

# Embedding of particle tracking data using hybrid quantum-classical neural networks

Carla Rieger<sup>1,\*</sup>, Cenk Tüysüz<sup>2,\*\*</sup>, Kristiane Novotny<sup>3</sup>, Sofia Vallecorsa<sup>4</sup>, Bilge Demirköz<sup>2</sup>, Karolos Potamianos<sup>3,5</sup>, Daniel Dobos<sup>3,6</sup>, and Jean-Roch Vlimant<sup>7</sup>

<sup>1</sup>ETH Zürich, Otto-Stern-Weg 1, Zürich, Switzerland

<sup>2</sup>Middle East Technical University, Dumlupınar Bulvarı, Çankaya, Ankara, Turkey

<sup>3</sup>gluoNNet, Avenue de Sécheron 15, Geneva, Switzerland

<sup>4</sup>CERN, 1 Esplanade des Particules, Geneva, Switzerland

<sup>5</sup>University of Oxford, Parks Rd, Oxford, United Kingdom

<sup>6</sup>Lancaster University, Bailrigg, Lancaster, United Kingdom

<sup>7</sup>Caltech, 1200 East California Boulevard Pasadena, California, United States

**Abstract.** The High Luminosity Large Hadron Collider (HL-LHC) at CERN will involve a significant increase in complexity and sheer size of data with respect to the current LHC experimental complex. Hence, the task of reconstructing the particle trajectories will become more involved due to the number of simultaneous collisions and the resulting increased detector occupancy. Aiming to identify the particle paths, machine learning techniques such as graph neural networks are being explored in the HEP.TrkX project and its successor, the Exa.TrkX project. Both show promising results and reduce the combinatorial nature of the problem. Previous results of our team have demonstrated the successful attempt of applying quantum graph neural networks to reconstruct the particle track based on the hits of the detector. A higher overall accuracy is gained by representing the training data in a meaningful way within an embedded space. That has been included in the Exa.TrkX project by applying a classical MLP. Consequently, pairs of hits belonging to different trajectories are pushed apart while those belonging to the same ones stay close together. We explore the applicability of variational quantum circuits that include a relatively low number of qubits applicable to NISQ devices within the task of embedding and show preliminary results.

## 1 Introduction

With the start of the High Luminosity LHC (HL-LHC), there will be many more simultaneous collisions (pile-up) leading to ambiguities and making the task of reconstructing the particle tracks highly complex [1]. To explore how (classical) machine learning techniques can tackle this task, the TrackML challenge has been launched on Kaggle [2]. The TrackML data set consists of more than 8,000 simulated collision events. Since its release, the data set has become an important benchmark for particle tracking algorithms. Graph neural networks, as novel approaches to track reconstruction, are applied within the HEP.TrkX project [3] and

---

\*e-mail: [carieger@ethz.ch](mailto:carieger@ethz.ch)

\*\*e-mail: [cenk.tuysuz@cern.ch](mailto:cenk.tuysuz@cern.ch)

its update, the Exa.TrkX project [4]. In both of these projects, the graph neural network has the general structure of iteratively applying a node and an edge network. The edge and node information can either take doublets or triplets of nodes into account. An important part in the data processing pipeline of the Exa.TrkX project forms the embedding of the detector measurement data. A feed-forward neural network with hidden layers – referred to as Multi-Layer Perceptron or MLP – acts as a non-linear projection onto a higher-dimensional embedding space. Hits belonging to the same trajectory are embedded close together, while those belonging to different trajectories are embedded far apart. This step improves the performance within the classification task. This work explores hybrid quantum-classical networks for embedding the simulated detector measurements of the TrackML dataset. It builds up on the promising approach of using quantum graph neural networks [5], [6], as well as the projects using classical neural networks which are mentioned above [3], [4], [7]. Several quantum circuit configurations have been explored that extend or replace parts of the classical MLP used within the Exa.TrkX project. The quantum-classical version utilizes the exponential size of the Hilbert space and explores the effects of entanglement on the embedding and inference tasks across different sized classical networks that aim to optimize the representation within several quantum circuits. The quantum circuits used are chosen to have possible applications on noisy intermediate-scale quantum (NISQ) devices [8]. Hence, the circuits build up on a low number of qubits and quantum gates.

