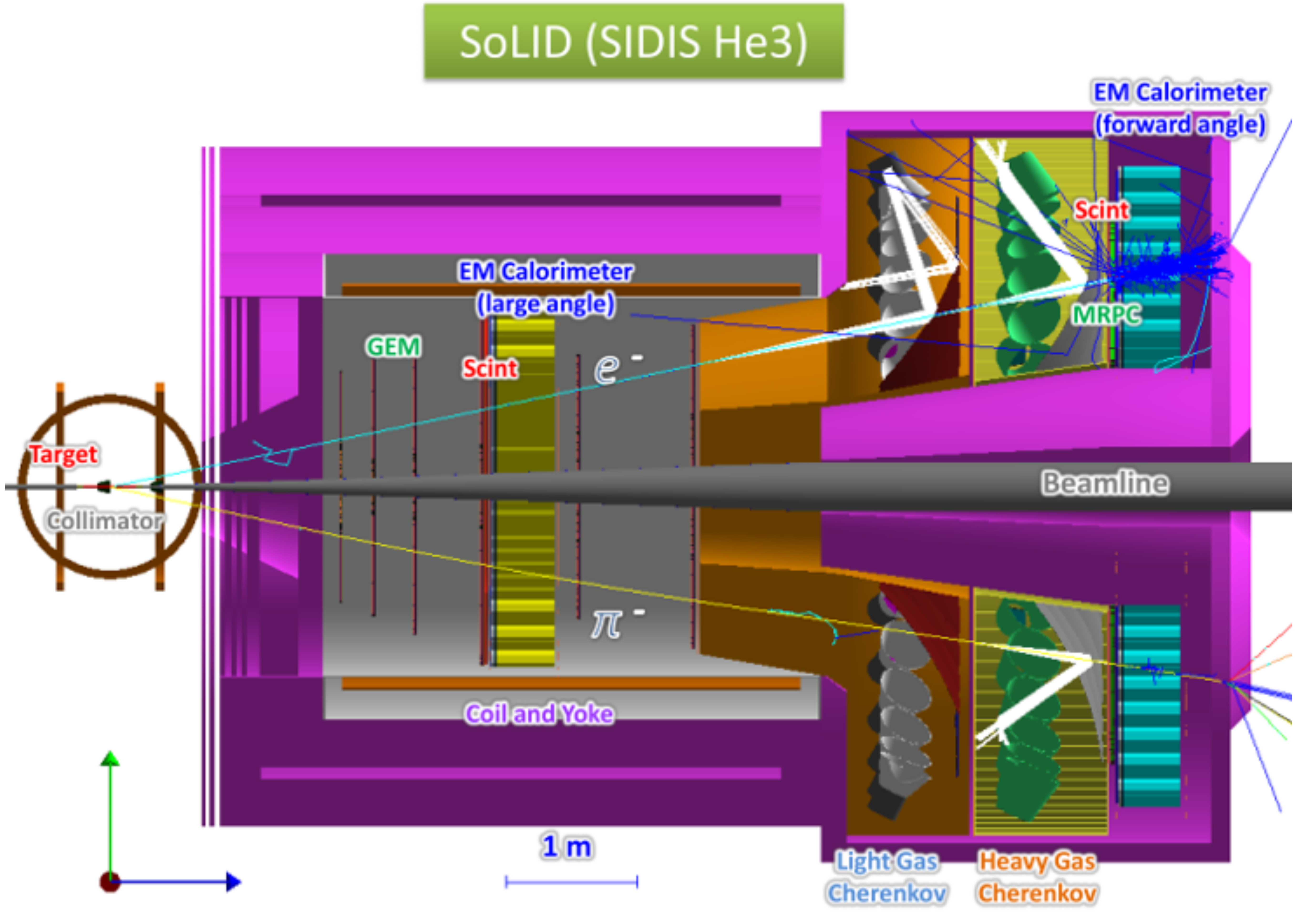
AI/ML For SoLID Tracking

Tracking with Artificial Intelligence

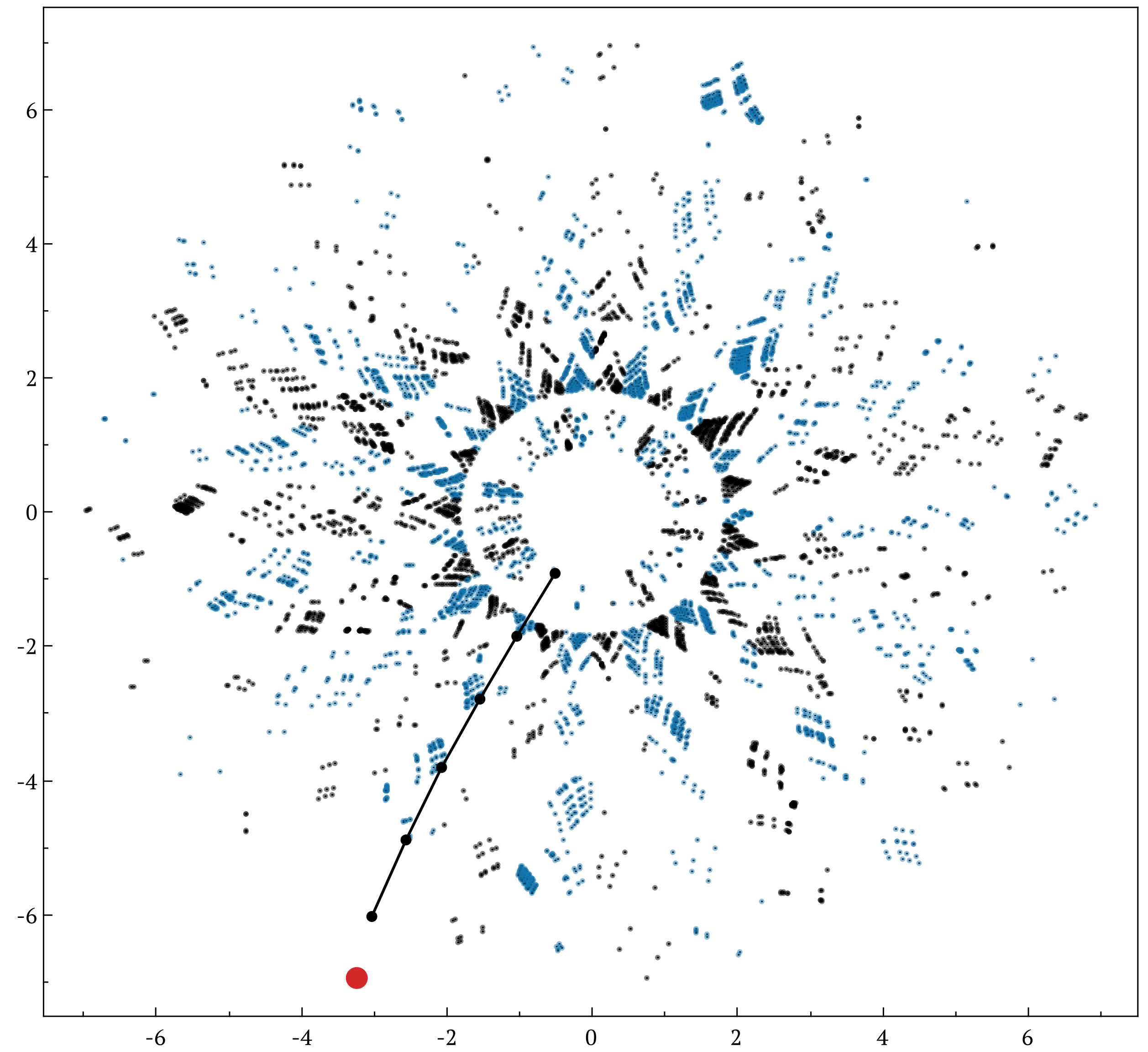
Gagik Gavalian (Thomas Jefferson National Lab)

SOLENOIDAL LARGE INTENSITY DEVICE

SoLID Detector

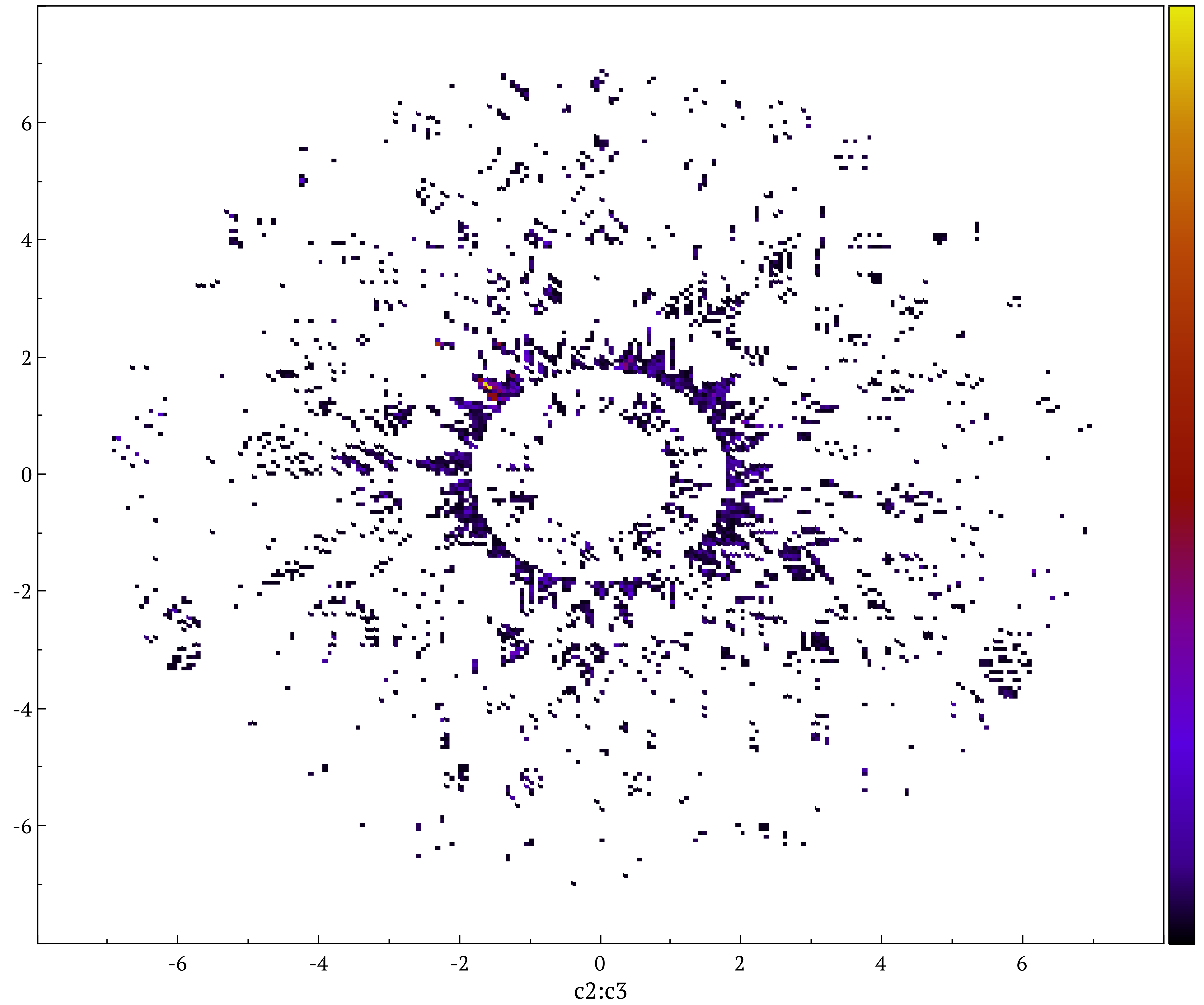


GEM Trackers:
Six planes of GEM trackers, each plane consists of 2 stereo views. The azimuthal space is divided into 30 chambers.



Event:
Track generated in simulation with background overlaid.

The task



The Task:

Find 6 hits that belong to a track

~7200 points (6 of them come from a simulated track)

There is one generated track in this event.

