# BESIII track finding algorithm based on edge-classifying GNN

*Mini-workshop on graph neural networks for tracking*

*3 June 2022*

Xiaoqian Jia, Xiaoshuai Qin, Teng Li, Xingtao Huang, Xueyao Zhang

Shandong University, China

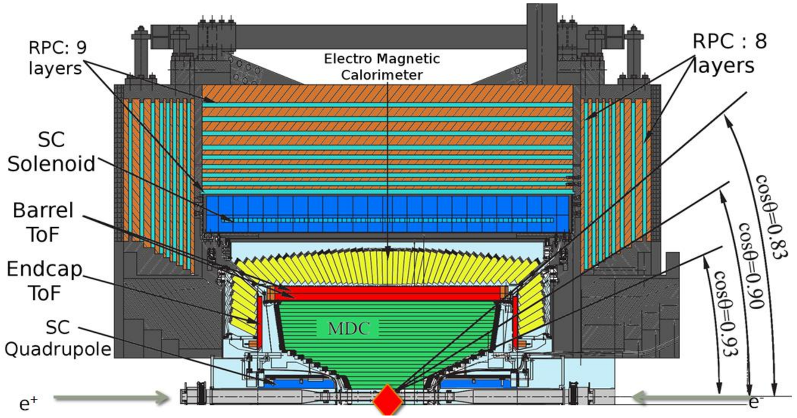
# Outline

- Motivation
- Pattern map based on MC simulation
- Graph construction method
- Edge classification
- Track building
- Summary

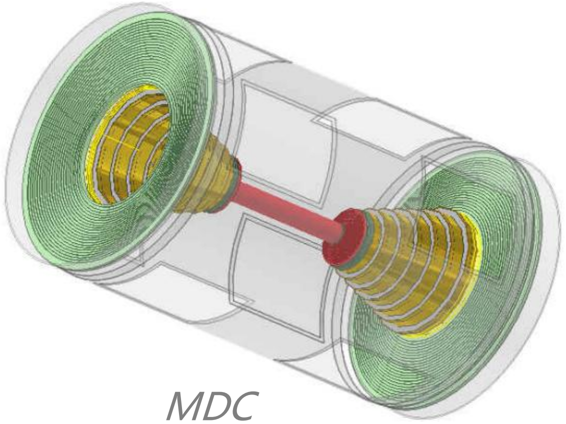
# Charged particle tracking at BESIII

- Beijing electron-positron collider (BEPCII)
  - Peak luminosity : $10^{33}cm^{-2}\ s^{-1}$
  - CMS: 2.0 - 4.95 GeV, τ -charm region
- Beijing Spectrometer (BESIII)
  - Study the electroweak and strong interactions
  - Search for new physics
- Main Drift Chamber (MDC)
  - 43 sense wire layers
  - dE/dx resolution : 6%
  - Momentum resolution : 0.5%@1GeV/c
- Tracking finding for low $P_T$ tracks is still challenging for traditional methods (e.g. pattern recognition)

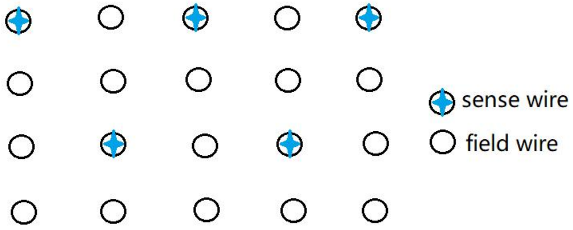
*Aerial view of the BEPCII*



*BESIII detector*



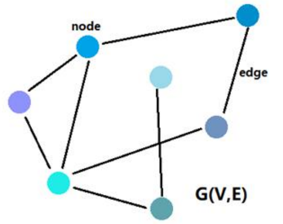
*MDC*



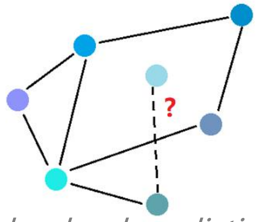
*Schematic drawing of the MDC cell*

# Graph Neural Networks (GNNs)

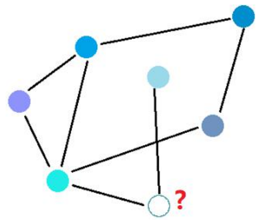
- ◆ GNN has great potential to boost tracking efficiency
- ◆ Basic working principle :
  - Graph
    - A graph is composed of nodes and edges presented in an adjacency matrix
  - Node embedding
    - Encode nodes so that similarity in the embedding space approximates similarity in the graph
  - Message passing
    - Aggregate information from neighboring nodes across edges to form new features on each node
  - Learn the node embeddings by iteratively combining the node information in a local neighborhood



