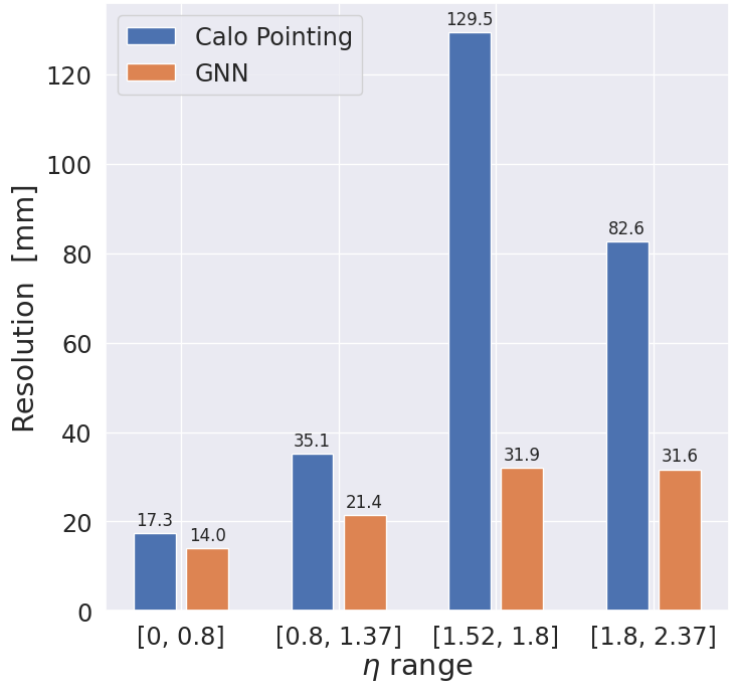
Improving Photon Pointing with Graph Neural Networks