## 2 Quantum Gates and Quantum Circuits

This section provides a short overview on how quantum computing can be included within classical neural networks. Each qubit in a gate-based quantum circuit is initialized in the  $|0\rangle$  state, pointing upwards in the Bloch sphere representation. Quantum gates can be applied on a single or multiple qubits that can get entangled. In the following, we will use for example  $R_X$  gates to encode classical parameters, i.e., angles  $\theta$ . Those values are for example given by the classical MLP output  $\hat{x}_i$ , as shown in Fig. 2. They act on a general quantum state  $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$  as follows [9]:

$$R_x(\theta)|\Psi\rangle = \exp\left(-i\frac{\theta}{2}\sigma_x\right)|\Psi\rangle = \begin{pmatrix} \cos(\theta/2) & -i\sin(\theta/2) \\ -i\sin(\theta/2) & \cos(\theta/2) \end{pmatrix} \cdot \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \quad (1)$$

Here,  $\sigma_x$  represents the respective Pauli matrix and  $X$  the rotational axis. The gates applied on the different qubits in a quantum circuit are designed in a reversible way, meaning the output of the gate fully determines its input. There is no loss of information in a noiseless quantum circuit. Gates with variable parameters can either encode incoming classical information (IQC), here in form of angle-encoding, or exhibit trainable parameters (PQC). In this case, gates exhibiting a variable parameter are initialized at random and optimized during the training procedure, similarly to classical weights in a neural network. In addition, a quantum circuit can include hidden dimensions consisting of additional qubits that do not encode input information. They can be entangled with any other qubit. Gates with trainable parameters may be applied on the hidden dimensions to expand the number of free parameters in the circuit. To convert back to a classical output, we compute the expectation value with respect to the  $\sigma_z$ -operator. Practically, this value is estimated numerically by evaluating the circuit for  $n_{shots} = 1000$ , as it is standard in PennyLane [10] and thus inferring the probabilities through the wave-function collapse.

### 3 The data set

The network has been trained on the TrackML data set which is publicly available [11]. The task is to classify the 3-dimensional hits in order to identify the trajectories of the single particles involved in the collision. The figure below (Fig. 1) displays how the hits, displayed as dots, are connected by edges and form trajectories. True trajectories are displayed in blue and false ones in red. Due to training time restrictions only a part of the TrackML data set has been processed to doublets in the same way as in the Exa.TrkX project, setting a cut at  $pT = 1$ . 10,000 doublets have been generated and are used for training the hybrid quantum-classical network (8,000 for training, 2,000 for validation). This is a relatively small number of samples to have a reasonable training time when working with quantum circuits. In further tests, more samples should be used for training. For the train-validation data split, the data has been shuffled and thus randomly assigned to one of the data sets to guarantee an expressive validation procedure. By observing the behaviour of the training and validation losses, overfitting can be seen and prevented. The validation loss is more informative than the actual training loss because the performance is evaluated on a separate, independent part of the data that is not involved in the optimization procedure. Hence, only the validation loss is presented.

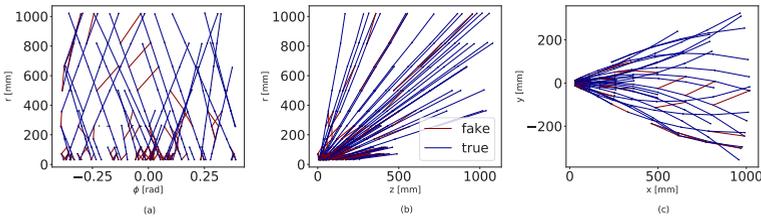


Figure 1: Illustration of the particle track reconstruction task. The black dots represent the simulated detector hits which are connected to trajectories. True edges are displayed in blue, false ones in red [5].