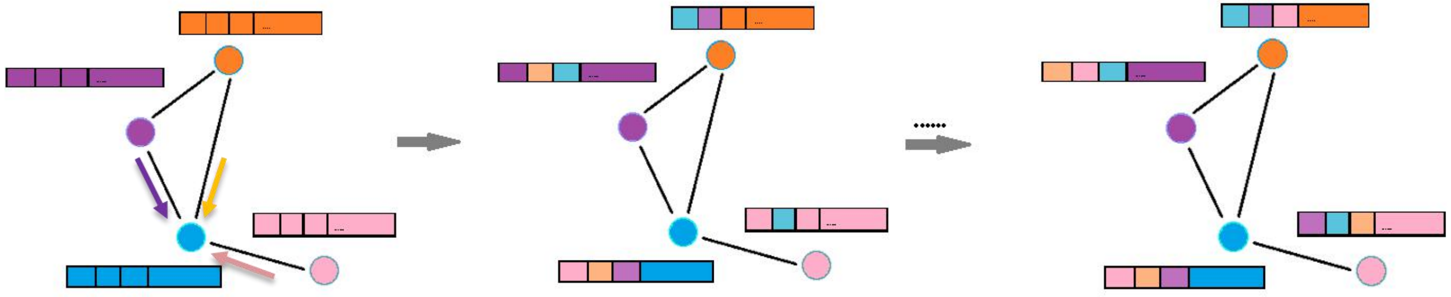
*Graph-level predictions*



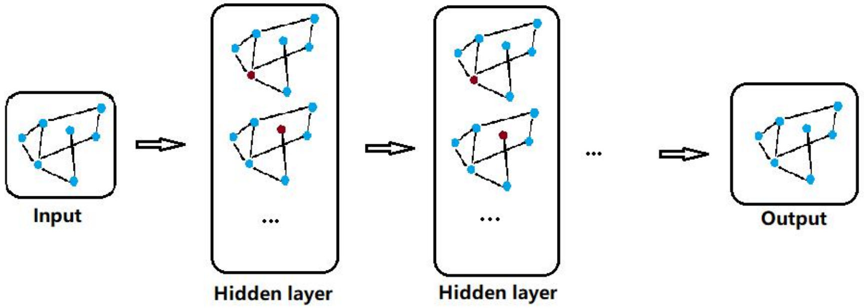
*Edge-level predictions*



*Node-level predictions*



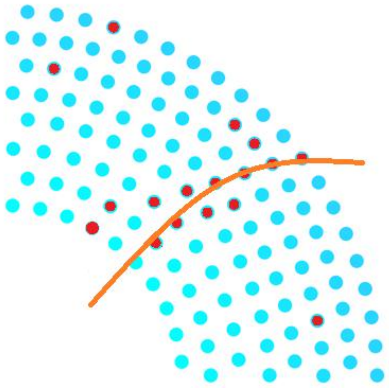
*Message-passing*



# Tracking based on edge-classifying GNN

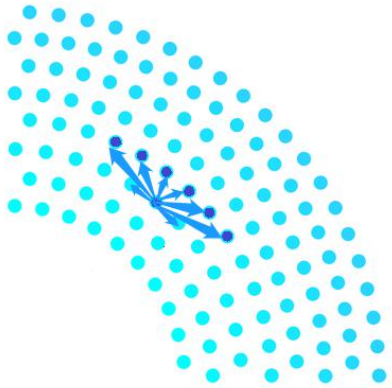
**MC simulation**

Charged particles leave hits in the MDC

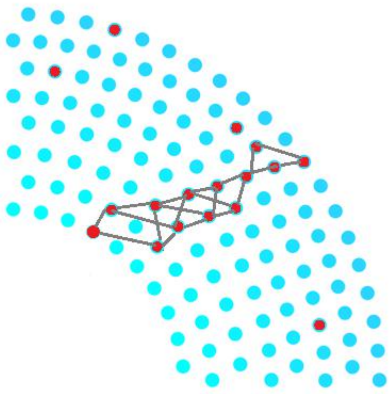


Pattern Map

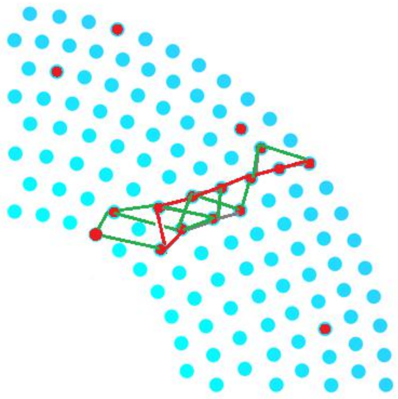
(A set of possible hit neighbour combinations)



