A TRACK FINDER ALGORITHM BASED ON GNN FOR THE MEG II EXPERIMENT

A. Venturini[1], A. M. Baldini[1], H. Benmansour [1], L. Bianco [1], G. Cavoto[2], F. Cei[1], M. Chiappini[1], L. Dispoto[1], M. Francesconi[3], L. Galli[1], G. Gallucci[1], F. Grancagnolo[4], E. G. Grandoni[1], M. Grassi[1], F. Ignatov[5], F. Leonetti[1], A. Oya[6], D. Nicolò[1], M. Panareo[4], A. Papa [1], F. Renga[2], S. Scarpellini[2], M. E. Tegano[1], Y. Uchiyama[6], and C. Voena[2]

[1]INFN Sezione di Pisa; Università di Pisa (Italy), [2]INFN Sezione di Roma; Università di Roma "Sapienza" (Italy), [3]INFN Sezione di Napoli (Italy), [4]INFN Sezione di Lecce; Università del Salento (Italy) , [5]Oliver Lodge Laboratory, Liverpool (UK) , [6]ICEPP, The University of Tokyo (Japan)

Introduction

The MEG II experiment is searching for the forbidden $\mu \to e\gamma$ decay with record sensitivity.
Central for the experiment success is the e^+ spectrometer, composed of a Cylindrical Drift Chamber (CDCH), a pixeled timing layer (pTC) and a superconducting magnet.
We discuss the implementation of a Graph Neural Network (GNN) classifier capable of improving the MEG II track finder, a track following algorithm with Kalman Filter, strongly reducing pile-up inefficiencies.

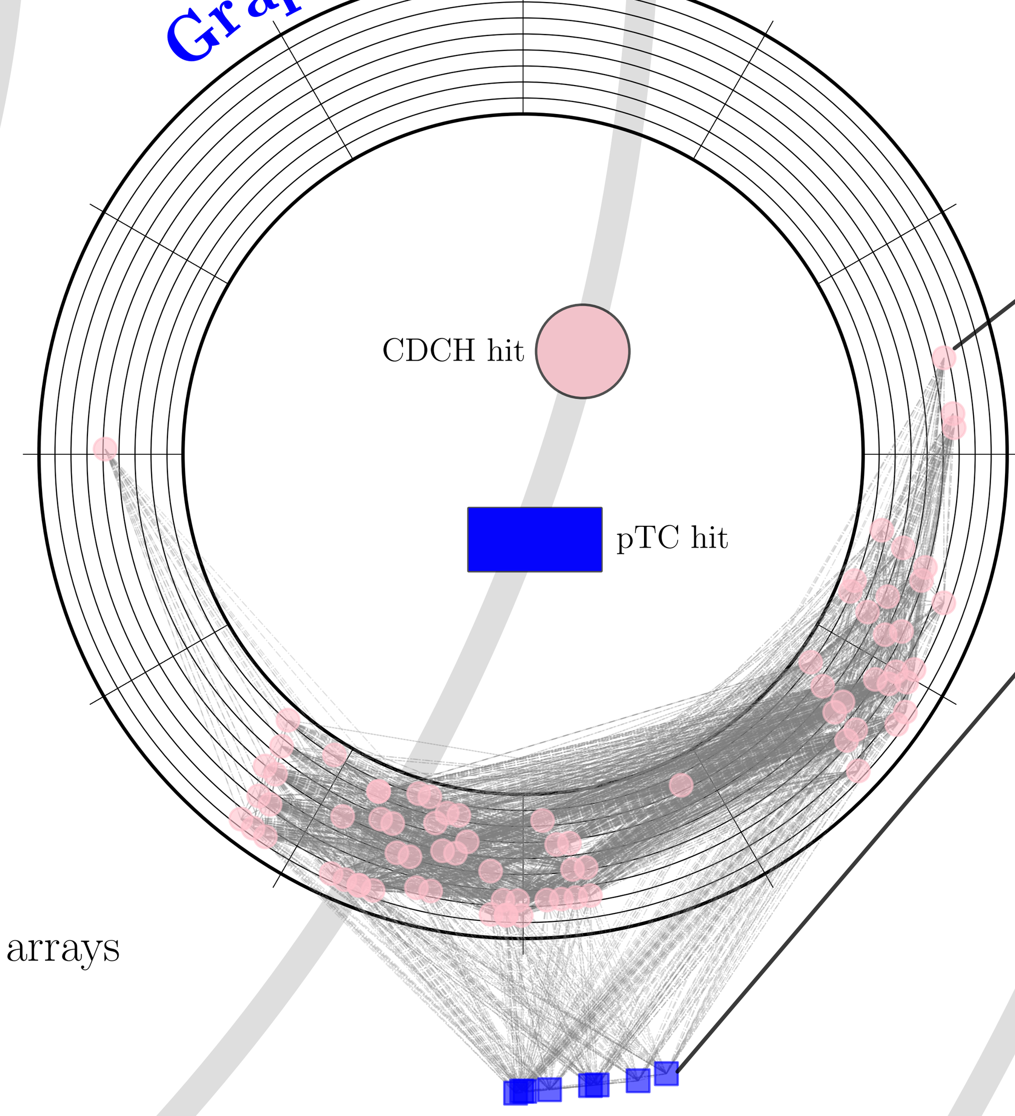
The CDCH:

- Single volume, ultra-light drift chamber
- 1728 sense wires in 9 concentrical layers
- Full stereo geometry

The pTC:

- 512 scintillating tiles ($12 \times 4 \times 0.5$ cm^3)readout with SiPM arrays
- $\sigma_{t_e} \approx 80$ ps for a single tile

Graph Building



Node Features:

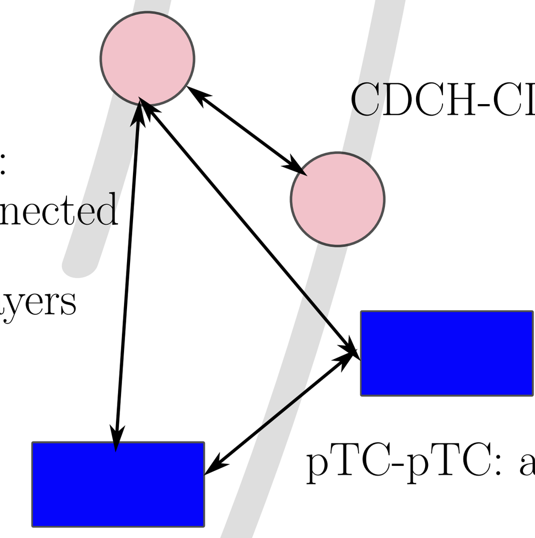
$$\vec{x}_i^{CDCH} = \{\overbrace{x_0, y_0, \phi, \theta}^{\text{wire geometry}}, \overbrace{z, t, \text{amplitude}}^{\text{CDCH hit features}}\}$$