Extract track parameters for the identified track (i.e. momentum and direction)

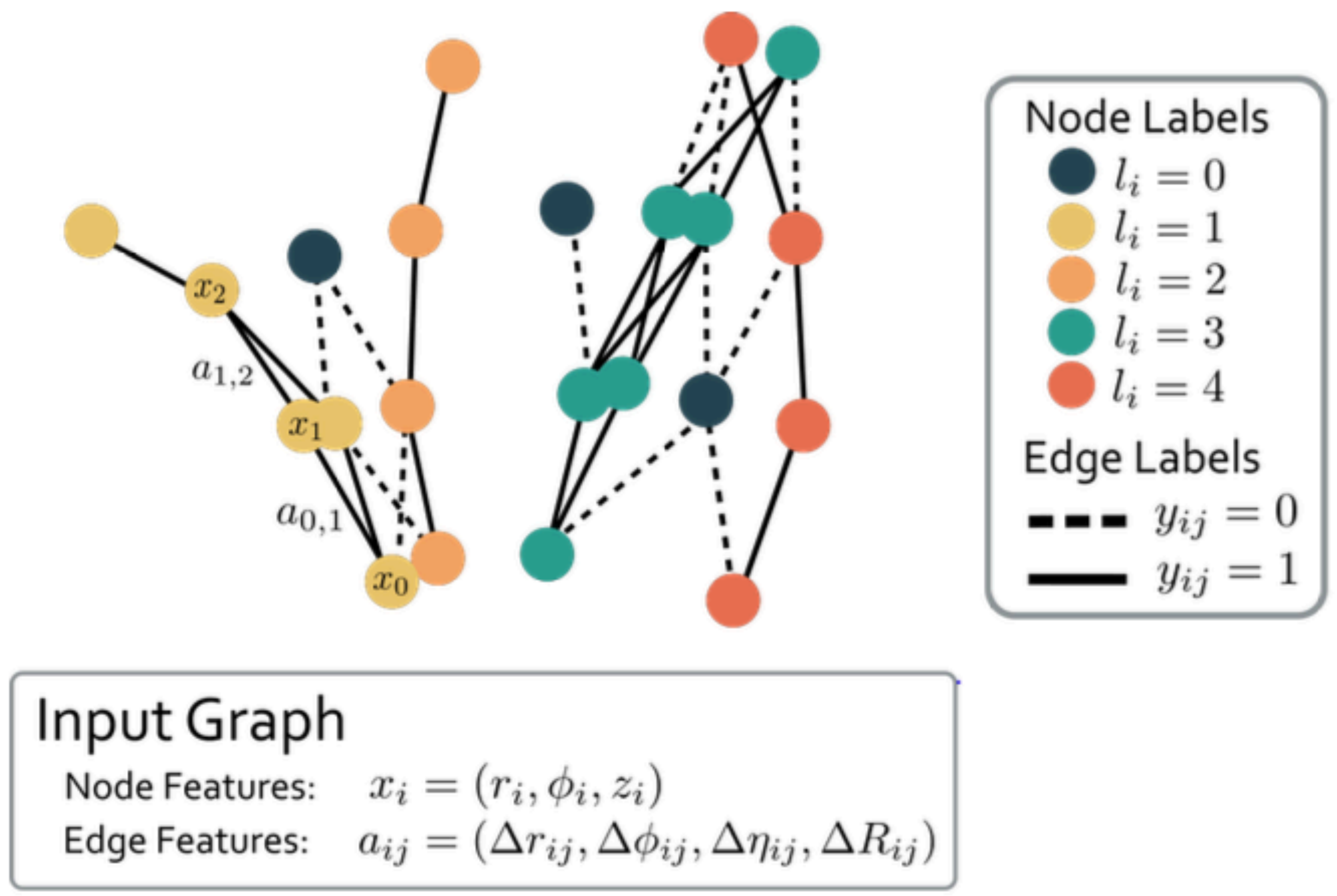
Strategy

Previous Experience:
Successful track reconstruction using AI/ML was developed in CLAS12; the developments were made 7 years ago. There were not many implementations of track reconstruction with AI. Uses convolutional neural networks (Complicated to implement and integrate), uses separate networks for denoising, classification, and track parameter extraction.

New methods:
In the past 5 years, there have been significant developments in HEP to reconstruct tracks using Graph Neural Networks (GNN). One-shot models: feed all points in the tracking detector and out-track candidates and parameters.
Will be easier to integrate with the EIC software.

Current stage of developments

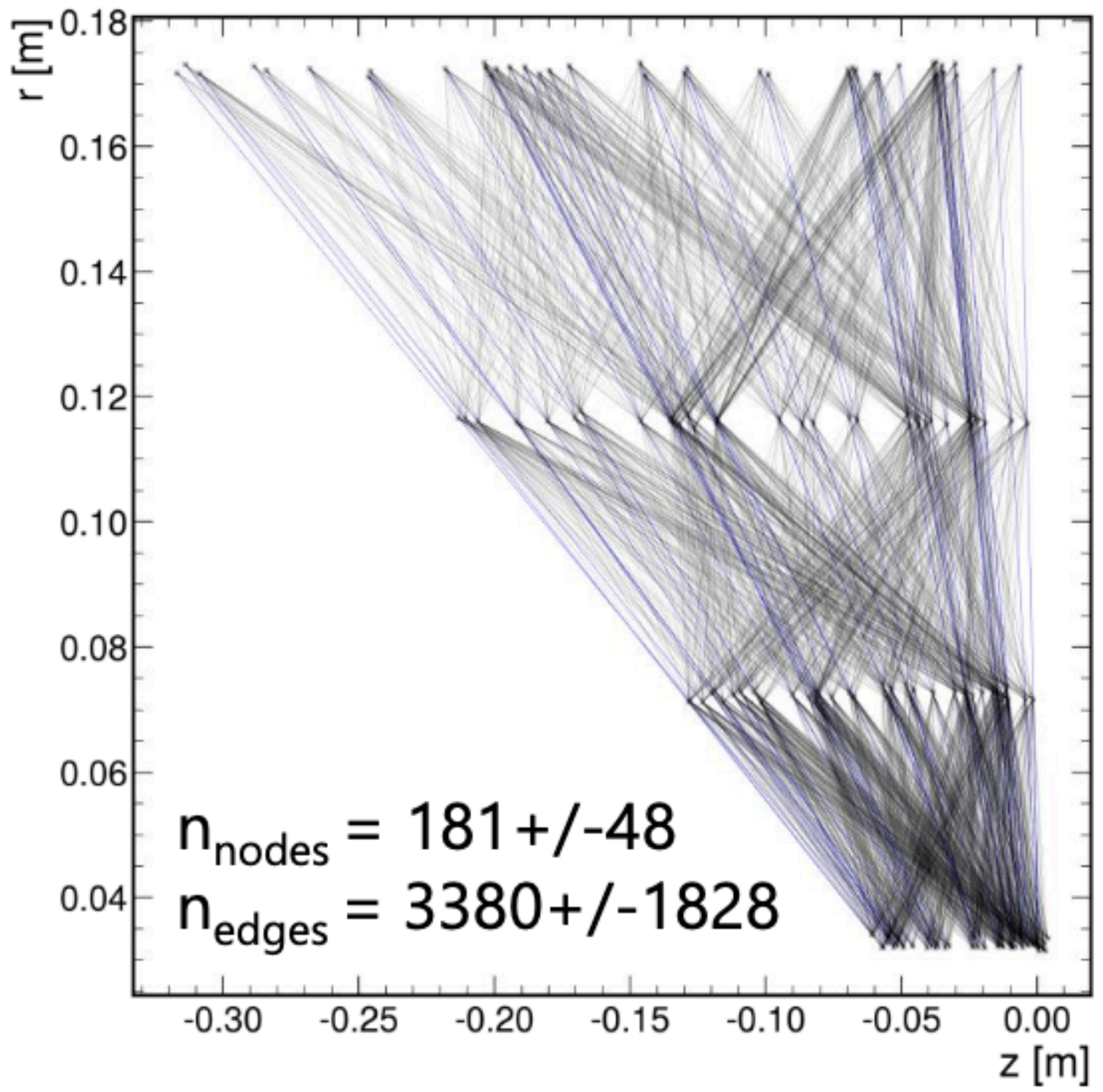
GravNet

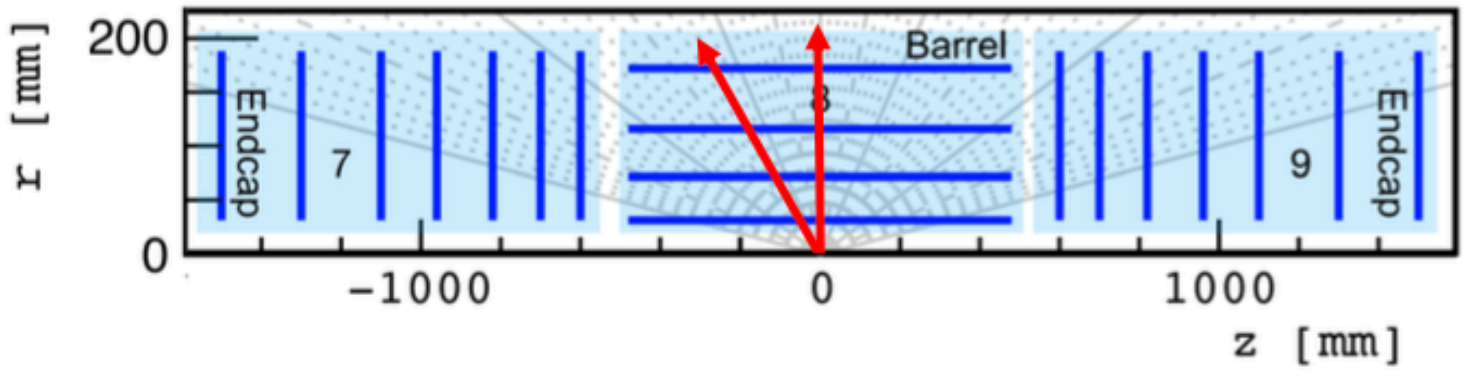


Example GNN-based tracking workflow

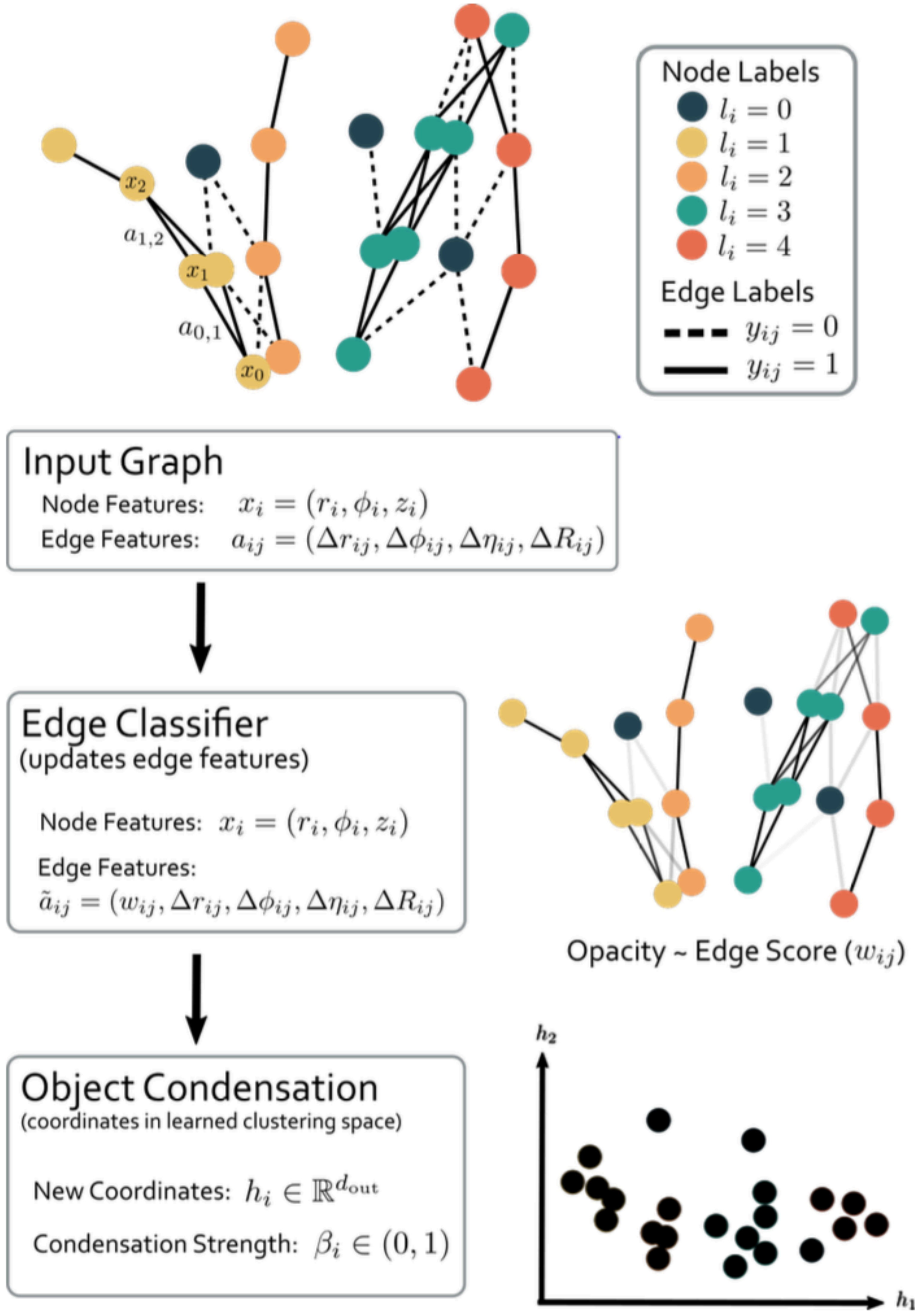
1) Graph Construction

Draw edges between particles detector hits based on some initial constraints, clustering, etc.



