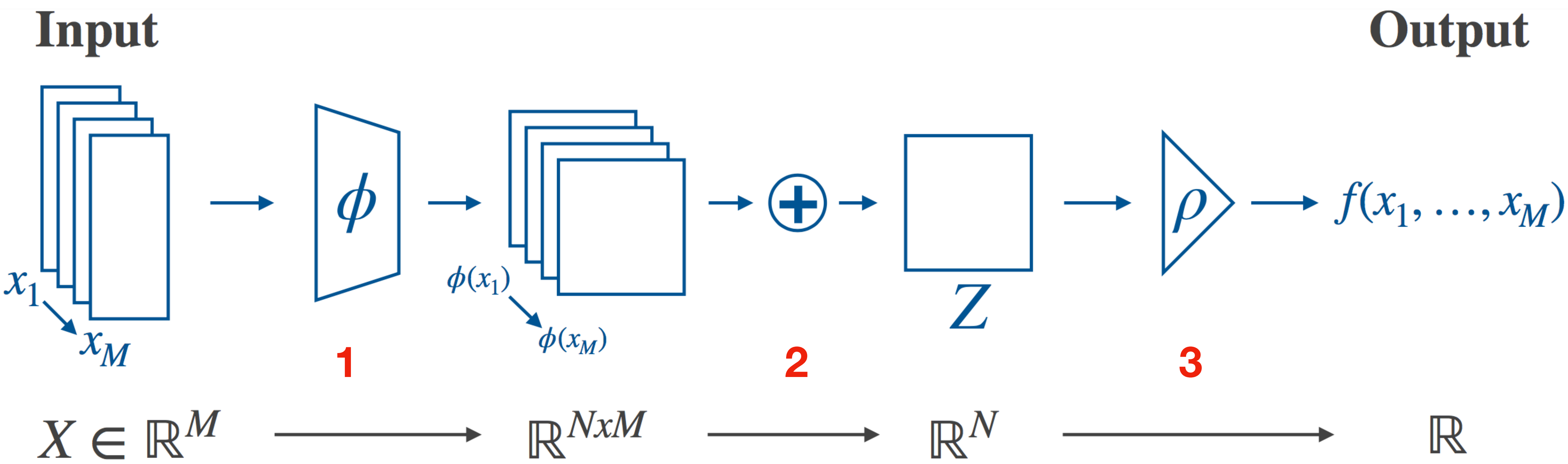
1. Transform inputs into some latent space
2. Destroy the ordering information in the latent space (+, μ)
3. Transform from the latent space to the final output

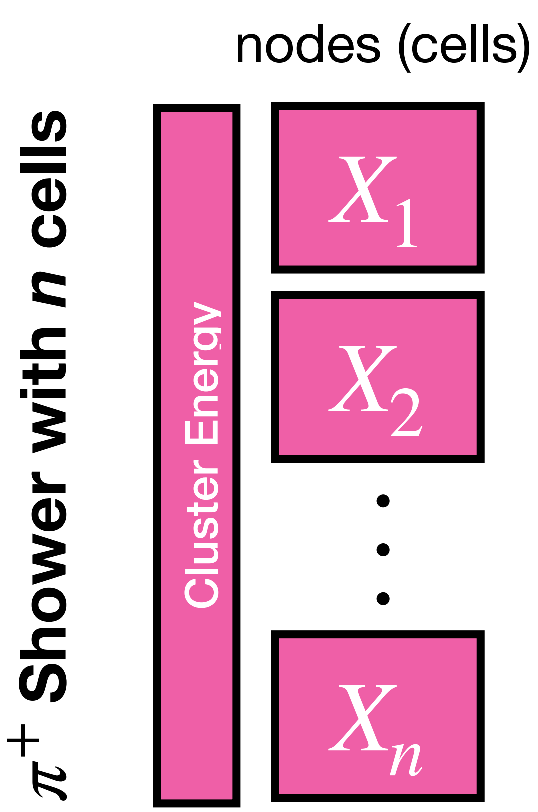
<u>Permutation Invariant</u>
Works well with point clouds
A GNN without edges

<u>arXiv: 1703.06114</u>

<u>arXiv:1810.05165</u>

Deep Sets





$$X_i = \begin{Bmatrix} \mathbf{E} \\ \mathbf{X} \\ \mathbf{Y} \\ \mathbf{Z} \end{Bmatrix} \in \mathbb{R}^4$$

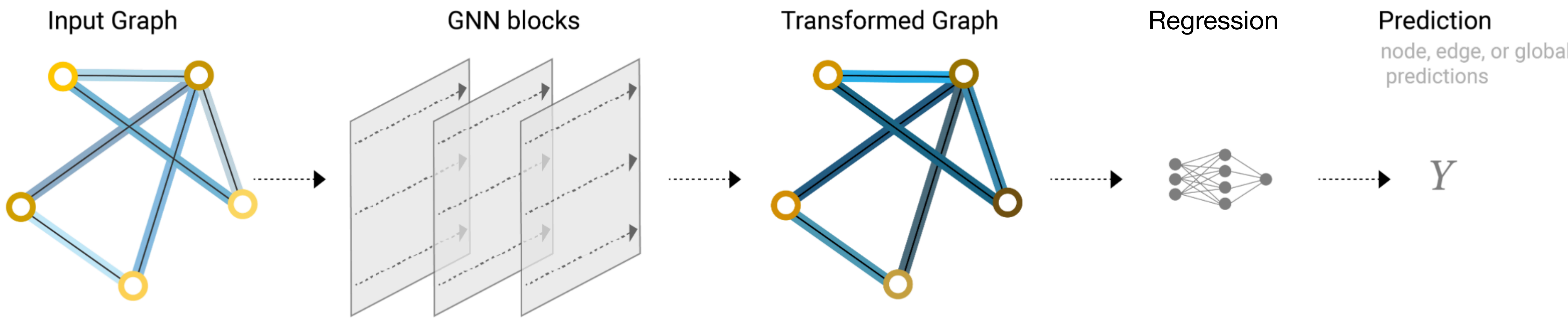
1. **Latent**
2. **Aggregation**
3. **Final Output**

Goal:

π^+ Energy Regression

Model uses energy and position information for energy regression

Graph Neural Network



- For each node in the graph, *gather* all the neighboring node embeddings (*messages*)
- *Aggregate* all messages via an aggregate function
- All pooled messages are passed through an *update function*, usually a learned neural network

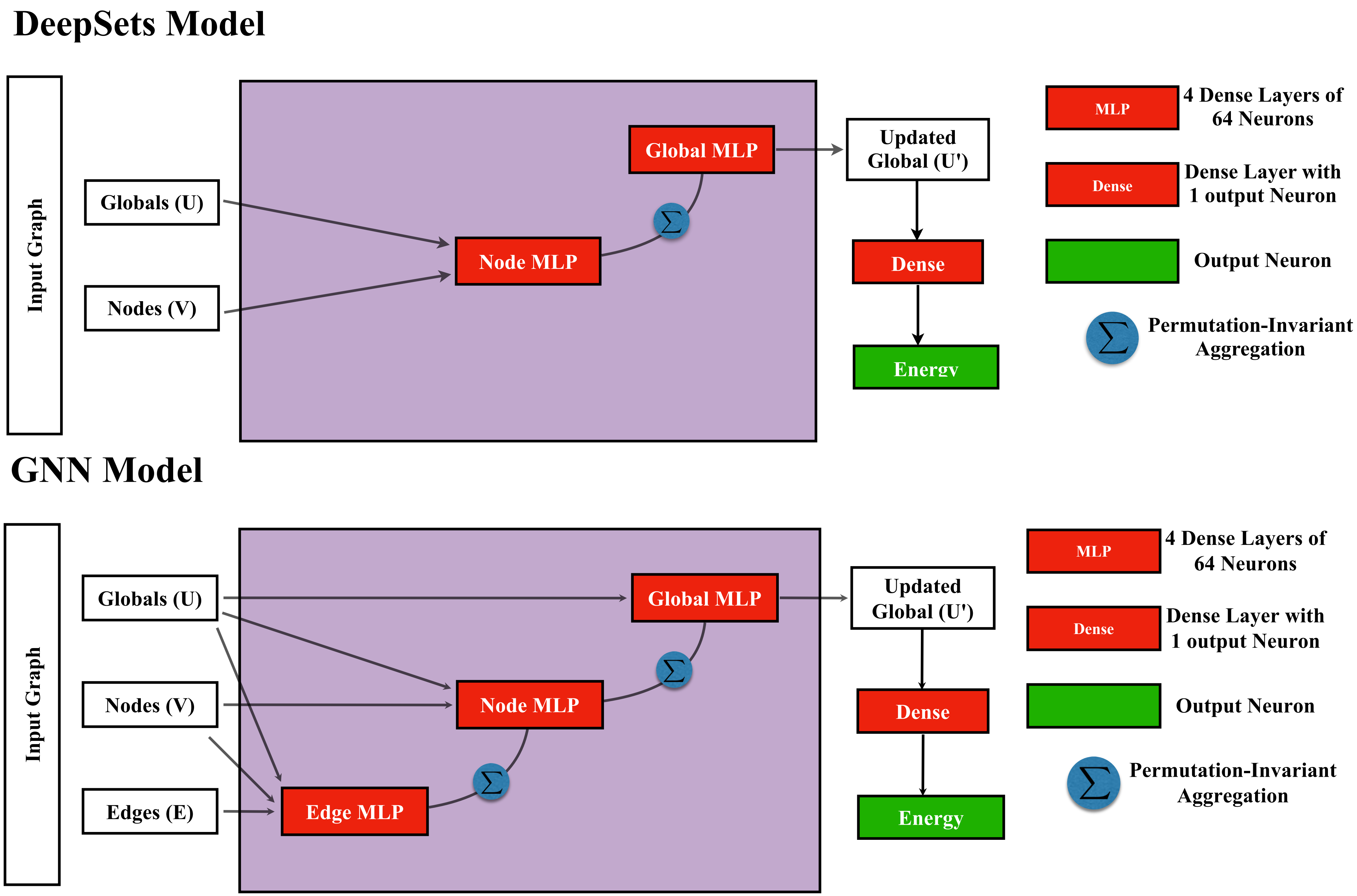
V Vertex (or node) attributes
e.g., node identity, number of neighbors