### 4 Hybrid Architecture

The architecture follows two different approaches that extend the classical MLP used for embedding by the Exa.TrkX project [4]. Both approaches follow the general structure as presented in Fig. 2 with different quantum circuits (QC) and various numbers of classical hidden layers ( $n_{layers}$ ). In the first approach, the classical MLP is combined with different quantum circuits, which act as a function, i.e., the output from the previous classical part is encoded using rotational gates, as described above. Hence, the output layer of the classical MLP projects onto the number of parameters  $n_{params}$  of the quantum circuits. Depending on the respective circuit, a certain degree of entanglement is present across the qubits. Quantum circuits that exhibit different metric values with respect to entanglement, expressibility and varying numbers of parameters [12] have been used to explore how those differences influence the training behavior of the whole network. The second approach using the quantum feature map follows again the same general structure as described in figure 2. The output of the classical MLP is encoded using rotational gates. Furthermore, this version includes trainable parameters within the quantum circuit. As before, the last classical network layer projects the output of the quantum circuit ( $n_{meas.}$ ) onto the preferred embedding dimension that is smaller or equal to  $n_{meas.}$ .

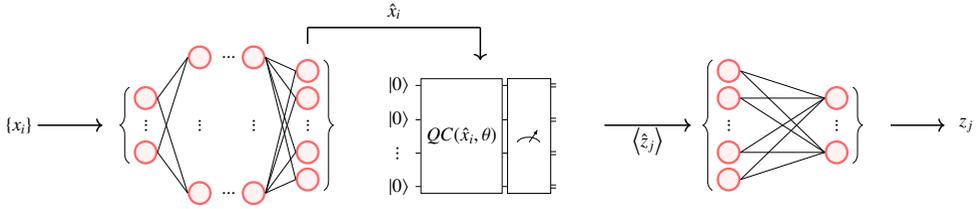


Figure 2: Illustration of the general structure of the hybrid quantum-classical MLP. The displayed quantum circuit (QC) acts as a placeholder for the quantum circuit architectures presented below. The QC encodes the incoming classical input  $\hat{x}_i$  and can include trainable parameters  $\theta$ .

### 4.1 The classical MLP

The classical MLP used in the Exa.TrkX project [4] has hidden layers of the dimension of  $n_{layers} \times 512$ , where 512 is the number of neurons per layer and  $n_{layers} = 10$ . There are additional input and output layers. The Exa.TrkX project uses an embedding/output dimension of 8. The classical embedding version performs well and forms a successful pre-processing step, which improves the performance of the classification GNN. The aim is to explore how incorporating quantum circuits changes the training behaviour and how the reduction of classical MLP layers within this hybrid model performs in simulations.

### 4.2 Quantum Circuit approach

The general architecture can be seen in Fig. 2. For this first approach  $n_{layers}$  is set to 10 and the quantum circuits from Fig. 3 act as an encoding function. The 3-dimensional input data from the TrackML data set is embedded into a 4-dimensional space. Due to the long training time for the simulation of the quantum circuit, only 4 and 8 qubit quantum circuits have been used. This limits the embedding space dimension for this 4-qubit circuit ( $\dim(\vec{z}) = 4$ ) to  $\dim(\vec{z}) = 4$ . For better comparison, the same embedding dimension was kept also for the 8 qubit circuits, where  $\dim(\vec{z}) = 8$ . The last classical layer is kept in all tests for the same reason. The circuits should be later extended to use more than 8 qubits and thus make an embedding into a higher dimensional space possible.