Shikai Qiu, Haichen Wang

University of California, Berkeley
Supported by USATLAS

April 14, 2021

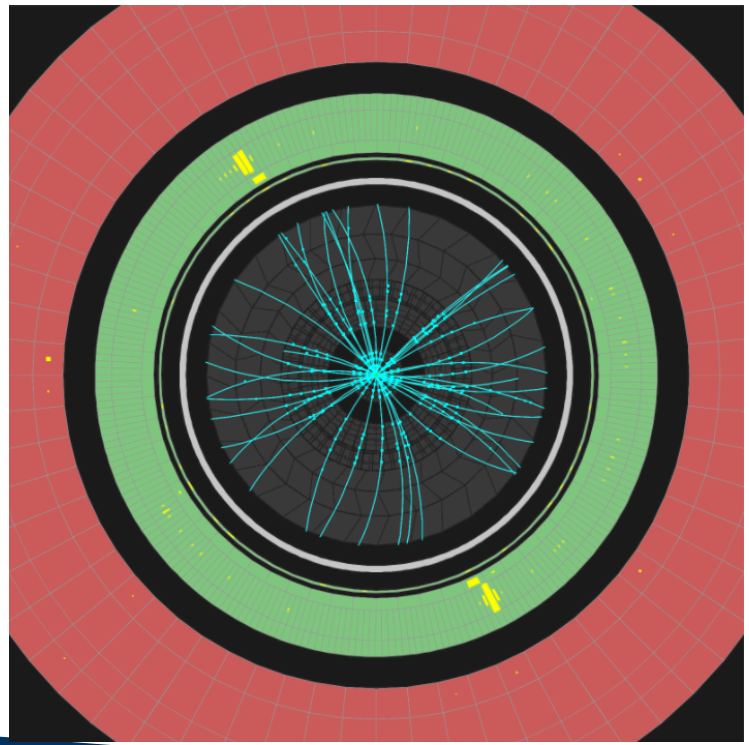
USATLAS SUPER Symposium



Berkeley
UNIVERSITY OF CALIFORNIA

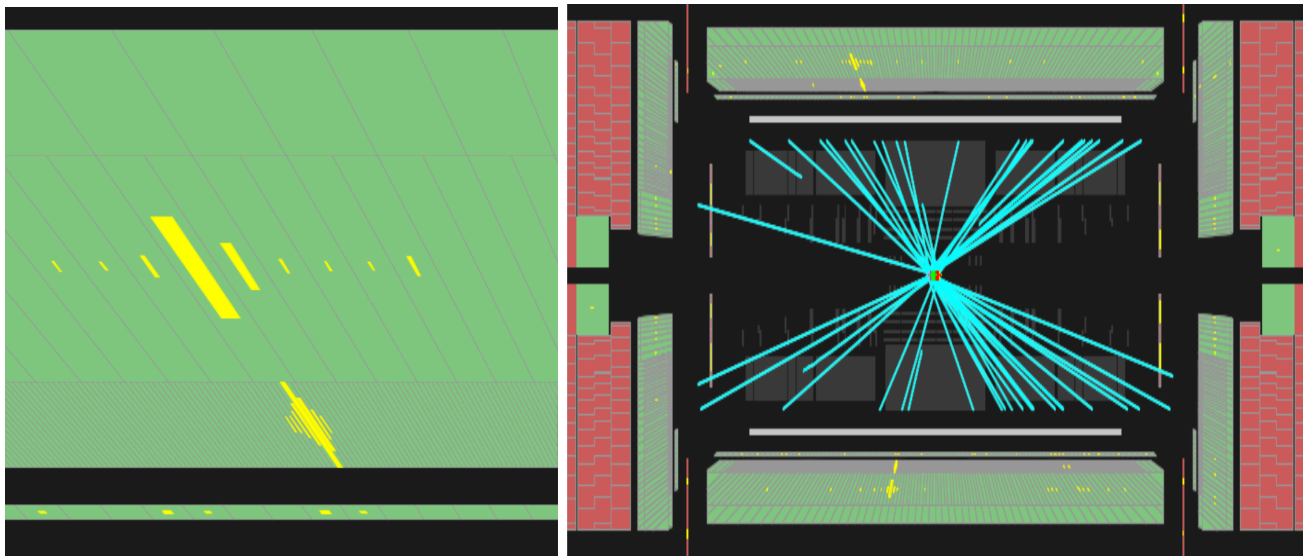
Motivation

- EM showers are naturally and faithfully represented by graphs, making GNN the candidate for a powerful performance tool in a wide range of tasks: energy calibration, photon direction measurement, e-γ identification, etc.
- We apply GNN to photon pointing as a proof of this general concept.



Background

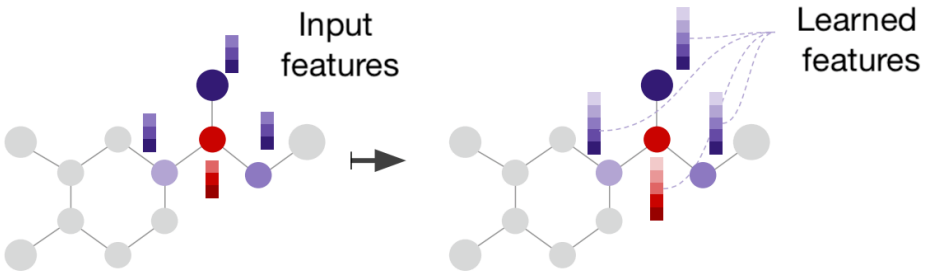
- **Photon Pointing:** measurement of the z-coordinate of a photon along the beamline.
- **Application:** measurement of photon direction, particularly important in $H \rightarrow \gamma\gamma$ studies; Search for non-pointing photons predicted by many BSM models; etc.



- **Current method:** measures centroids of the EM shower in the first and second EM calorimeter layers, and connect the dots.

Graph Neural Networks

- A graph G consists of nodes $X = \{x_i\}$ and edges $\mathcal{E} = \{e_{ij}\}$.
- A GNN models a function from a graph G to some output y by
  1. Propagate and process information locally over the graph, e.g:



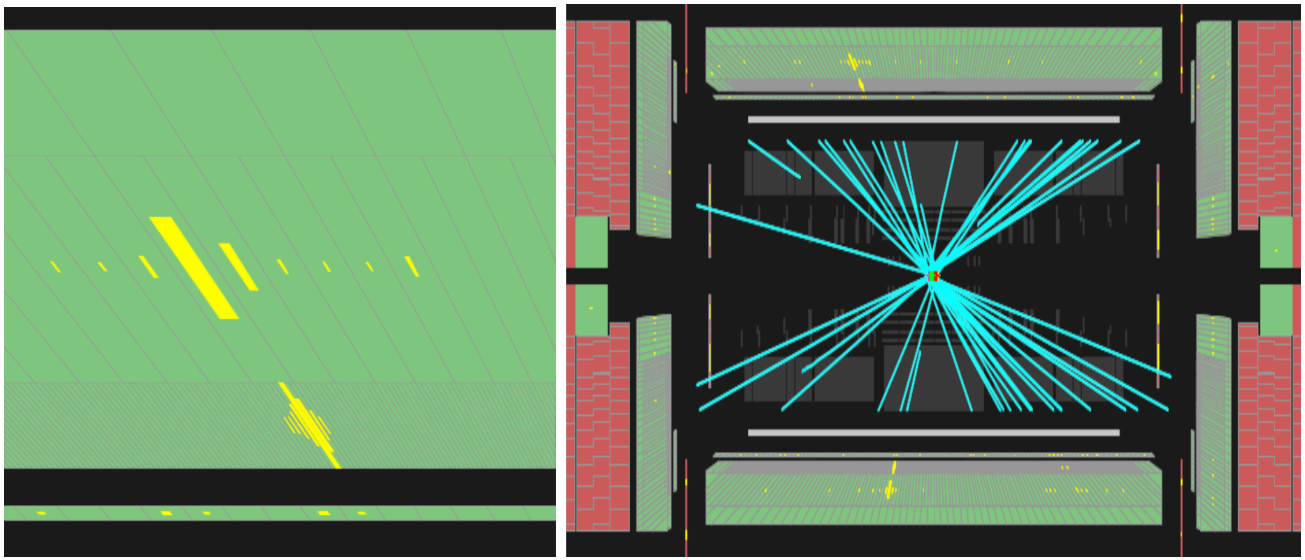
$$x_i \leftarrow \text{NodeUpdate}_{\text{NN}}(x_i, \{x_j, e_{ij}\}_{j \neq i})$$
$$e_{ij} \leftarrow \text{EdgeUpdate}_{\text{NN}}(e_{ij}, x_i, x_j)$$

  2. Perform a readout operation, e.g:

$$y \leftarrow \text{Readout}_{\text{NN}}\left(\sum_i x_i\right)$$

Graph Representation of an EM Shower

The most faithful representation of a shower is a graph whose nodes represent individual cells in the calorimeter, with node features being the coordinate of the cell and the amount of energy deposit.