E Edge (or link) attributes and directions
e.g., edge identity, edge weight

U Global (or master node) attributes
e.g., number of nodes, longest path

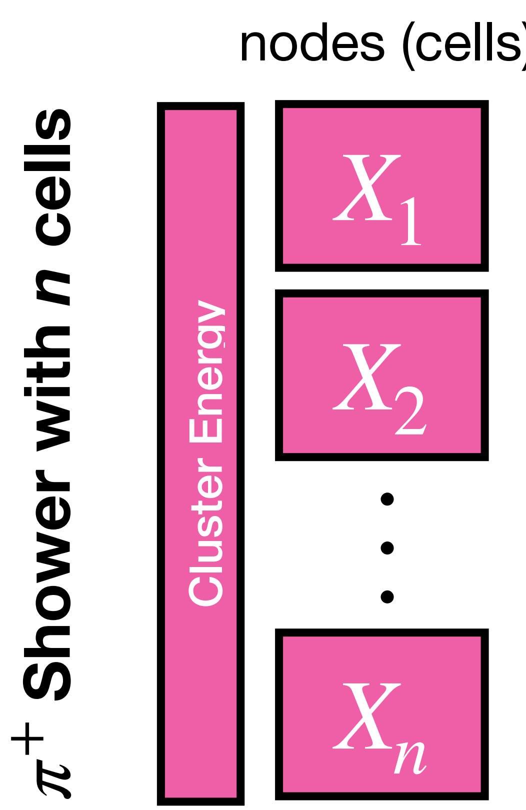
Using k-nearest neighbors

Obligatory Model Schematics



- In *theory,* DeepSets can learn *everything* a GNN can
- We encode geometric information *directly* in the GNN

Varying the Input Features

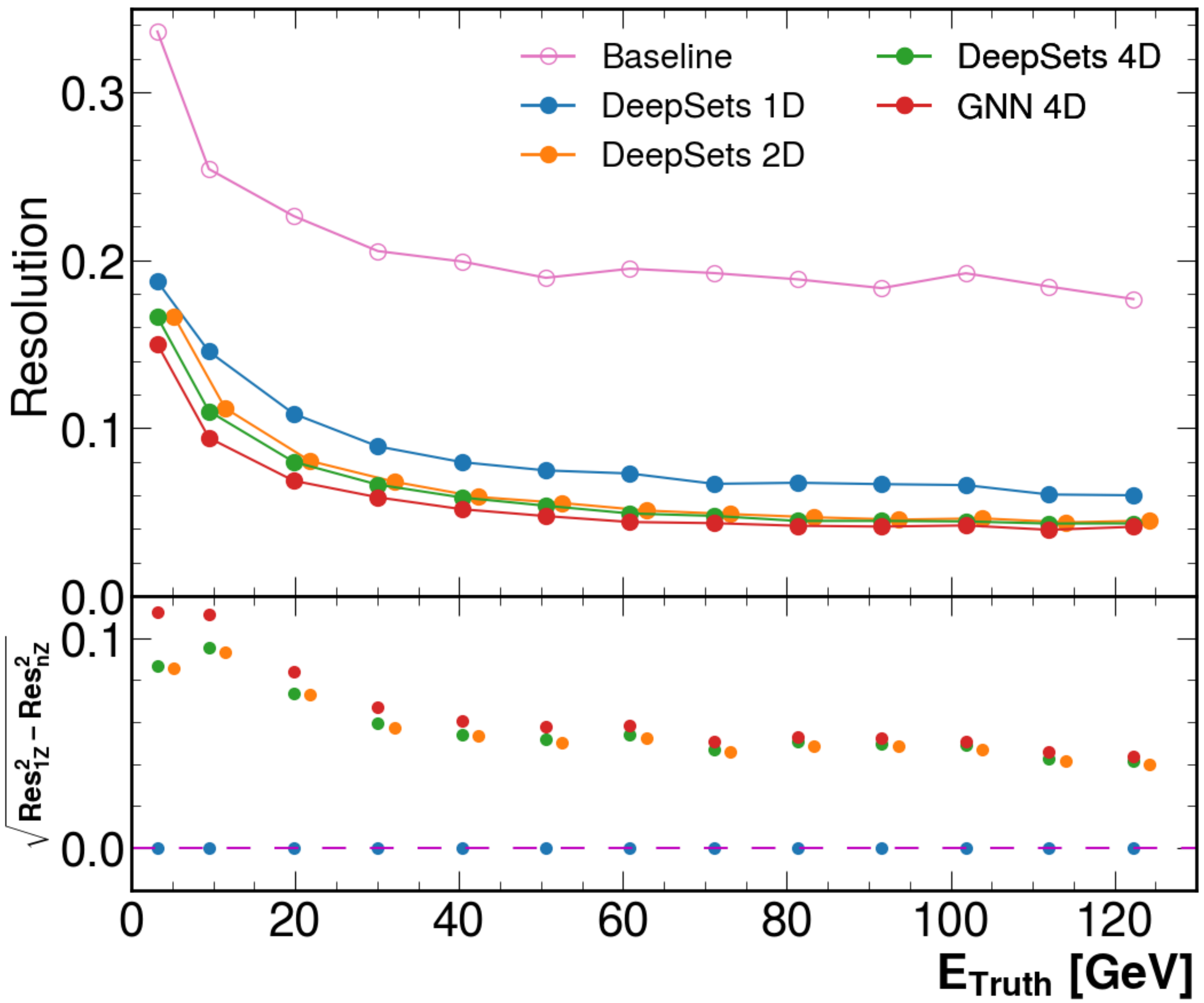


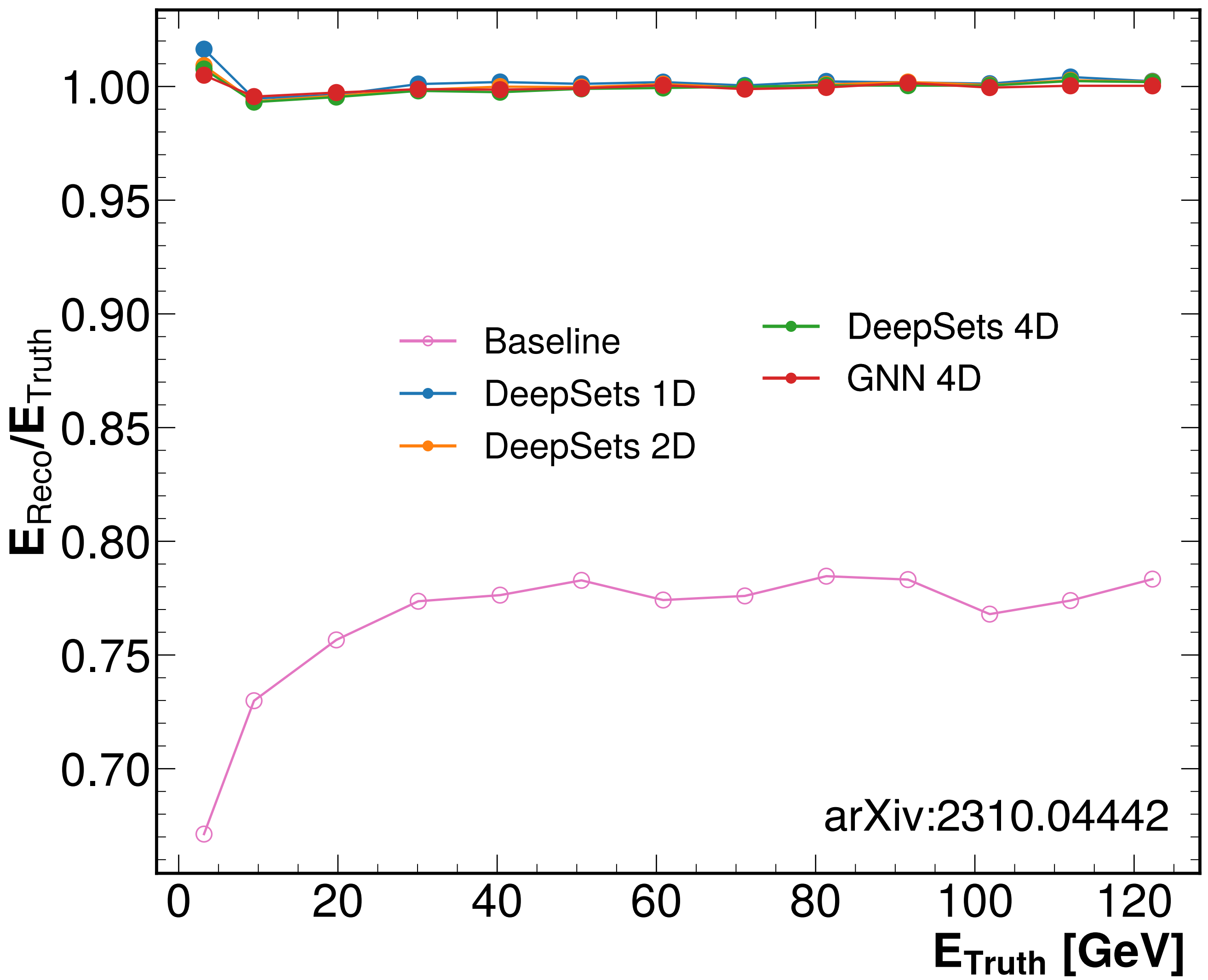
$$X_i = \left\{ \begin{matrix} \mathrm{E} \\ \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{matrix} \right\} \in \mathbb{R}^4$$

