GNN-Based Tracking Reconstruction for the Fermilab Muon g-2 Experiment

Bingzhi Li

之江实验室
ZHEJIANG LAB

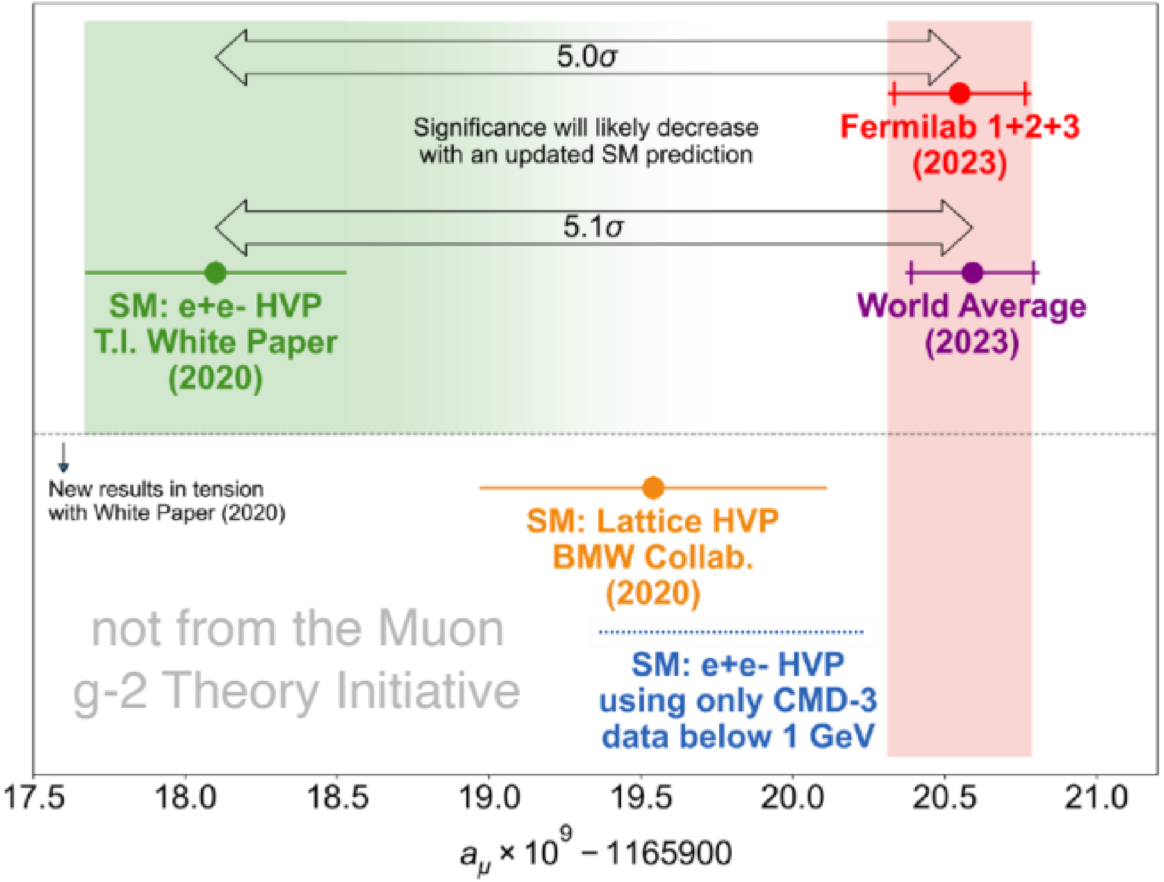
之江实验室 ZHEJIANG LAB

Contents

之江实验室 ZHEJIANG LAB

Brief introduction to the Muon g-2 Experiment

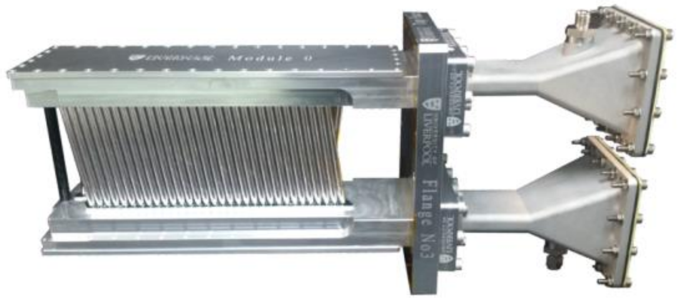
- The Fermilab muon g-2 experiment aims to measure the muon anomalous magnetic moment a_μ to 140ppb
- The Run 1-3 measurement precision is ~200 ppb
- Data collection is finished on 2023 ~ 21.9x BNL data
- 5σ discrepancy between Fermilab along result and SM value (white paper)
- Lattice HVP calculation and CMD3 experiment results reduce the discrepancy
- Expect results to be published by 2025

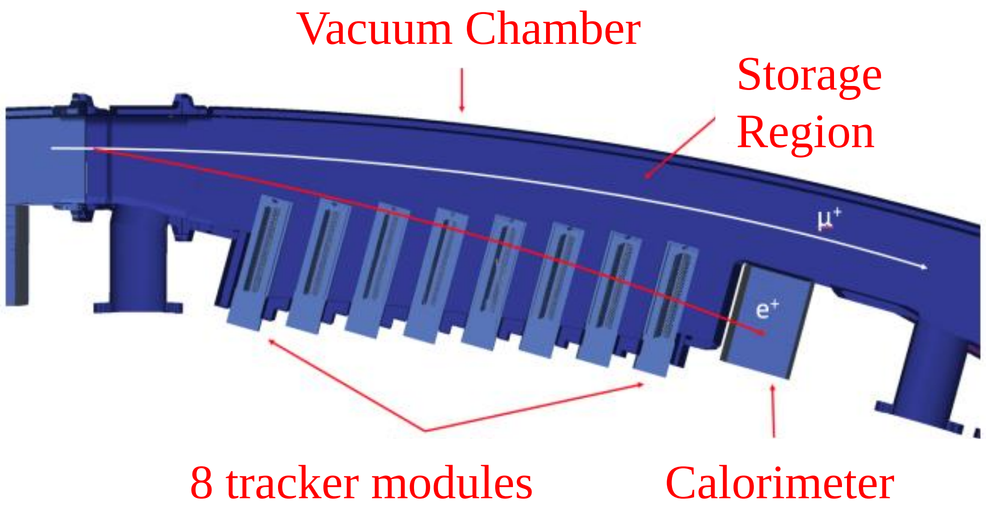


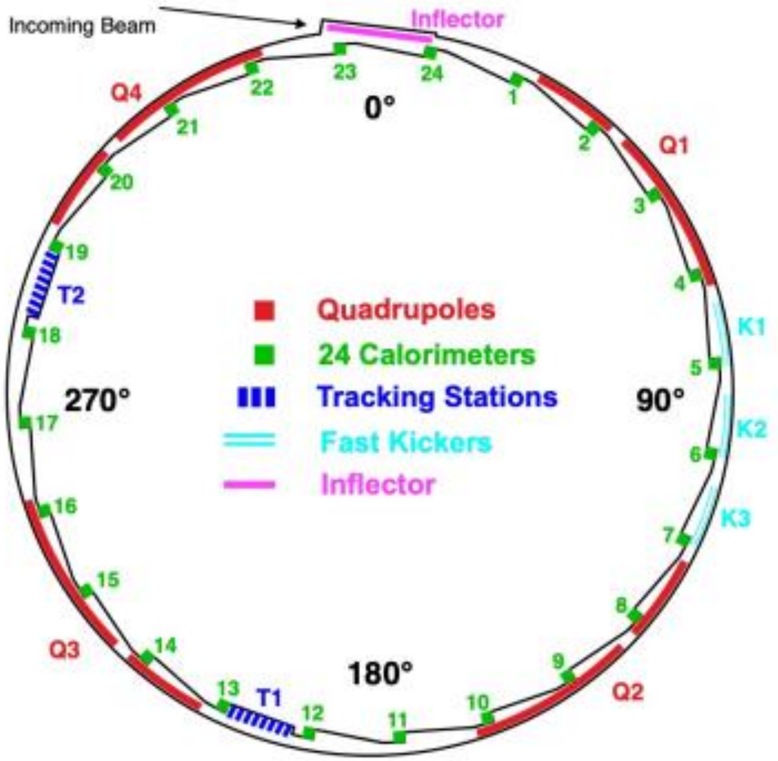
Tracker Detector in the Muon g-2 Experiment

- 50:50 Argon Ethane gas straw tube
- Time resolution 650ps / Position resolution 0.1mm
- Tracker system:

  2 station × 8 module × 2 view × 2 layer

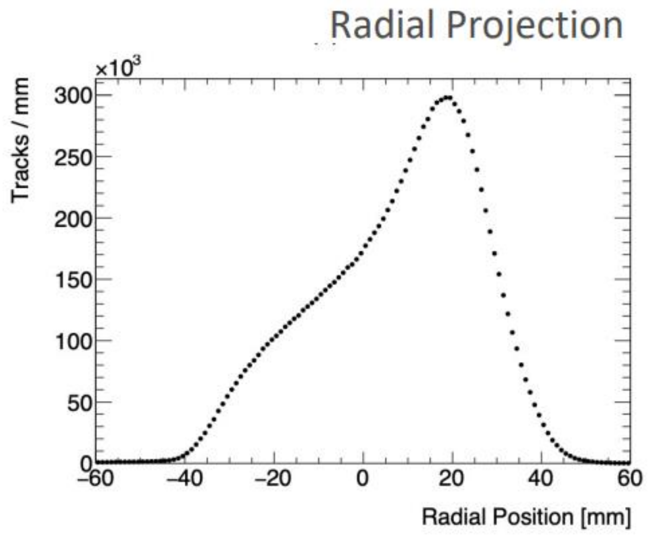


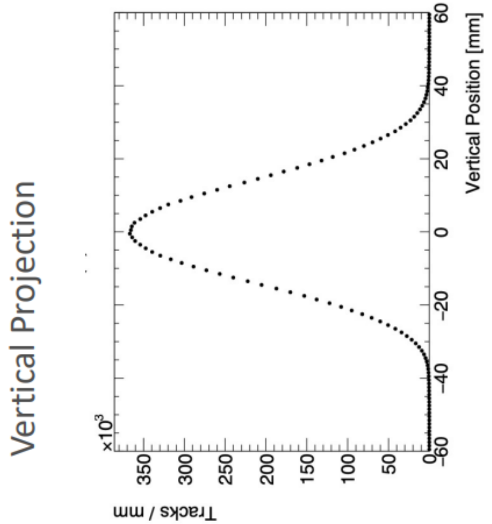


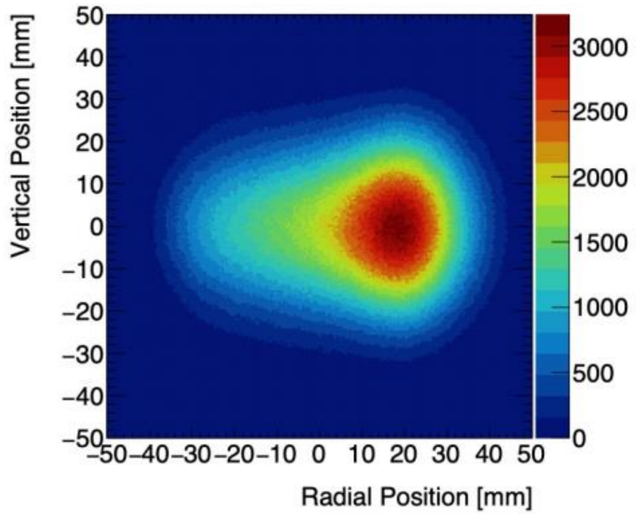


Motivation of the Tracker System

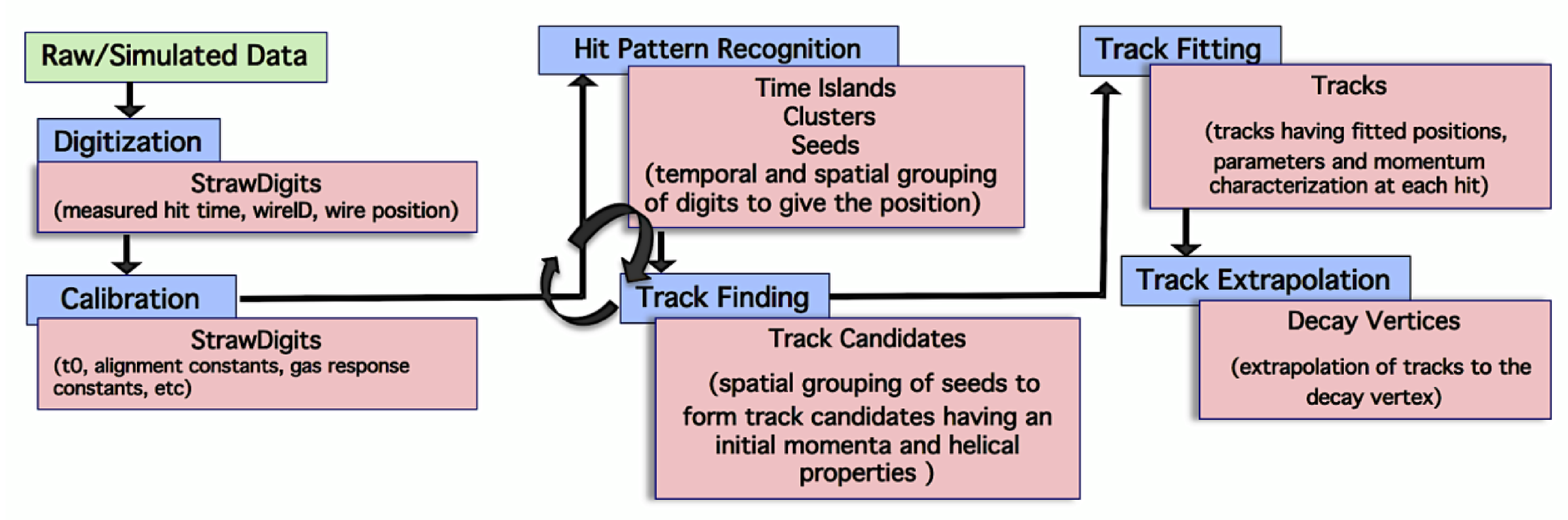
- **Beam Position:** The main role of the tracking detector is to measure the muon-beam spatial profile, along with projections in the radial and vertical directions
- **Beam Momentum:** Make an independent measurement of positron momentum
- **Beam Dynamics:** Characterize position and width CBO modulations, horizontal and vertical

