University and INFN of Ferrara

Updates and final results on the Quantum Graph Neural Network project for particle tracking in HEP

3rd Workshop on Quantum Computing @ INFN

Matteo Argenton
margenton@fe.infn.it

Laura Cappelli

Concezio Bozzi

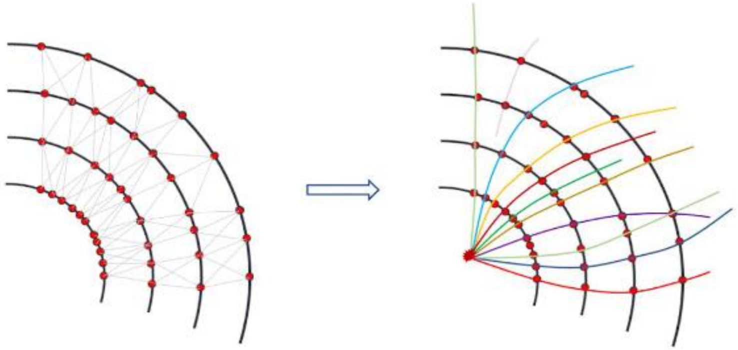
ICSC
Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

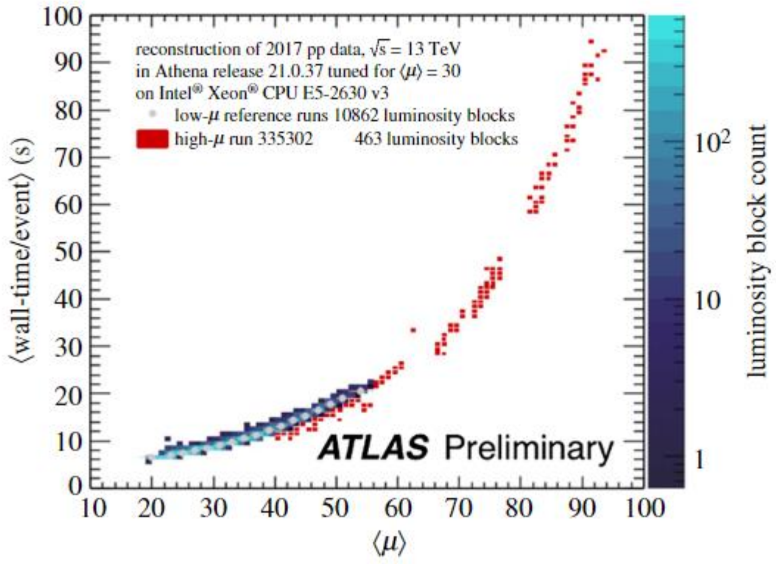
FERRARIAE UNIVERSITAS · EX LABORE FRUCTUS · 1391

INFN
Istituto Nazionale di Fisica Nucleare

Particle tracking

Tracking, the reconstruction of particle trajectories starting from particle hits in different trackers' layers, is already a **computationally demanding task** in the major experiments at CERN

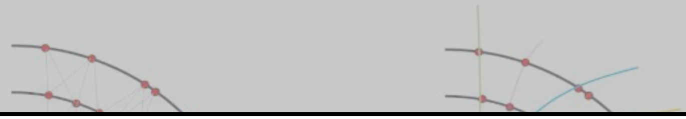




- LHC is entering High-Luminosity in Run 4.
- Particle tracking in CMS and ATLAS will need to account for an increment of a **factor 3-5 in the number of primary proton-proton interactions, averaging 140-200** in future runs.
- Although targeting different levels of luminosity, planned upgrades for the LHCb and ALICE detectors in Run5.

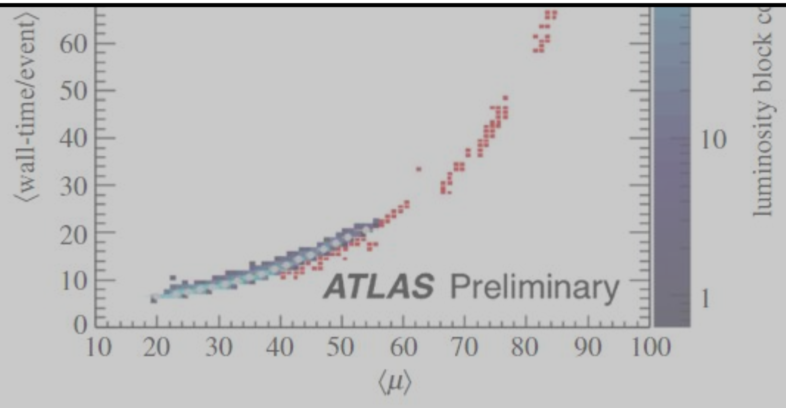
Particle tracking

Tracking, the reconstruction of particle trajectories



Research in particle tracking is very much alive, and oriented to new technologies

- Modern ML methods (e.g. GNNs, transformers, …)
- Quantum technologies (QGNNs, non-variational methods, …)



number of primary proton-proton interactions, averaging 140-200 in future runs.

- Although targeting different levels of luminosity, planned upgrades for the LHCb and ALICE detectors in Run5.

The Quantum Computing perspective

Why:

QC offers a an entirely new computing paradigm with built-in

- **parallelism** (coherent action on superimposed states)
- **exponentially-scaling** Hilbert space in the **linear** number of qubits

But:

QC is in its NISQ (Noisy Intermediate Scale Quantum) era:

- **Limited number of qubits and quantum logic gates (depth)**
- **Error mitigation but no error correction (no fault tolerance)**