Want understand what information is most relevant for the Energy Reconstruction and ML-based software reconstruction

We train Deepsets models on E, E+Z, E+XYZ (1D, 2D, 4D)

Energy Regression: Feature Dimension

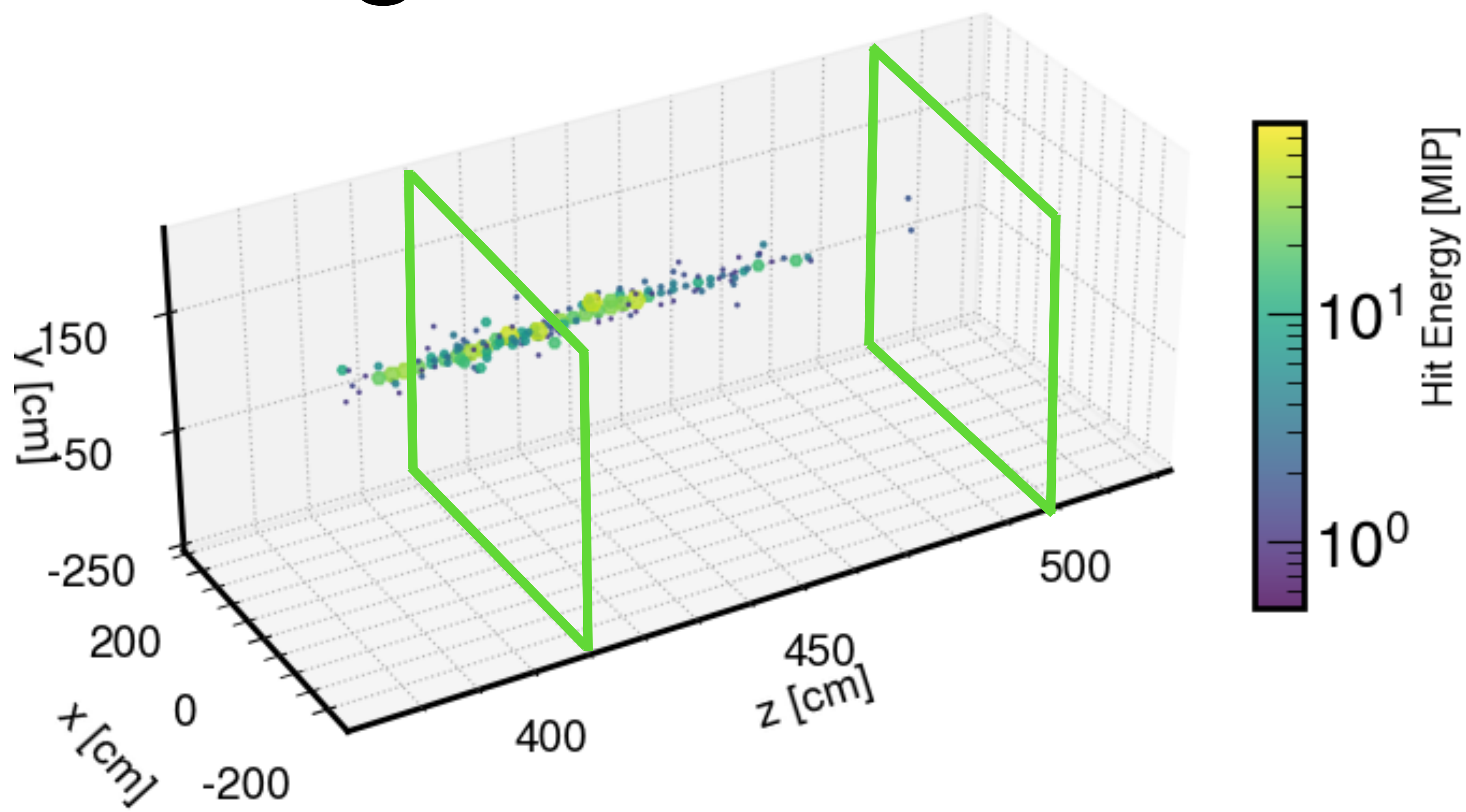




- Biggest improvement after the inclusion Z information (2D →4D less impactful)
- Less sensitive to fluctuations of the EM fraction of the shower
- Energy scale within 2% of truth (1% after 10 GeV)
  - *Effective Software Compensation!*