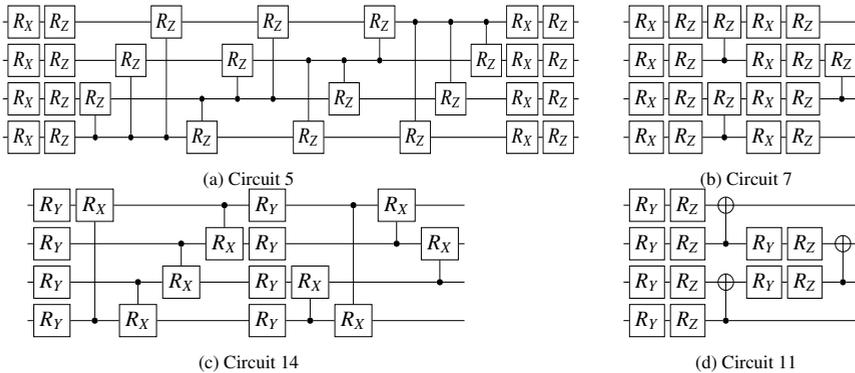


Figure 3: The 4-qubit quantum circuits [12] that have been used within the general architecture (Fig.2). Each of the (conditional-) rotational gates can encode incoming classical information as angles on the respective qubit (as in equation 1). Moreover, the displayed 2-qubit gates, i.e., the conditional rotational and the CNOT gates, can entangle the two qubits they are applied on.

The different quantum circuits displayed in Fig. 3 have been used as QC in the hybrid MLP architecture displayed in Fig. 2. Those four circuits have been chosen due to their differences with respect to entanglement and expressibility as depicted in [12]. Entanglement is measured using the Meyer-Wallach entanglement measure [13], [12]:

$$Q(|\psi\rangle) = \frac{4}{n} \sum_{j=1}^n D(\iota_j(0)|\psi\rangle, \iota_j(1)|\psi\rangle) \quad (2)$$

with  $D(|u\rangle, |v\rangle) = \frac{1}{2} \sum_{i,j} |u_i v_j - u_j v_i|^2$  and  $\iota_j(b)|b_1, \dots, b_j, \dots, b_n\rangle = \delta_{b,b_j}|b_1, \dots, b_{j-1}, b_{j+1}, \dots, b_n\rangle$ . Thus, the entanglement value for a given quantum circuit is defined as:

$$\mathbf{Ent} = \frac{1}{|S|} \sum_{\theta_j \in S} Q(\psi_{\theta_j}), \quad (3)$$

where  $S = \{\theta_j\}$  is given as the ensemble of sampled circuit parameter vectors. The expressibility of the circuits is measured as the Kullback-Leibler-divergence ( $D_{KL}$ ) between the estimated probability distribution of the fidelities  $F$  of the respective quantum circuit and the one of Haar random state [12]:

$$\mathbf{Expr} = D_{KL}(\hat{P}_{P_{QC}}(F, \theta) || P_{Haar}(F)). \quad (4)$$

Another relevant metric that should be tested in future studies is for example the Fisher information spectrum [14]. The (conditional) rotational gates in Fig. 3 encode the output of the classical MLP as angles, i.e. the output of the MLP is optimized in a way to learn the encoding on the qubits within the given quantum circuit structure. Thus, the quantum circuit acts as a function:  $QC_{id} : \mathbb{R}^{n_{param.}} \rightarrow \mathbb{R}^{n_{meas.}}$  for  $id \in \{5, 7, 11, 14\}$  and projects the input to the number of measurements in the circuit. There is a further classical down-projection layer applied that makes it possible to embed into a lower-dimensional embedding space (i.e., for the 8-qubit circuit). The layer is kept for all versions to make them comparable.