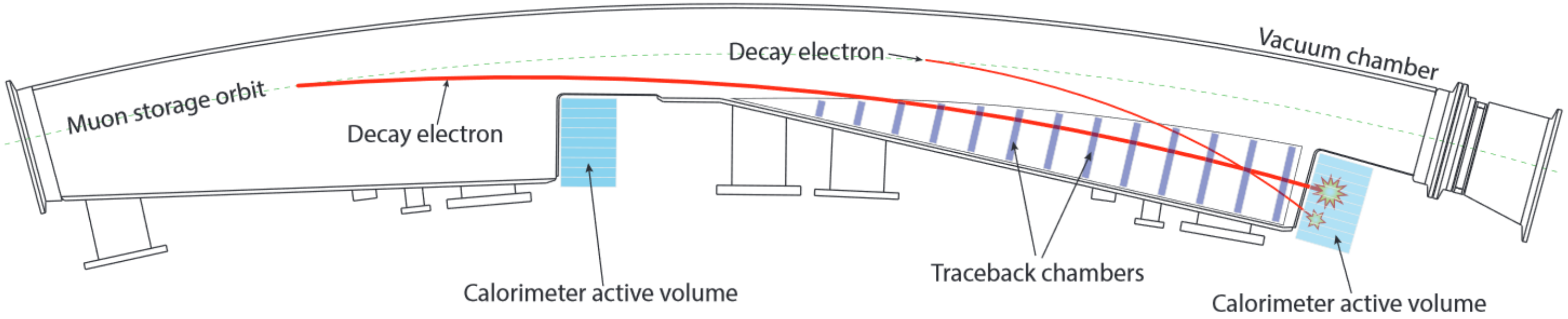






Default Tracking Reconstruction Workflow



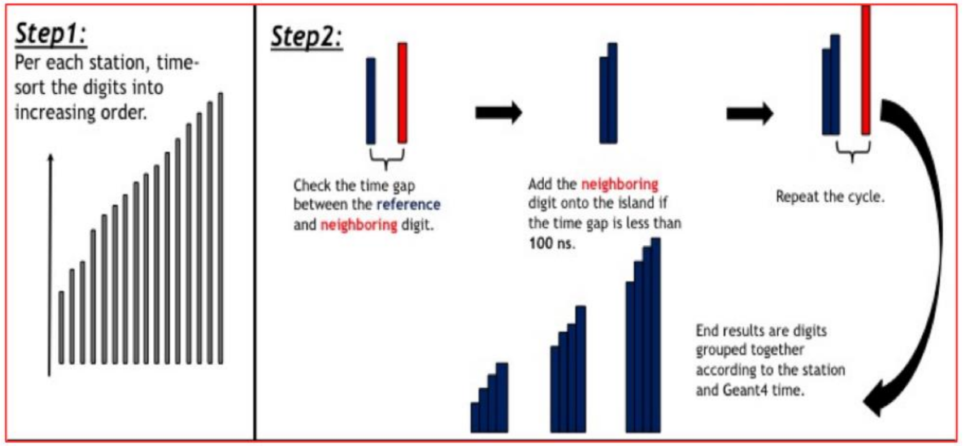


taken from Fermilab muon g-2 internal note 215

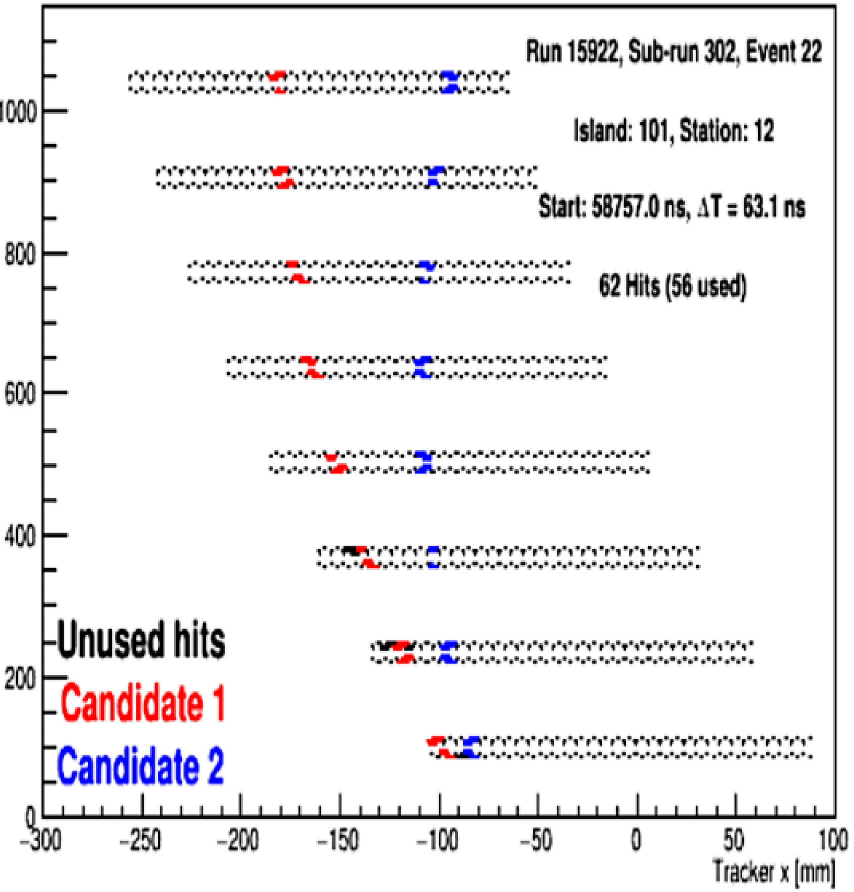
之江实验室 ZHEJIANG LAB

Default Tracking Reconstruction Workflow

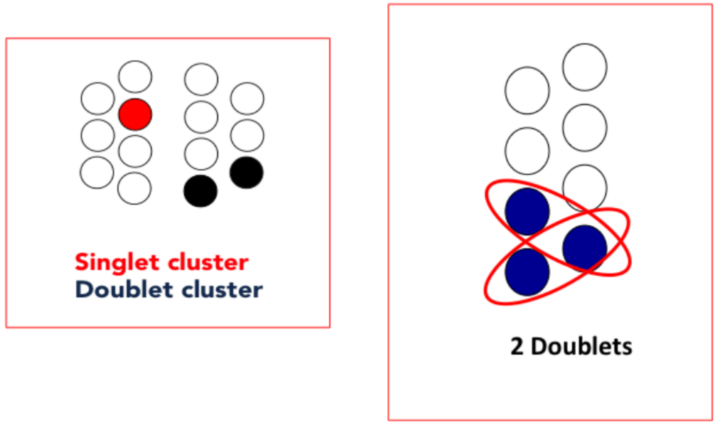
Time island



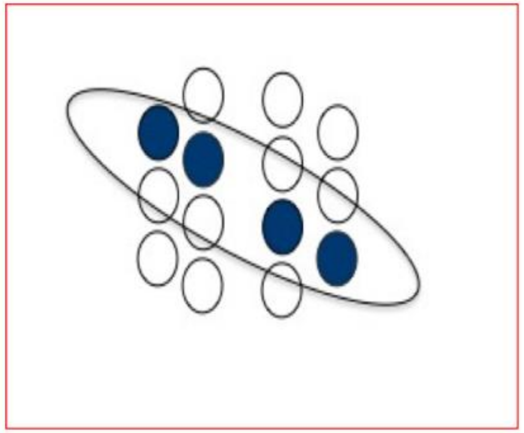
Track Finding & Fit



Clustering

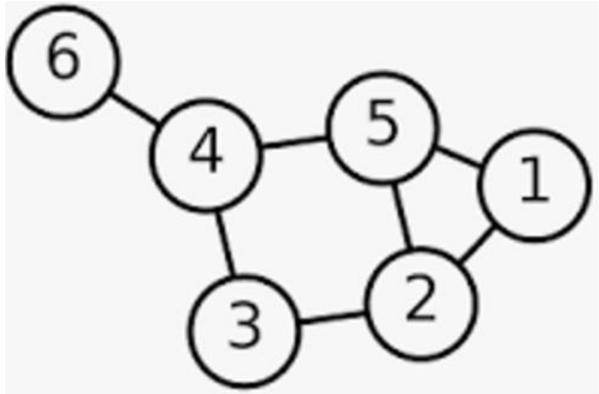


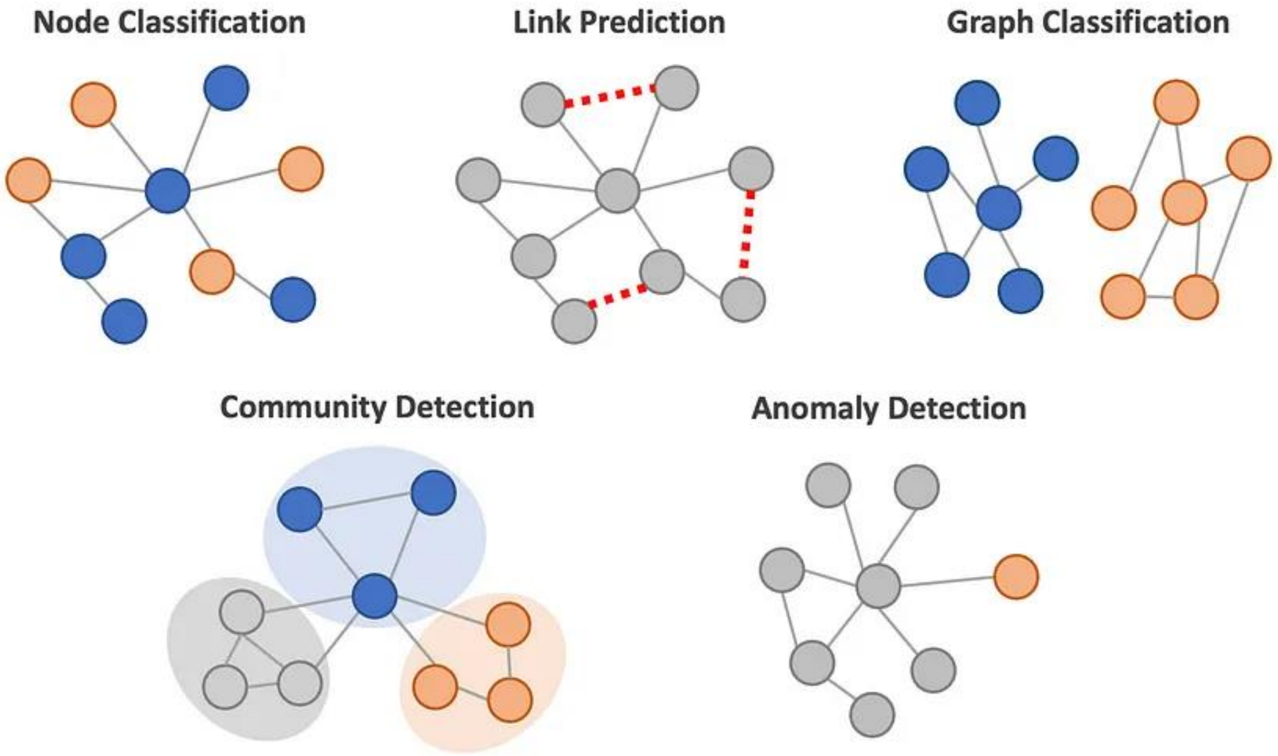
Seeding

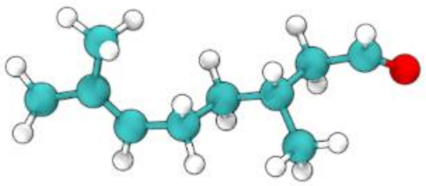


Graph Neural Network

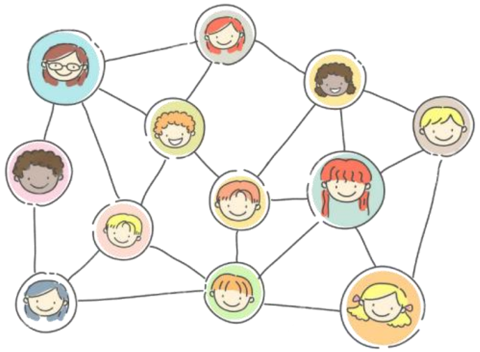
A graph represents the relations (edges) between a collection of entities (nodes)