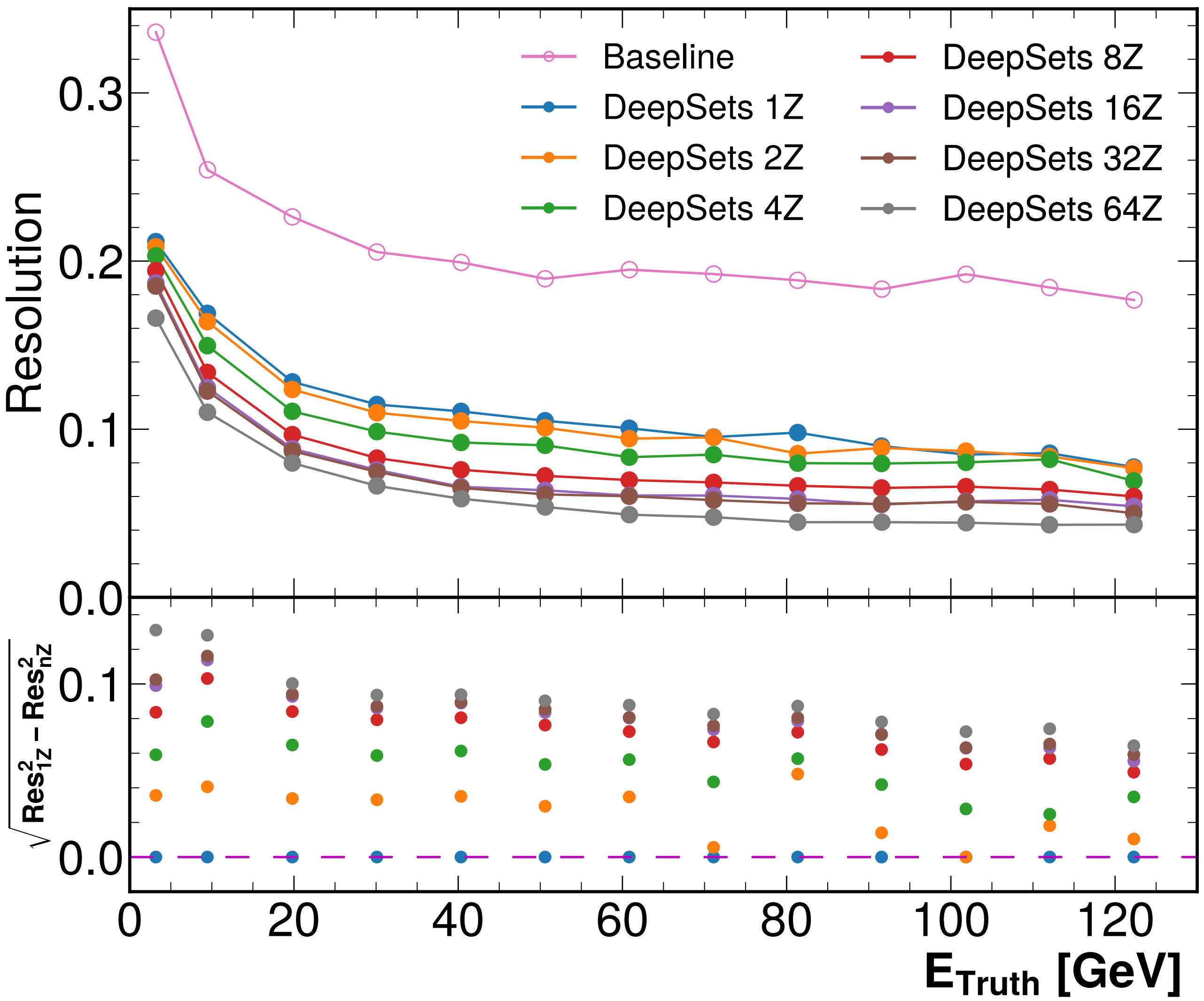
Data Processing for Models

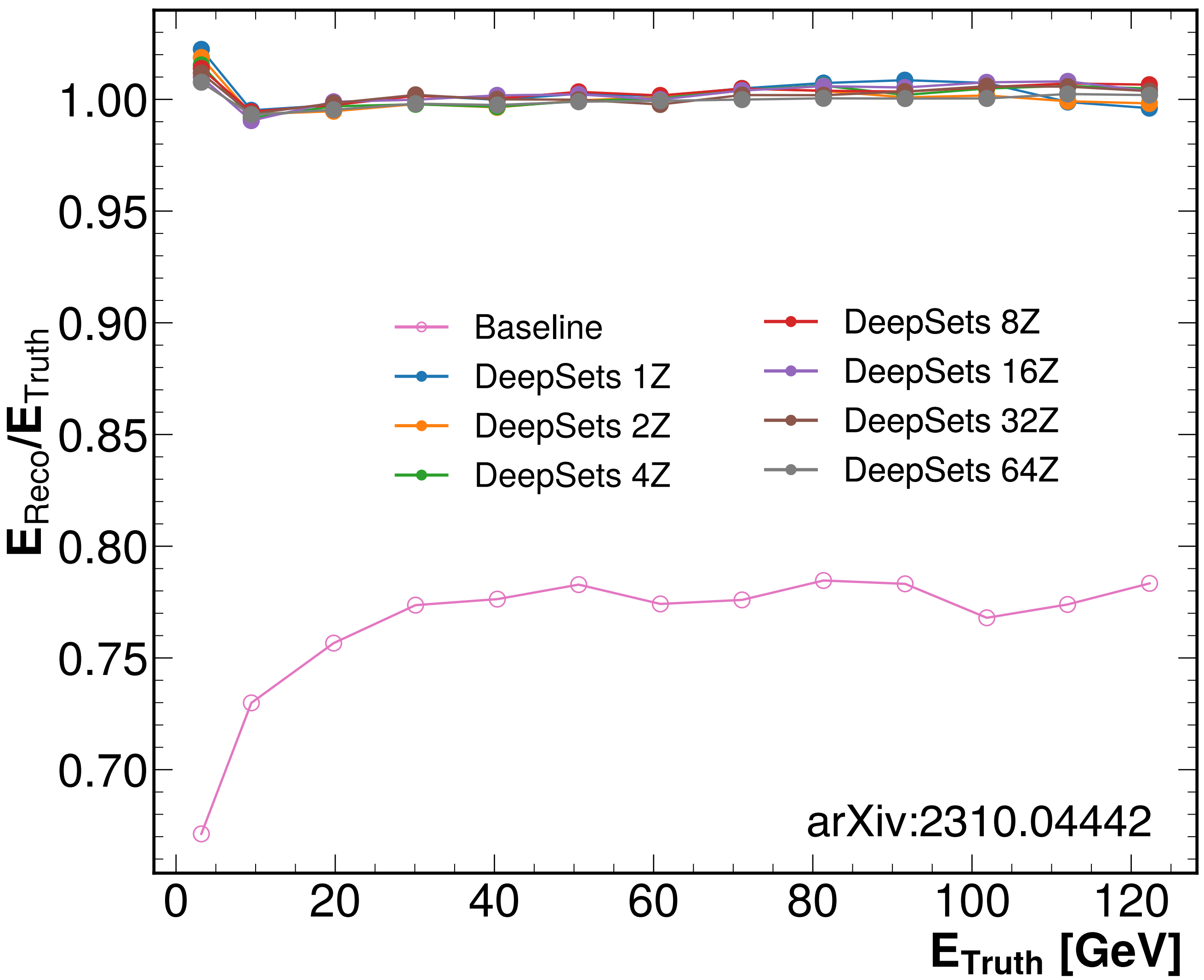
4-Layer Configuration



- Full point-cloud readout is *unrealistic* for final detector
- Segment the calorimeter N=1-64 layers
- Run regression, identifying optimal longitudinal configuration

Energy Regression: Number of Layers





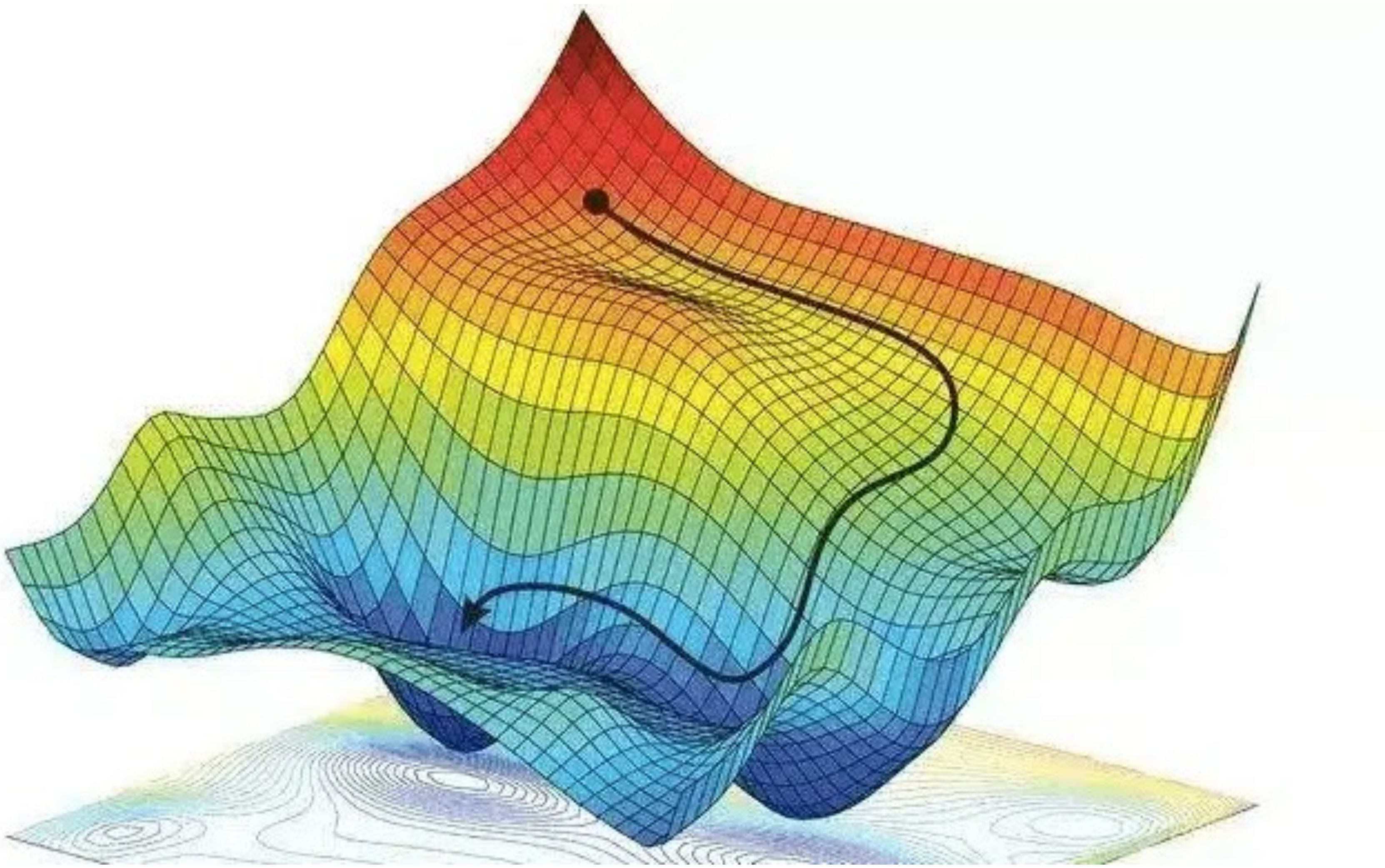
- 1-Layer configuration w/ Deepsets outperforms baseline
- Intuitive increase in performance as N_z increases
- Software compensation does incredibly well

ML for Energy Reconstruction

https://arxiv.org/abs/2310.04442

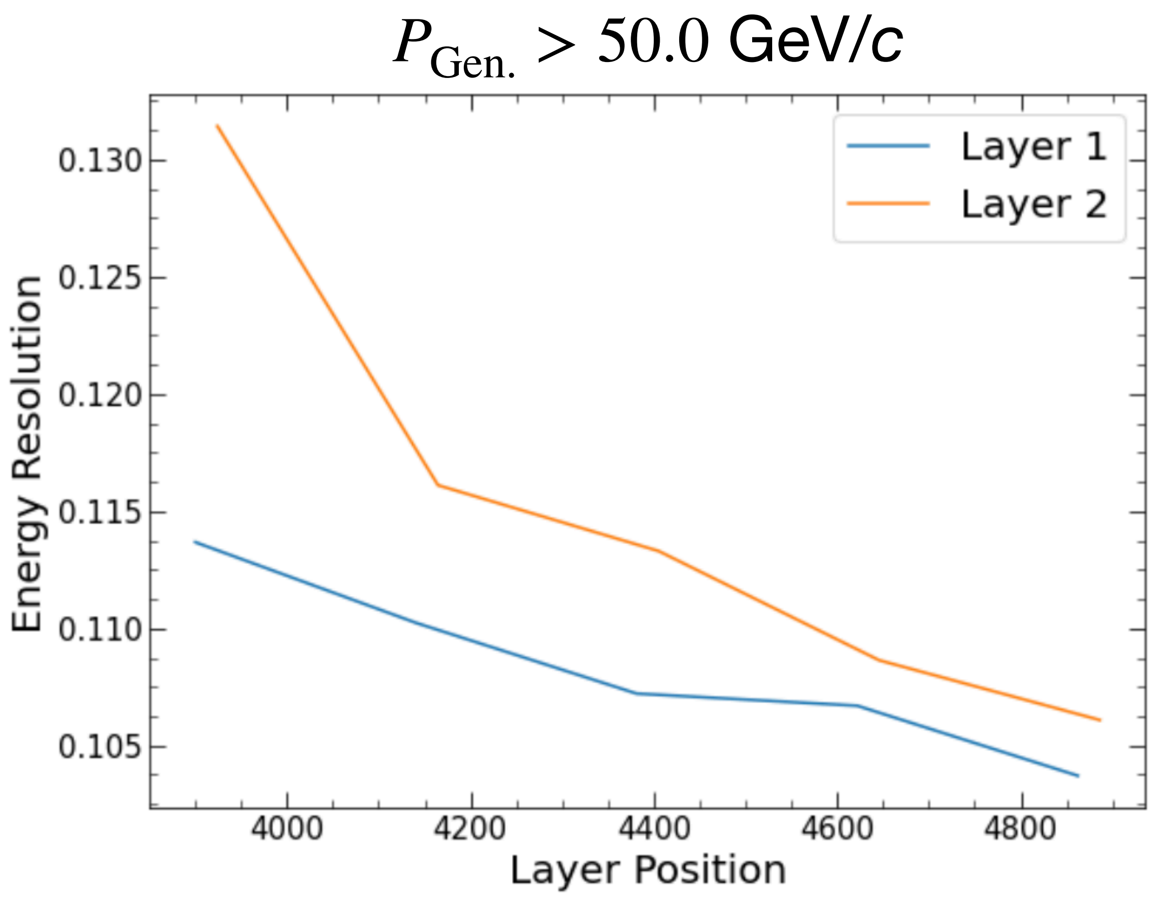
- DeepSets & GNNs extremely effective models for software compensation
  - Energy scale within 1% of unity
- Every Resolution most effected by longitudinal information. Less sensitive to transverse segmentation
- Can easily regress ϕ, η