Construct the graph based on the Pattern Map



Classify the graph edges



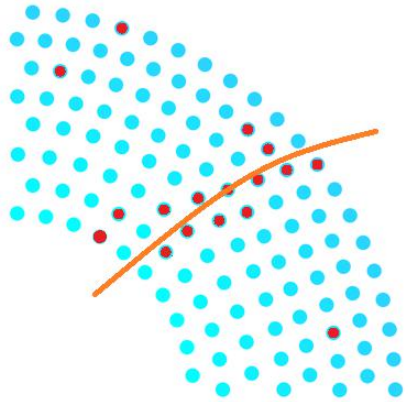
***High classification score***

*→the edge belongs to a true particle track*

***Low classification score***

*→it is a spurious or noise edge*

Build track candidates

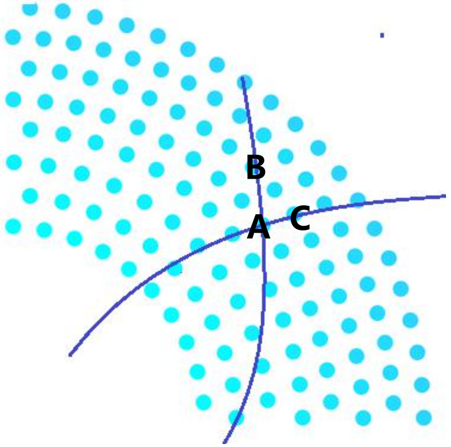


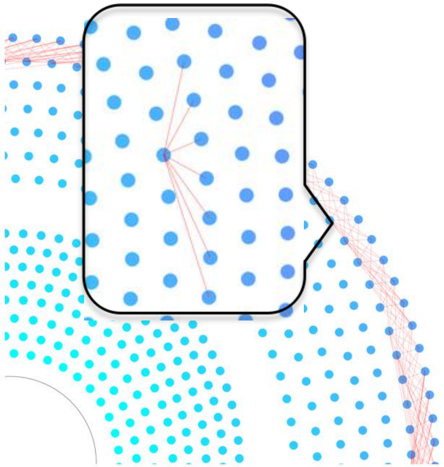
Track fitting

# Pattern Map based on MC simulation

*To reduce the number of fake edges during graph construction based on MC simulation*

- Definition of valid neighbors
  - Hits on the same layer
    - Two adjacent signal wires on the left and right
  - Hits on the next layer
    - The collection of sense wires that could potentially represent two successive hits on a track
- MC sample used to build pattern map
  - Two million single tracks from BESIII MC truth information
  - 10 charged particles ($e^{\pm}$, $K^{\pm}$ , $\mu^{\pm}$, $p^{\pm}$, $\pi^{\pm}$)
  - 0.05GeV/c < P< 3GeV/c
- To reduce the size of the graphs, the Pattern Map is further reduced based on a probability cut