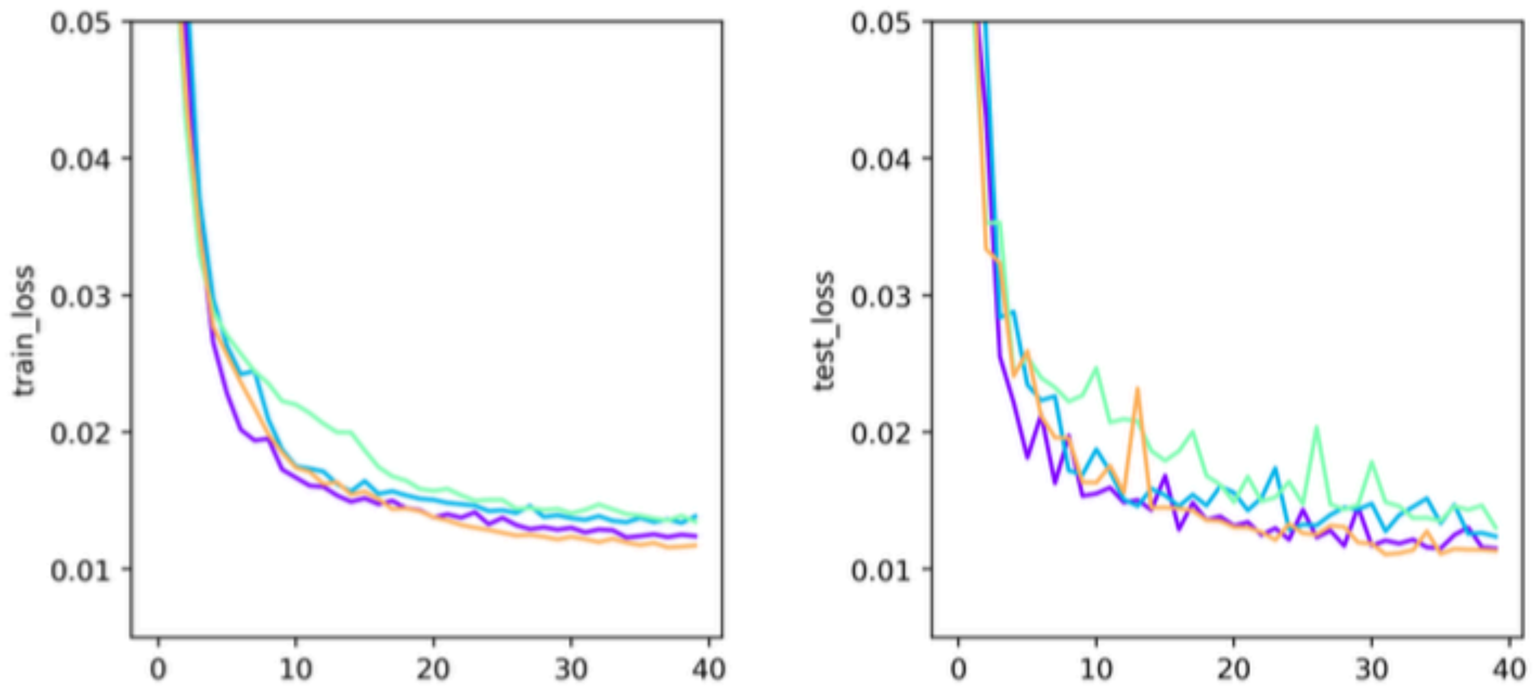


GravNet

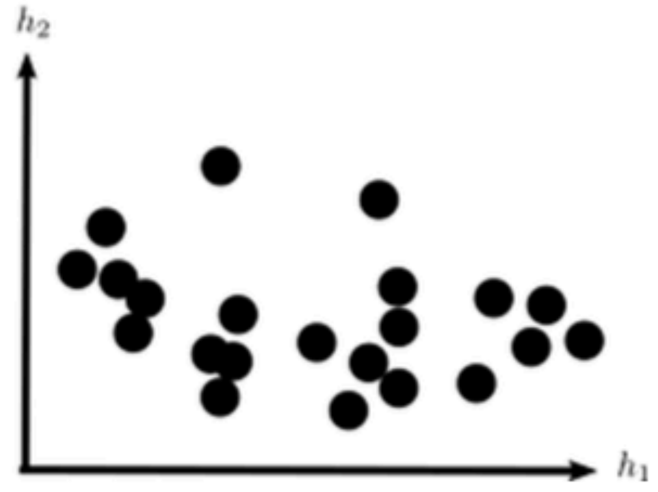


2) GNN Inference

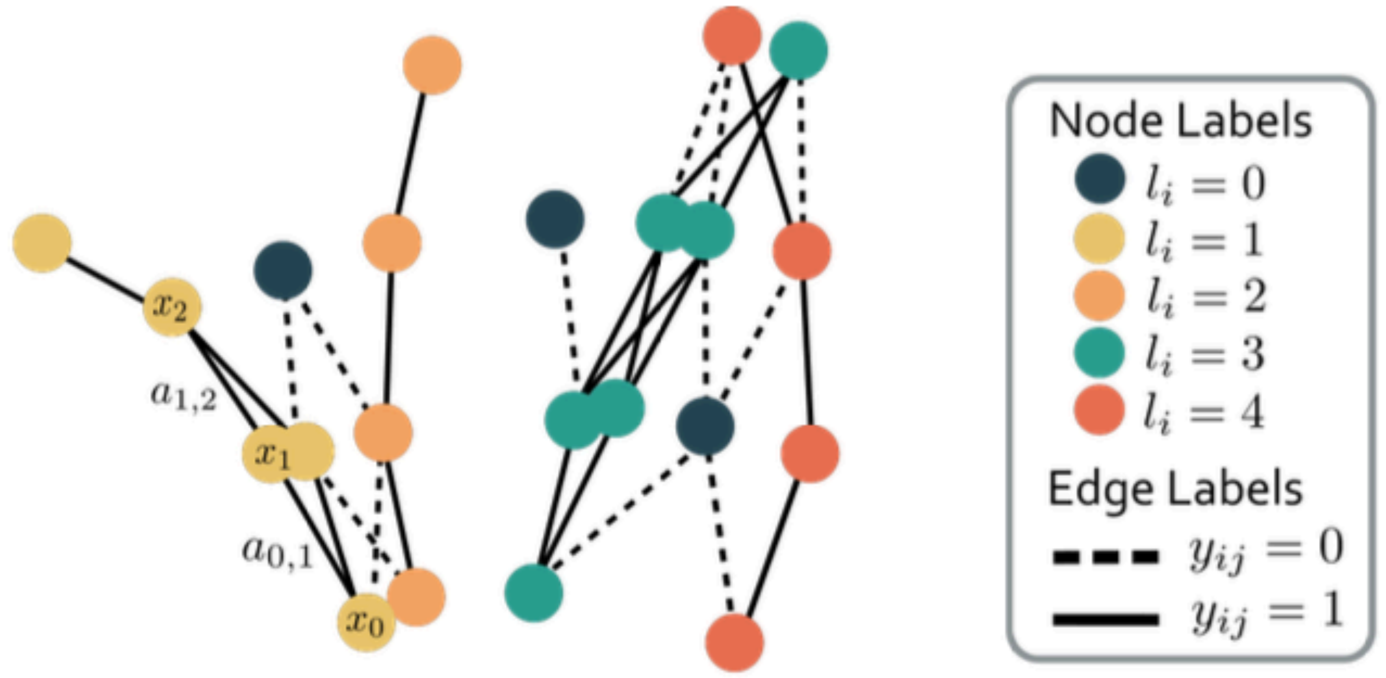
Train a GNN to classify the edges (binary cross entropy) **and** cluster signals belonging to the same particle (object condensation)



Output: track hit coordinates in a learned clustering space



GravNet



Input Graph

Node Features: $x_i = (r_i, \phi_i, z_i)$

Edge Features: $a_{ij} = (\Delta r_{ij}, \Delta\phi_{ij}, \Delta\eta_{ij}, \Delta R_{ij})$

↓

Edge Classifier
(updates edge features)

Node Features: $x_i = (r_i, \phi_i, z_i)$

Edge Features:
$\tilde{a}_{ij} = (w_{ij}, \Delta r_{ij}, \Delta\phi_{ij}, \Delta\eta_{ij}, \Delta R_{ij})$

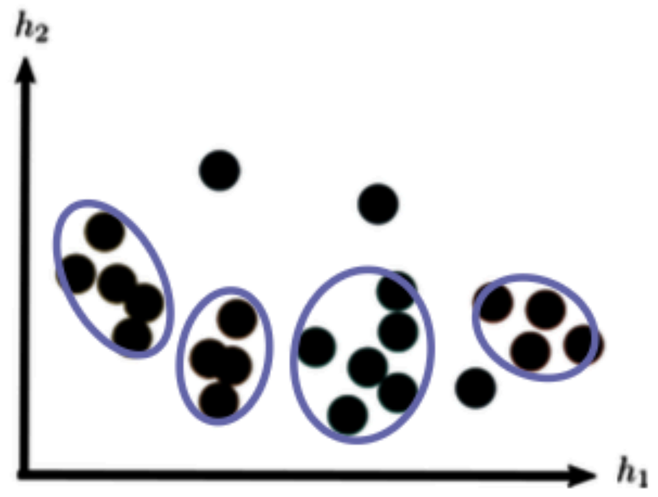
Opacity ~ Edge Score (w_{ij})

↓

Object Condensation
(coordinates in learned clustering space)

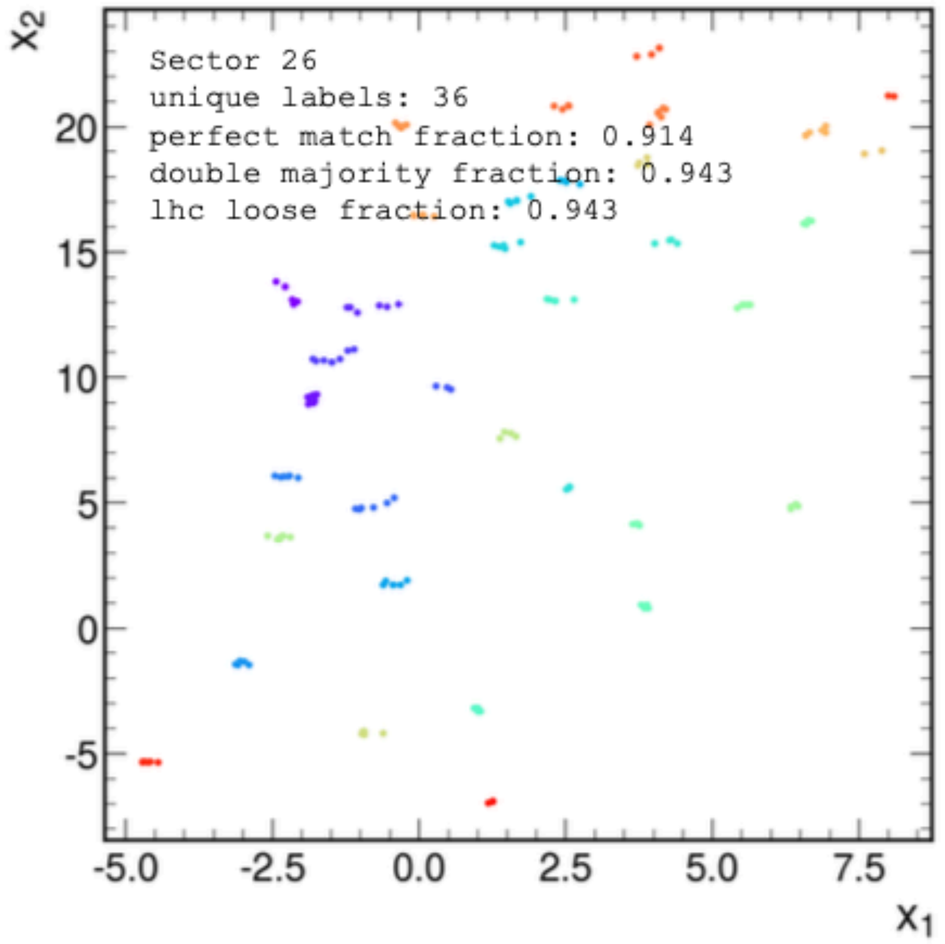
New Coordinates: $h_i \in \mathbb{R}^{d_{out}}$