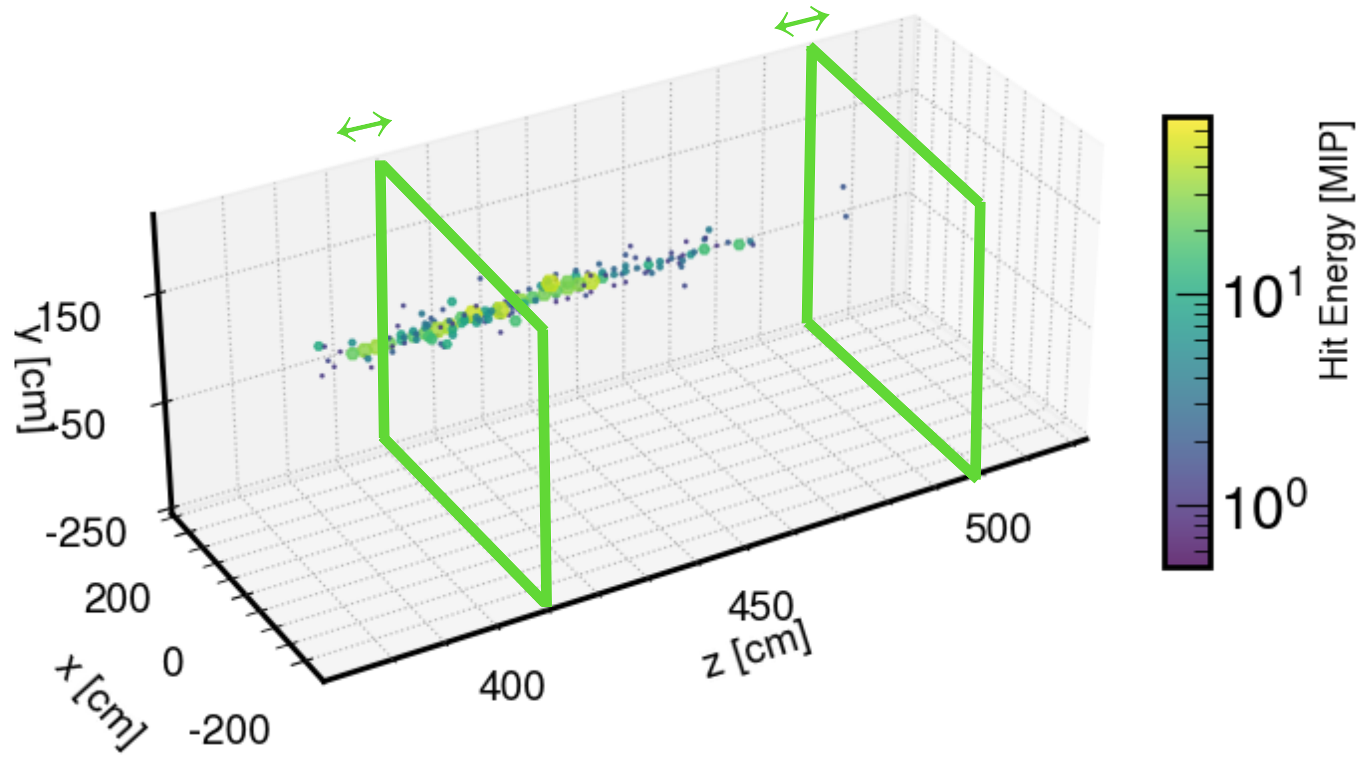
Conditioning Model l_z



Can we use gradient-optimization techniques to optimize our detector design?

Conditioning Models





- Similar to the 1-64L study, we re-group the point cloud
- Models have the same point-cloud input as before
- **Addition of l_z input**
  - position of longitudinal boundaries
- For every event, 5 random configurations of layers are created

Conclusions

- GNNs and DeepSets towards optimal segmentation and energy reconstruction
  - Emphasis on the importance of optimizing longitudinal segmentation
  - arXiv:2307.04780

- These models can lay a foundation for gradient optimized detector design
  - MODE_2023

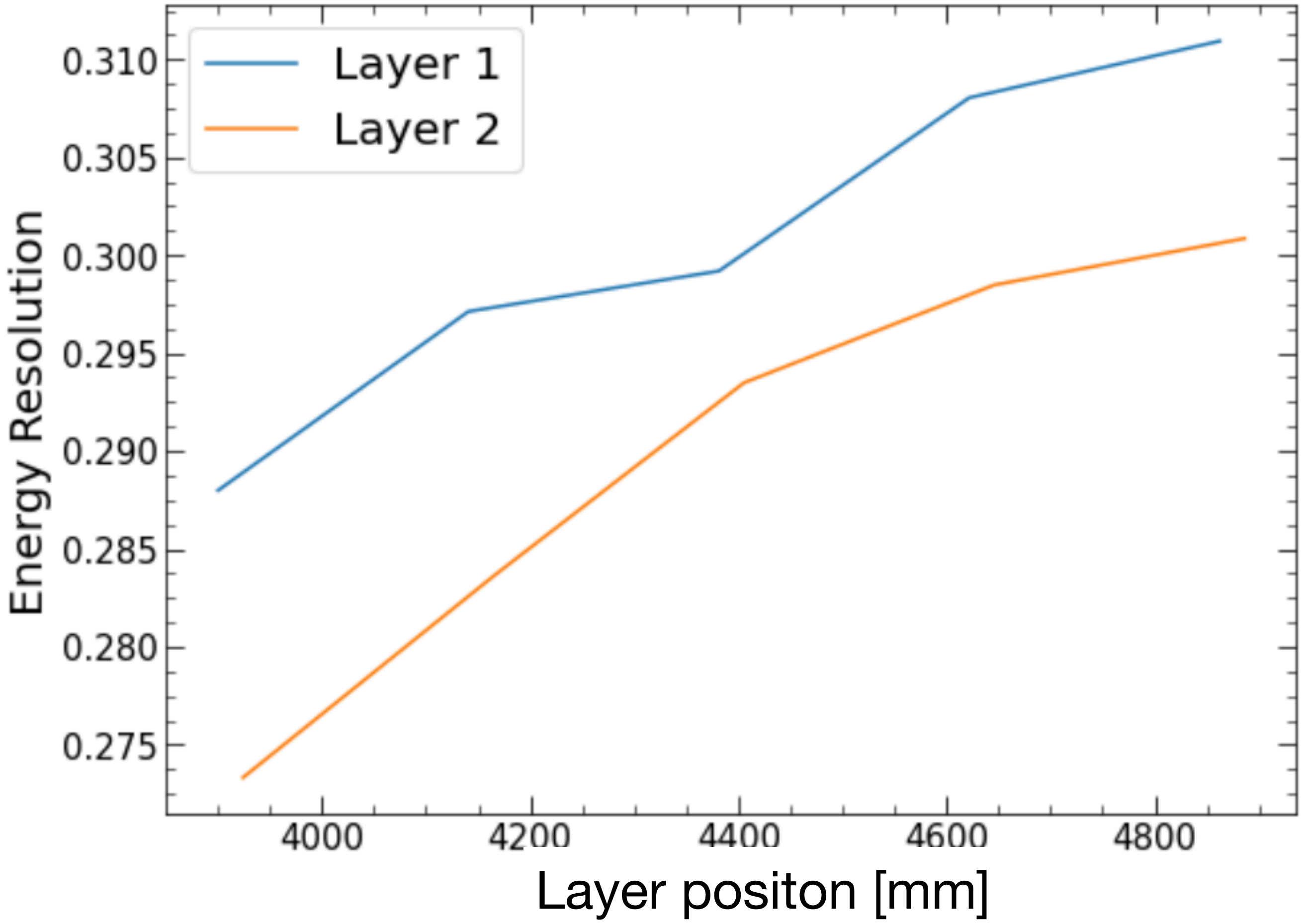
- Score based generative models using point clouds are ideal for fast calo-sim at the EIC
  - arXiv:2307.04780