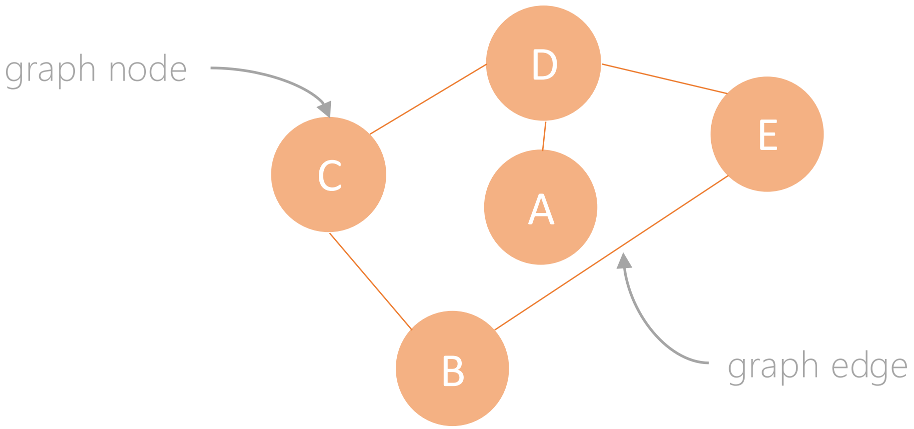
- **Quantum machine learning (QML) as a framework to learn from data using QC**
- **Variational QML algorithms interesting in NISQ computing**
  - QNNs and in this case hybrid QGNNs [1]

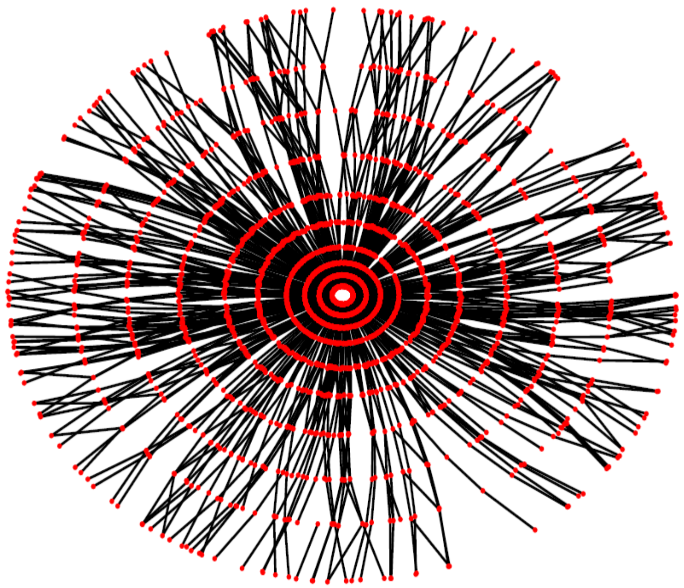
[1] first implemented by Tüysüz et. Al. https://doi.org/10.48550/arXiv.2109.12636

QGNNs for particle tracking

Input: graphs

A collision event can be represented as a graph:





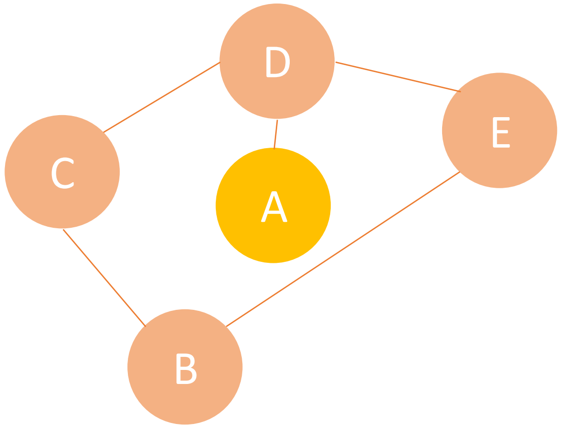
this is a simple graph → **nodes** → hits in a detector layer

this is a simple graph → **edges** → track segments

HEP graphs → hits in a detector layer, track segments

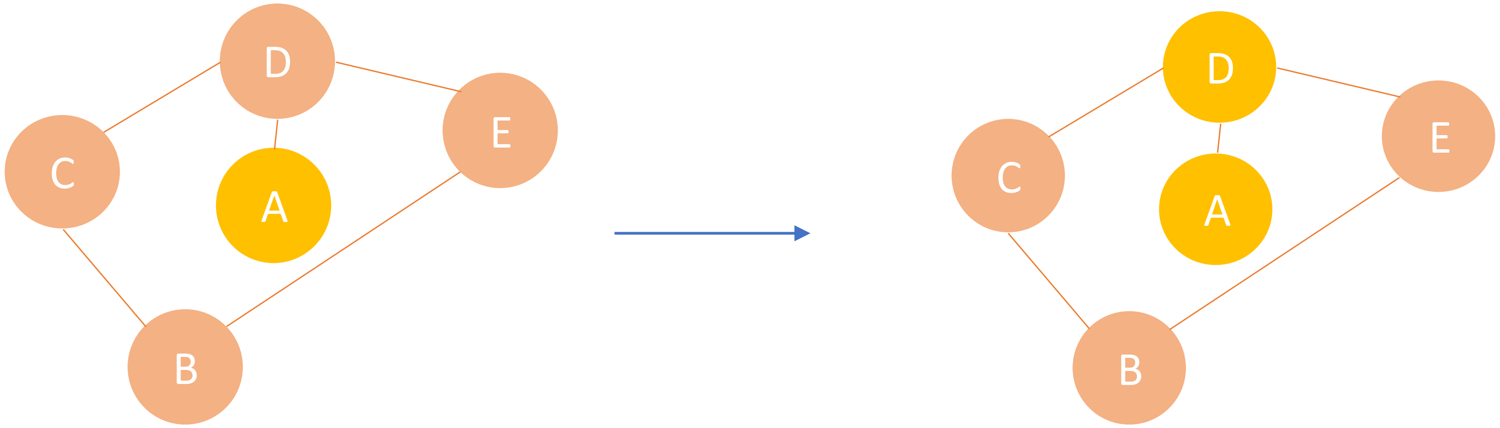
Graphs and graph neural networks

Information is propagated through the graph (with an attention mechanism: some nodes can be made more important than others)