### 4.3 Quantum Feature Map approach

The second approach includes a quantum circuit with trainable parameters. The initial architecture of the quantum circuit was adapted from [15]. The overall structure of the hybrid network follows again Fig. 2. In this version, the output of the classical MLP is encoded repeatedly using rotational gates. In between, entangling gates are applied that include optimizable parameters. Various numbers of hidden dimensions can be included. To keep the size of the circuit as small as possible, only one hidden dimension was used for testing purposes. The hidden dimension, here, is a single qubit that is not used for encoding or decoding but is entangled with other non-hidden dimensions and includes trainable parameters in the form of variable angles of rotational gates. Furthermore, this approach can be extended to more qubits and hence encode more input parameters as well as include more trainable parameters, i.e., within the hidden dimensions. The quantum feature map (QFM) circuit is displayed in Fig. 4 and exhibits 74 parameters. The  $R_X$  gates encode the input of the 4-dimensional previous layers and the remaining parameters are variable and hence randomly initialized and optimized during training. Just like before, each qubit is initialized in the  $|0\rangle$  state. The first 4 qubits are the ones measured at the end of the quantum circuit, the 5th one forms the hidden dimension of the circuit. In Fig. 4, it is visible how the input (IQC) is repeatedly applied in every iteration. This encoding is done via rotational gates:

$$|\Psi_i\rangle = R_X(\hat{x}_i)|0\rangle, \text{ for } 1 \leq i \leq 4. \quad (5)$$

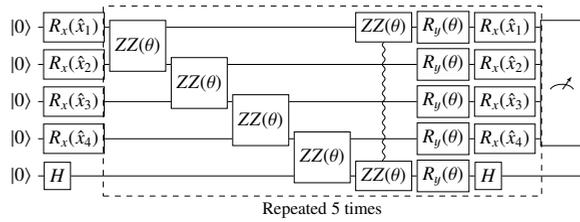


Figure 4: Quantum circuit of the quantum feature map approach used within the general architecture in Fig. 2. Adapted from [15]. The variational 2-qubit gate is defined as  $ZZ(\theta) = \exp(-i\theta\sigma_z^{\otimes 2}/2)$ , additionally we apply the Hadamard gate on the hidden layer, which has the following effect on the initial qubit state  $H|0\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$  in the computational basis. The measurement is only taken of the first 4 qubits, but the hidden layer can have nevertheless an influence on the outcome since it is entangled via the ZZ-gates. The amount of entanglement depends on the value of  $\theta$ .

## 5 Results and implementation details

The different architectures of the hybrid quantum-classical neural network described above have been tested and the results are shown in detail below. The simulations of the hybrid quantum-classical network structure have been done in Python using PyTorch [16] and quantum computing libraries such as PennyLane [10] and Qiskit [17]. It is a supervised training procedure. The data set consists of doublets, which are two 3-dimensional points (hits) in a Euclidean space that form nodes connected via an edge. Each of these doublets have a label that defines whether they are true edges, i.e., the corresponding hits belong to the same trajectory or not. The aim of the embedding is to represent the hits in a feature space where the true doublets are close together and the false ones are further apart. This is achieved by using the hinge embedding loss during training, similarly to the Exa.TrkX project. The hinge embedding loss, available in PyTorch [16], is defined for each sample  $s_n = (x_i, x_j, y_{i,j})$  as follows, where  $\Phi(x_k, \theta)$  represents the embedded hit  $x_k$  given by the model including trainable parameters  $\theta$ :

$$loss(s_n) = \begin{cases} \max\{0, \|\Phi(x_i, \theta) - \Phi(x_j, \theta)\|_2\}, & \text{if } (x_i, x_j) \text{ belong to the same trajectory} \\ 0, & \text{if } (x_i, x_j) \text{ belong to different trajectories.} \end{cases} \quad (6)$$

The n-th sample  $s_n$  is described by the two hits in the original space, denoted as  $(x_i, x_j)$ . To each of these doublets belongs a label  $y_{(i,j)} \in \{\pm 1\}$  that indicates if the two points belong to the same trajectory  $y_{(i,j)} = 1$  or not,  $y_{(i,j)} = -1$ . The hinge embedding loss function favors the case in which two hits of to the same trajectory are embedded close together. Other loss functions that act similarly (i.e., cosine embedding loss) have also been tested.