END

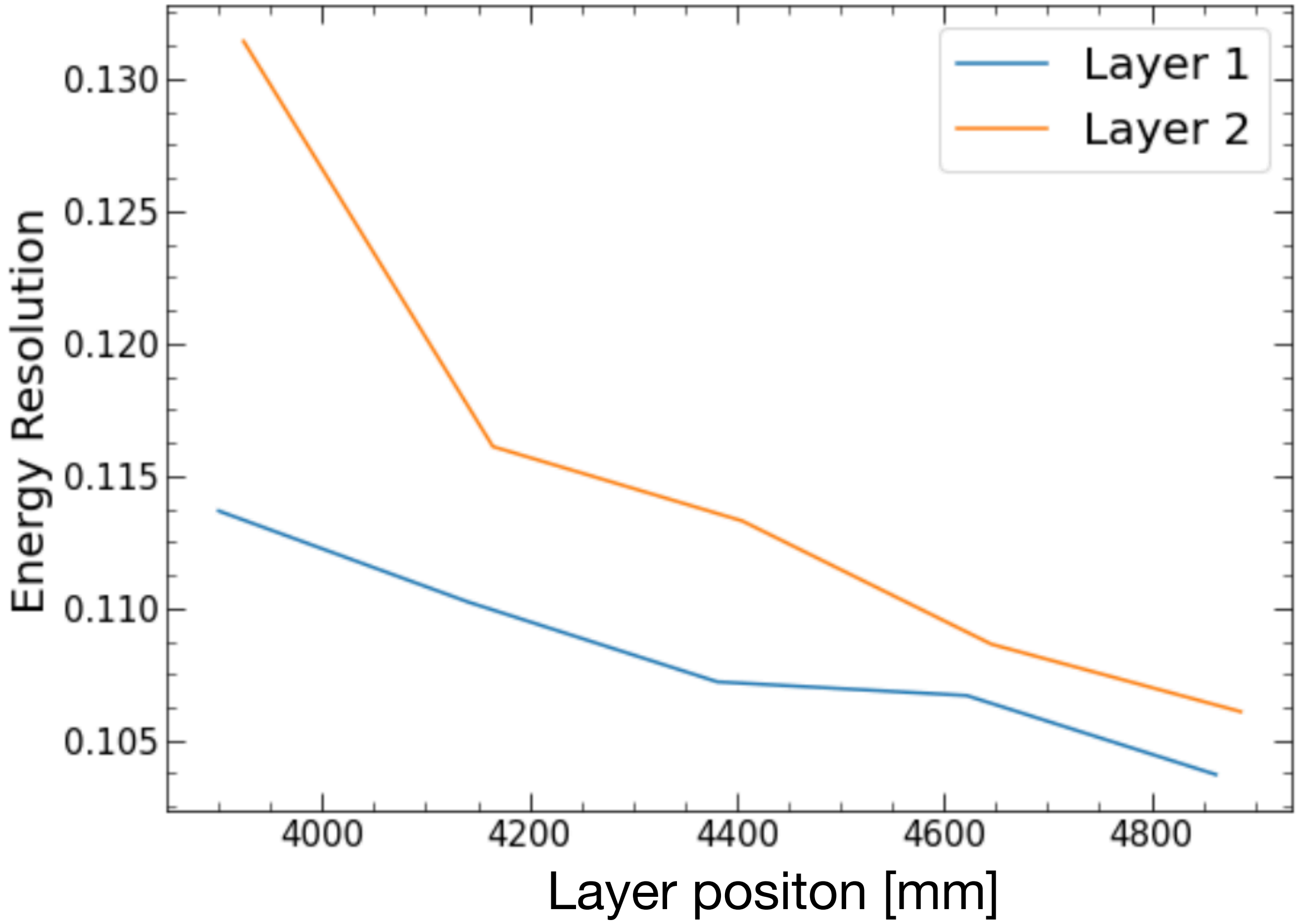
Backup

$$\sigma_E = f(z_1,\ z_2, \vec{x})$$

$P_{\mathrm{Gen.}} < 10.0$ GeV/c

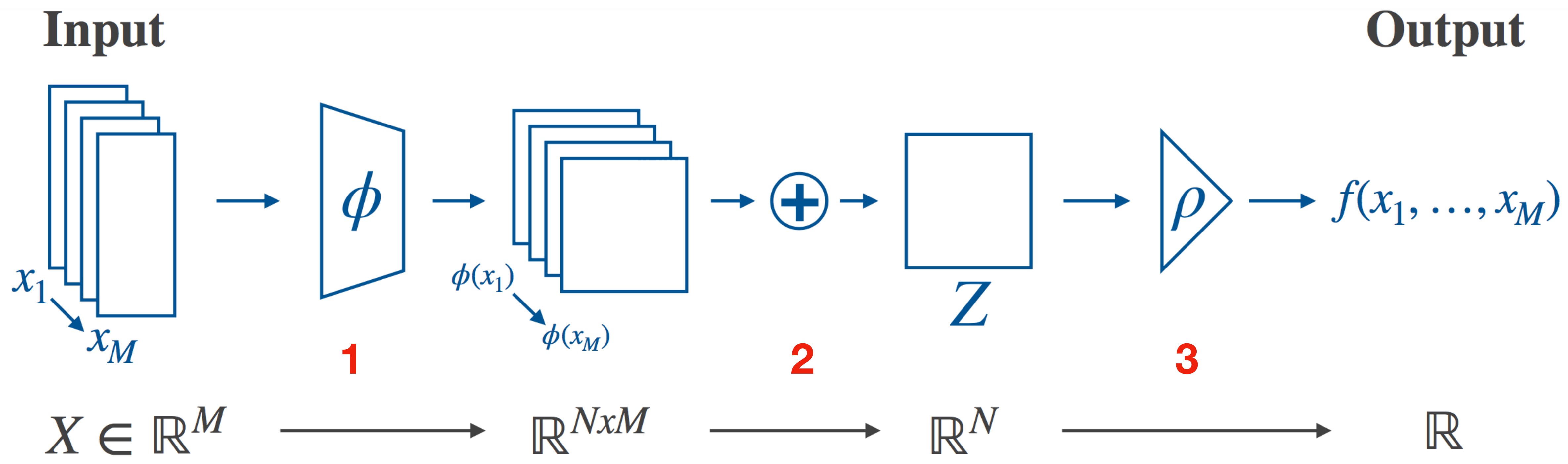


$P_{\mathrm{Gen.}} > 50.0$ GeV/c



We have a differentiable function for energy resolution

Deep Sets



1. Transform inputs into some latent space
2. Destroy the ordering information in the latent space (+, μ)
3. Transform from the latent space to the final output