Graphs and graph neural networks

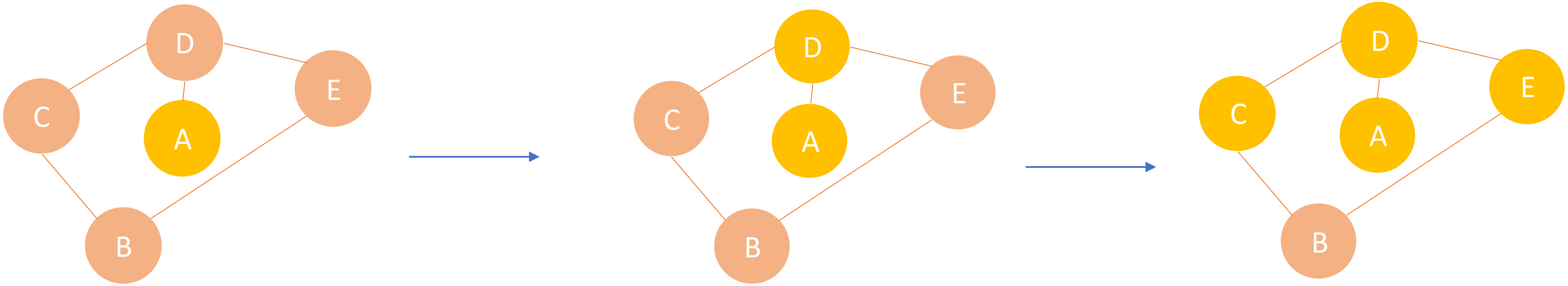
Information is propagated through the graph (with an attention mechanism: some nodes can be made more important than others)



The information associated with node A (e.g. the coordinates of a specific hit) is propagated to its **neighbors**

Graphs and graph neural networks

Information is propagated through the graph (with an attention mechanism: some nodes can be made more important than others)



The information associated with node A (e.g. the coordinates of a specific hit) is propagated to its **neighbors**

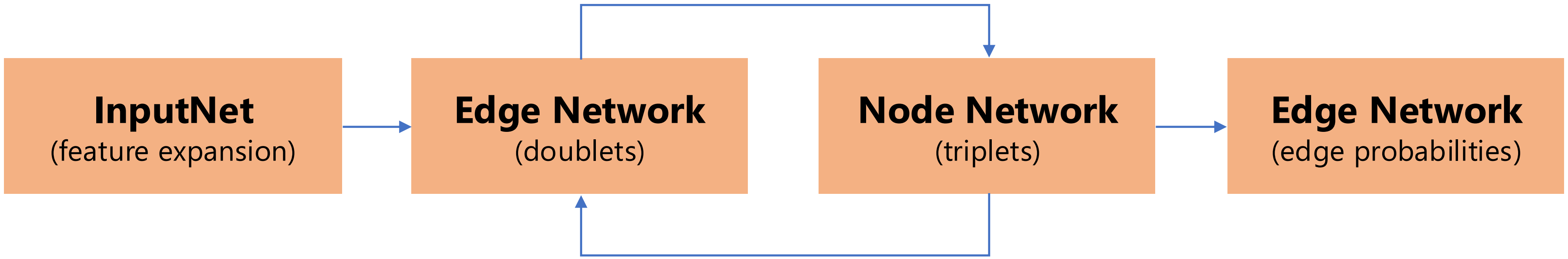
Graphs and graph neural networks

Graphs are represented by data structures such as adjacency and feature matrices which are fed to a graph neural network

A GNN performs information propagation

Edge Network → each edge in the graph is associated with a probability of being a physical link between two hits (doublet)

Node Network → each node aggregates the information from its neighbors (triplet)

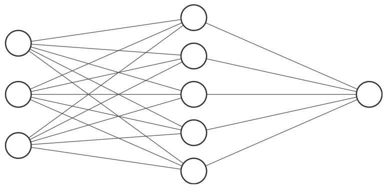


the number of iterations is related to the flow of information between the nodes of each graph

Research: Quantum GNNs

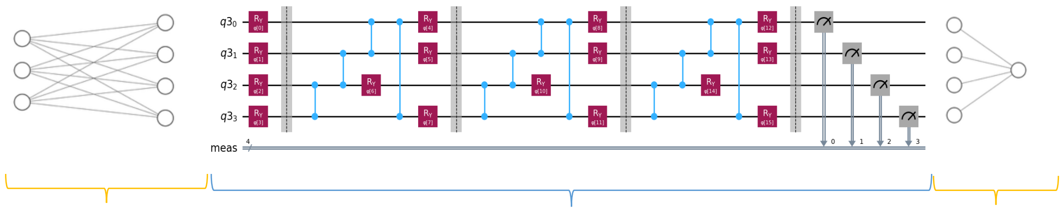
Classically the Multi Layer Perceptron (MLP) inside the GNN is a stack of dense fully connected layers