Condensation Strength: $\beta_i \in (0, 1)$

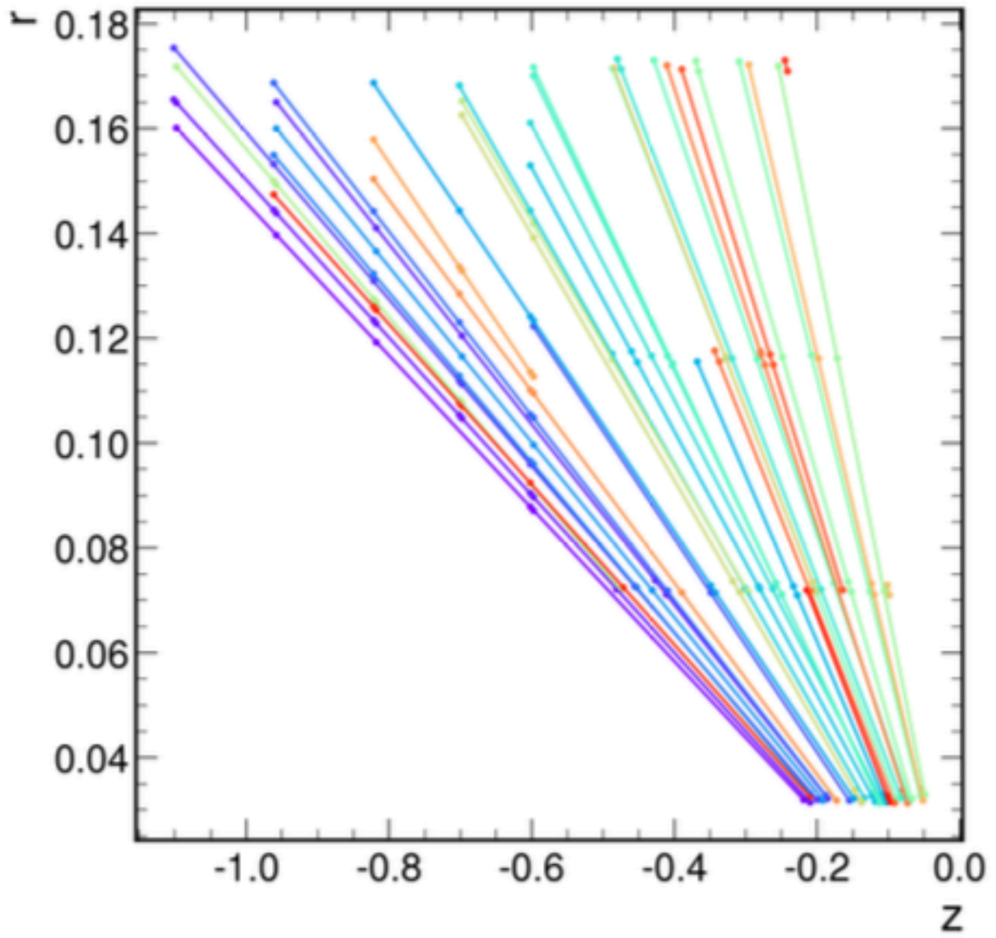


3) Postprocessing

Cluster hits in the learned coordinate space to form track candidates!

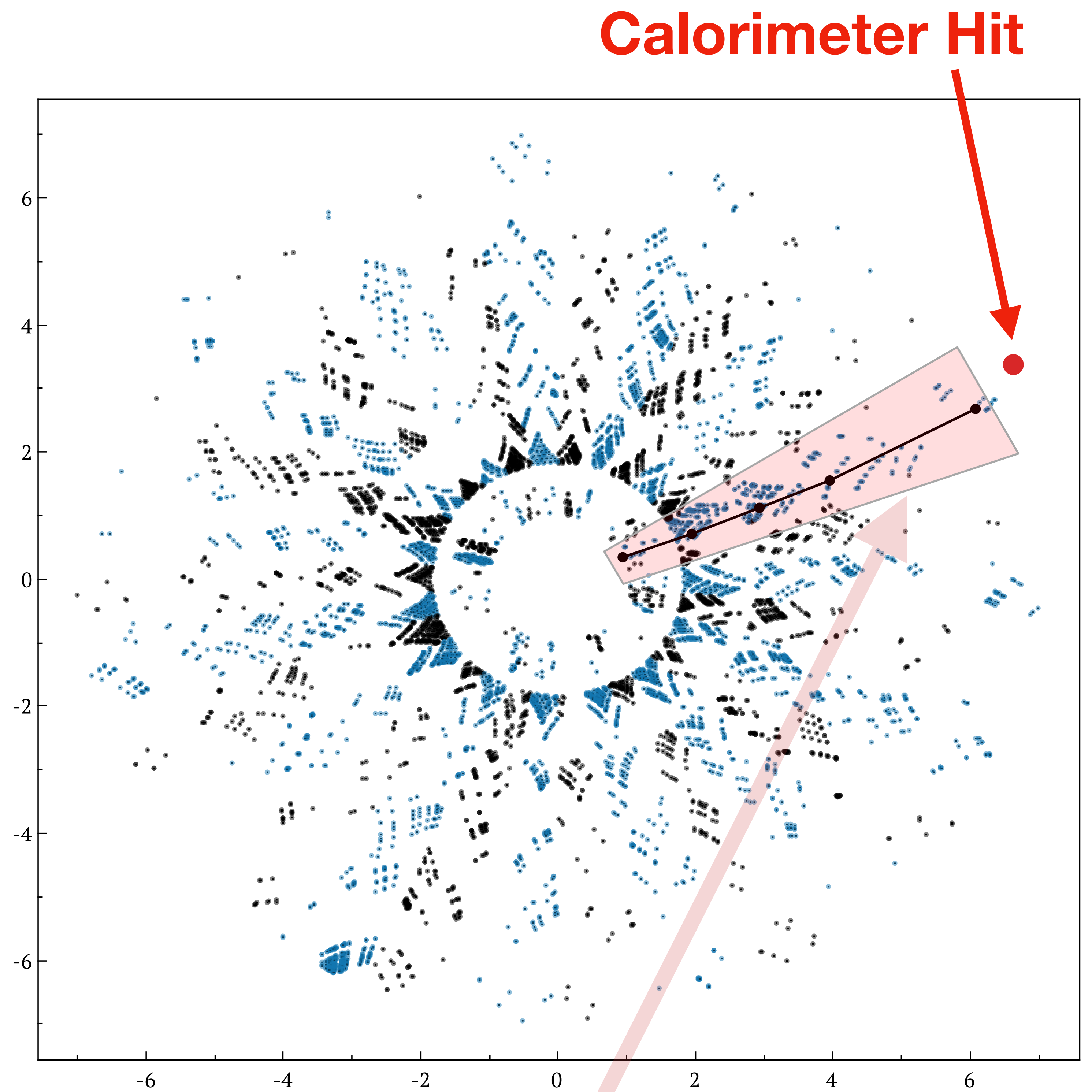


GNN outputs 2D coordinates for each hit



Colors are cluster labels generated by DBSCAN (not the GNN)

SoLID GNN (GravNet)



The Narrow region matching the position of the calorimeter hit is considered for training and track identification.

Selection of Region:
The region of possible track candidates is chosen based on the hit in the calorimeter.

Training:
The points in the selected region are passed to the networks with label “0” for background hits and “1” for hits belonging to the track.

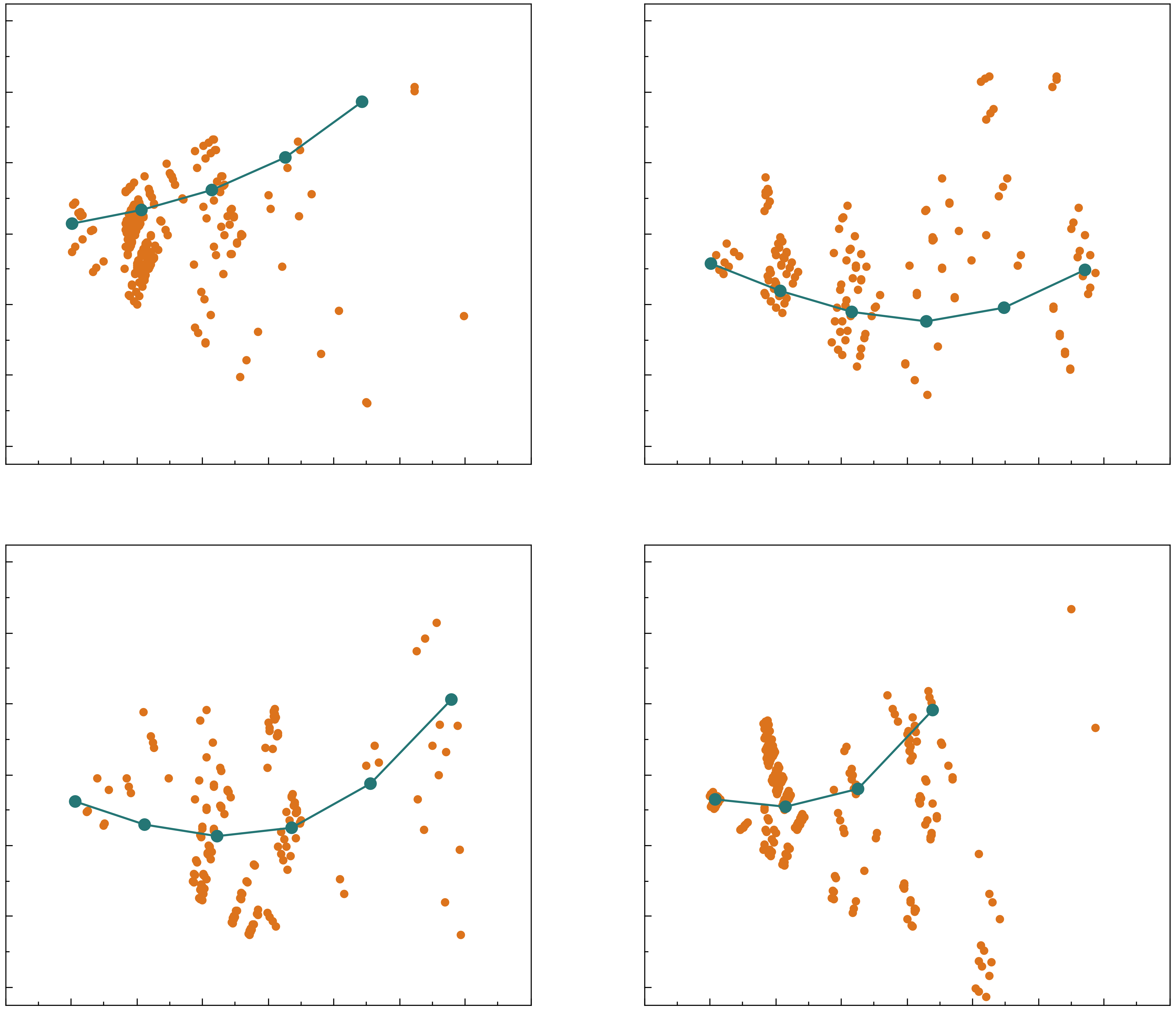
$$X = (x_i, y_i, z_i)$$

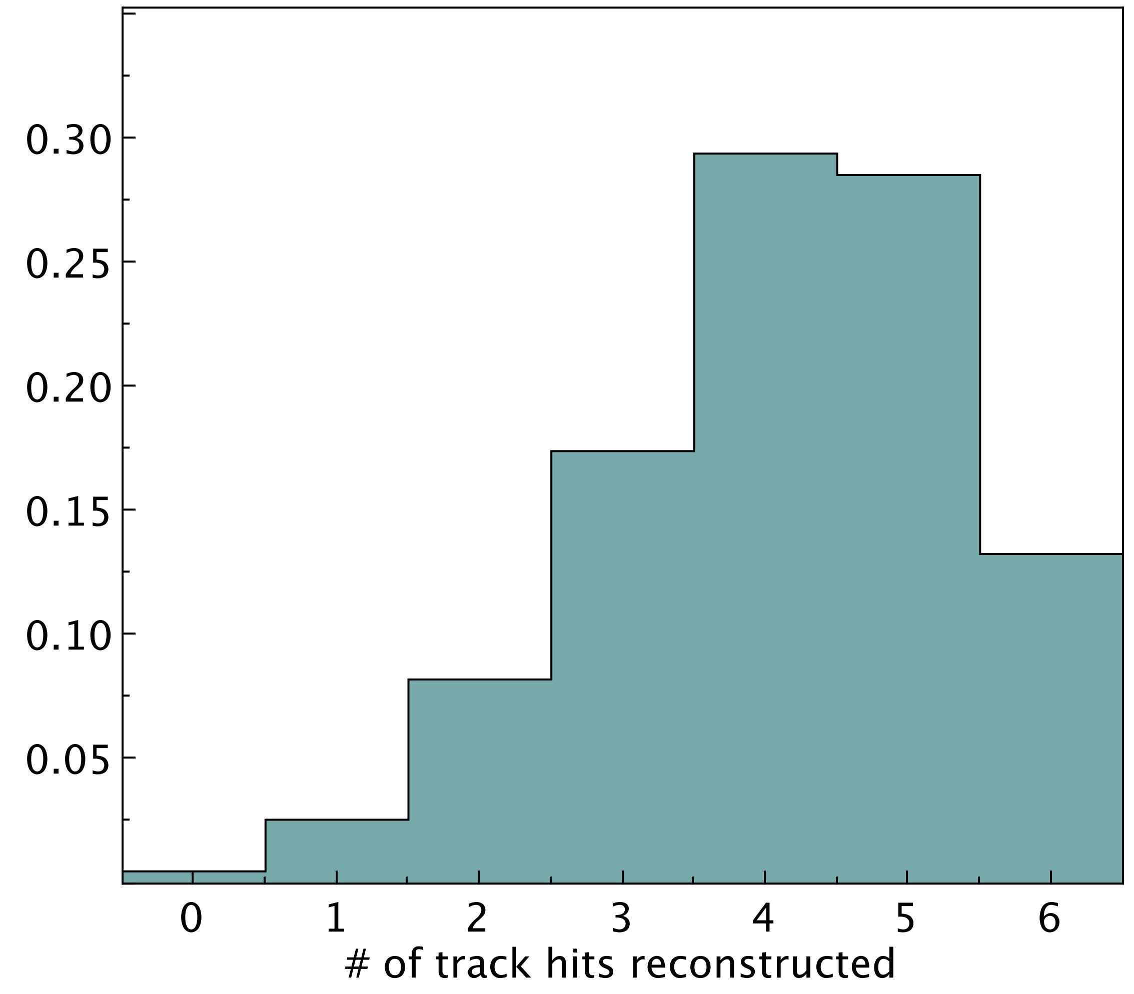
Future Improvements:

- Other features, such as charge and time, can be used in the future; they are currently not available in the data set.
- Currently, there is no rule on edge construction, which should be added.
- The current algorithm used is the K-means clustering. DBSCAN claims better performance

SoLID GNN (GravNet)

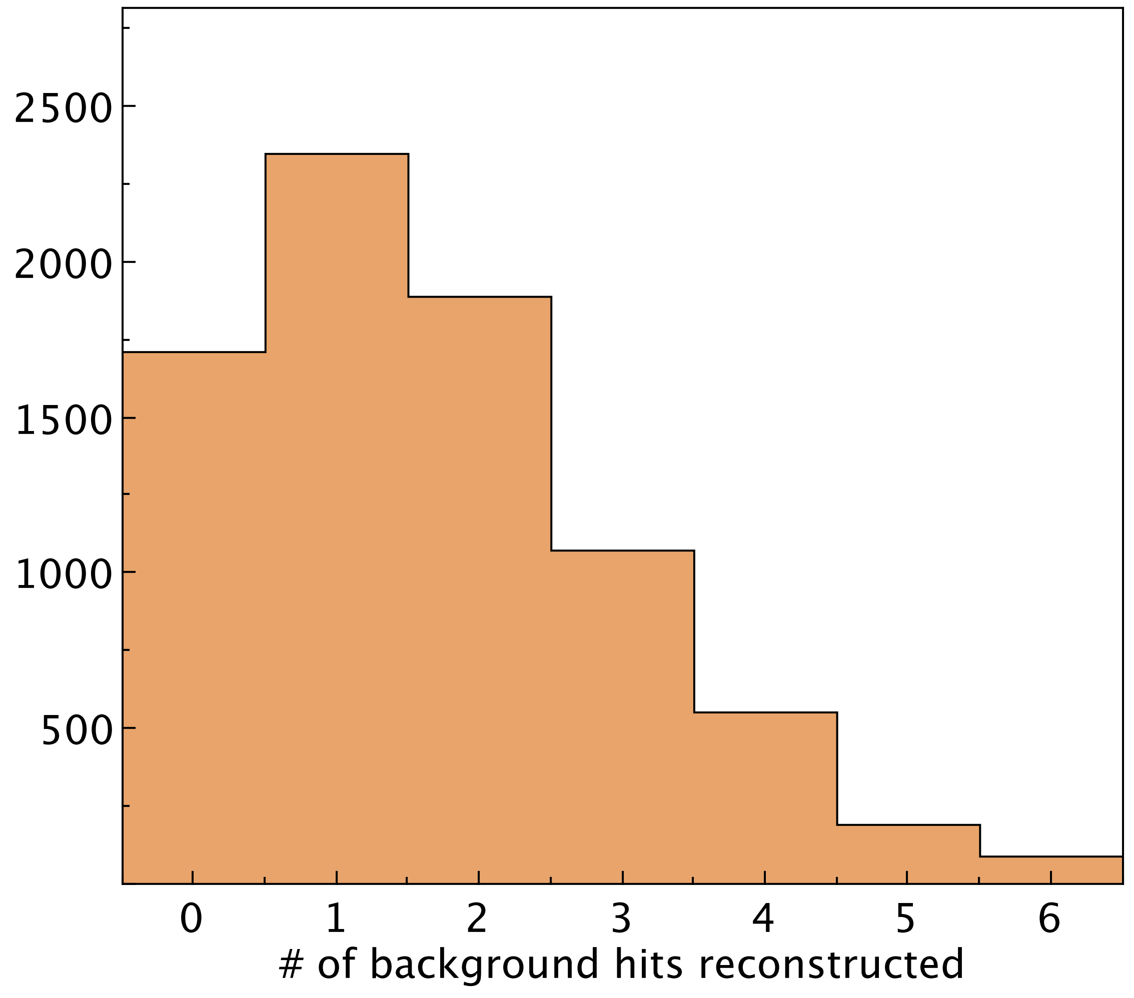
Example Reconstructed Tracks from one of the segments





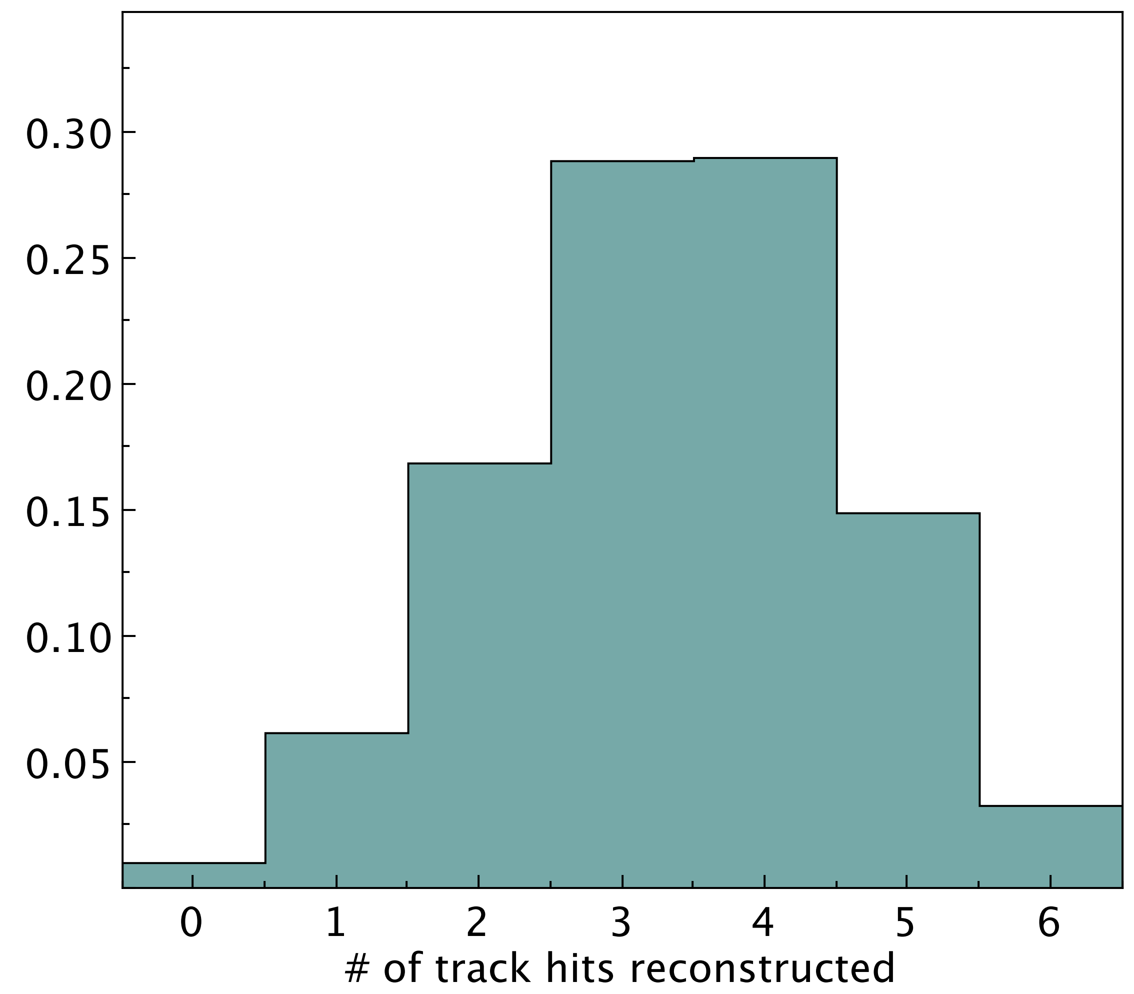
Post-Processing:
After initial inference, the identified points (hits) can be used to recover missed hits.

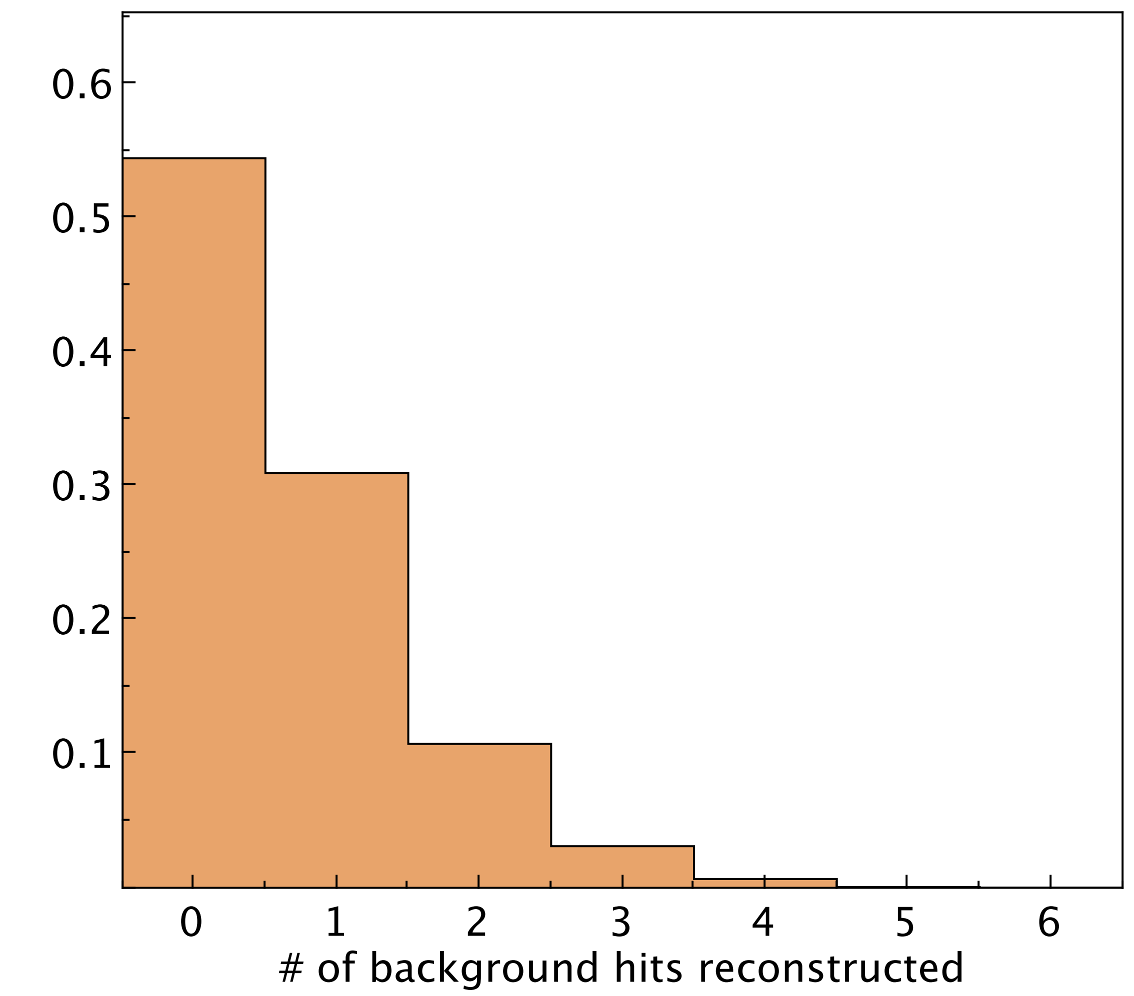
~200 Hz



SoLID GNN (GravNet)