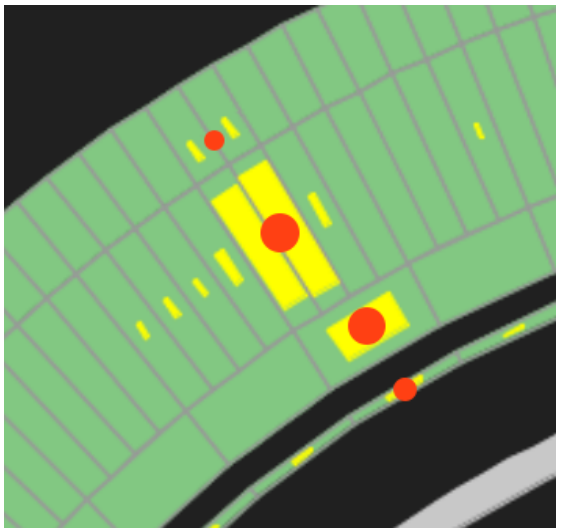
Each cell with non-zero energy deposit is a node in the graph

Graph Representation of an EM Shower

As a proof-of-principle project, we use a simpler graph with only 4 nodes, each representing a layer of the EM calorimeter including the pre-sampler. Each node is connected to each other.

- Node features: layer-index, spatial coordinates, energy deposit, and shower shape variables(rough descriptions of shower shape, e.g: width of the shower in η.)
- Edge features: relative separation

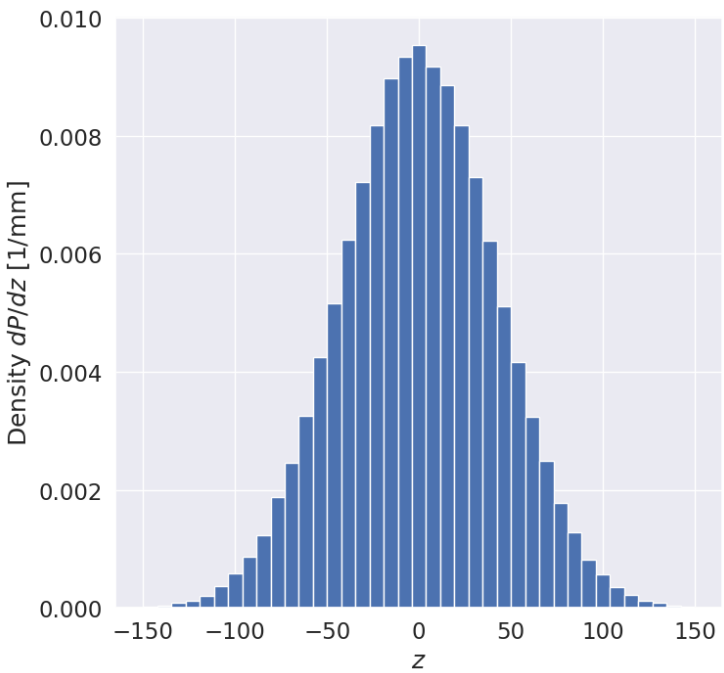
$$x_i = (i, \eta_i, \phi_i, E_i, s_i)$$
$$e_{ij} = (\eta_i - \eta_j, \phi_i - \phi_j)$$

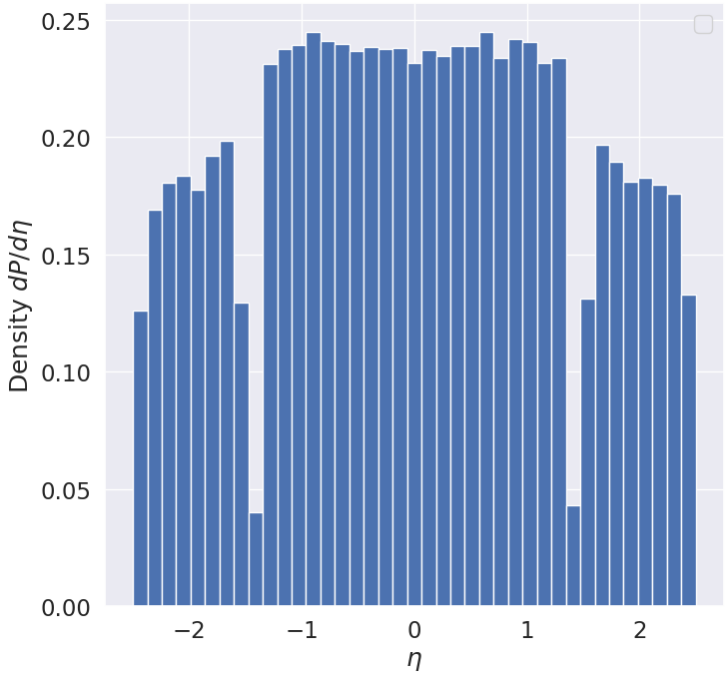


A shower represented by a graph of 4 nodes, edges not shown.