$$\vec{x}_i^{pTC} = \{\overbrace{x, y, z, t}^{\text{pTC hit features}}\}$$

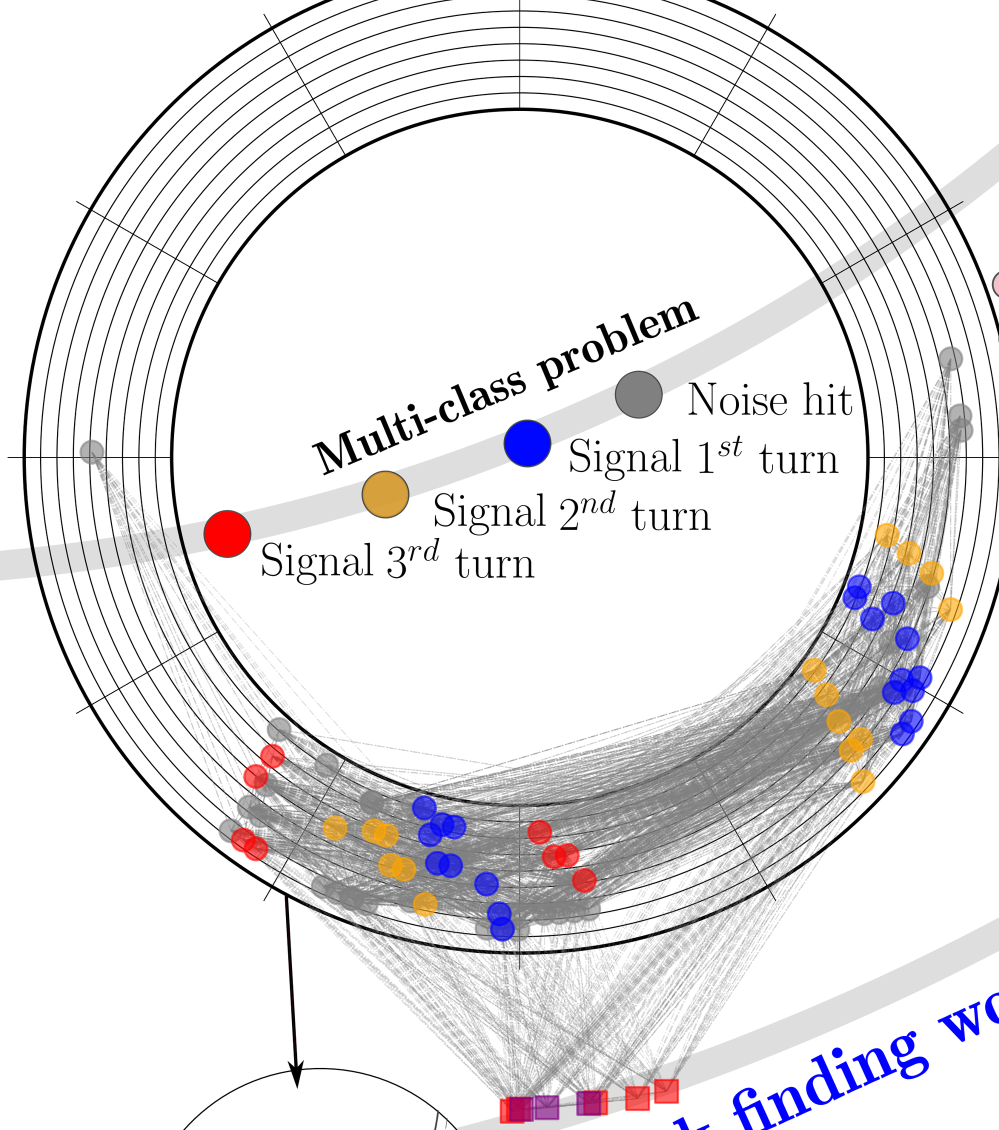
Edge features:

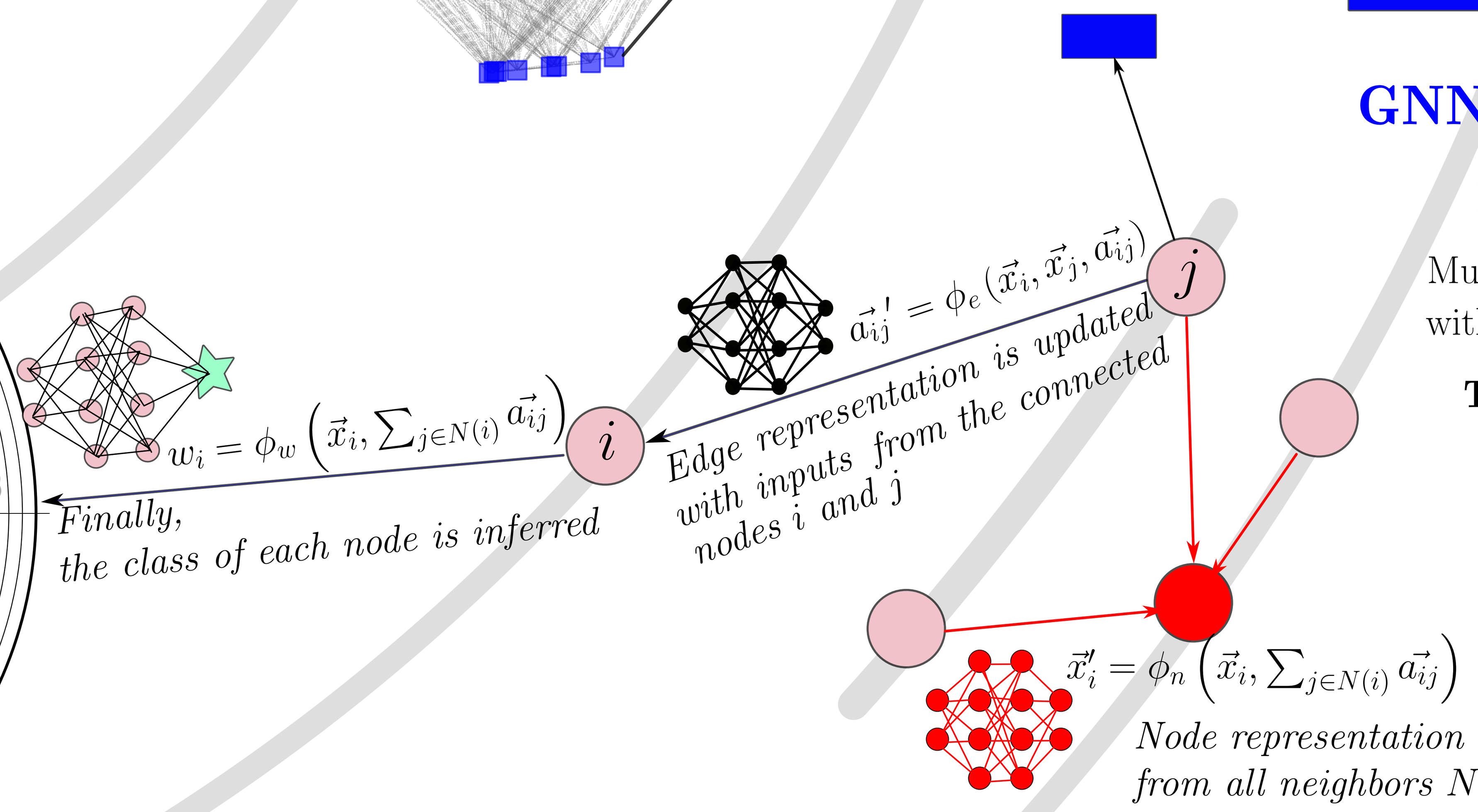
$$\vec{a_{ij}} = \{\Delta x_{ij},\ \Delta y_{ij},\ \Delta t_{ij}\}$$