### 5.1 Training the quantum circuit construction

The quantum circuit architecture (displayed in Fig. 2 and 3) was trained using ADAMAX optimizer [18], a learning rate of 0.001, batch size of  $N = 100$ . The classical MLP has  $n_{layers} = 10$  where each of the layers consist of 512 neurons. In table 1, a comparison between the quantum circuits that have been trained is displayed. The training time increases with increasing number of gates in the 4-qubit circuits. The 2-qubit gates are more computationally expensive. The difference in number of gates and number of parameters occurs because some of the gates (e.g. the CNOT gate) do not have a variable parameter. The listed quantum circuits have been chosen due to their different values regarding the entanglement and expressibility metrics. The presented values in Table 1 have been reproduced and the metrics are described

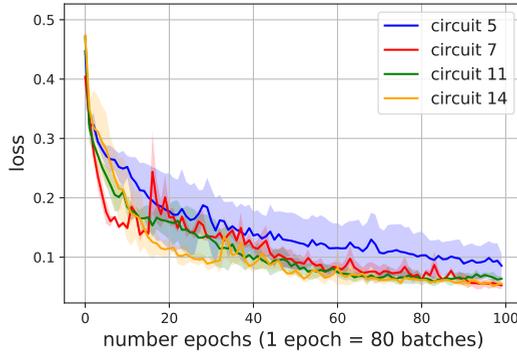


Figure 5: Validation loss of the hybrid network with the respective quantum circuit within the general architecture.

Table 1: Comparison between the different 4-qubit circuit parameters. Circuits from [12]. The error on the metric values is of the order  $O(10^{-4})$ .

Circuit	Parameters $n_{params}$	Entanglement (higher value preferred)	Expressibility (lower value preferred)	Training time (average per batch)
5	28	0.290	0.051	$37 \pm 8s$
7	19	0.212	0.104	$20 \pm 4s$
11	12	0.538	0.139	$14 \pm 4s$
14	16	0.545	0.011	$16 \pm 4s$

in more detail in [12]. Due to the favourable metric values of circuit 14, a good training performance is expected for this circuit. This is confirmed in when comparing the validation loss.

In Fig. 5, the mean values of 3 independent runs with different initial random states and the respective standard deviation as error bars are shown. We observe a strong initialization dependence for circuit 5 that also exhibits the largest number of parameters of the tested circuits, in detail discussed below. Circuit 14 has the best converging behavior towards the lowest validation loss in this comparison. While circuit 7 converges to a similarly low value, the validation loss function exhibits higher and lower spikes. This could be explained by the training strategy using mini-batches of size 100. Circuit 11 converges to a comparable value of validation loss but seems to be converging more slowly until epoch 40. An unfortunate random initialization might explain the initially slow convergence rate of circuit 7 and especially 5. The training results could be improved, by repeatedly applying the respective quantum circuit [12]. By doing this, the entanglement value increases as the KL-divergence decreases, i.e., the expressibility of the circuit increases (4.2) and thus the flexibility of the quantum circuit in representing random states. This means the quantum circuit is able to represent a wider range of states. The improvement when increasing the number of repetitions of the circuit can be seen in Fig. 6. Hence, repeating the best performing circuit 14 from before can improve the validation loss and score displayed above. At the same time such it also increases the number of gates in the circuit which leads to longer training times.

### 5.1.1 Barren plateaus

Even though the hybrid quantum-classical neural network version uses shallow quantum circuits including only up to 8 qubits and the number of parameters is of  $O(10)$ , the training performance is highly dependent on the initialization of the circuit parameters. In some cases, the training and validation losses do not converge at all within the 100 epochs that

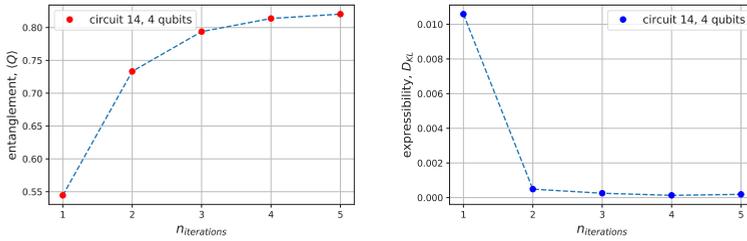


Figure 6: Change in entanglement (left) and expressibility (right) value for quantum circuit (QC) 14 which is repeatedly applied on the qubits for  $n_{iterations}$ . Ent./Expr. calculated as in [12].