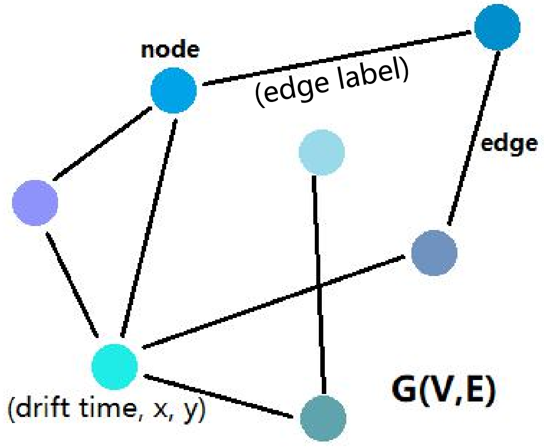


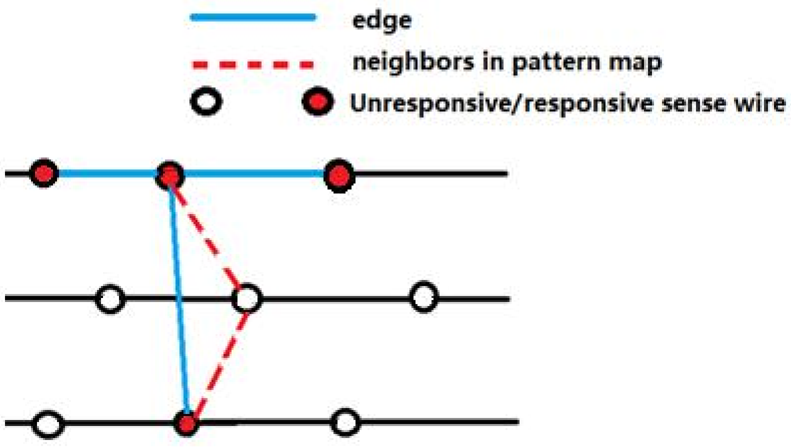


*A wire on layer13 and tits neighbors on layer14*

# Graph construction

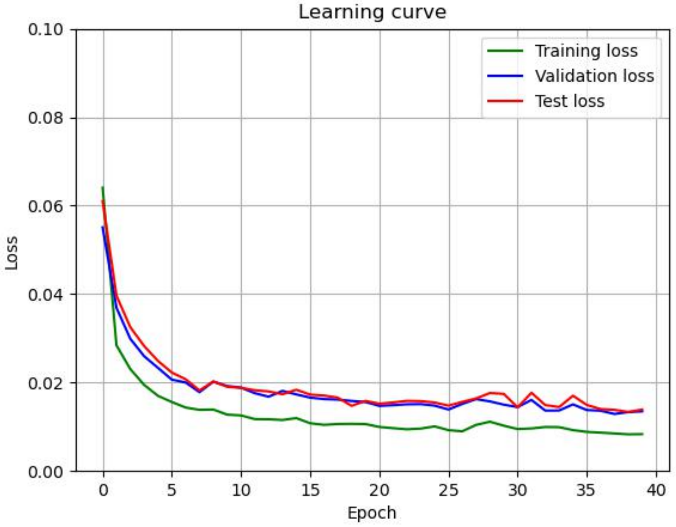
- Training sample
  - Single-particle ($e^\pm$, $K^\pm$ , $\mu^\pm$, $p^\pm$, $\pi^\pm$ ) MC sample
  - $0.2$ GeV/c < p < $3.0$ GeV/c
  - Mixed with BESIII random trigger data as background (~45% hits)
  - Train: Validation: Test = 4: 1: 1
- Edge assignment based on Pattern Map
  - Hit with its neighbors on the same layer
  - Hit with its neighbors on the next layer
  - Hit with its neighbors' neighbors on one layer apart
- Node features
  - Raw drift time
  - Position coordinates (x, y) of the sense wires
- The final graph representation:
  - Node features, adjacency matrices, edge labels