Dataset

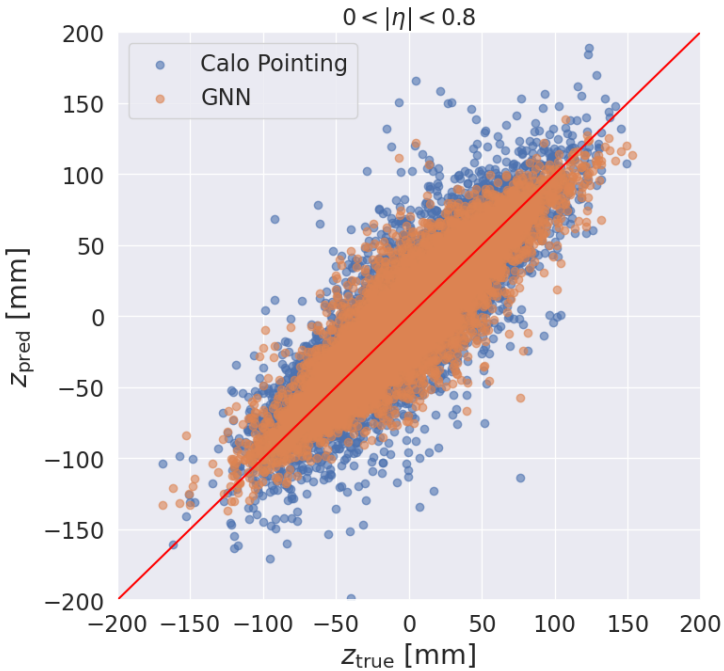
We construct a dataset of 23.4 million electron shower events from a Monte Carlo(MC) Simulated $Z \rightarrow ee$ dataset. The dataset is split at random into training, validation, and testing set with a ratio of 8:1:1. z roughly follows a Gaussian distribution with zero mean and $\sigma \approx 40$mm.



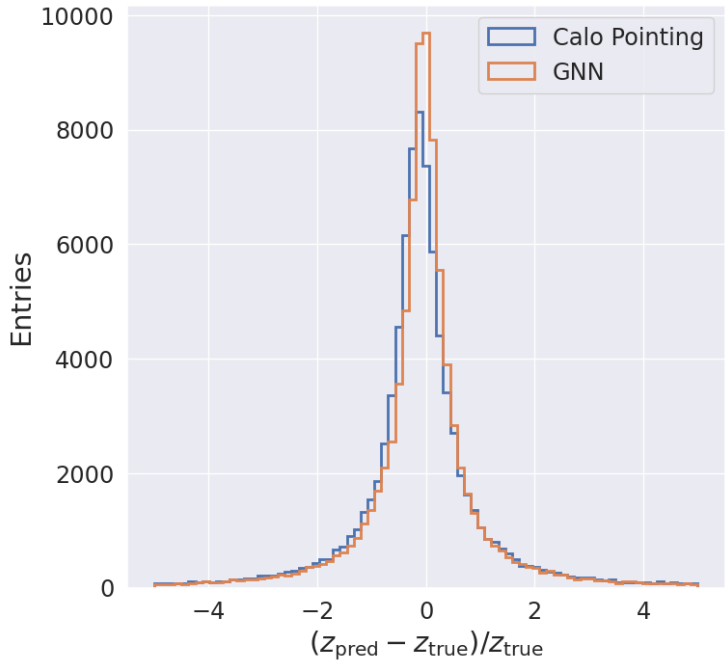


Prediction and resolution

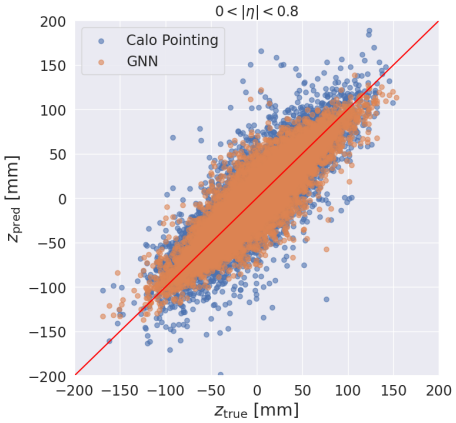
- GNN significantly improves the resolution, i.e, average prediction error.



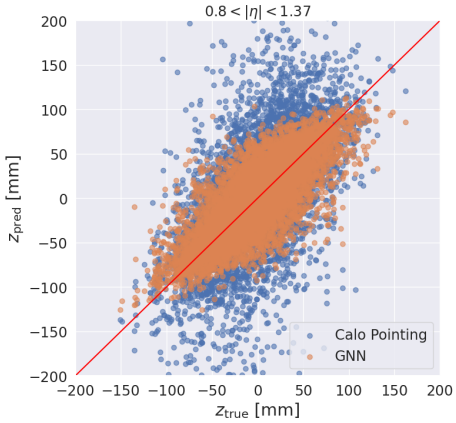
Prediction v.s Truth, $|\eta| \in [0, 0.8]$