Edge Connectivity:



Node inference





GNN Architecture for MEG II

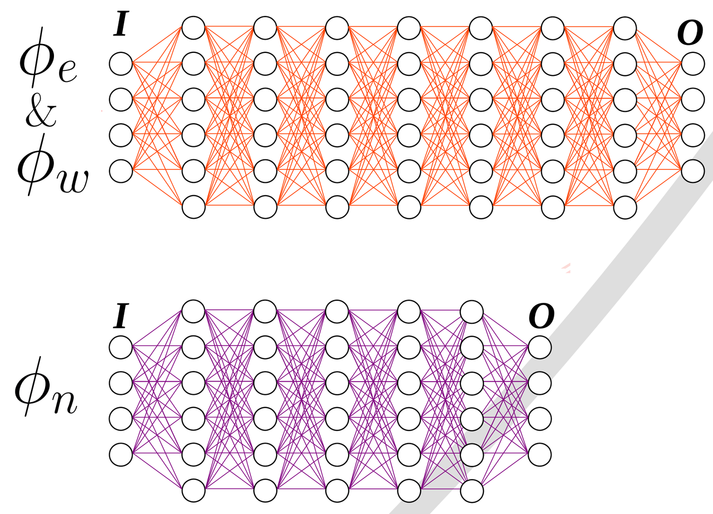
Heterogeneous Graph:

Multiple detector types (pTC, CDCH) with different features.

Three MLP Networks:

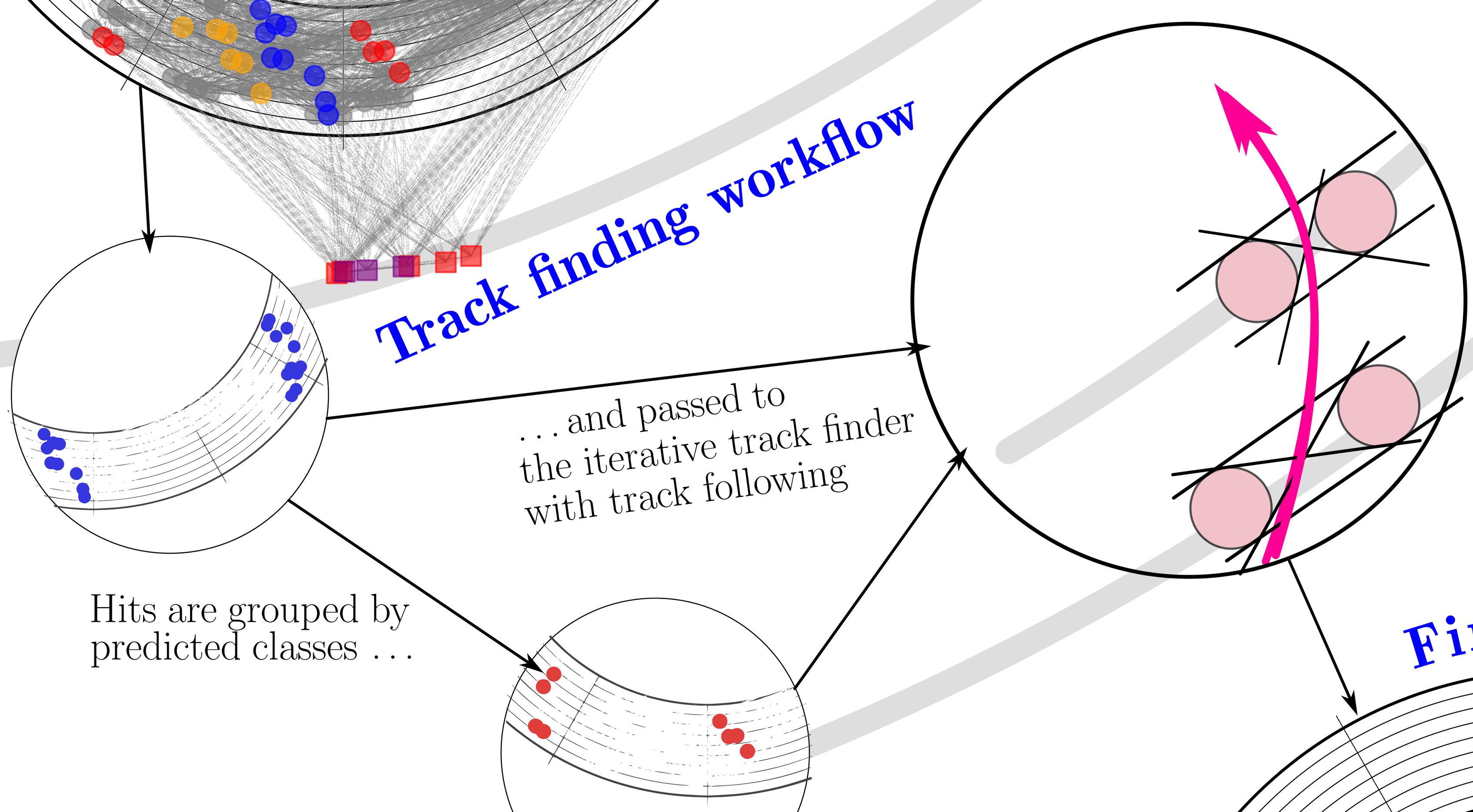
- ϕ_e: Edge update (4 networks)
- ϕ_n: Node representation (2 networks)
- ϕ_w: Classification (2 networks)

Network Architecture:

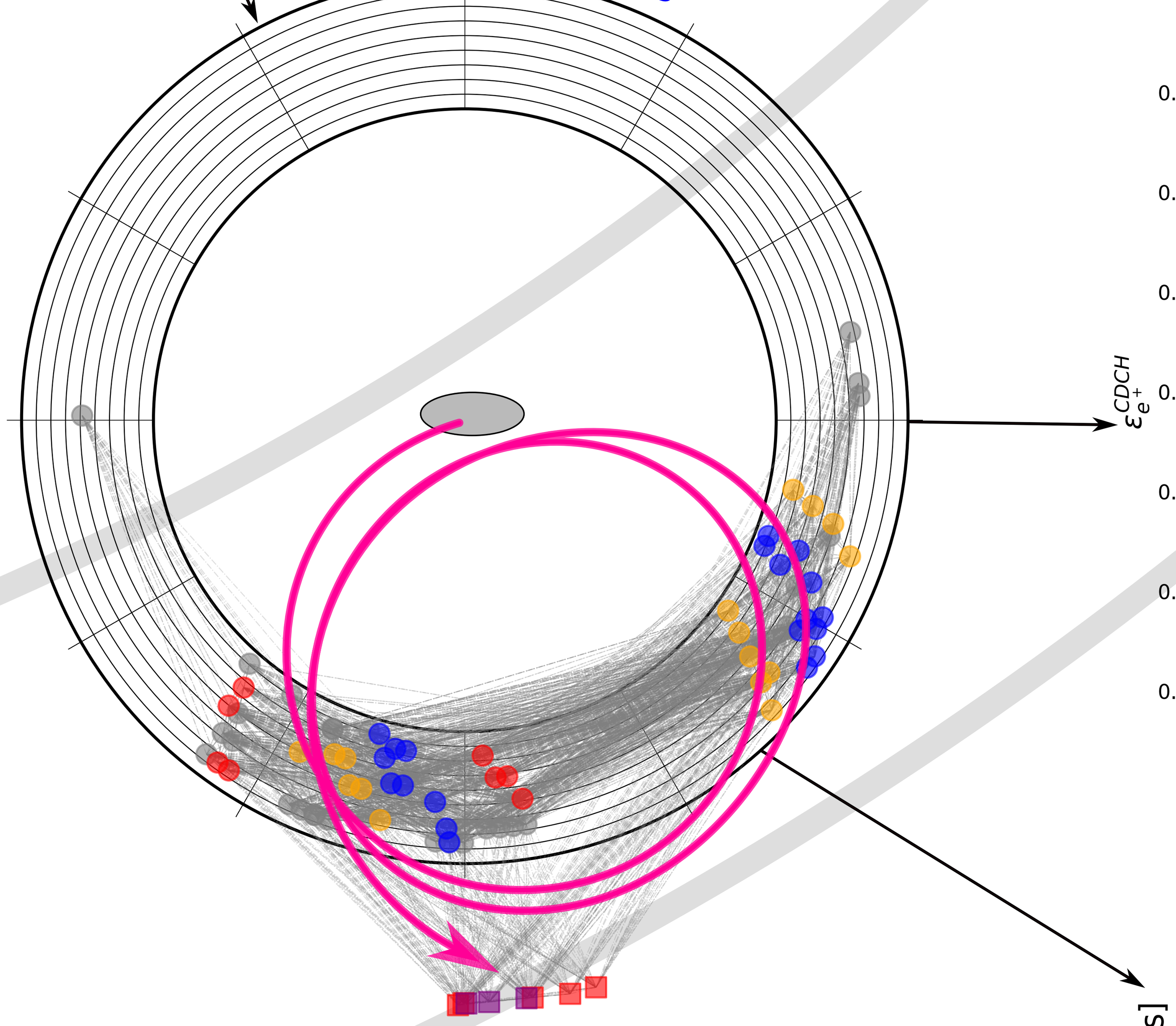


- ϕ_e & ϕ_w: 7 fully con
- ϕ_n: 5 fully connected
- 75 nodes per layer
- ReLU activation +
- Message passing up
- Built with PyTorch and `pytorch-geometric` library
- GNN integrated in the MEG II framework (C++ based) with ONNX