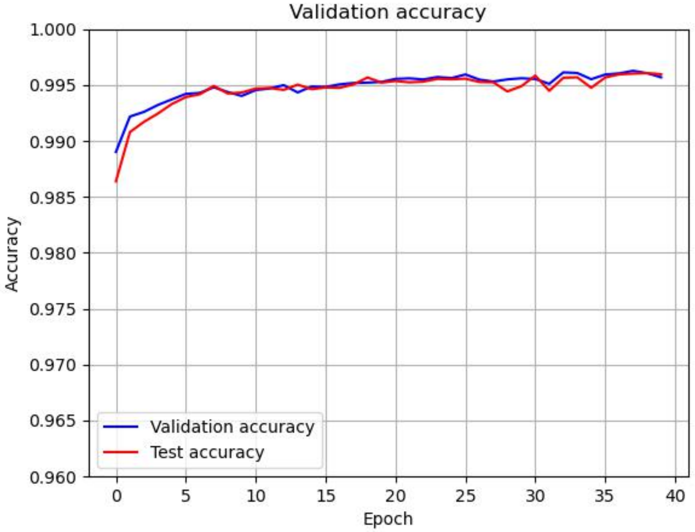




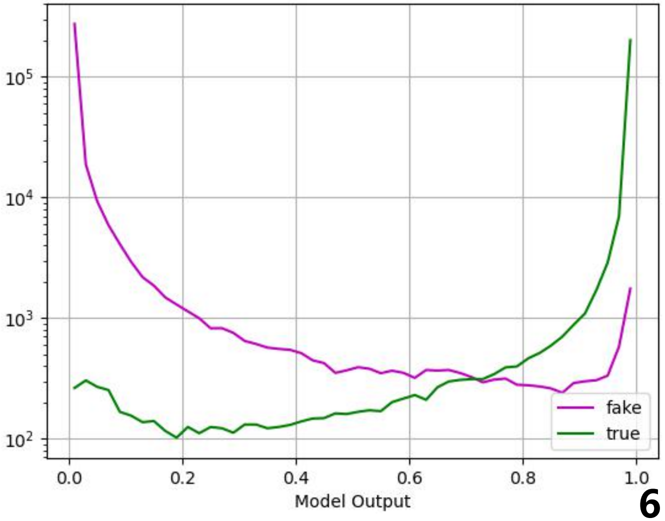
# GNN edge Classifier based on PyTorch

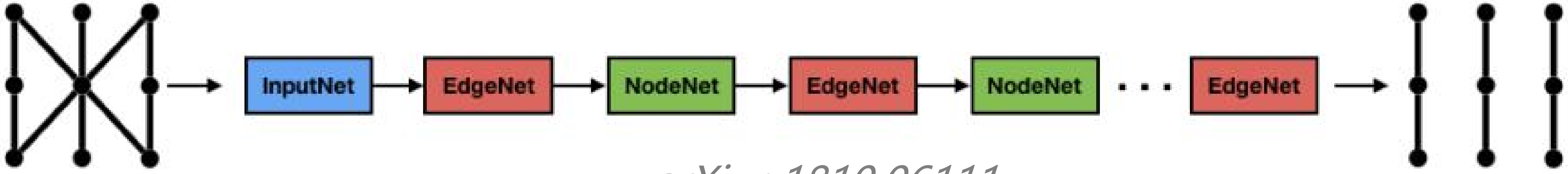
- Input network
    - Node features embedded in latent space
- Graph model
    - Node network and Edge network
    - 8 graph iterations
    - MLPs
    - Tanh activation





**~99.5% classification accuracy (threshold = 0.7)**

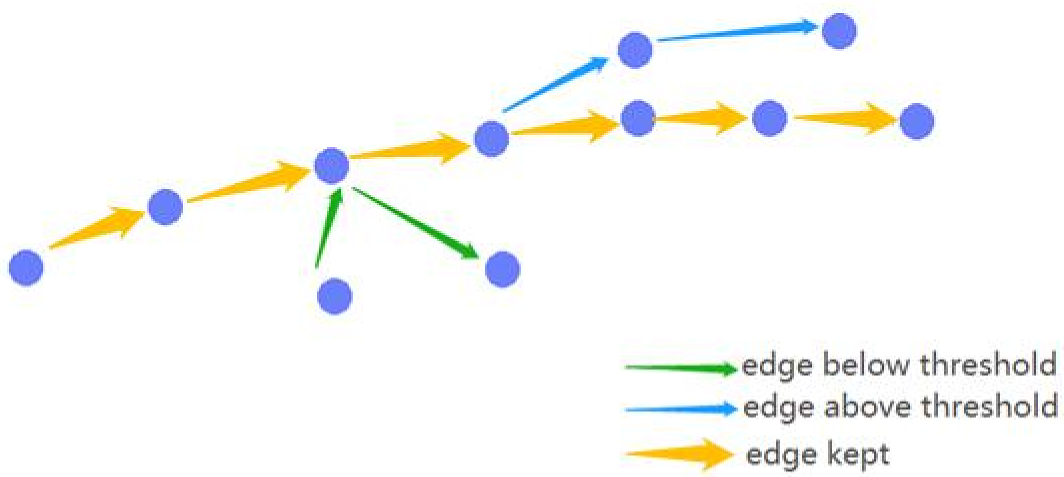


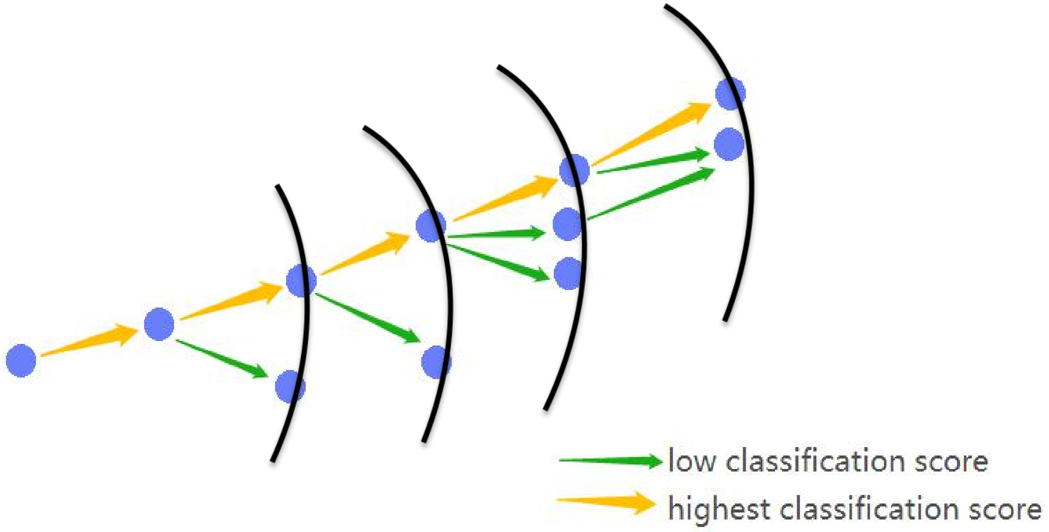


*arXiv : 1810.06111*

# Preliminary track building method

- Events with single tracks
  - Build the longest track
    - Connect true edges which form the longest track
  - Maximum classification score
    - Connect edges with the highest classification score