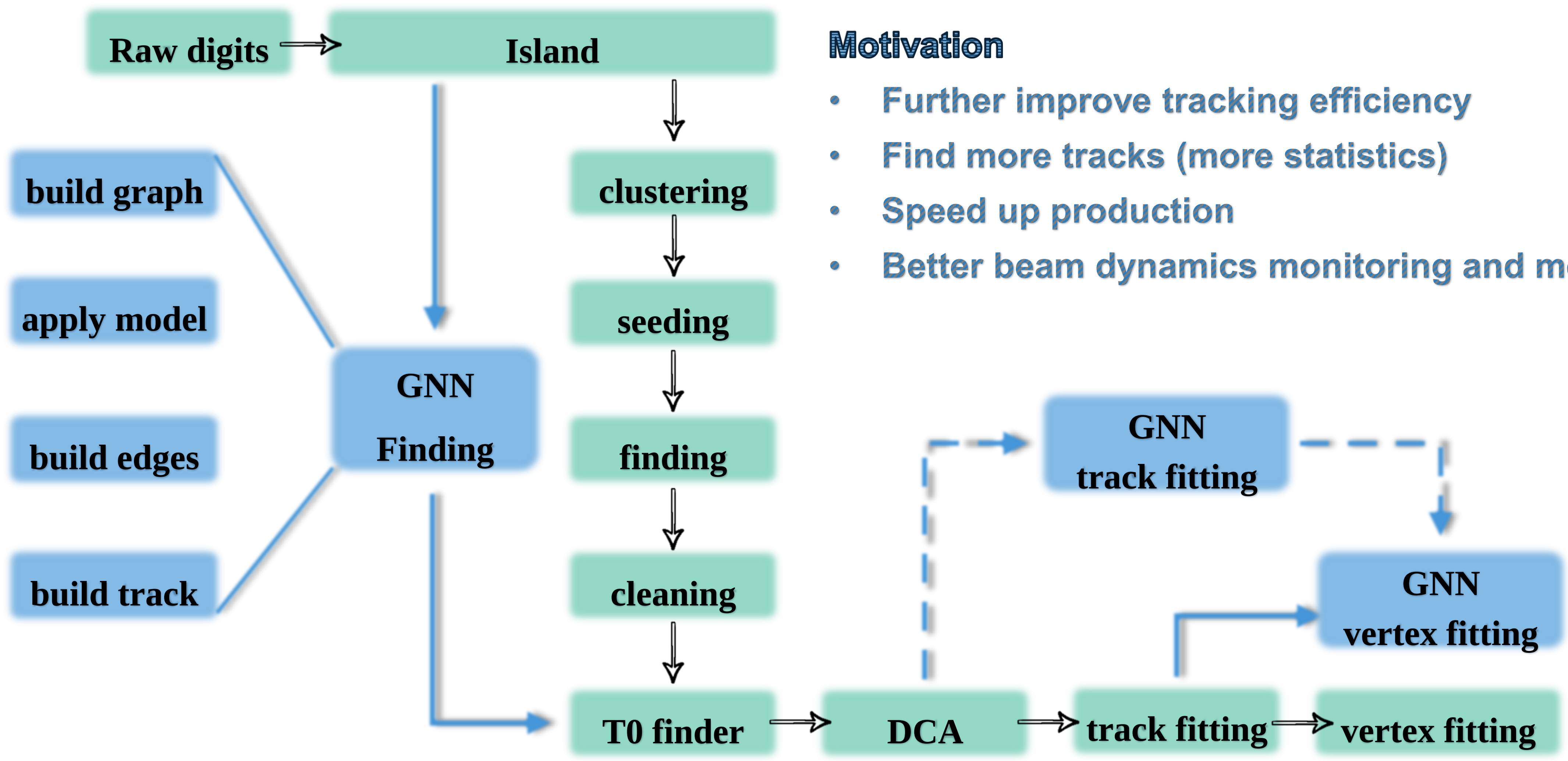
Molecules as graphs



Social networks as graphs

之江实验室 ZHEJIANG LAB

GNN V.S. official workflow



Motivation

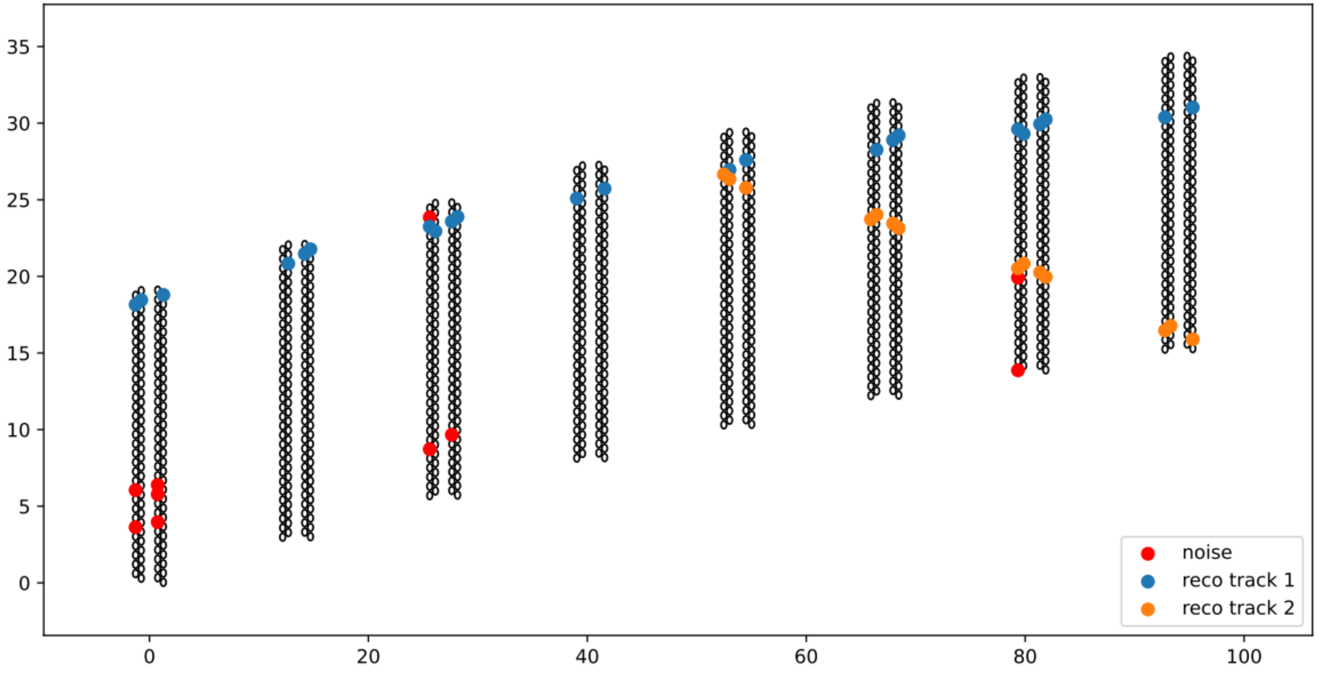
- Further improve tracking efficiency
- Find more tracks (more statistics)
- Speed up production
- Better beam dynamics monitoring and measurement

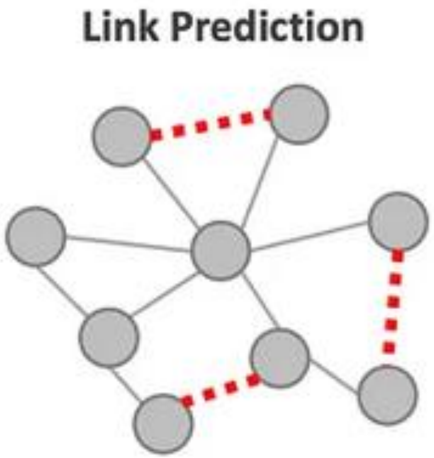
之江实验室 ZHEJIANG LAB

Datasets

A ML model is only as good as the data it is trained on

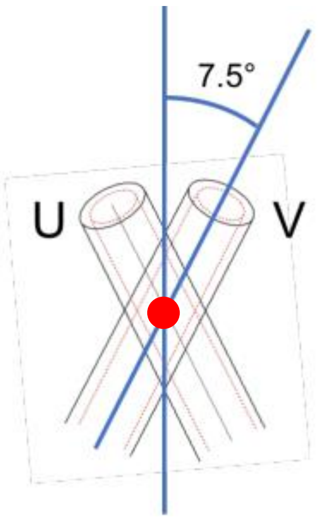
- Use part of Run3 track data as the dataset: 150,000 time island
- Graph Nodes: track hits
- Graph Edges: relation between every two track hits
- Task: Edge Classification + Node Clustering = Track Finding





之江实验室 ZHEJIANG LAB

Build the Graphs:



Node: tracker hits
Node features: {#layer, #straw, x, y, $t - \bar{t}$, hit width}

x, y are the positions in the cross plane

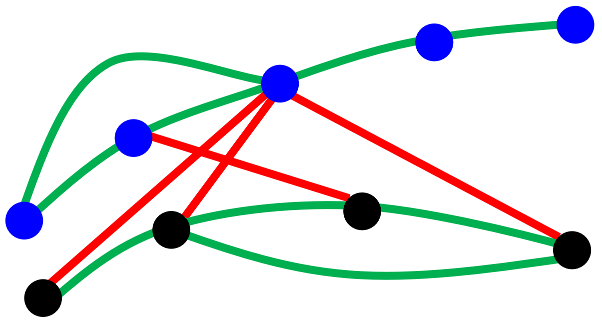
Edge: fully connected edges

Edge features: { |Δlayer|, |Δstraw|, $|\Delta x|$, $|\Delta y|$, $|\Delta(t - \bar{t})|$, | Δ width|, $\frac{|\Delta x|}{\sqrt{\Delta x^2+\Delta y^2}}$, $\frac{|\Delta y|}{\sqrt{\Delta x^2+\Delta y^2}}$ }