<u>Permutation Invariant</u>
Works well with point clouds
A GNN without edges

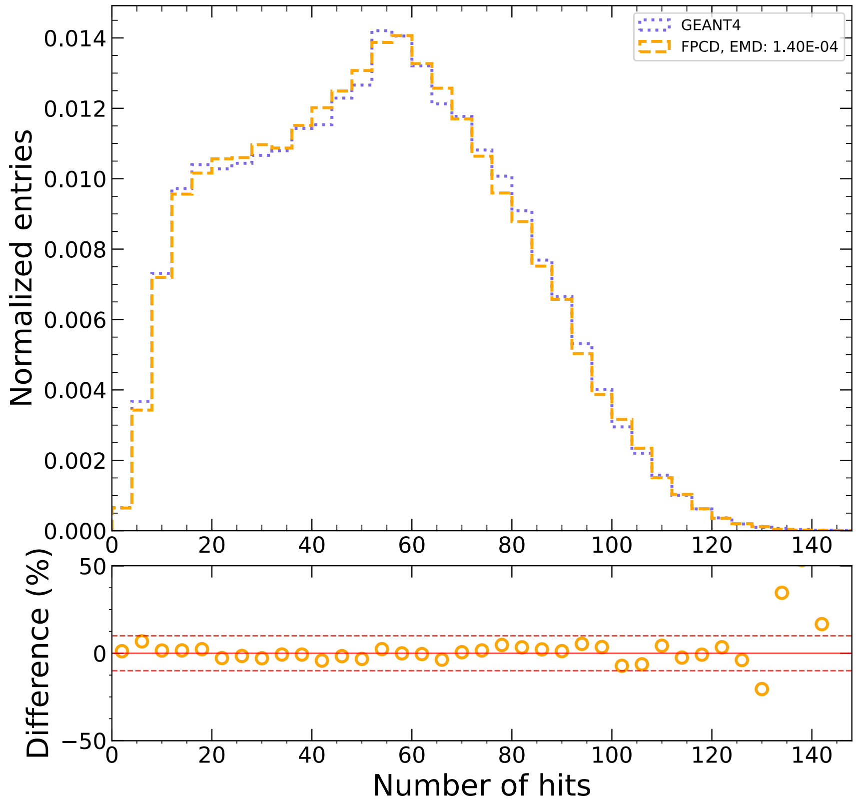
<u>arXiv: 1703.06114</u>

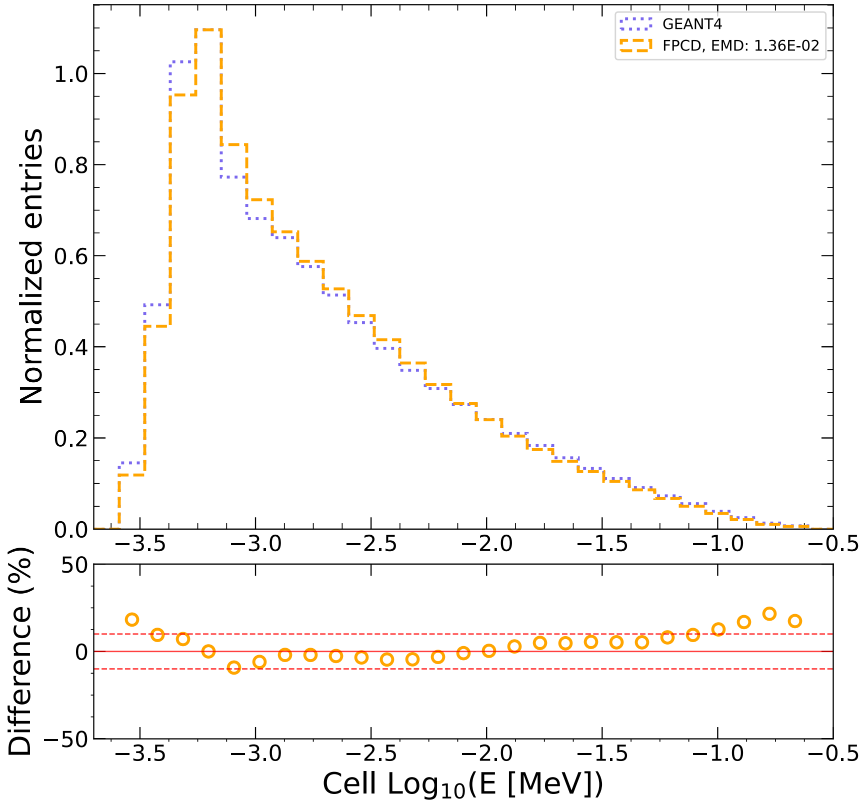
<u>arXiv:1810.05165</u>

2 Diffusion Models

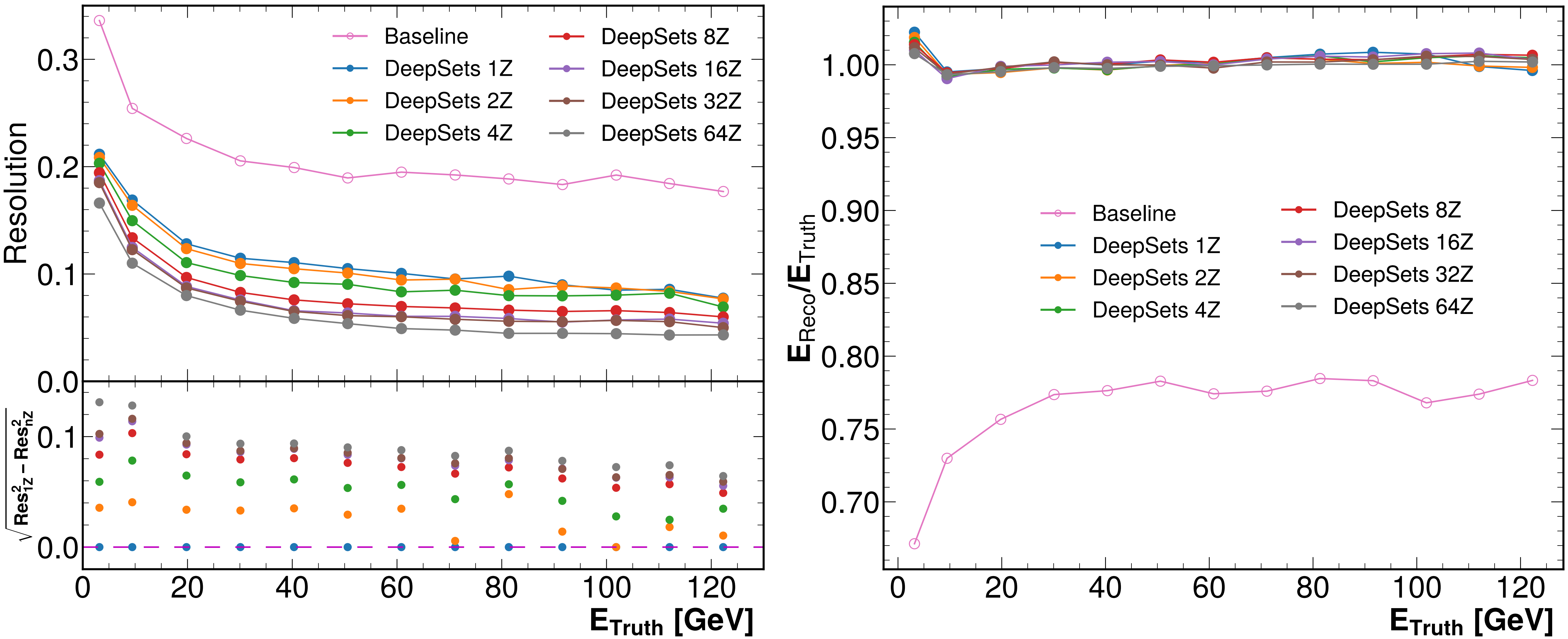
1. Take as input $P_{\text{Gen.}}$, learns N_{hits}

2. Take as input N_{hits}, learns E_{cell}, X_{cell}, Y_{cell}, Z_{cell}



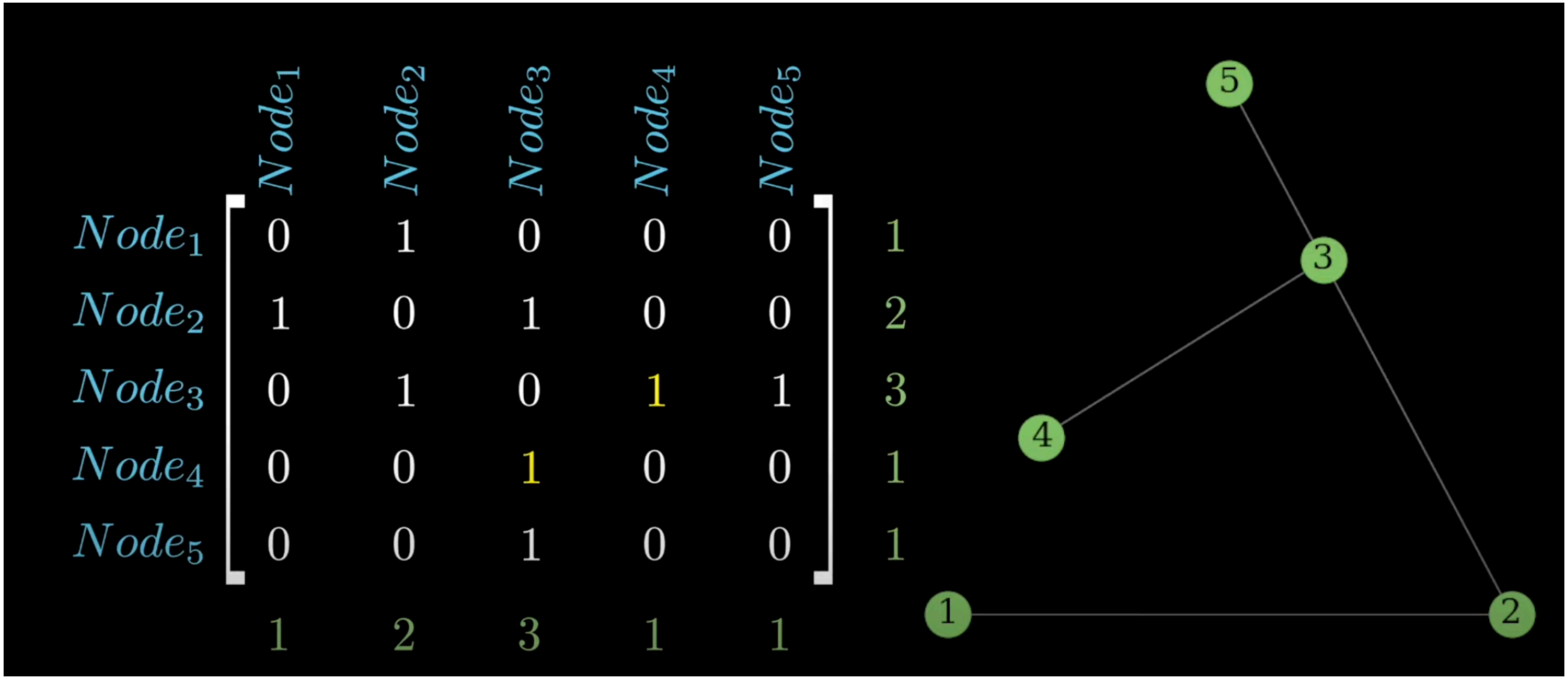


Energy Regression Results



- Geant4 Simulation of single π^+ showers
- Condition model on position of longitudinal segmentation

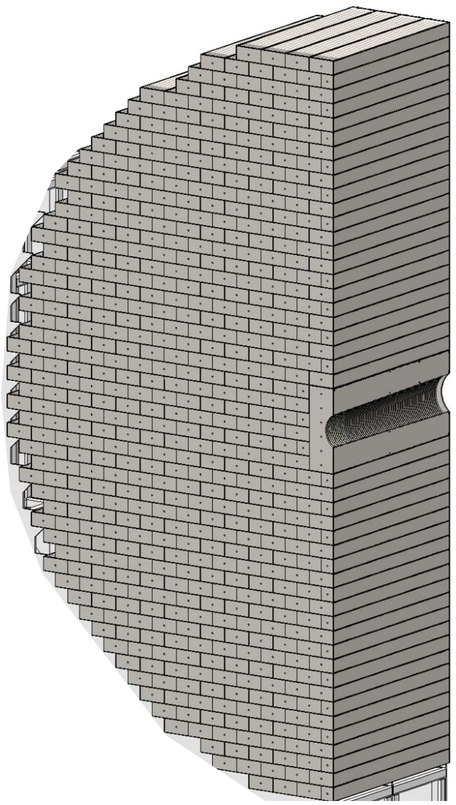
Adjacency Matrix



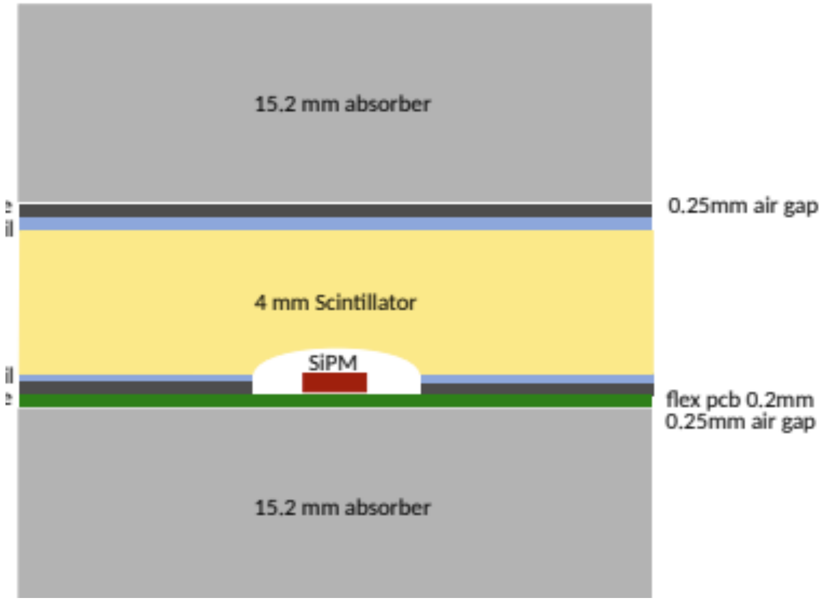
Simple Example of Adjacency Matrix for GNN

In our case, we use Keras K-Nearest neighbors algorithm in cartesian coordinates

Forward HCal



- High-granularity iron-scintillator calorimeter
- Forward region, 1.2 < η < 3.5
- Sampling calorimeter comprised of 0.3 cm scintillator tiles sandwiched between 2.0 cm steel plates