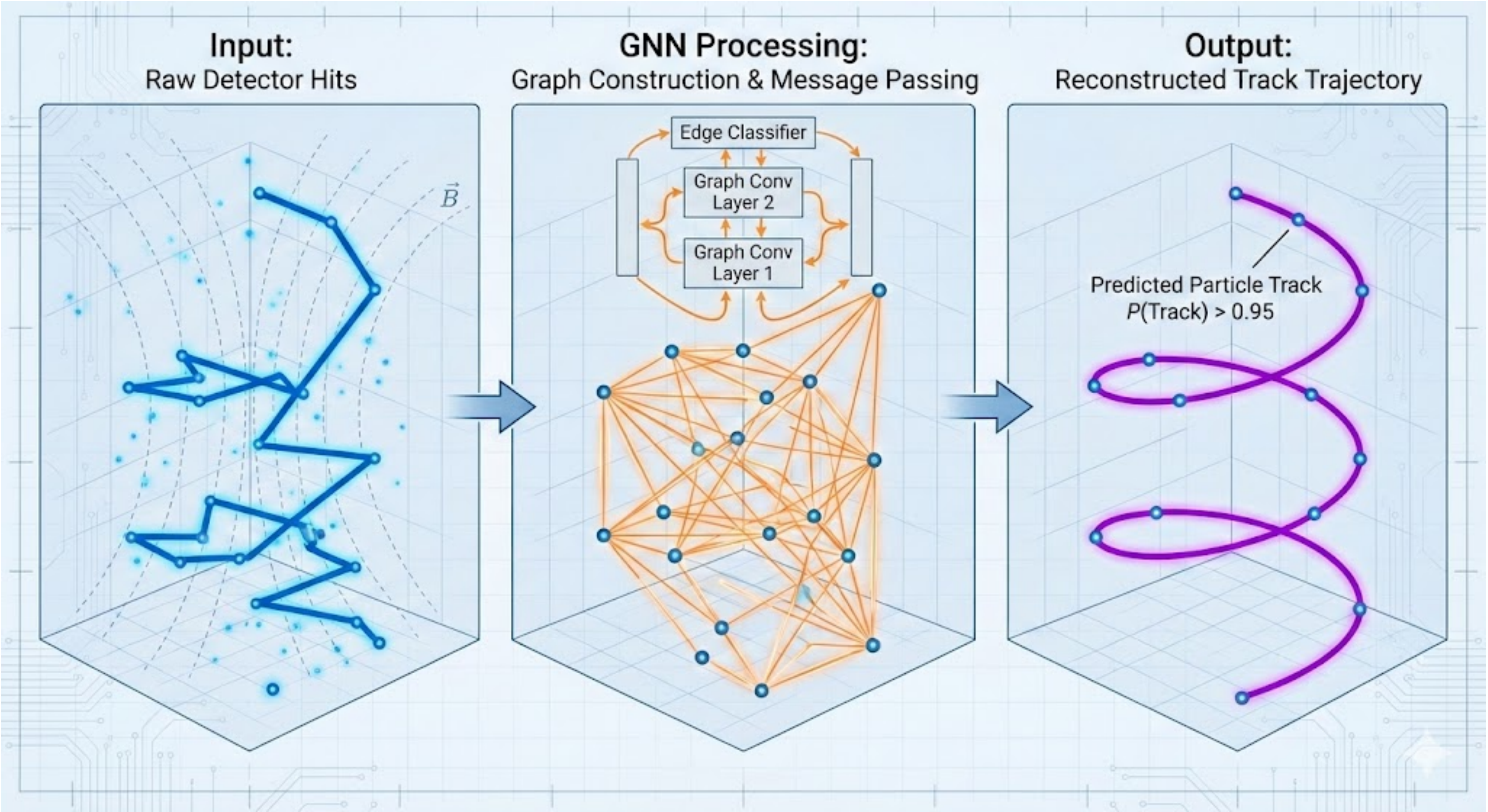
GNN Prediction





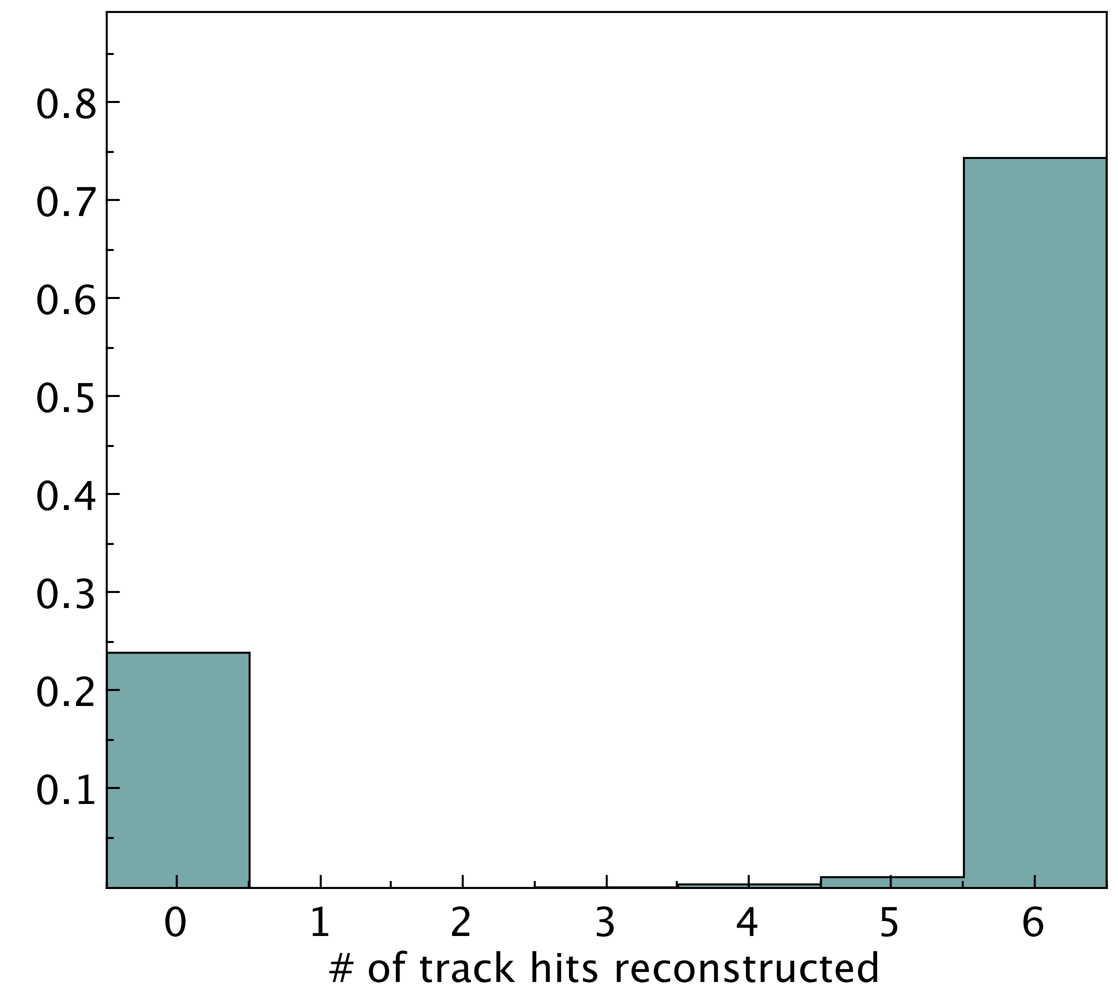
1/2 Lum

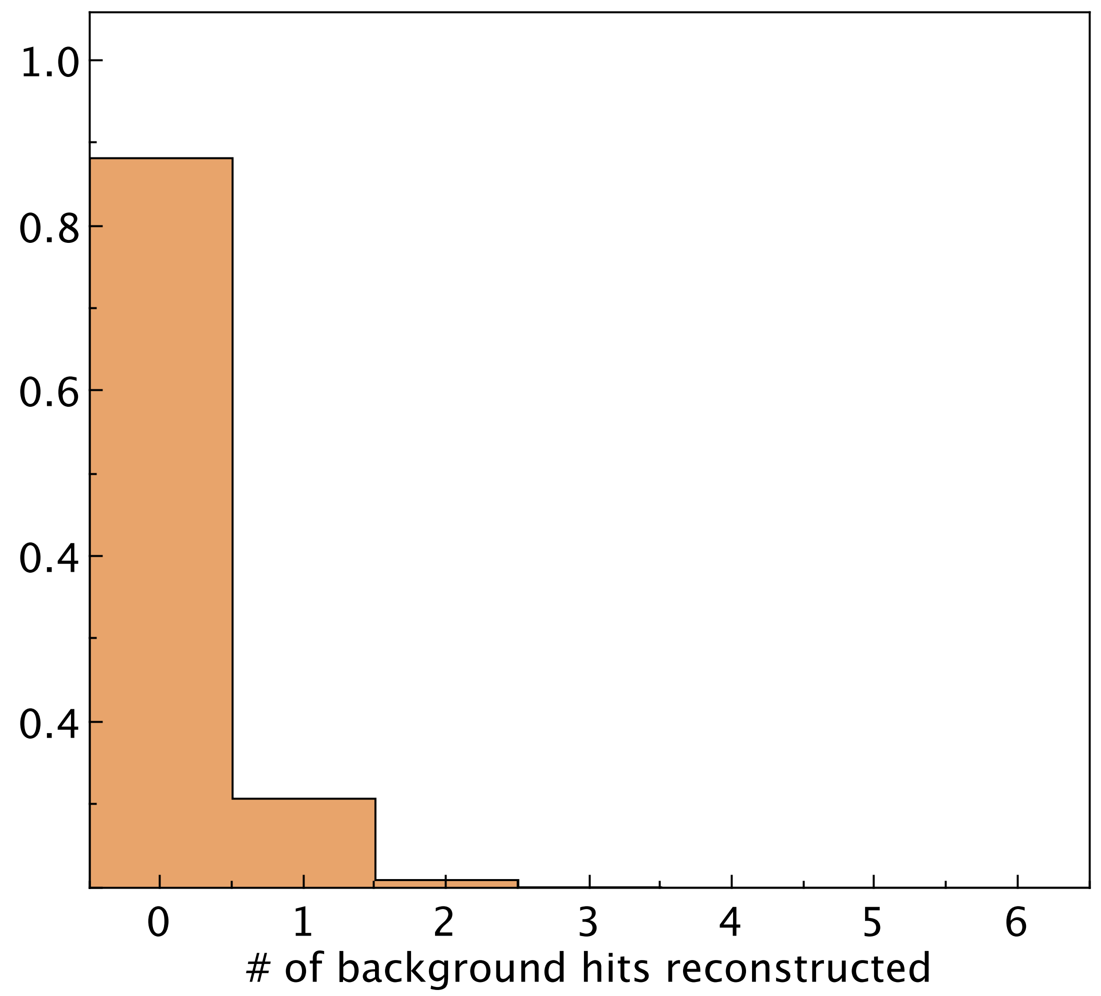
Helix Fit:
The reconstructed hits can be used to do a Helix Fit (3 or more points) and collect points that are close to the helix. The collected points are then run through the network and infer points belonging to a track.