Edge label: fake edges / truth edges

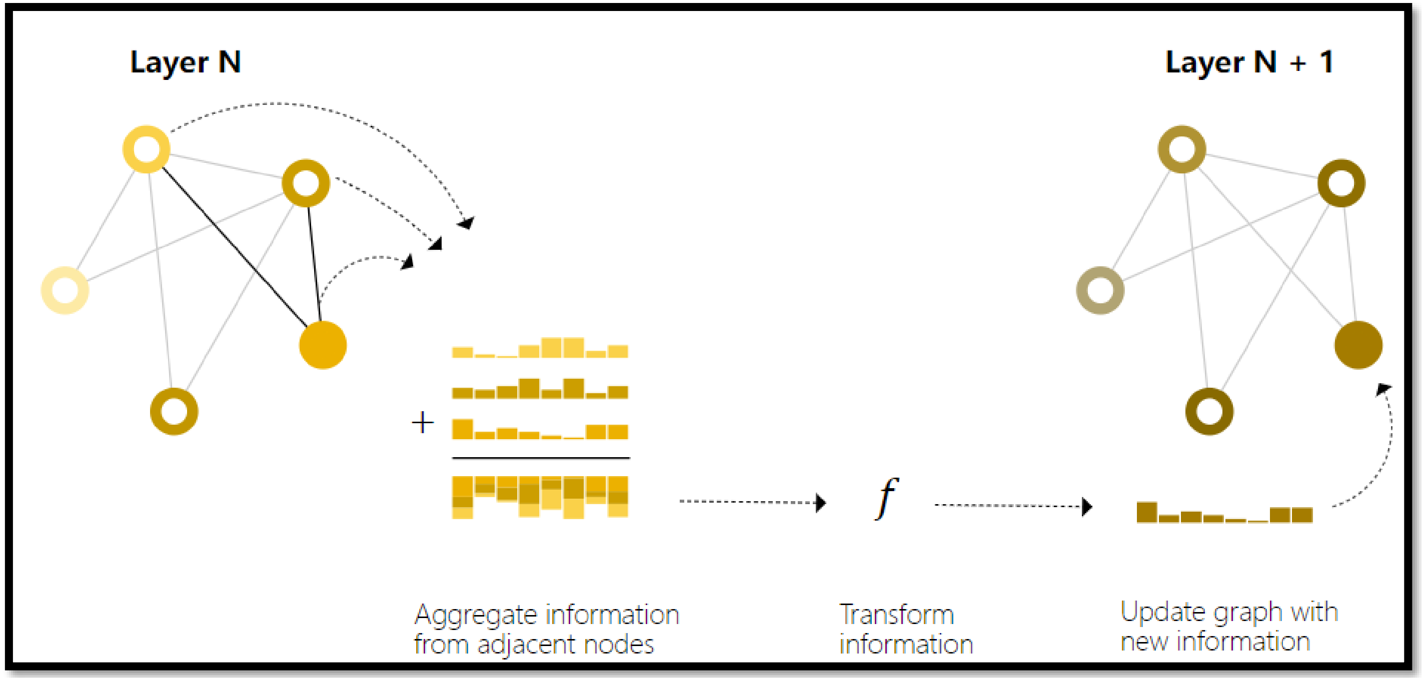
Truth edge:
edge between the nodes which are belong to the same reconstructed track

Fake edge:
edge between the nodes which are not belong to the same track



GNN Track Finding: Message Passing

- **Method: Message Passing**
    1. **For each node in the graph, gather all the neighboring node features**
    2. **Aggregate all features via an aggregate function (like sum)**
    3. **All pooled messages are passed through an update function**



之江实验室 ZHEJIANG LAB

GNN Track Finding: Architecture

1. Embedding the node features and edge features into a high-dimension latent space to get the node representation h^0 and edge representation x^0
2. The messages between node i and node j are generated through a network ϕ_m

$$m_{ij}^l = \phi_e(h_i^l, h_j^l, x^l)$$

3. The edge representation will be updated through a network ϕ_x with residual connection

$$x_i^{l+1} = x_i^l + \phi_x(m_{ij}^l) \cdot x_i^l$$

4. The node representation will be updated through a network ϕ_h with residual connection

$$h_i^{l+1} = h_i^l + \phi_h\left(h_i^l, \sum w_{ij} m_{ij}^l\right) \text{ where } w_{ij} = \phi_w(m_{ij}^l)$$

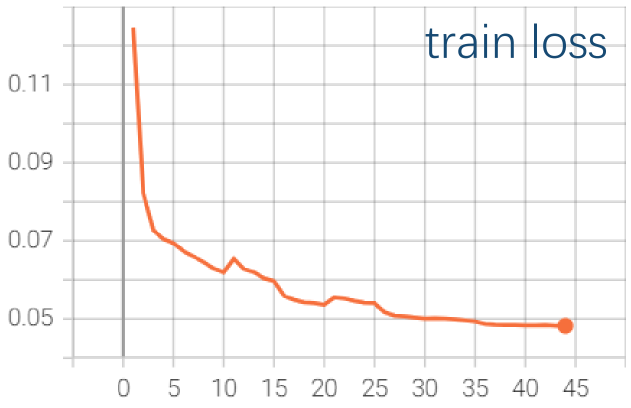
5. Repeat step 2-4 for L layer, then use a classifier network ϕ_c to judge whether the edges exist between node i and node j: $s_{ij} = \phi_c(h_i^L, h_j^L)$

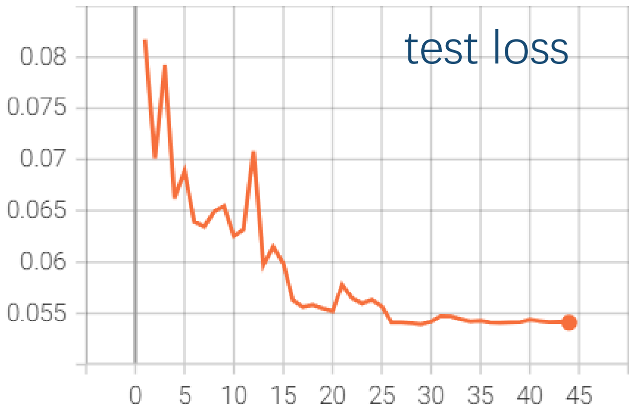
- Loss Function: Binary Cross Entropy loss

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(p_i) + (1 - y_i) log(1 - p_i)]$$