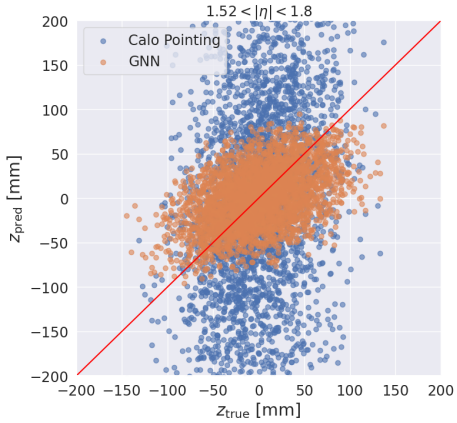
Pull plot, $|\eta| \in [0, 0.8]$



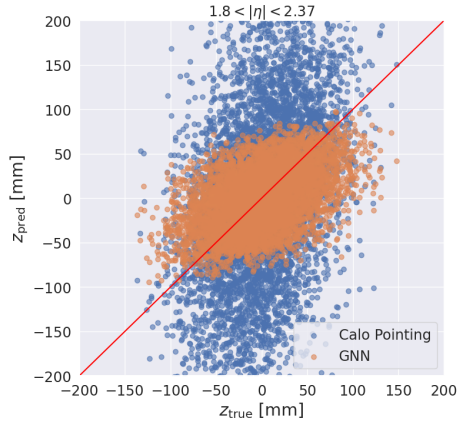
$|\eta| \in [0, 0.8]$



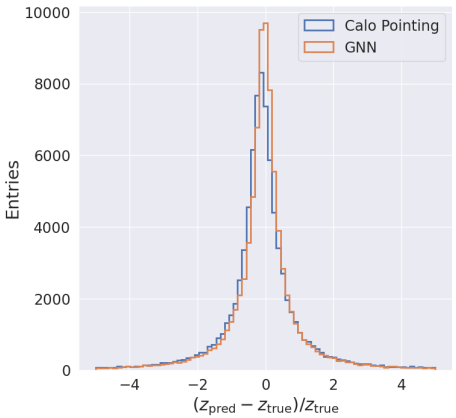
$|\eta| \in [0.8, 1.37]$



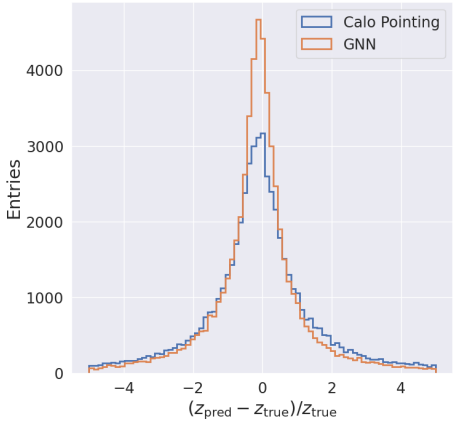
$|\eta| \in [1.52, 1.8]$



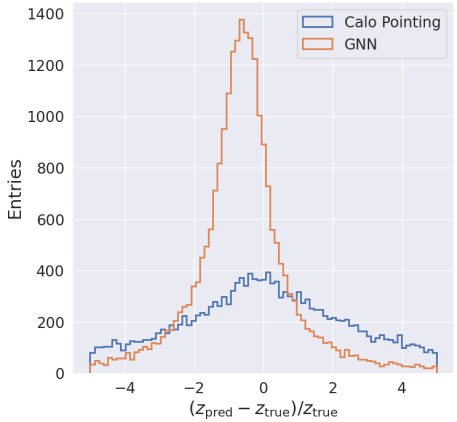
$|\eta| \in [1.8, 2.37]$



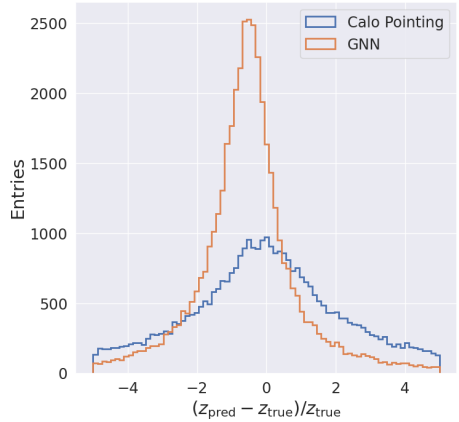
$\text{med}_{\text{GNN}} = -5\%$
$\text{med}_{\text{Calo}} = -13\%$



$\text{med}_{\text{GNN}} = -12\%$
$\text{med}_{\text{Calo}} = -11\%$



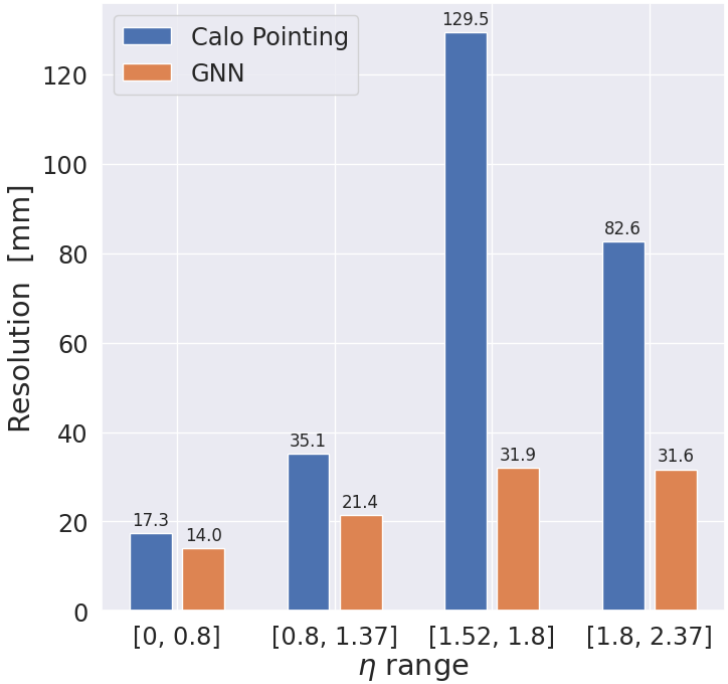
$\text{med}_{\text{GNN}} = -63\%$
$\text{med}_{\text{Calo}} = -0.08\%$

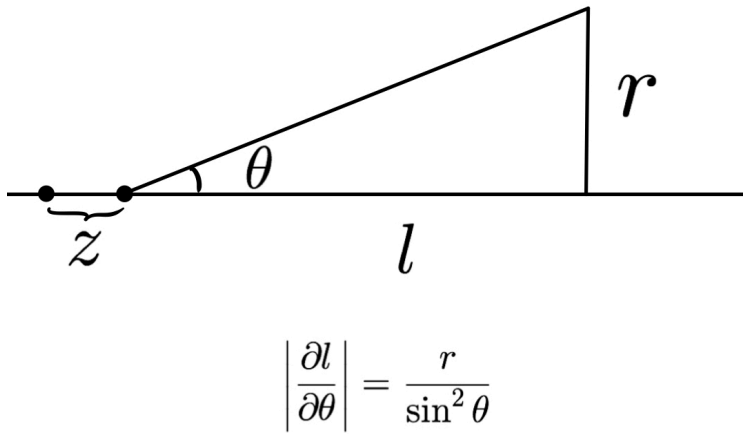


$\text{med}_{\text{GNN}} = -56\%$
$\text{med}_{\text{Calo}} = -6\%$

As $|\eta|$ increases, resolution becomes worse for both models, and GNN develops a bias.
med = median.