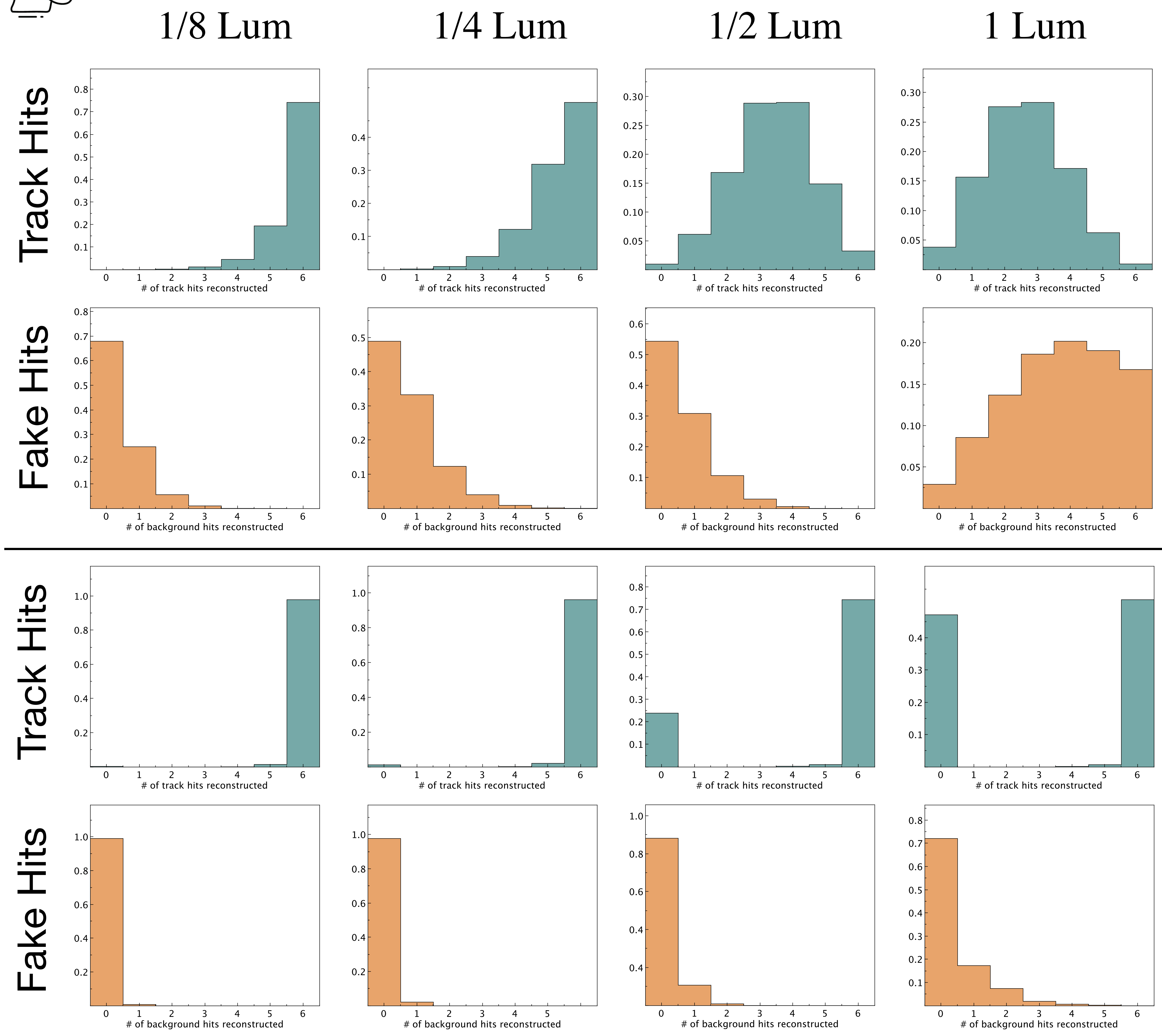
With 2 hits reconstructed, the Helix is fit using 2 hit positions and another point at the middle of the target. (Shown in later plots)

After Helix Fit





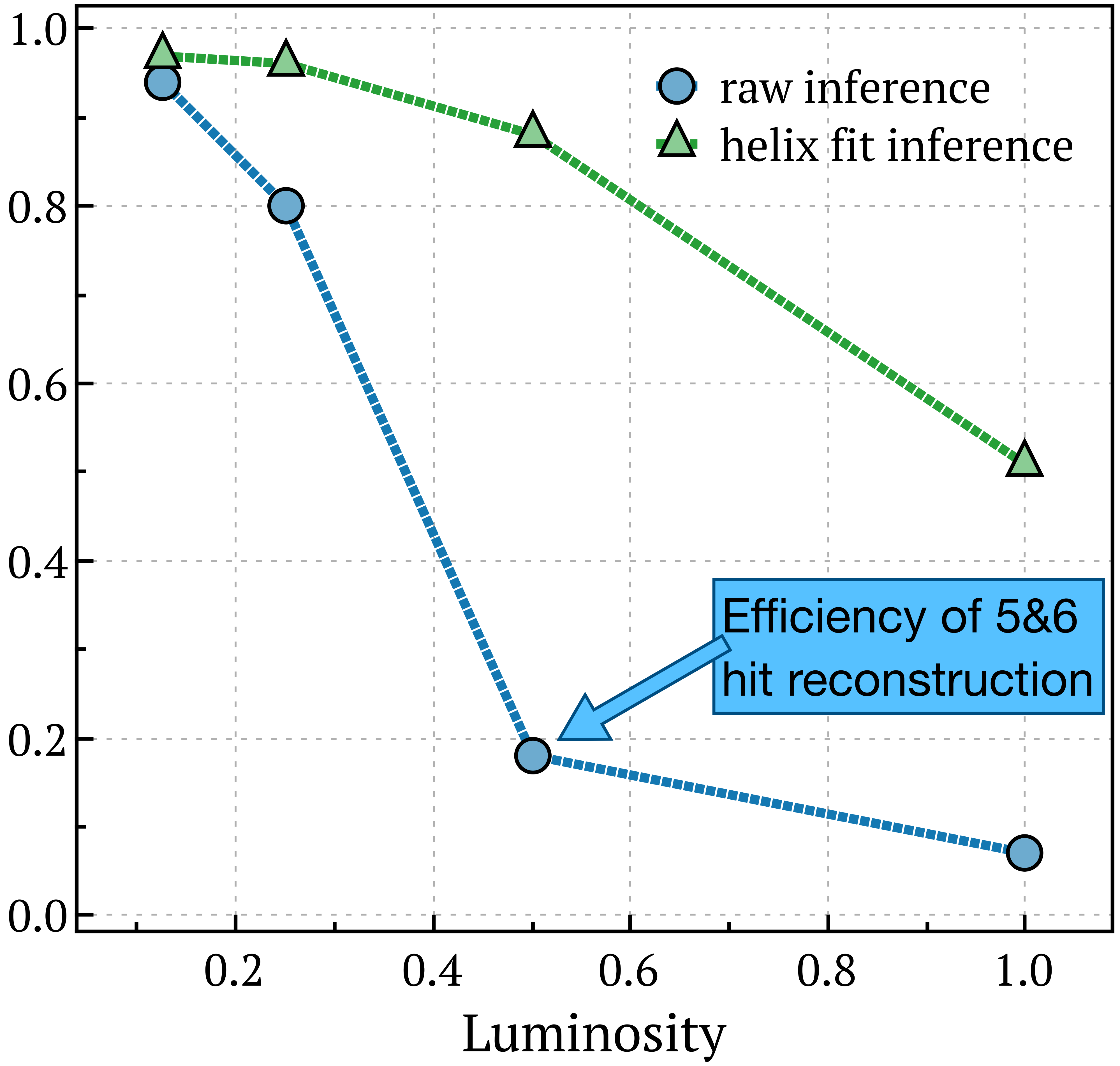
SoLID GNN (GravNet)



Raw Inference From the Network:
Number of hits reconstrcuted per track and number of background hits reconstructed

Reevaluated Inference:
The reconstructed points were fit with Helix, and hits with the closest approach to the helix are considered for a second pass through the network.

SoLID GNN (GravNet)



Full Luminosity:
Taking all hits in the selected segment.

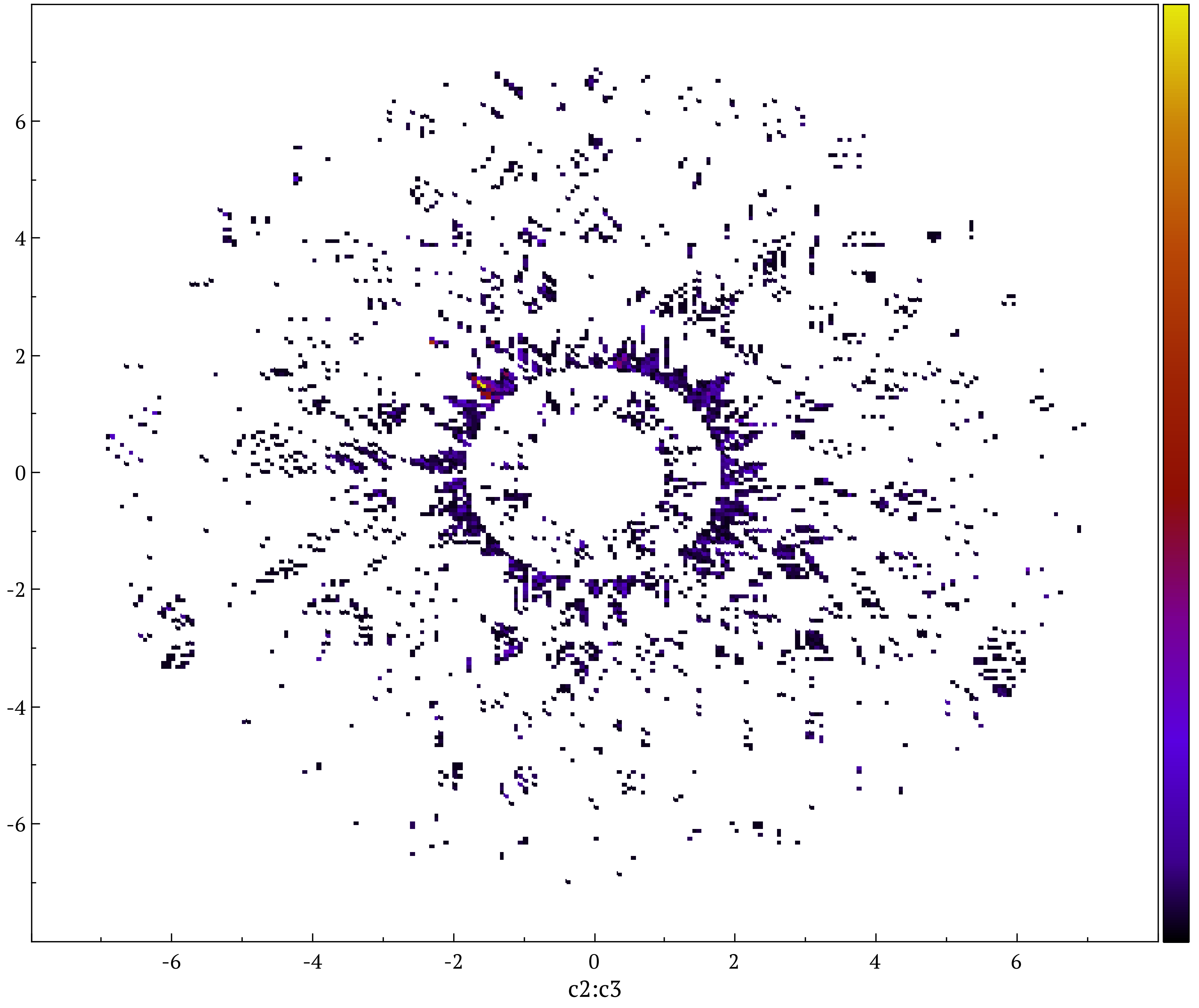
1/2 Luminosity:
In a given region, choose half of the background hits to mix with the signal hits.

1/2 Luminosity:
Can also be achieved by choosing the region more tightly, currently, it's a very loose empirical cut based on the position of the hit in the calorimeter.

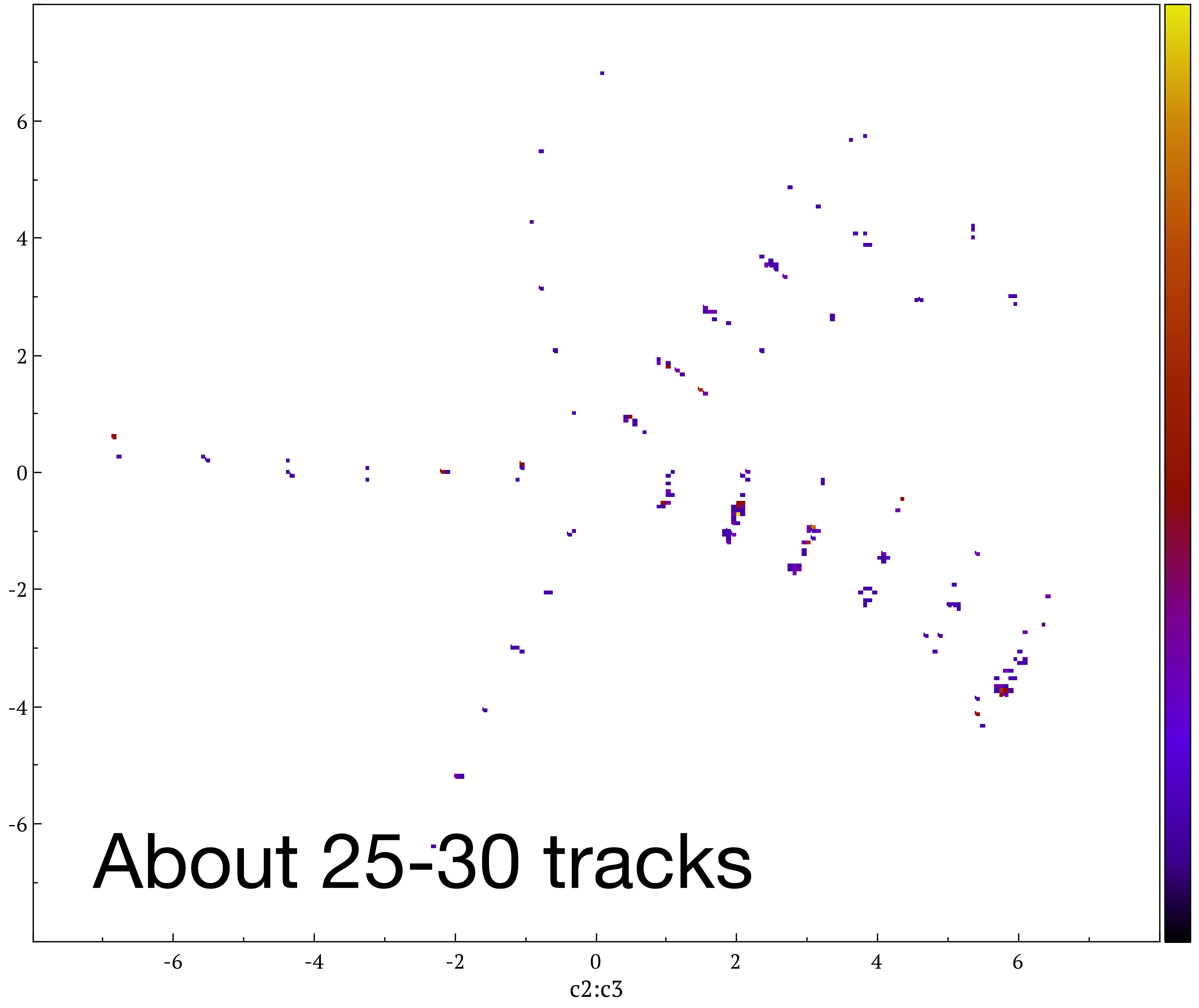
1/2 Luminosity:
Can also be achieved by preselecting the hits, based on their ADC or TDC values (maybe, the raw data values were not available in the sample)