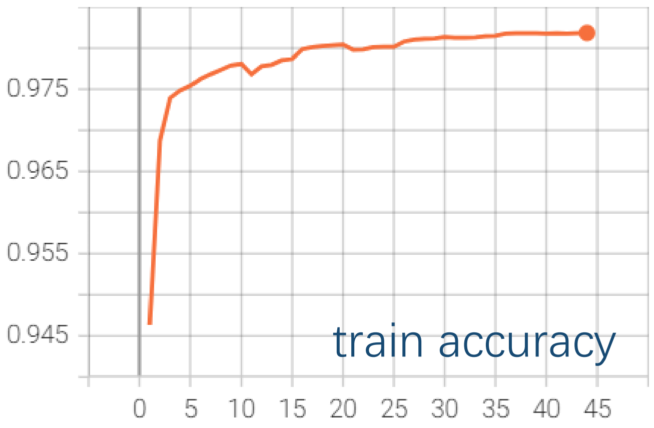
GNN Track Finding: Edge Classification

Training and Performance of the GNN edge classification

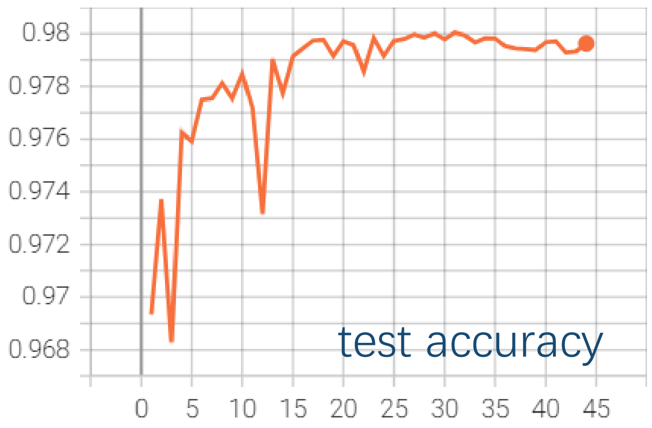




train accuracy vs ep



test accuracy



- **Quickly converge around 25 epoch**
- **Edge classification accuracy: 98%**

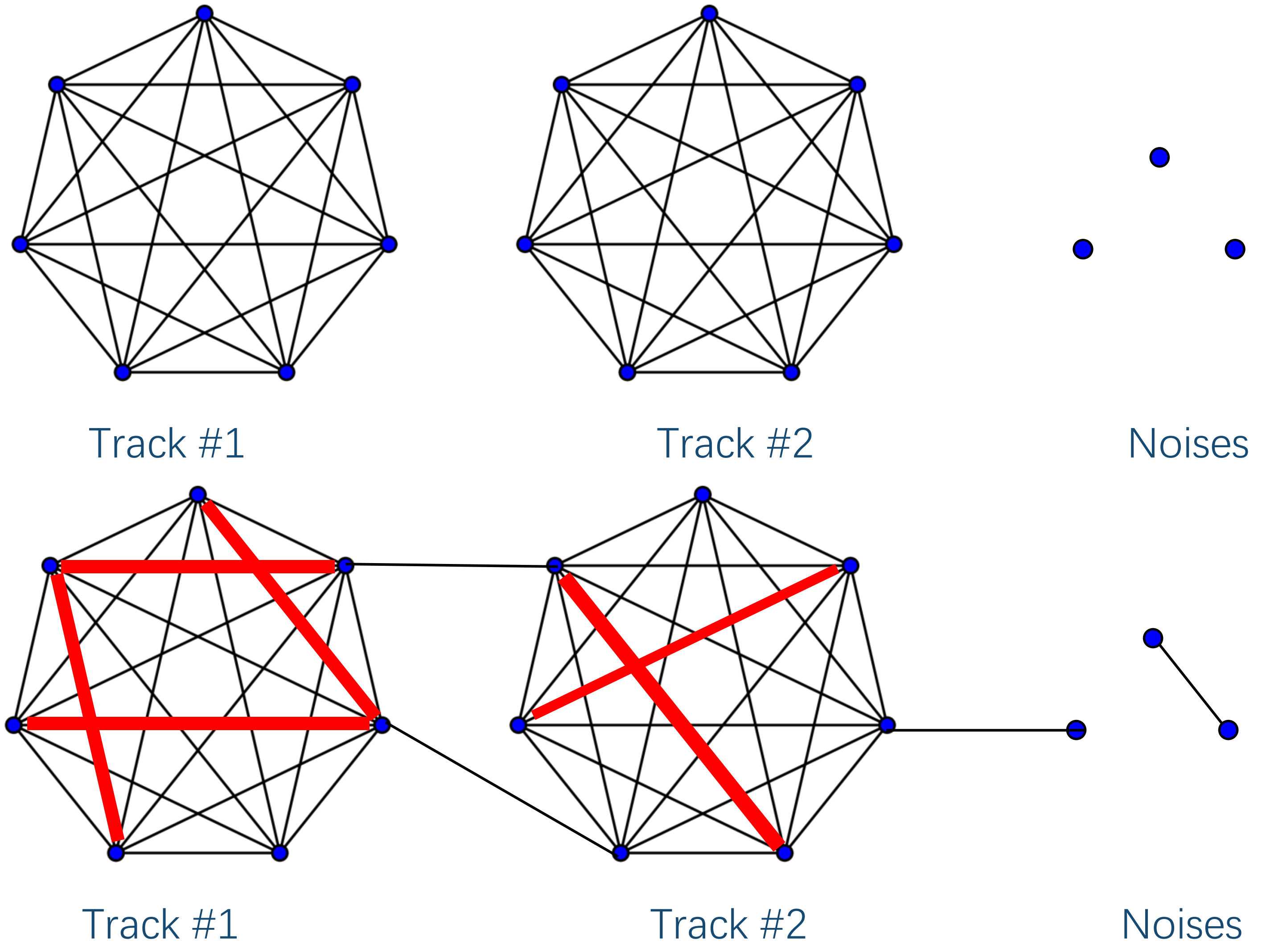
之江实验室 ZHEJIANG LAB

GNN Track Finding: Node Clustering

Remove the fake edges according to the GNN model

Perfect condition:
- All fake edges are removed
- All hits belong to a track are connected to each other
- No link connection between different tracks and noises

Practice condition: 98% accuracy
- truth edges → fake edges, be removed
- fake edges → truth edges, remaining
- Hard to use a simple complete condition cut to form track



之江实验室 ZHEJIANG LAB

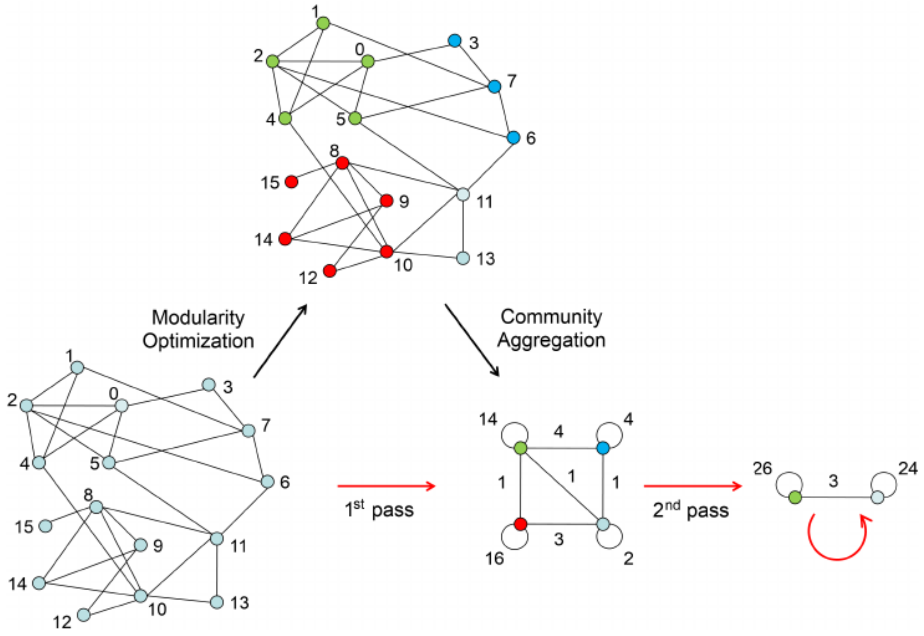
GNN Track Finding: Node Clustering

- The fully connected graph method has both high efficiency and robustness:
  - ✓ The edge accuracy is ~98%
  - ✓ Even one hit disconnected with its neighbor hit, it not likely loss connection with all the other hits in the same track
  - ✓ The noise hit is hard to get connection with most/all the track hits

Adopt a classical graph analysis method to select the tracks from the output graph:

Louvain Algorithm

Louvain algorithm is an unsupervised community discovery algorithm based on modularity: $Q = \sum_c [\frac{\Sigma_{in}}{2m} - (\frac{\Sigma_{tot}}{2m})^2]$

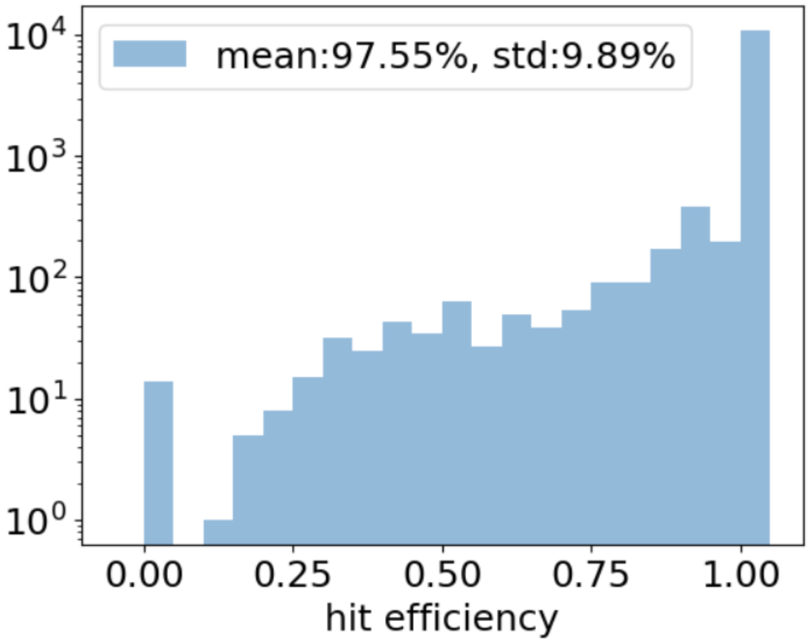


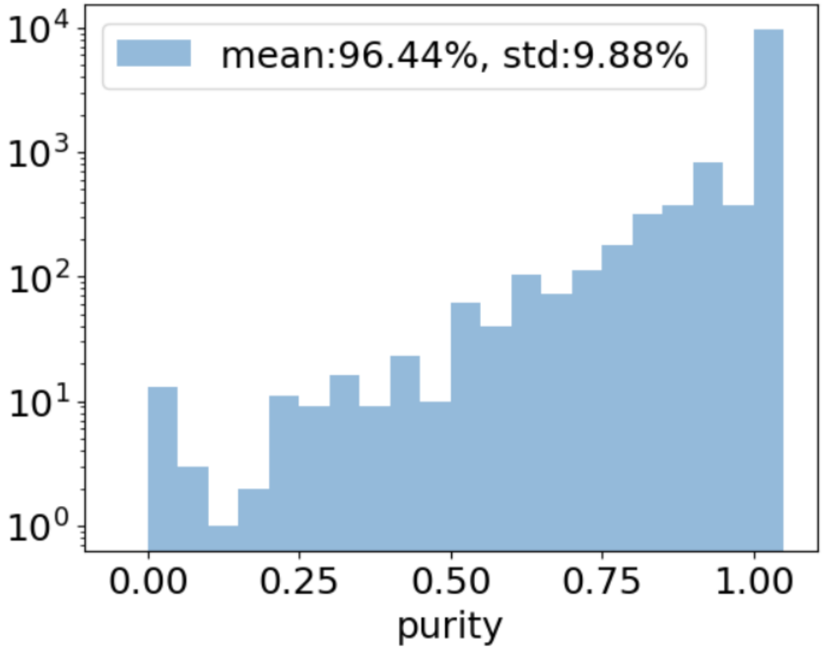
之江实验室 ZHEJIANG LAB

GNN Track Finding: Efficiency and Purity

$$\varepsilon = \frac{N_{GNN\ matched\ hits\ in\ a\ track}}{N_{total\ truth\ hits\ in\ this\ track}}$$

$$p = \frac{N_{GNN\ matched\ hits\ in\ a\ track}}{N_{total\ GNN\ predicted\ hits\ in\ this\ track}}$$