Classical $\equiv$



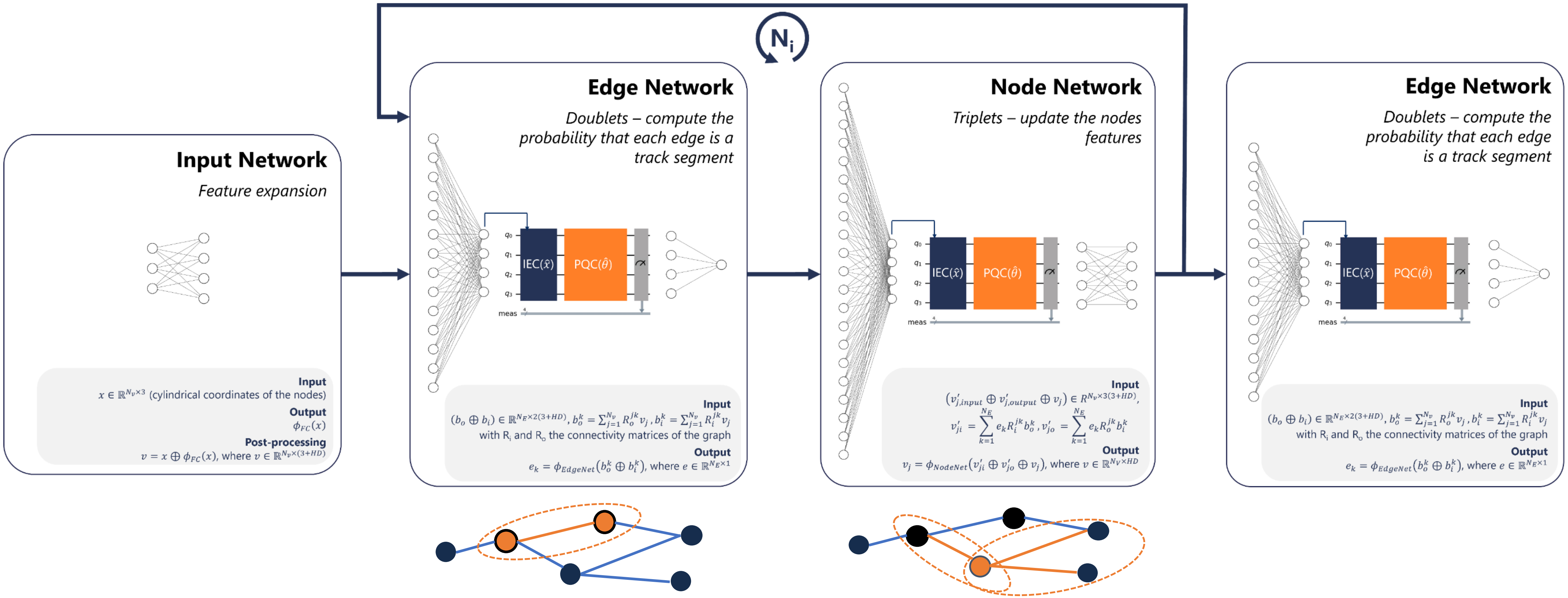
Exploration of the possibility of using a hybrid architecture employing parametrized quantum circuits with MLPs:

Hybrid $\equiv$



encoding layers — circuit — readout

Hybrid QGNN model



STAGE 1:
CHARACTERISATION
OF THE ORIGINAL QGNN ARCHITECTURE

Our implementation(s)

We have been working on an efficient implementation of the QGNN model using different frameworks

- We use the **TrackML** dataset, which provides collision events **simulated with HL-LHC conditions** in a generic tracker
- Jax + Flax + Pennylane is the most promising version of our software (up to an order of magnitude less training time –from a few days to a few hours – compared to Torch + Qiskit and TensorFlow Quantum + Cirq)
- We study the hybrid QGNN model in terms of **accuracy** and other metrics for **increasing pileup values**
  - we perform some small sized studies on IBM quantum hardware [2]

[2] Access to the IBM Quantum Services was obtained through the IBM Quantum Hub at CERN under the CERN-INFN agreement contract KR5386/IT.

PENNYLANE

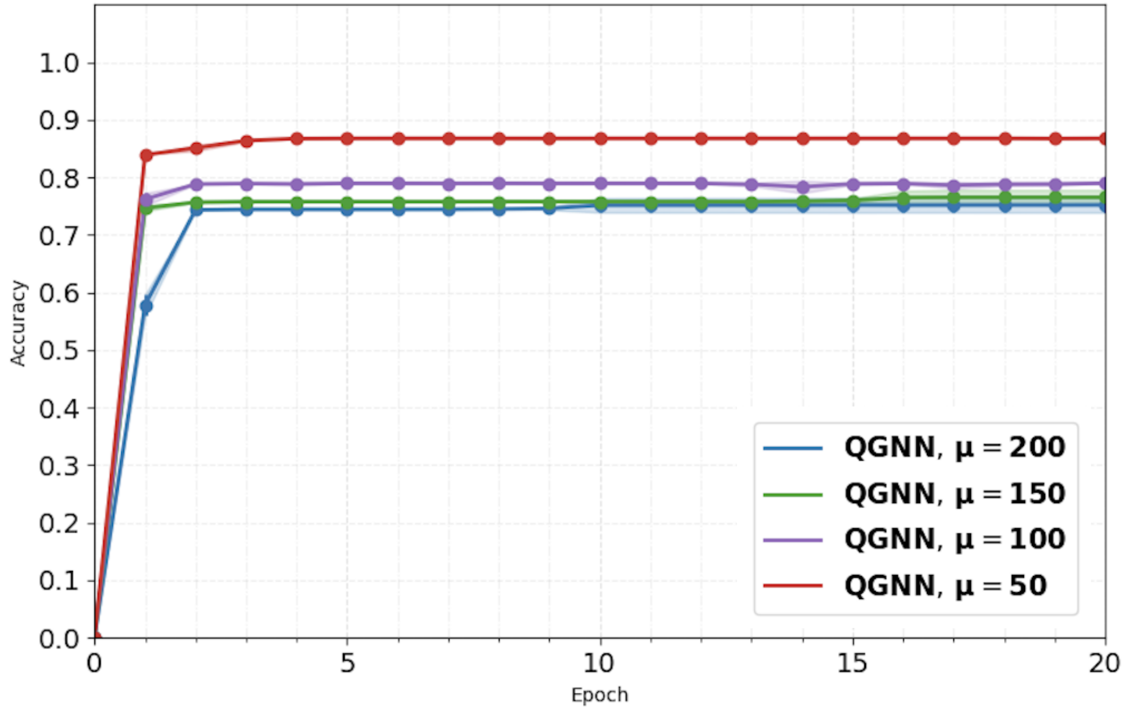
Phase 1 accuracy results:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

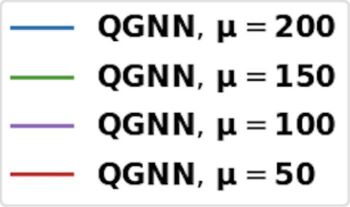
Accuracy is, as expected, higher with lower pileup

- The dataset is increasingly unbalanced for decreasing pileup
- Error bars are obtained by k-folding