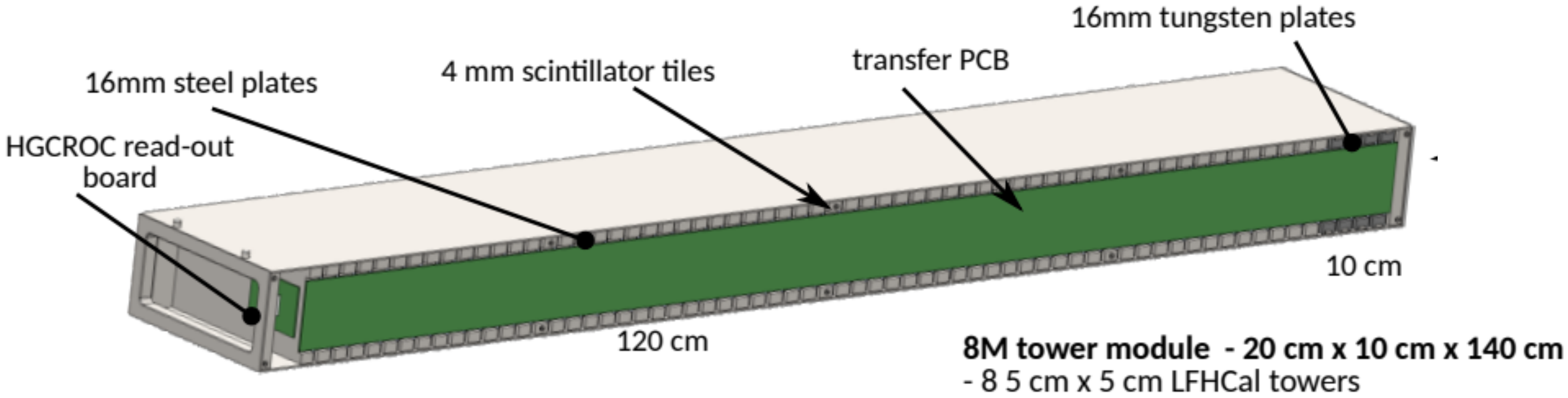
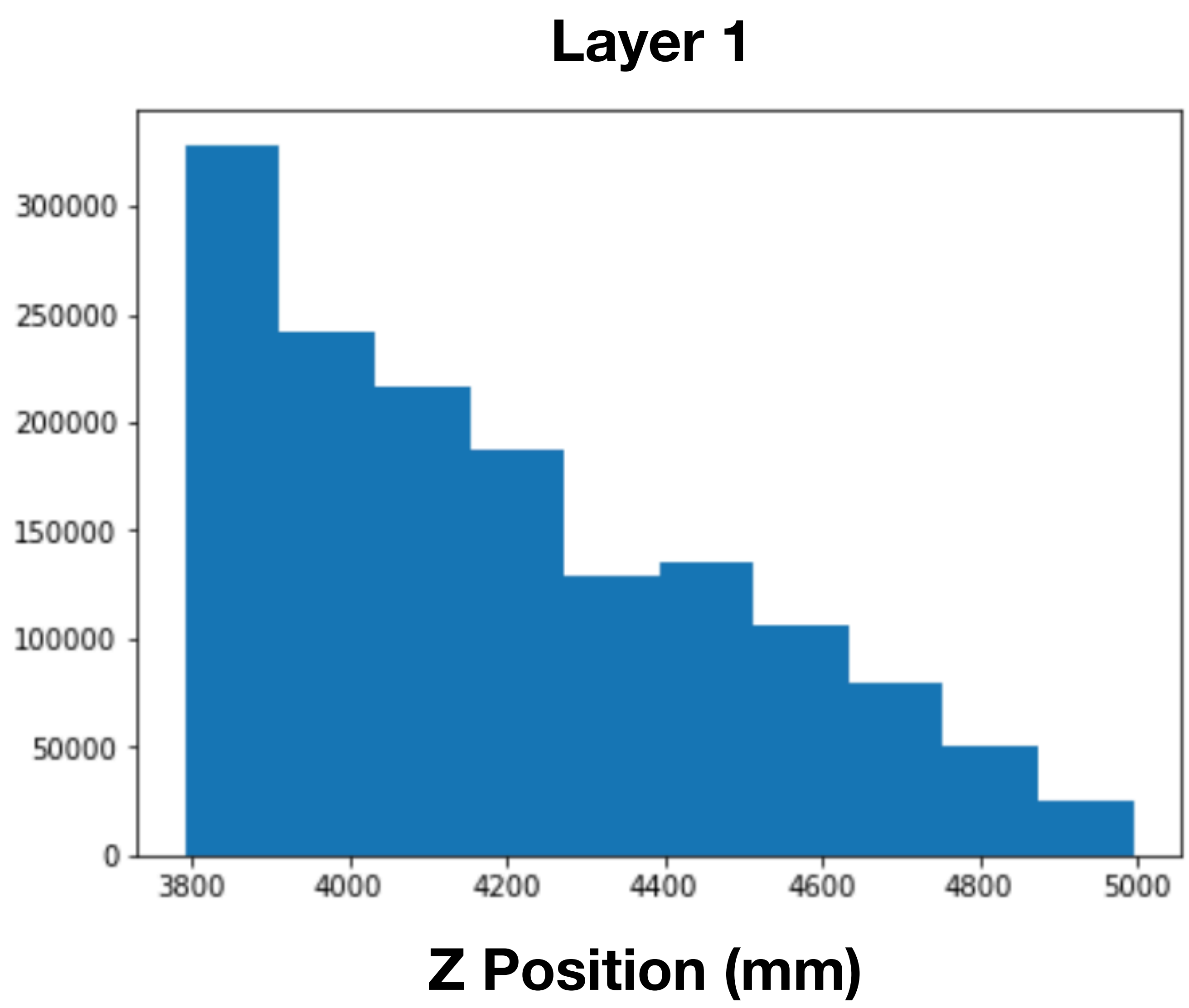
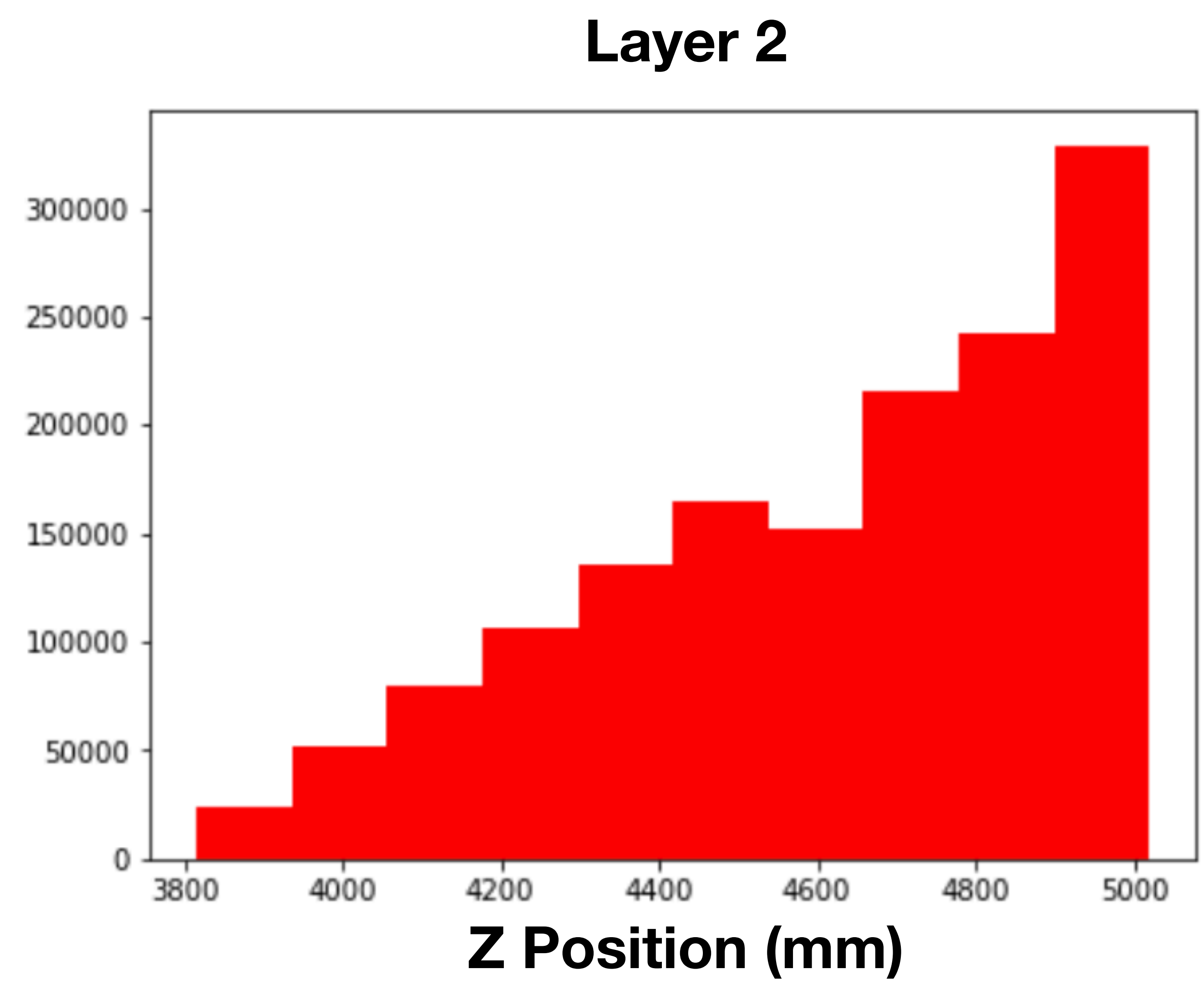


Figure Courtesy ePIC

Longitudinal Segmentation





MSE Loss Plot

$1.0 < P_{\mathrm{Gen.}} < 125.0$ GeV/c