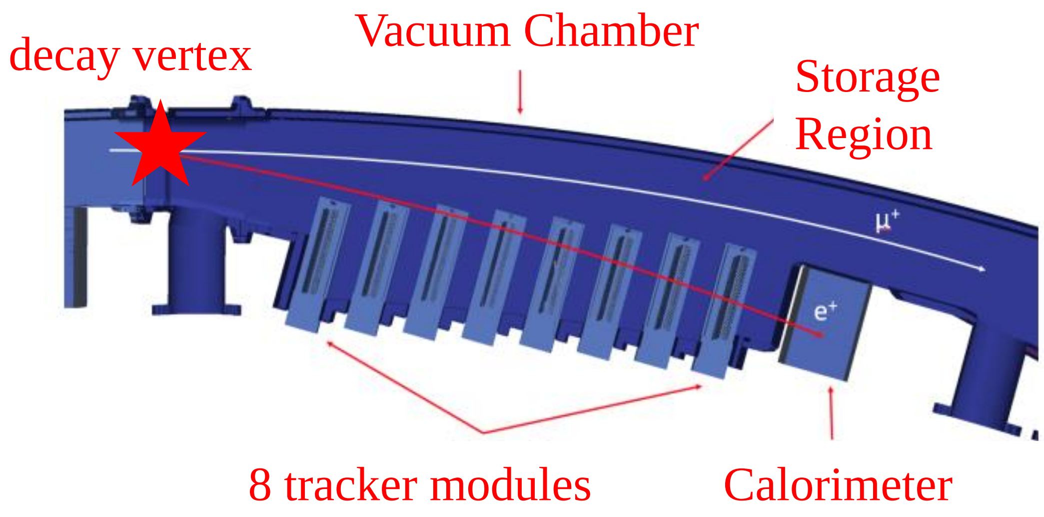




- **Promising result:**
  - ✓ **Efficiency and purity are both >95%**
  - ✓ **In the simulation study, the default track method found more #tracks (106% compared to the truth)**
  - ✓ **The GNN method found 94% #tracks compared to the default track method**
  - ✓ **The GNN method seems to fix the overestimation (94%×106%=99.6%) using the default track recon data**
  - ✓ **Will do more detailed study to further prove that**

之江实验室 ZHEJIANG LAB

GNN Vertex Fitting:

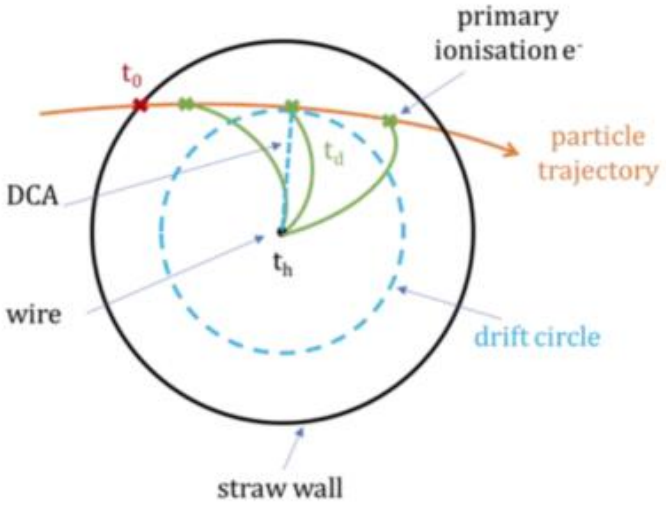


- Use fitted tracks to extrapolate the vertex
- Use GEANE package in GEANT4 to do the extrapolation: **2.5s process ~400 time island**
- Want to use GNN and GPU to speed up this progress: **12G 2080Ti: 8s ~ 35,000 time island, ~25x faster**
- Increase the memory and update the GPU can make it even faster (100x or even 1000x)

Input data:

- Track finding features
- Add high level features

(momentum, position and drift time, DCA)



之江实验室 ZHEJIANG LAB

GNN Vertex Fitting: Architecture

- Same structures like finding step 1 to 4
- Aggregate the final node representation h^L and then use a regression network to predict the vertex info (time, position, momentum)
- 3 individual GNN are trained to extract time, position and momentum separately

Regression Loss Function:

Huber Loss:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, & |y - f(x)| \le \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & |y - f(x)| > \delta \end{cases}$$

SmoothL1 Loss:

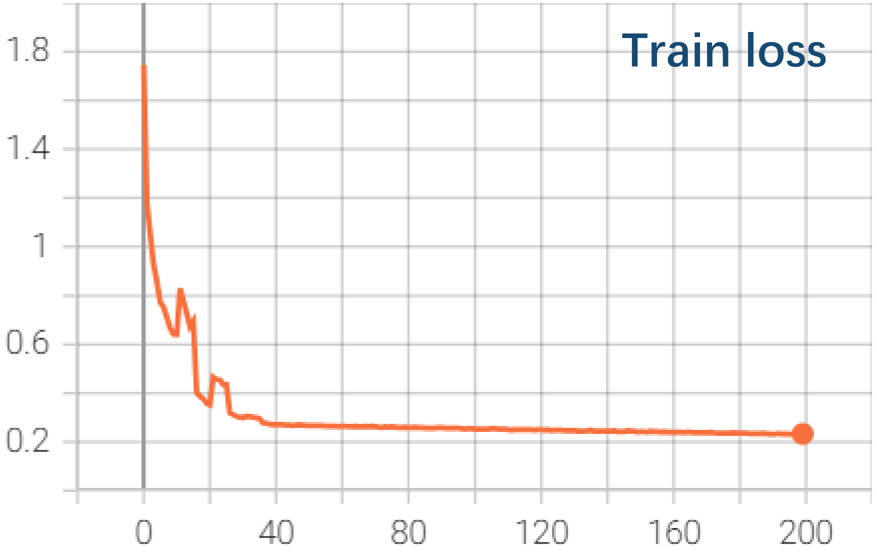
$$loss(x, y) = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} .5 * (y_i - f(x_i))^2, & if\ |y_i - f(x_i)| < 1 \\ |y_i - f(x_i)| - 0.5, & otherwise \end{cases}$$

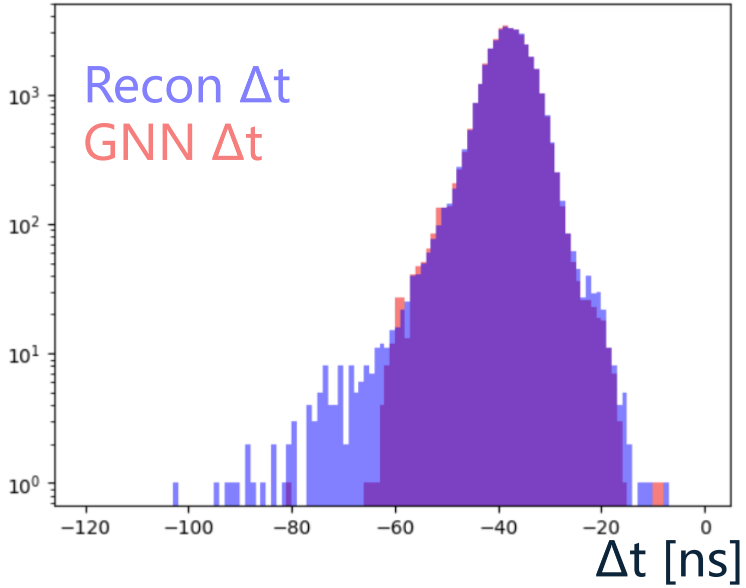
Log Cosh Loss:

$$L(y, y^p) = \sum_{i=1}^{n} \log(\cosh(y_i^p - y_i))$$

之江实验室 ZHEJIANG LAB

GNN Vertex Fitting: Time

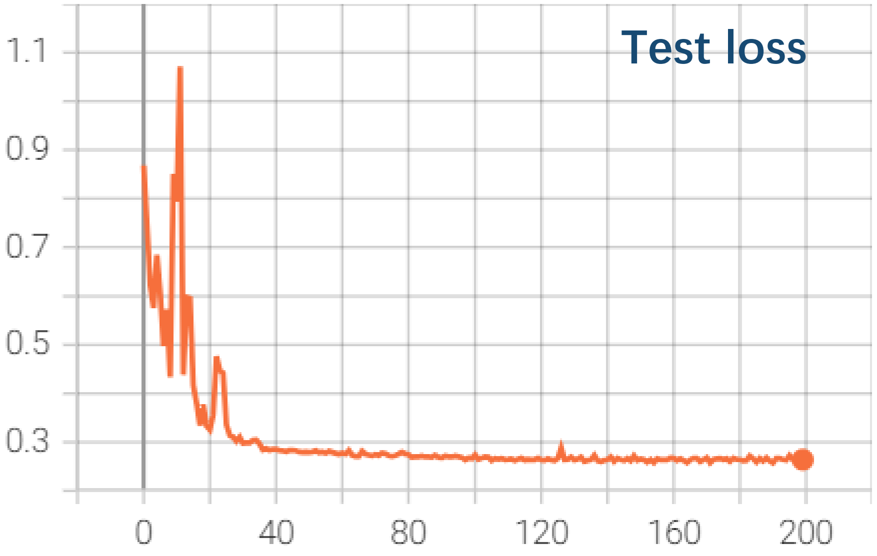


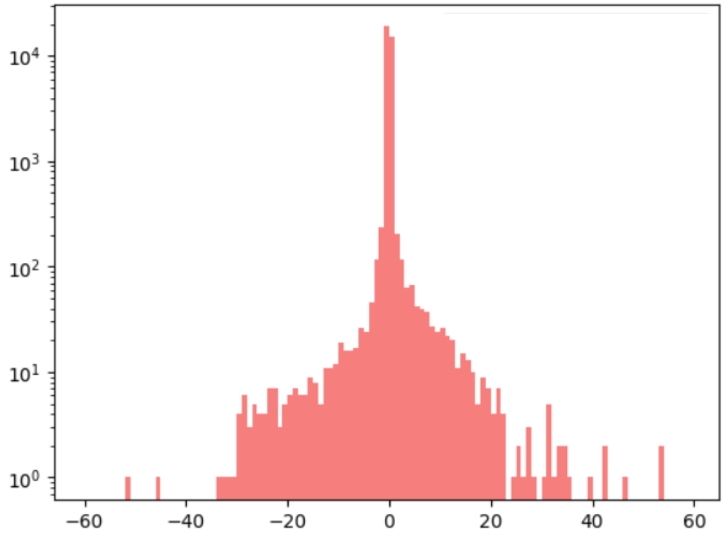


$\Delta t = t_v - \bar{t}_t$

t_v : vertex time

$\bar{t}_t$: mean time of track



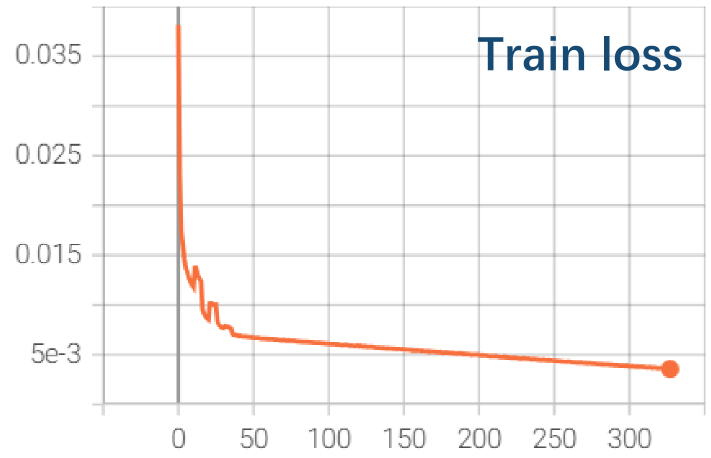


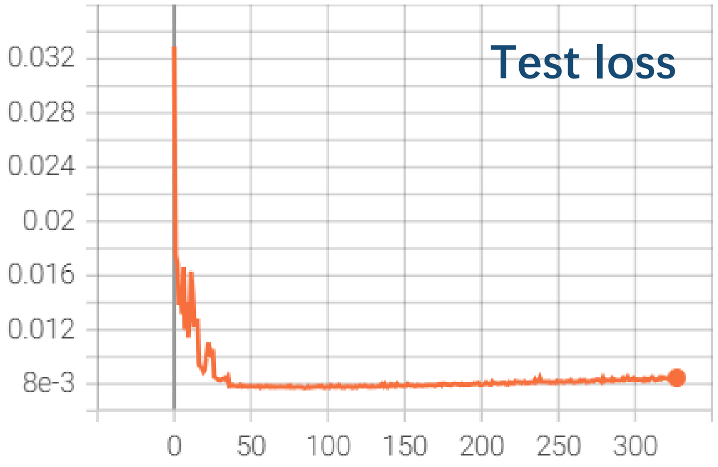
Difference between GNN and recon vertex time: 0.03ns $\pm$ 2.01ns

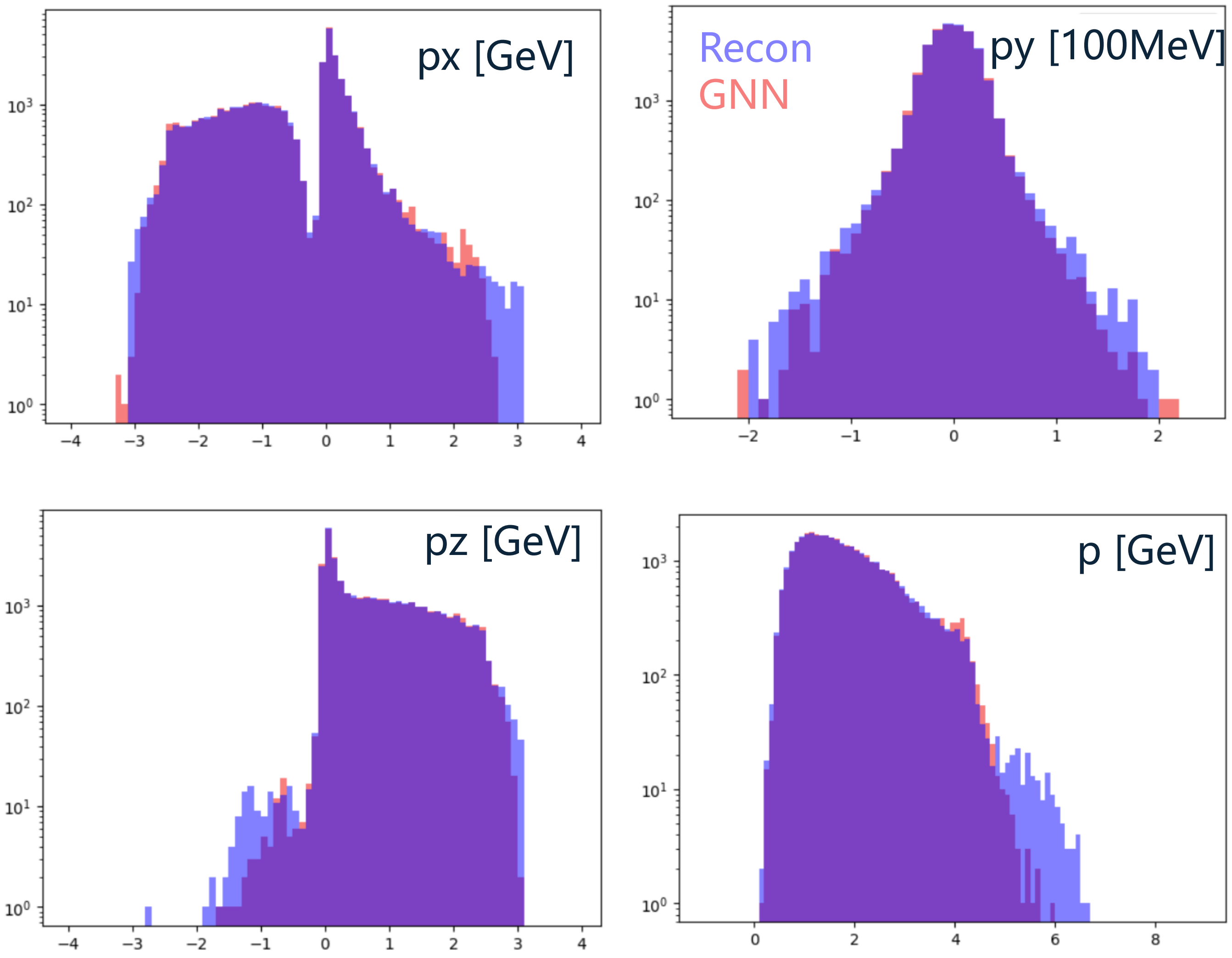
Similar time resolution between recon and simulation

之江实验室 ZHEJIANG LAB

GNN Vertex Fitting: Momentum







- Distributions of momentum are almost similar, mismatch mainly at the tail
- The mean Δmomentum are <5 MeV level, the RMS are at ~100MeV level

Summary and Outlook

- Tracker is one of the key detector system component of the muon g-2 experiment
  - ✓ Tracking measurement and reconstruction paly a important role in various aspects of the experiment: Field, Beam Dynamics ...
- GNN in tracking is motivated by current limitations in tracking efficiency and speed
- Developed a fully connected GNN workflow (Finding & Fitting) to solve the challenges
  - ✓ Preliminary result based on reconstruction data is promising
- Further studies needed:
  - ✓ GNN for Track fitting
  - ✓ Verify the GNN workflow in simulation data
  - ✓ Test the feasibility of GNN methods in analysis
  - ✓ Rather than just staying at the training itself

Thanks