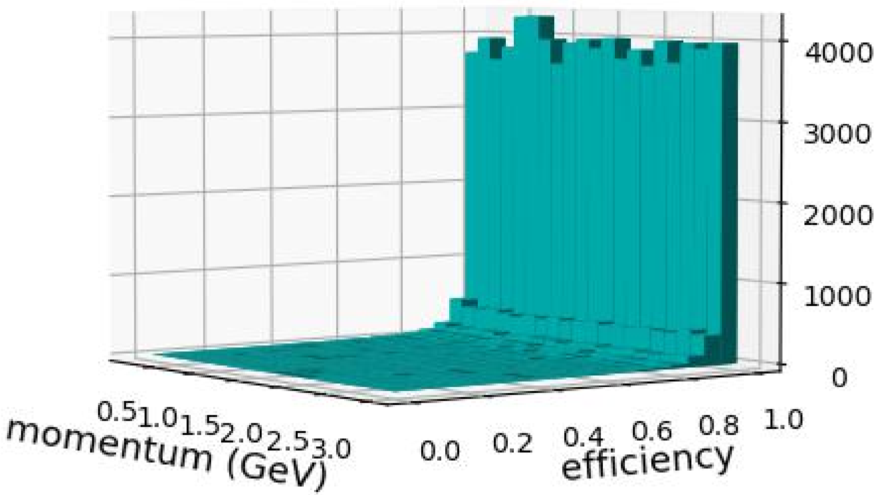
# Preliminary results

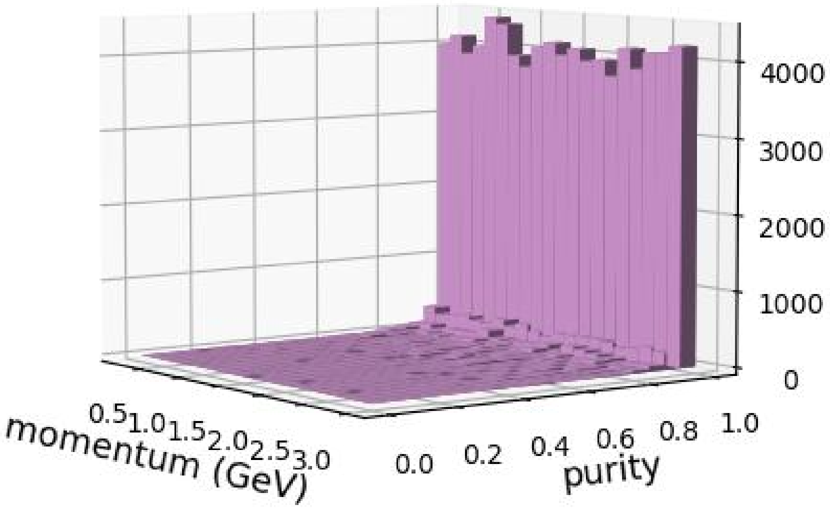
◆ Test sample

- Single-particle MC event sample
- Mixed with BESIII random trigger data as background (~45% hits)
- 0.2 GeV/c < p < 3.0 GeV/c

◆ Hit selection performance

- The preliminary results show that GNN provides high efficiency and purity of hits selection
  - Efficiency: Selected true hits / Total true hits
  - purity : Selected true hits / Total selected hits