SoLID GNN (GravNet)

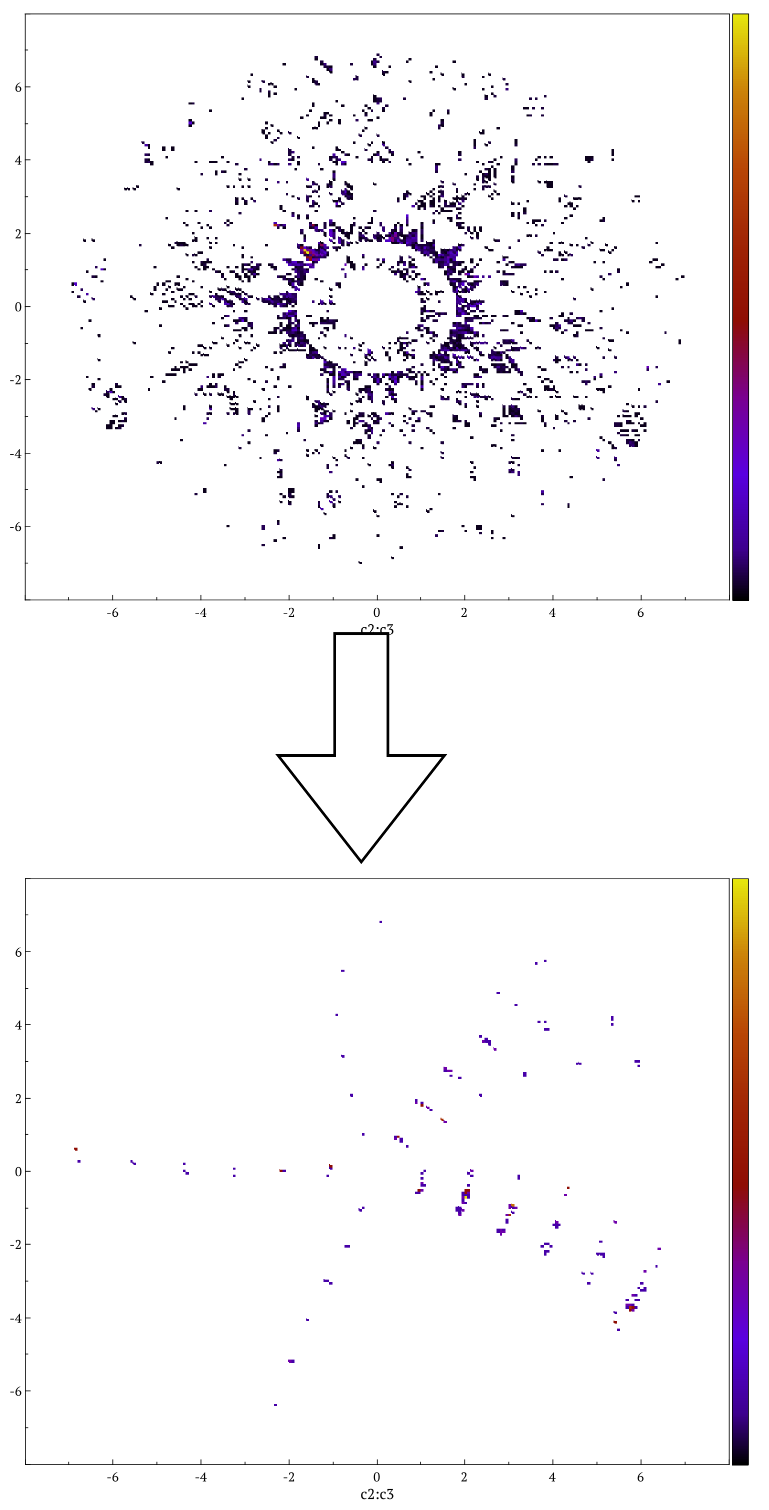
All raw hits in the event,
Only background hits included.



Tracks in the data sample
matching background hits.

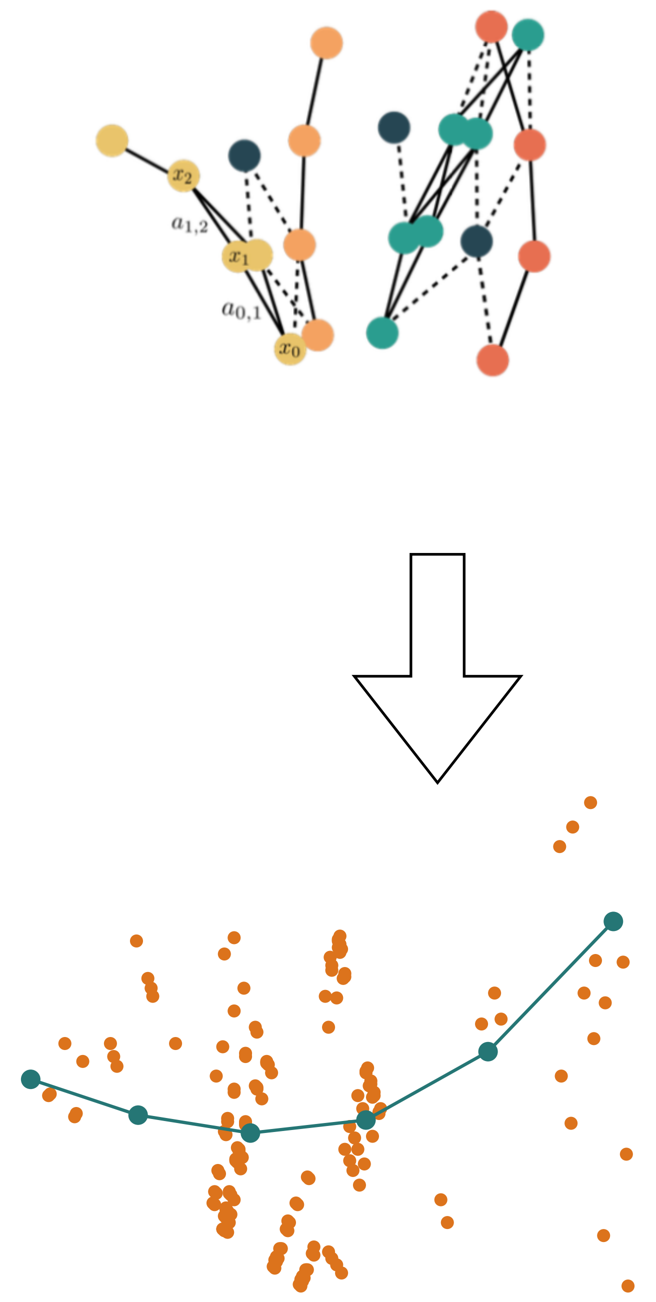


Strategy



Train a Neural Network to identify regions with potential tracks

GravNet



Use GravNet to identify tracks in each isolated region

Procedure:

- Identifying regions of interest with high accuracy and purity will lead to a significant reduction in data volume from DAQ.
- Each region will be analyzed for tracks using GravNet.
- Inefficiencies in hit reconstruction can be corrected by a simple helix fit.
- The network can be extended to provide track parameters for identified track candidates.

Denoising:

- The Denosing (region finding) network exists, having been developed after the talk was written. Details later.

The takeaway

The Results:

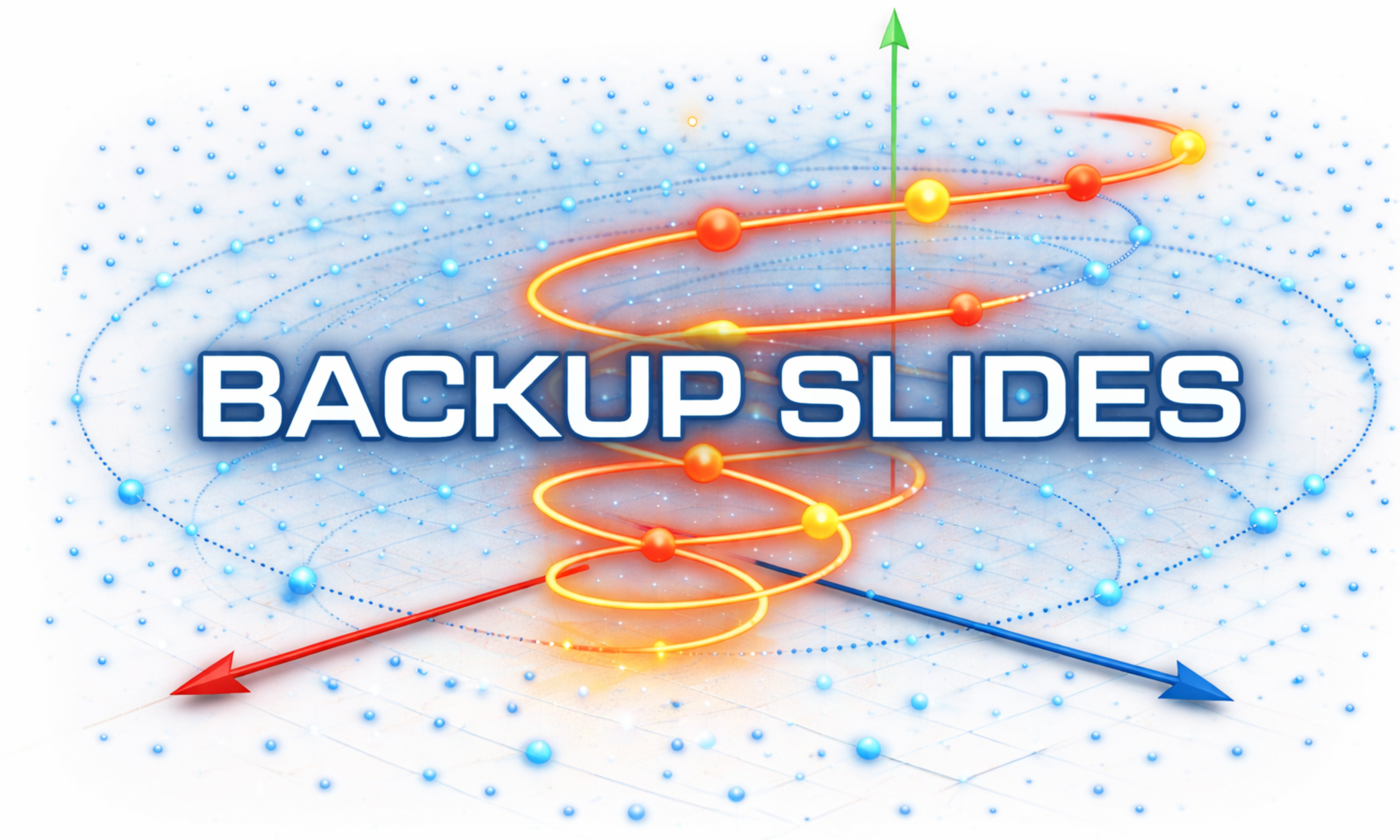
- The results presented in this talk are preliminary, based on only one week of investigation.
- The lack of quantitative results is deliberate; things may change.

Path Forward:

- Initial investigation of track classification using GNNs shows encouraging results.
- To implement full AI/ML track recognition, a pre-processing of data is necessary to identify the region of interest.
- Post-Processing is needed to correct for missing hits and possibly validate the predictions with a fit.
- More features are needed (such as time and charge for each hit), to clean the sample passed to the network

Synergy:

- The developed method is fairly generic and can be used for other detector systems (current and upcoming experiments)


BACKUP SLIDES