Interpretation: resolution



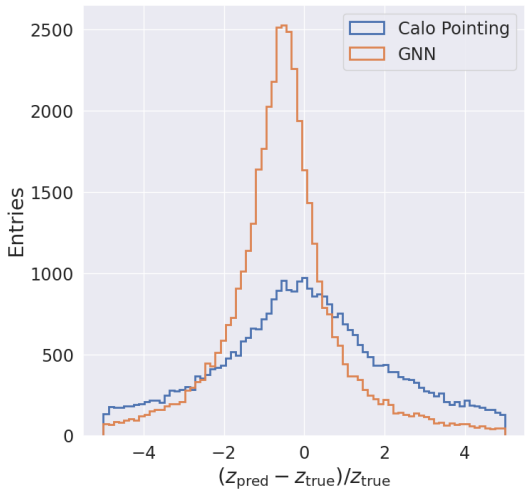


$$\left|\frac{\partial l}{\partial \theta}\right| = \frac{r}{\sin^2\theta}$$

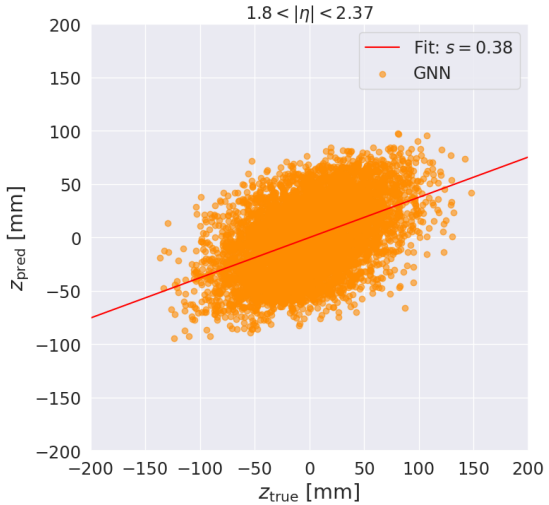
Prediction in the forward region is intrinsically more difficult as z becomes more sensitive to angular resolution.

Interpretation: bias

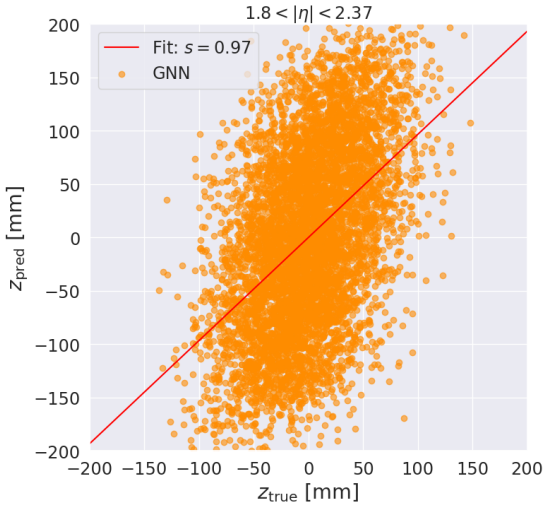
- The bias is an artifact of the chosen mean-squared-error objective function. We are working to find a better objective function, but we can already correct for the bias.
- We fit the slope $z_{\text{pred}} \approx \text{slope} \times z_{\text{true}}$ in different η bins.
- For each prediction z_{pred}, we look up its η value and determine the slope s in that bin.
- Scale the prediction with the inverse slope $z_{\text{pred}} \leftarrow z_{\text{pred}}/s$.



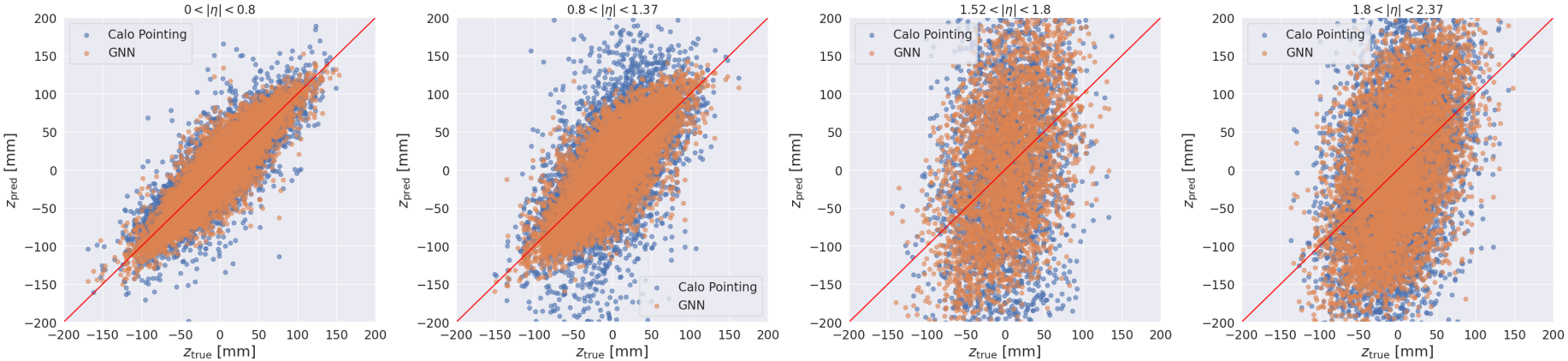
$|\eta| \in [1.8, 2.37]$
$\text{med}_{\text{GNN}} = -56\%$