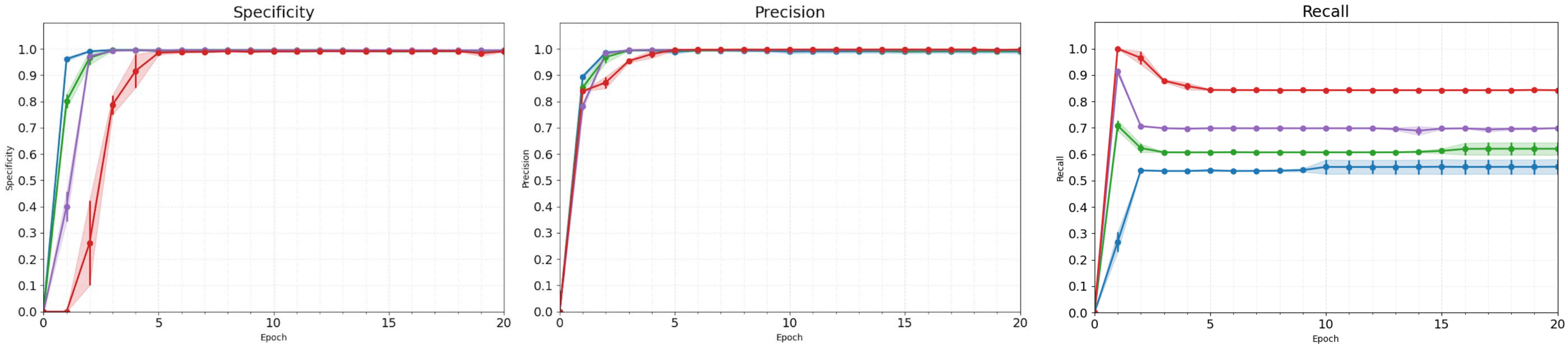
Validation set accuracy



Training the QGNN



Other metrics show that the QGNN is able to correctly recognize fake edges, but struggles with true edge classification



$$specificity = \frac{TN}{TN + FP}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

End of Phase 1 status

Tracking is definitely not a low hanging fruit for QML

- HEP events are far too big to be handled by a full quantum GNN in this NISQ era
- TrackML is a dataset that fits ML quite nicely, but can be an overshoot for QML

DATASET

The GNN architecture we are studying is not state-of-the-art anymore

- There are much more complex classical GNN oriented tracking pipelines (e.g. Atlas ACORN and LHCb etx4velo)

GNN

The number of circuits is a limiting factor in this hybrid architecture

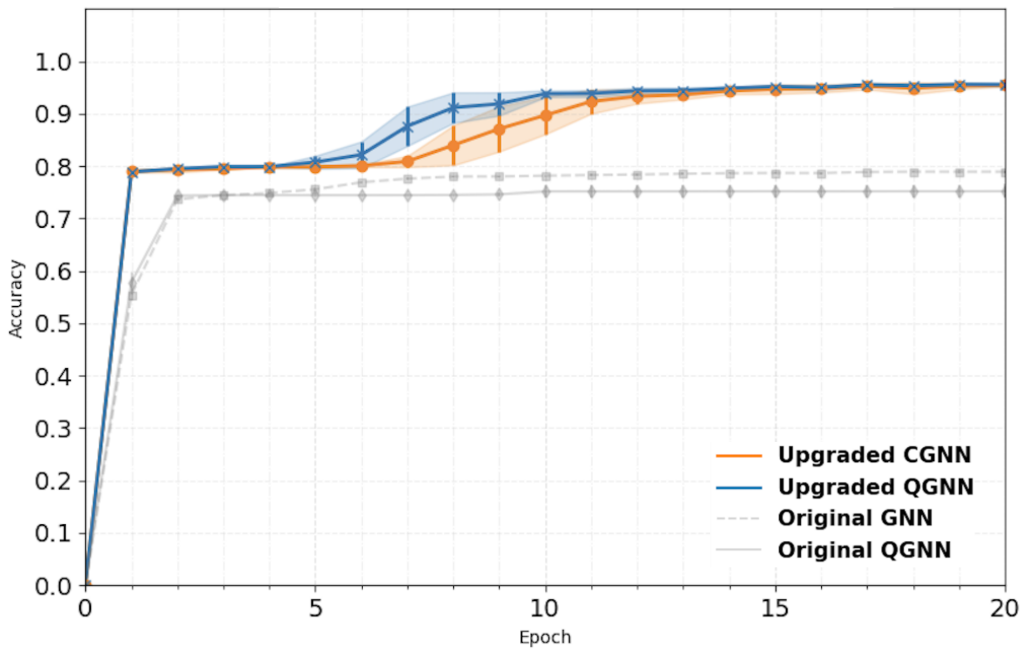
- Angle encoding is limiting the number of classical features from and to the MLPs
- Even using available quantum hardware is challenging because of Q/C integration

QGNN

STAGE 2:
UPGRADES TO THE QGNN ARCHITECTURE

Phase 2 upgrades

Validation set accuracy



GNN:

- Ad-hoc residual connections in GNN
- Wider and deeper MLPs

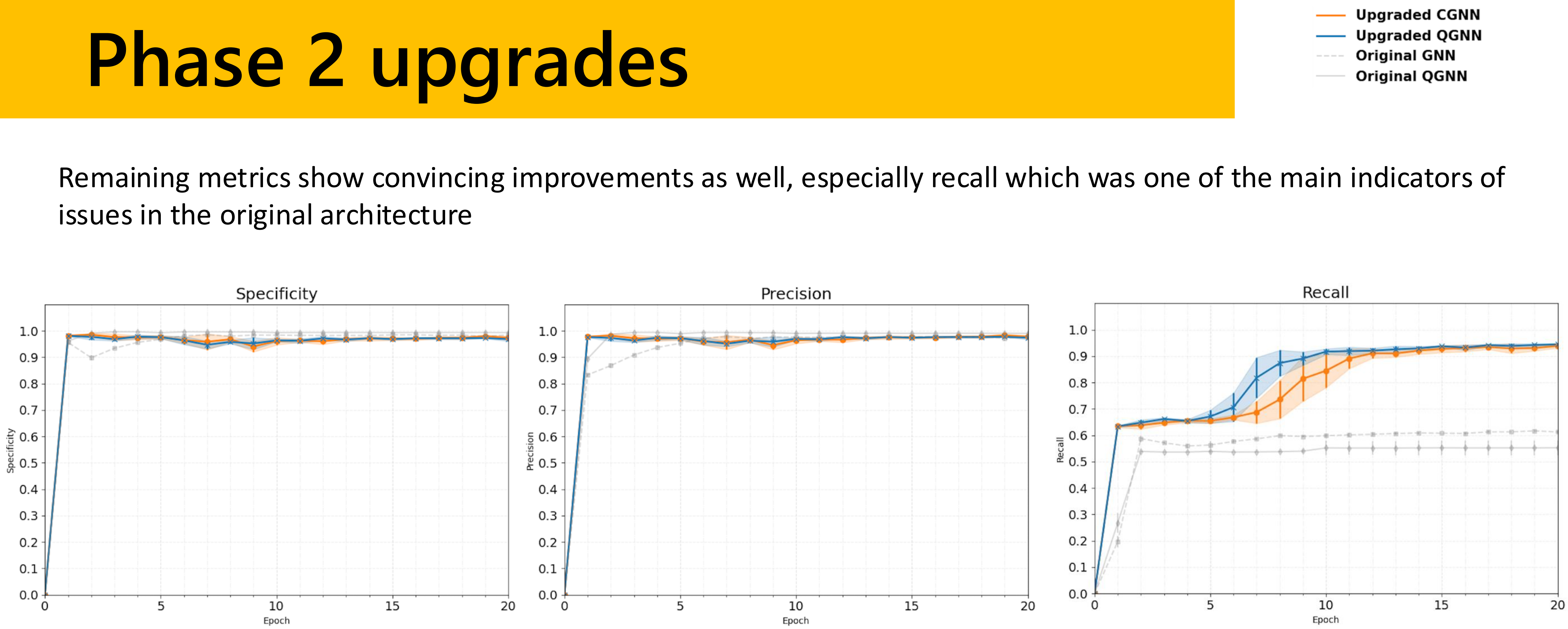
QGNN:

- Switch to amplitude encoding from angle encoding

- **The QGNN shows much improved metrics (here accuracy) at pileup 200**
- The QGNN is able to match the accuracy of the GNN
- There is an interesting earlier convergence which leaves space to extensions of this work

Phase 2 upgrades

Remaining metrics show convincing improvements as well, especially recall which was one of the main indicators of issues in the original architecture



$$specificity = \frac{TN}{TN + FP}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

our repo

Conclusion and prospects

- **An upgraded and self-consistent QGNN architecture for HL particle tracking has been developed** as an answer to the limitations of the original proposal

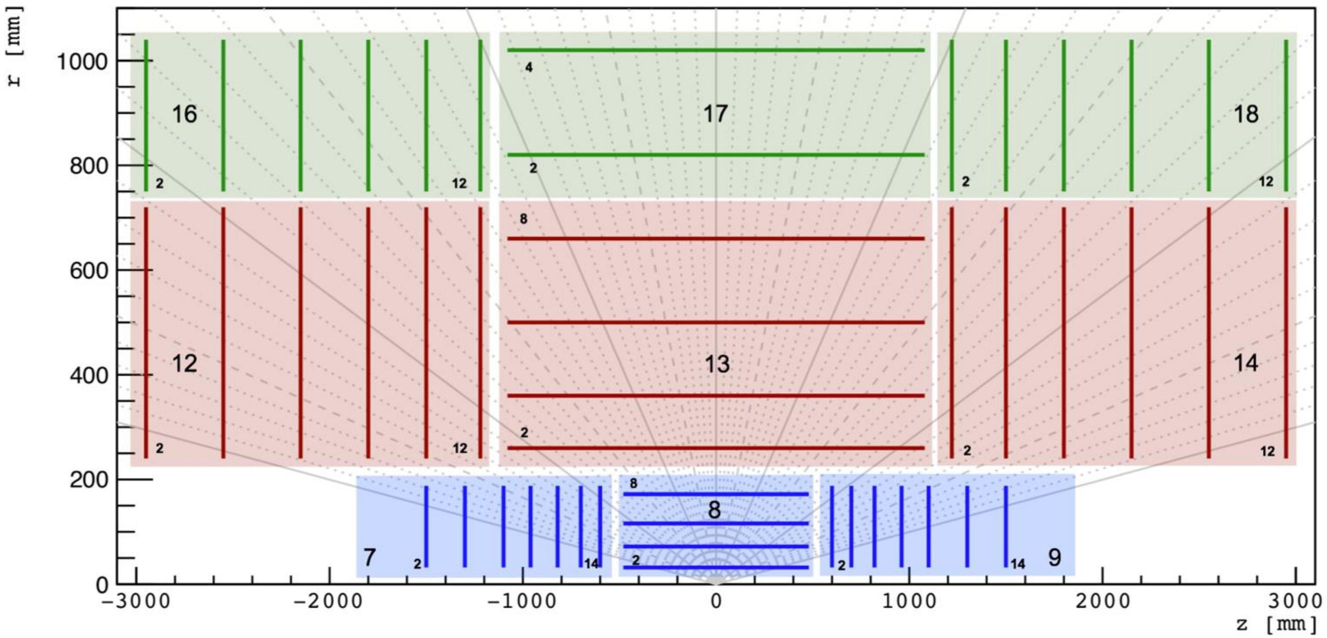
- All learning metrics now indicate that the task is being correctly learned using preprocessed TrackML dataset

- **This work helps shed light into the critical aspects of NISQ QML for experimental HEP**:
  - Full quantum Tracking at HL-LHC is unfeasible with current quantum resources
  - Hybrid variational algorithms are very useful in preparing the ground for future research
  - To further understand the role of PQCs in GNNs for this dataset more qubits need to be able to be simulated

Thank you for your attention

Backup – detector and dataset

The dataset we use comes from the TrackML Kaggle challenge [2]