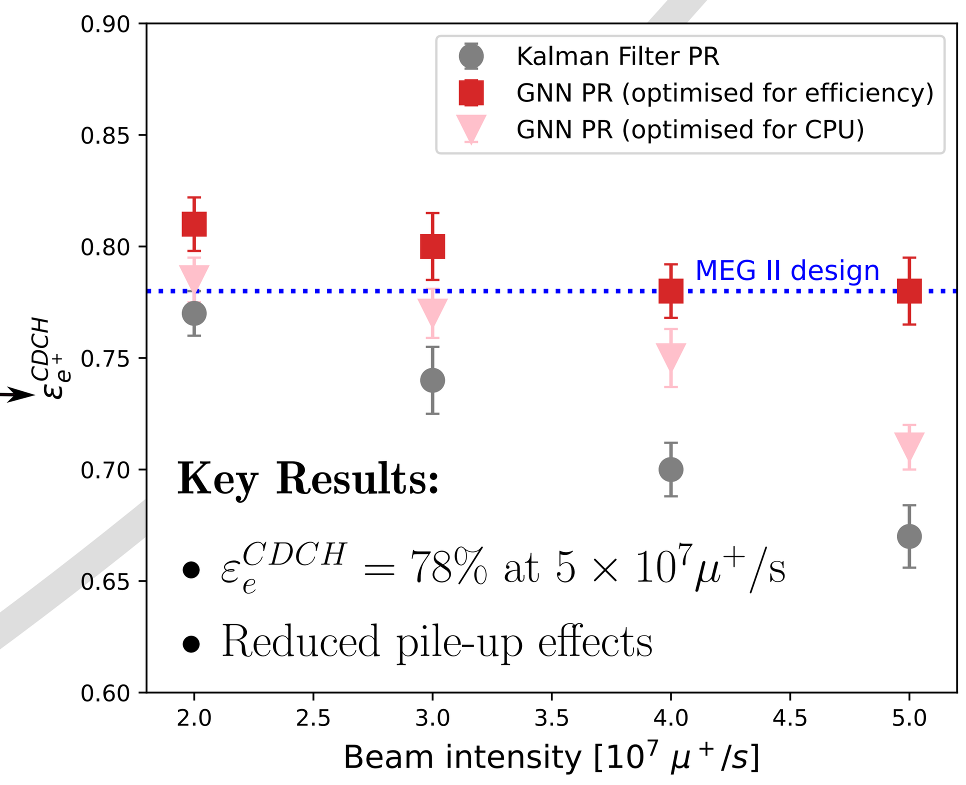
Track finding workflow



Final tracking



Tracking Efficiency Results



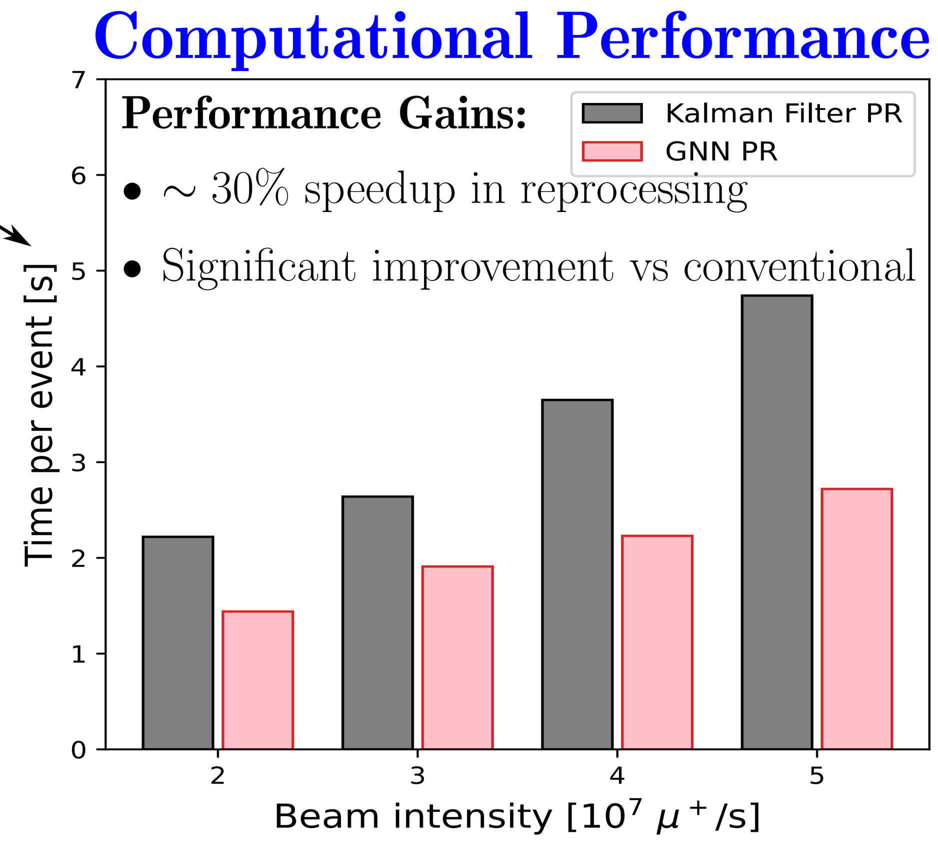
Key Results:

- $\varepsilon_e^{CDCH} = 78\%$ at $5 \times 10^7 \mu^+/s$
- Reduced pile-up effects

Computational Performance

Performance Gains:

- $\sim 30\%$ speedup in reprocessing
- Significant improvement vs conventional



Training Details

Dataset:

- $\gtrsim$ 100k events of real data collected at $5 \times 10^7 \mu^+/s$ beam intensity
- Supervised learning: labeling via conventional reconstruction
- Skip events with zero tracks to avoid biases

Validation

- Validation dataset:20k events at $5 \times 10^7\ \mu^+/s$
- $\sim 70\%$ of noise hits are correctly discarded
- $> 90\%$ of true signal hits are correctly forwarded to the next stage
- Accuracy around $85\%-90\%$ over all categories

Predicted Labels \ True Labels	Noise	Turn1	Turn2	Turn3	Turn4
Noise	66.71%	6.73%	7.48%	7.75%	7.56%
Signal	33.29%	93.27%	92.52%	92.25%	92.44%