have been used for training. It is visible in Fig. 7 how much the training success depends on the initialization of the parameters, i.e., convergence of the loss function within the first 100 epochs. The loss function shows the behaviour of 3 independent runs with different random state for initialization of the parameters in the classical MLP. Those random parameters act at the same time as the input to the quantum circuit and thus lead in case of run 3 to an unfavourable convergence behaviour, while for run 2 the initialization seems to only slow down the convergence in comparison to run 1. Several approaches exist to avoid such barren plateaus [19]. Furthermore, in this case it could help to lower the learning rate as it was done within the QFM approach. The small spikes could be explained by the mini-batch optimization procedure when using ADAMAX optimizer. Run 3 was omitted in Fig. 5 for better visualization.

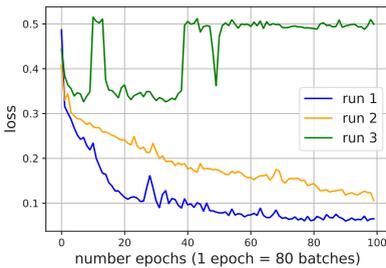


Figure 7: Validation loss of 3 independent runs of hybrid network using circuit 5 with 4 qubits.

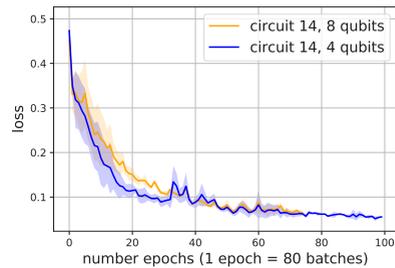


Figure 8: Validation loss of hybrid architecture with circuit 14 using 4 and 8 qubits.

### 5.1.2 Expanding the number of qubits

The initial circuits are all constructed using 4 qubits, corresponding to 4 qubits in the quantum circuit. If all qubits are measured, the output is also 4-dimensional, and the input data set is embedded into a 4-dimensional space. Thus, enlarging the quantum circuit allows to embed into higher dimensional spaces. To compare the different performance in validation loss and running time of the 8-qubit quantum circuit to the 4-qubit one, the last layer projects the output of the quantum circuit ( $n_{meas.} = 8$ ) onto a 4-dimensional embedding space. Fig. 8 shows the validation loss of the two versions of circuit 14 with 4 and 8 qubits. They exhibit a similar training performance. The 8-qubit circuit converges slightly slower than the 4 qubit one. In table 2 it is visible, how both circuits exhibit a similar entanglement value, but the 8-qubit circuit has a slightly higher expressibility. The plot shows the mean value of

Table 2: Comparison between the different 4-qubit and 8-qubit circuit parameters. As before, the error on the metric values is of the order  $O(10^{-4})$ . Ent./Expr. calculated as in [12].

Circuit	Parameters $n_{params}$	Entanglement	Expressibility	Training time (average per batch)
14 (4 qubits)	16	0.545	0.011	$0.28 \pm 0.07min$
14 (8 qubits)	32	0.544	0.062	$1.44 \pm 0.5min$

3 independent runs of one batch per epoch and the error bars display the respective standard deviation regarding the 3 runs. Due to the difference in number of qubits and number of parameters, the training time per batch for the 8-qubit circuit is much longer. In this case, the performance difference does not justify increasing the qubit number when projecting onto a 4-dimensional embedding space due to the difference in simulation times. The 8-qubit version becomes important if one makes use of the higher number of measurements.

### 5.2 Training the quantum feature map approach

The quantum feature map network was trained using similar specifications as before. Again, ADAMAX was used as optimizer and a batch size of  $N = 100$ . The embedding dimension here is also 4, limited due to the number of qubits within the quantum circuit and training time constraints. The main difference is that the learning rate was lowered to  $lr = 1e^{-4}$ .