- only the barrel region (8,13-17) is considered
- selection:

```
pt_min: 1. # GeV
phi_slope_max: 0.0006
z0_max: 100
n_phi_sections: 1
n_eta_sections: 1
eta_range: [-5, 5]
```

[2] https://www.kaggle.com/competitions/trackml-particle-identification

Backup – Input Graphs

Pileup 200

Graph with 5653 hits, 8837 edges, 53% true

Pileup 150

Graph with 4223 hits, 5630 edges, 58% true

Pileup 100

Graph with 2728 hits, 3117 edges, 71% true

Pileup 50

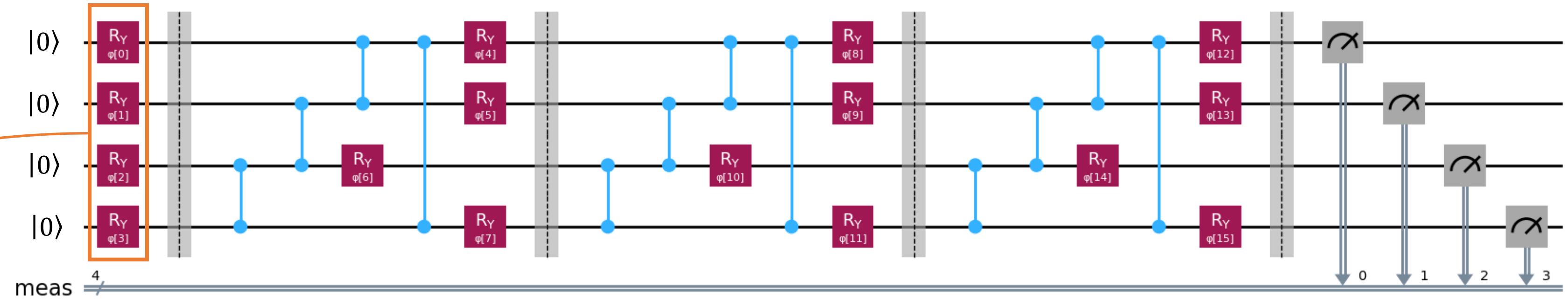
Graph with 1512 hits, 1553 edges, 83% true

Pileup 10

Graph with 291 hits, 240 edges, 98% true

Quantum circuits and angle encoding

The crucial step is the embedding of classical information into the quantum circuit

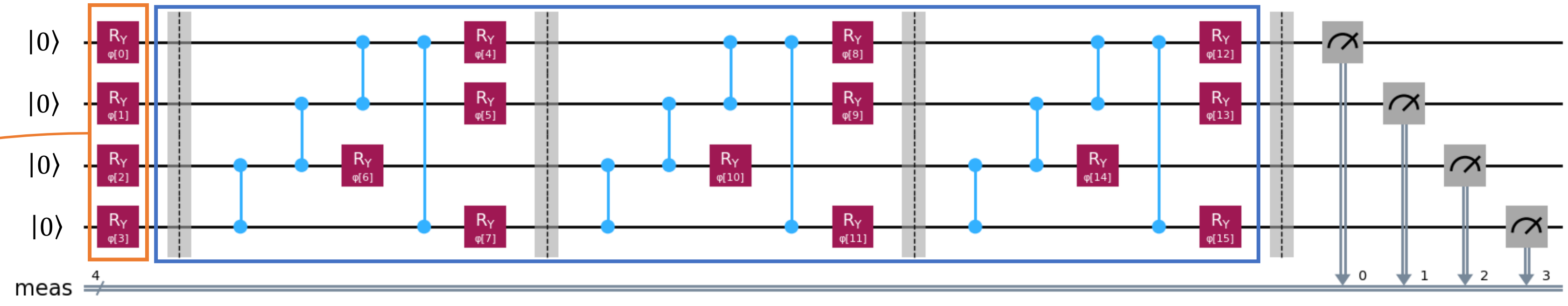


the output of the encoding layers is embedded as rotation angles

these parameters are used by the encoding $R_y(\theta)$ gates to rotate the initial $|0000\rangle$ state to a new state in the Hilbert space

Quantum circuits and encoding

The crucial step is the embedding of classical information into the quantum circuit



the output of the encoding layers is embedded as rotation angles

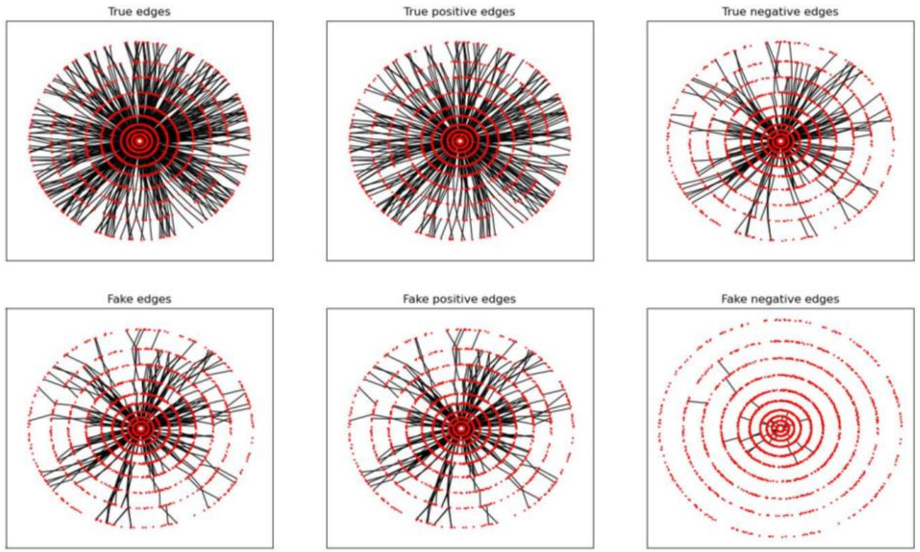
these parameters are used by the encoding $R_y(\theta)$ gates to rotate the initial $|0000\rangle$ state to a new state in the Hilbert space