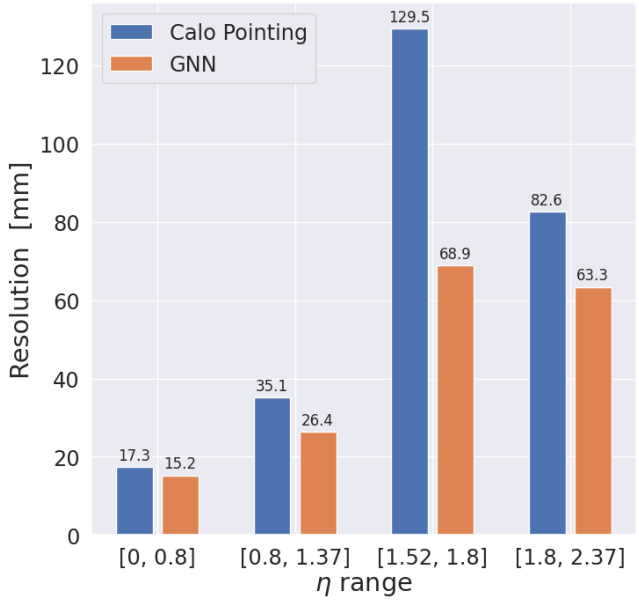


Before correction



After correction





GNN still achieves better resolution after correction.

Conclusion

- Graphs are well suited to represent EM showers, making GNN a powerful performance tool in a wide range of tasks: energy calibration, photon direction measurement, e-γ identification, etc.
- As a proof of principle, we applied GNN to photon pointing.
- We show that GNN improves the resolution over the existing method, and its current bias can be corrected.
- Future directions include using a higher-resolution graph representation of the shower, e.g, where nodes are individual cells in the calorimeter.

Acknowledgements

- Thanks to USATLAS and the USATLAS SUPER Program for funding and supporting this research.
- Thanks to my advisor Professor Haichen Wang for his guidance and support.
- Thanks to my collaborators, Jake Austin, Shuo Han, Xiangyang Ju, and Ryan Roberts, for their support and insightful discussions.

Back up

Definition of resolution

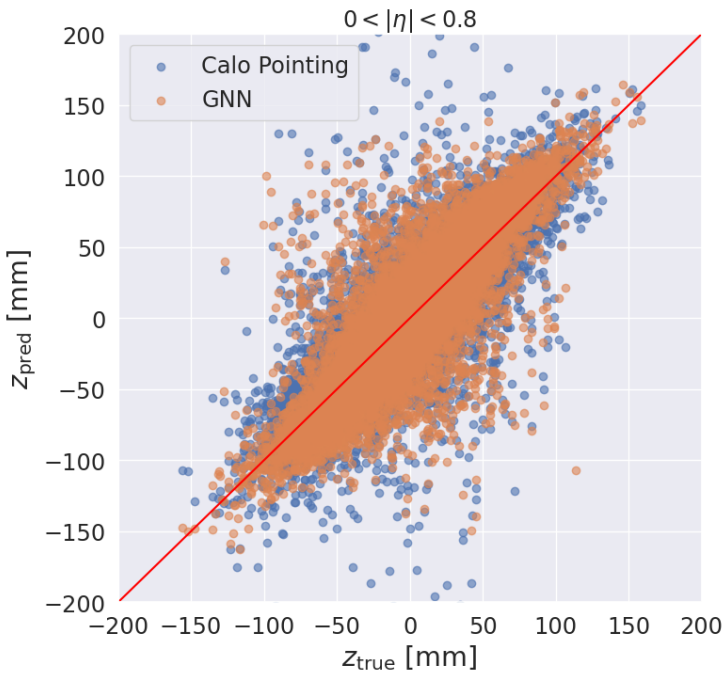
We define resolution by the mean absolute error

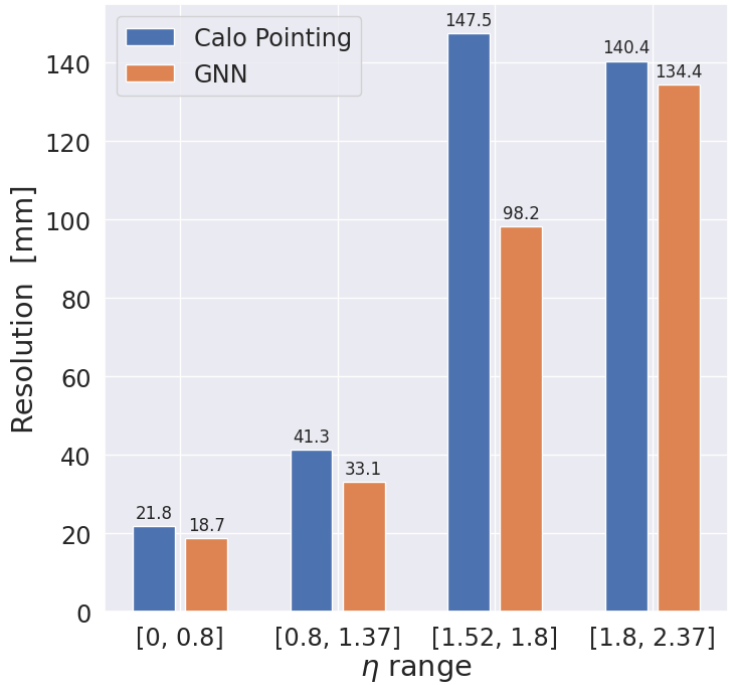
$$\text{Resolution} = \frac{1}{N}\sum_{i=1}^{N} |z_{\text{pred}}^{(i)} - z_{\text{true}}^{(i)}|, \tag{1}$$

where the sum is over the test sample.