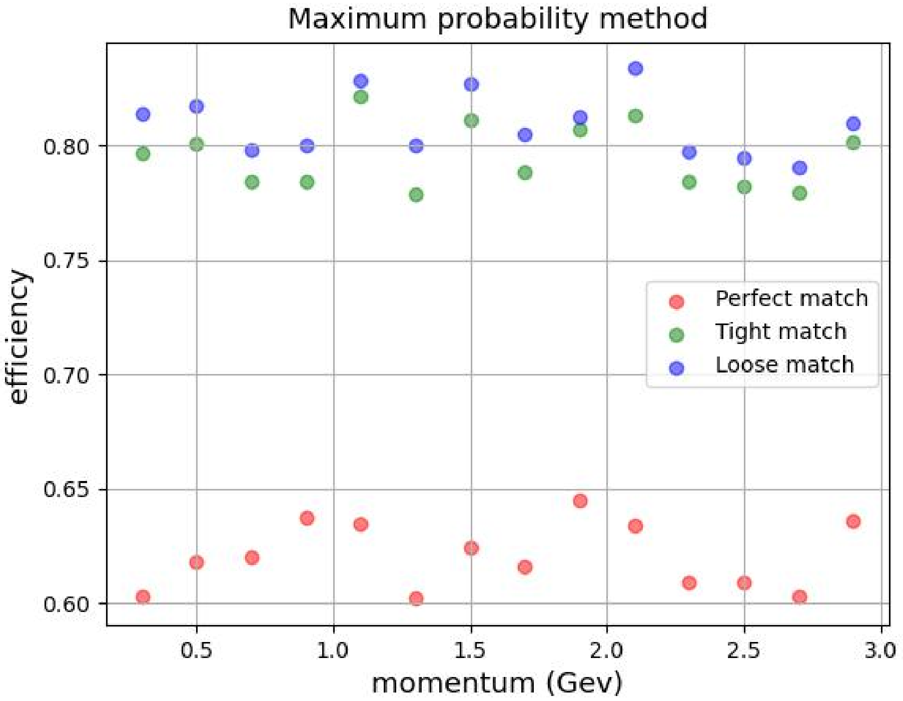


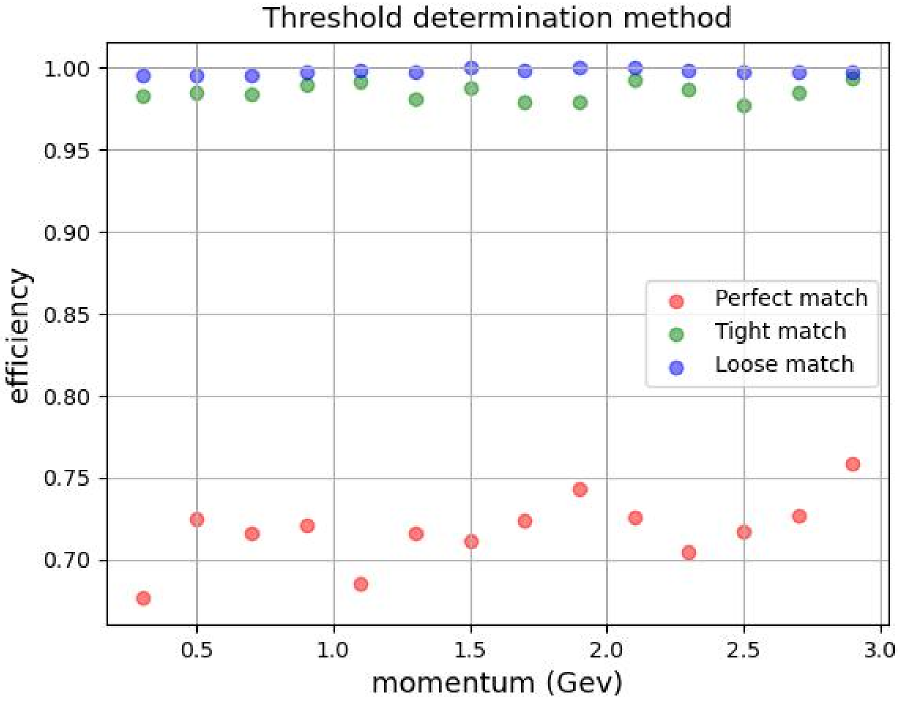


# Preliminary results

◆ Track reconstruction efficiency

- Perfect / Tight / loose match efficiency : the number of reconstructed tracks containing over 100% / 75% / 50% of hits from the same particle, divided by the total number of particles