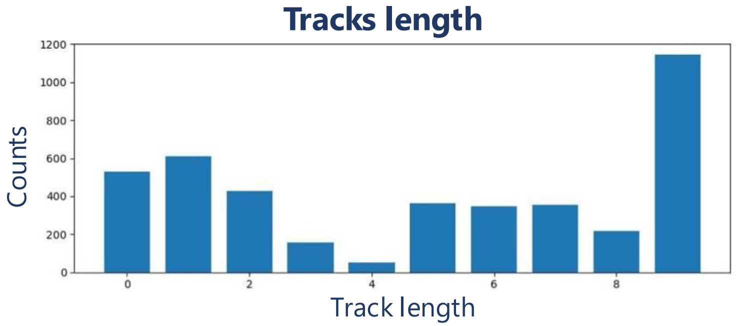
The second part of the circuit is called PQC (Parametrized Quantum Circuit) and its free parameters are the ones we train

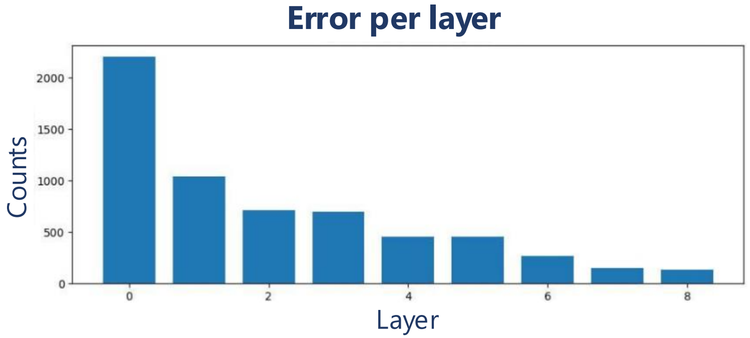
Phase 1 errors

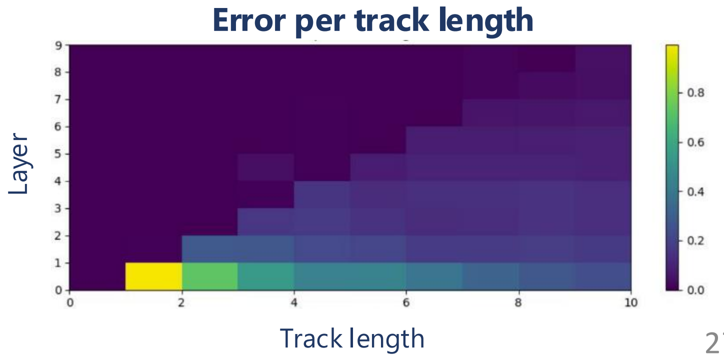
- In particular the majority of the errors occur in the innermost layers of the detector

- This is an expected behavior since because in layers 0-1 we find the vast majority of the combinatorial for the track segment candidates





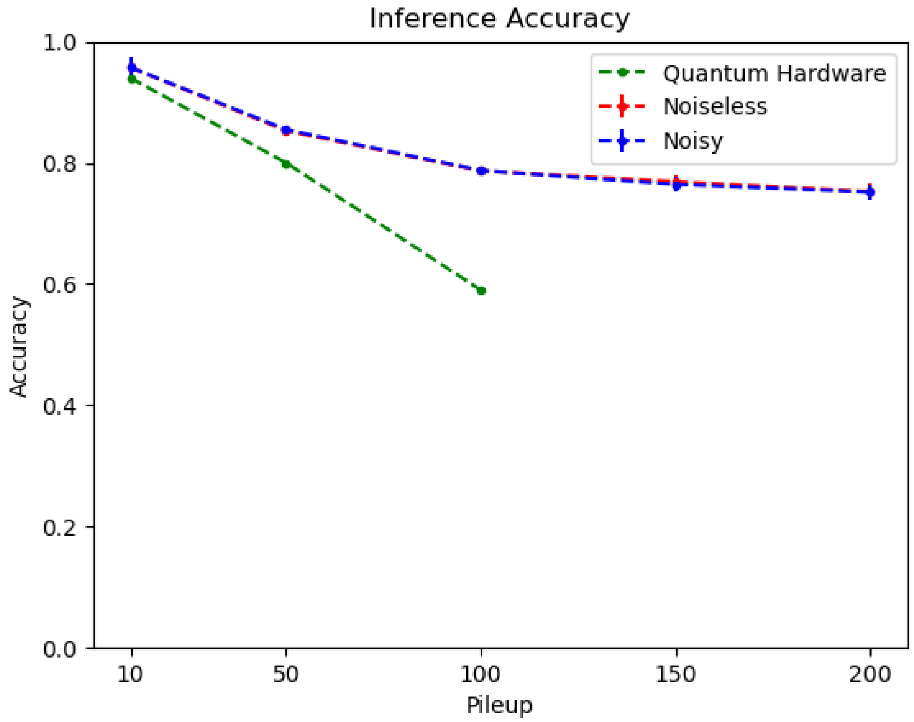




Inference on Hardware

We tested the QGNN model on different backends

- ideal noiseless simulator
- Qiskit Aer noisy simulator
- IBM Quantum hardware (IBM_Osaka)



- There is no significant difference between the results for noiseless and noisy simulated values, the two curves are essentially overlapped
- Test set is reduced for inference on IBM Quantum Hardware due to limitations in QPU time and resources availability

Further studies from Phase 2

Characterization of expressivity enhancement schemas for PQC

For **amplitude encoding** I am now testing a parallel schema where:

$$|\Psi>_{parallel} = |\Psi> \otimes |\Psi> \quad \text{with:} \quad |\Psi> = \frac{1}{\sqrt{2^N}} \sum_{i=0}^{2^N} x_i \, |i>$$

- 64 features can be accommodated by the amplitudes of the state $|\Psi>$ of 6 qubits.
- A state $|\Psi> \otimes |\Psi>$ needs 12 qubits which is a manageable number of qubits
- **Accuracy is still high, but this encoding shows no critical improvement in the learning metrics**.