The model does work for photons

We apply the trained model on MC simulated $Z \rightarrow ee\gamma$ events to predict pointing value of photons. After similar bias correction, the model still outperforms calo pointing.





Why there is a bias and how we can reduce it

The bias is largely an effect of the chosen loss function. Let x denote the input to the model(the shower graph). The model minimizes the loss function defined by the average squared error between its prediction and truth over the training set,

$$\mathcal{L} = \mathbb{E}_{(x,z)\sim p(x,z)}\left[(f(x) - z)^2\right], \tag{2}$$

where f denotes the map from input x to the model's prediction for z. The global minimum of the loss function is obtained by setting $\delta\mathcal{L}/\delta f = 0$.

$$\begin{aligned}
\delta\mathcal{L} &= \mathbb{E}_{x\sim p(x)}\left[\mathbb{E}_{z\sim p(z|x)}\left[\delta(f(x) - z)^2\right]\right] && (3)\\
&= 2\mathbb{E}_{x\sim p(x)}\left[\mathbb{E}_{z\sim p(z|x)}\left[\delta f(x)(f(x) - z)\right]\right] && (4)\\
&= 2\mathbb{E}_{x\sim p(x)}\left[\delta f(x)(f(x) - \mathbb{E}_{z\sim p(z|x)}[z])\right] && (5)\\
& && (6)
\end{aligned}$$

This shows $\delta\mathcal{L}/\delta f = 2p(x)(f(x) - \mathbb{E}_{z\sim p(z|x)}[z])$. Therefore the optimal solution f^* is given by $f^*(x) = \mathbb{E}_{z\sim p(z|x)}[z]$, the conditional expectation value of z given x.

Why there is a bias and how we can reduce it

- Since $p(z|x) \propto p(z)p(x|z)$, when the input information x is compatible with a large range of z, which can happen when $|\eta|$ is large, the model would predict an average value of z determined by $p(z)$, the prior z-distribution, and $p(x|z)$, which captures the kinematics and the detector effects such as noise. $p(z) \propto \exp\left(-z^2/2\sigma_z^2\right)$ suppresses large $|z|$ contribution exponentially in $z^2/2\sigma_z^2$, which biases the model to predict low values. We alleviated this effect by re-weighting the training examples so that z follows a uniform distribution within ± 80mm.
- On the other hand, we have no control over the $p(x|z)$ term. For example, in the very forward region, the finite resolution of the photon/electron direction of travel translates to an increasingly large resolution on its z-coordinate. There x will have little mutual information with z and $p(x|z)$ will approach a constant for all z. Then $f^*(x) = 0$ since $p(z)$ is symmetric. Clearly, if we can supply the model with input that has higher mutual information with the true z, e.g: using calorimeter cells as nodes, the bias can be further reduced. Finally, an alternative way to reduce the bias is to change the loss function, which is yet to be explored.

Model architecture

The model can be divided into an encoder and a decoder . The encoder starts with an embedding layer which maps each node and edge raw feature vector into a latent representation $x_i, e_{ij} \in \mathbb{R}^H$, followed by multiple encoder blocks with separate learnable parameters, where each block updates the latent representation of the nodes and edges via

$$x_i' = \mathbf{FeedForward} \circ \mathbf{Attention}(x_i, \{(x_j, e_{ij})\}_{j \neq i}), \tag{7}$$

$$e_{ij}' = e_{ij} + \mathbf{MLP} \circ \mathbf{LayerNorm}(e_{ij}, x_i', x_j'), \tag{8}$$

where $\mathbf{FeedForward}$ is a feed-forward network with a residual connection, MLP is a Multilayer Perceptron, and $\mathbf{Attention}$ is the self-attention operation that routes information from the rest of the graph to the node i. The encoder is very similar to the Transformer encoder architecture, except that it's extended to incorporate edge information. We found using edge features and performing edge updates to be a crucial component of the model. Without it, the model takes much longer to train and achieves worse performance.

Model architecture

Likewise, the decoder consists of several blocks. Each block performs a pooling operation over the latent representations of the nodes and egdes produced by the encoder and updates an output latent state $h \in \mathbb{R}^H$ via

$$\tilde{x} = \mathrm{Affine}(\mathrm{Pool}_X(\{x_i\})), \tag{9}$$

$$\tilde{e} = \mathrm{Affine}(\mathrm{Pool}_E(\{e_{ij}\})), \tag{10}$$

$$h' = \mathrm{FeedForward}(h + \tilde{x} + \tilde{e}), \tag{11}$$

where Affine represents an affine transformation, and h is initialized to the zero vector. We use Global Attention Pooling for both Pool_X and Pool_E. Finally, an MLP is used to map the final output latent state h into a pointing prediction $\hat{z} = \mathrm{MLP}(h)$.