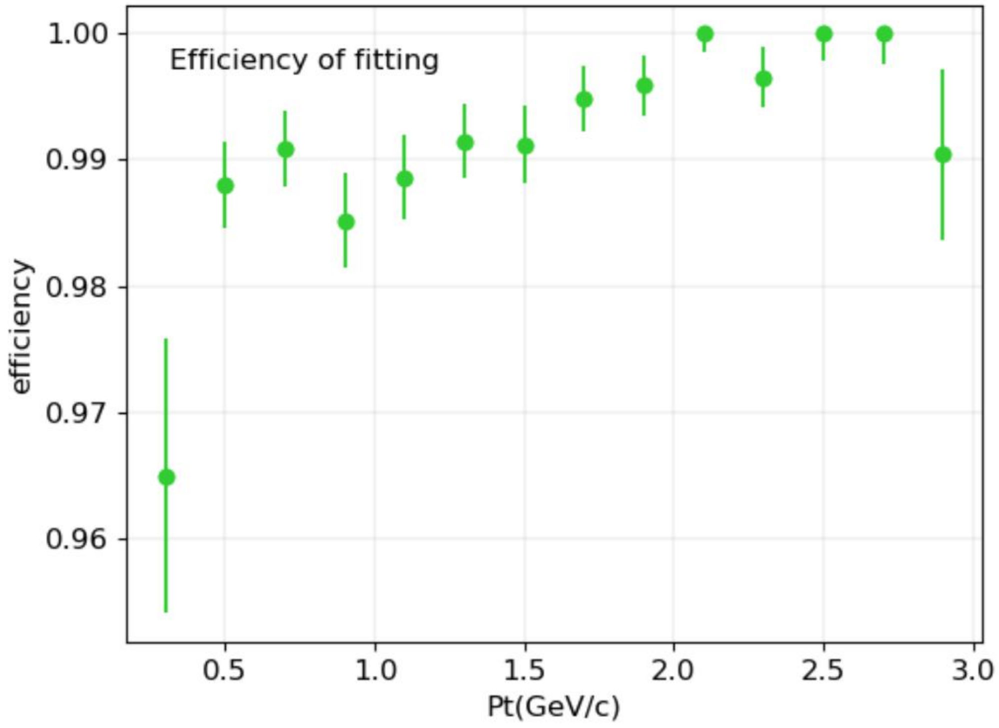
**The threshold determination method shows much better performance**

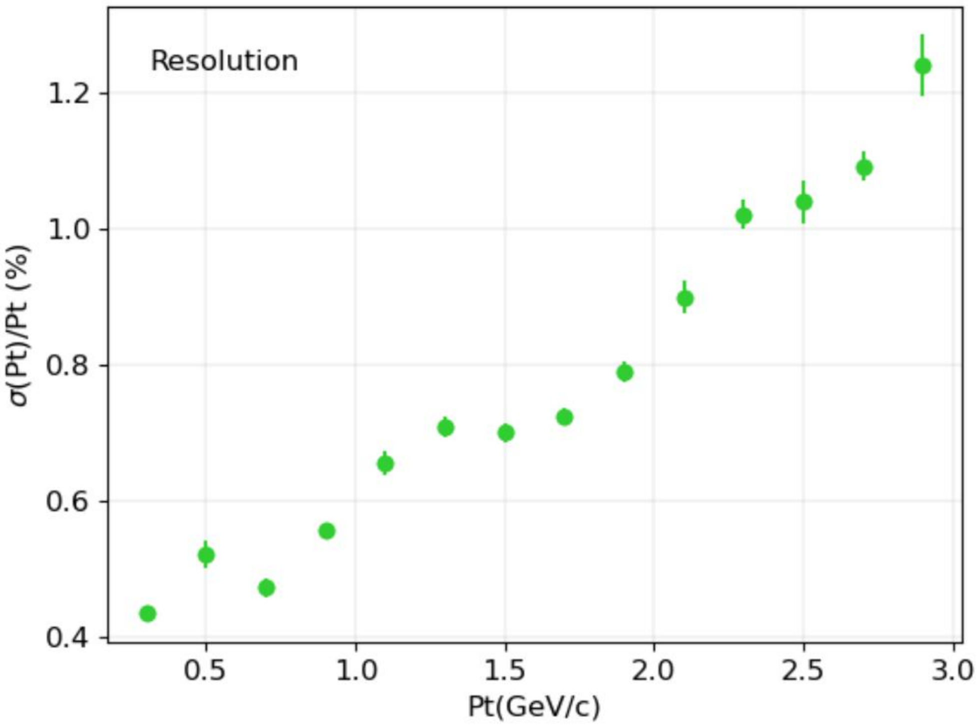
# Preliminary results

- ◆ To evaluate the tracking performance, track fitting algorithm based on Genfit ( a Generic Track-Fitting Toolkit) is used to reconstruct the particle momentum

*https://github.com/GenFit/GenFit*

- ◆ Particle reconstructed performance
  - Good resolution performance can be obtained





# Summary

- A track finding algorithm prototype based on edge-classifying GNN at BESIII is under development
    - A graph construction algorithm based on a hit pattern map is studied
    - GNN model is studied and tested to distinguish true hits on track from fake hits
    - Preliminary track building method is studied
- Preliminary results on BESIII MC data shows promising performance
- Outlook
    - Further optimization of the model is needed

      To boost performance for low $P_T$ tracks
    - Advanced track building algorithms is needed for multi-track events

      e.g. clustering with unsupervised learning