The QFM model exhibits additional trainable parameters within the quantum circuit. Thus, it is an interesting question how the performance changes if the size of the classical MLP is reduced. Hence, a range of combinations of layers in comparison to the classic 10-layer case have been tested. The parameters and metric values with respect to the entanglement and expressibility metrics are shown in Table 3 in comparison to circuit 14. As

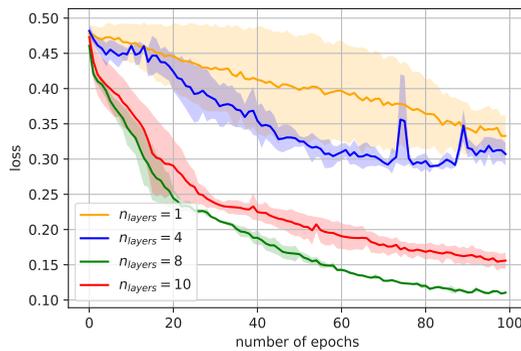


Figure 9: Validation loss of the quantum feature map circuit with variable numbers of layers  $n_{layers}$  of the classical MLP.

displayed in Fig. 9, the 8-layer version seems to perform better or at least is comparable to the 10-layer version within the first 100 epochs. The 4-layer version converges to a higher validation loss and thus seems to reach its expressiveness quite early and converges to a local minimum. The 1-layer version performs similarly to the 4-layer one, the expressiveness in this case is also limited. Moreover, both variants (1- and 4-layer) depend on the initialization for training success which is indicated by the error bars. For this model, the learning rate had to be lowered to  $lr = 1e^{-4}$ . When training using  $lr = 1e^{-3}$  as before, it exhibits an

Table 3: Comparison of relevant metric parameters of the respective quantum circuits within the general model. To get the respective training time, we set  $n_{layers} = 10$  (classical layers). Ent./Expr. calculated as in [12].

Circuit	Parameters $n_{params}$	Entanglement	Expressibility	Training time (average per batch)
QFM (5 qubits) ( $n_{iteration} = 5$ )	74	0.772	0.001	$5min38 \pm 8s$
14 (4 qubits) ( $n_{iteration} = 1$ )	16	0.545	0.011	$16 \pm 4s$

unfavorable convergence behavior. Comparing the validation loss of the 8 and 10-layer version using the quantum feature map approach to the quantum circuit 14 (10 classical layers) from before, they have a similar performance within the first 100 epochs, even if the QFM circuit exhibits favorable metric values (Table 3). In the presented range of epochs circuit 14 seems to perform better. However, this behaviour could change when training the QFM model for more epochs. Regarding the training time per batch, circuit 14 trains much faster and completes 100 epochs within  $O(10^5s)$ . In comparison to that, the QFM version needs  $O(10^7s)$ . This difference is due to the much higher number of quantum gates and parameters to be optimized. Also, for the QFM approach, the entanglement and expressibility of the circuit increases as the number of  $n_{iterations}$  increases, as shown in Fig. 10. This is accompanied by an increased training time. In Fig. 11 we compare the hybrid

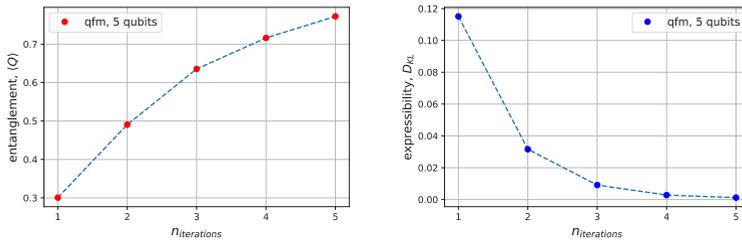


Figure 10: Change in entanglement (left) and expressibility (right) value for quantum feature map QC for different  $n_{iterations}$ . Calculated as in [12].

version to different sized classical MLPs. Regarding the best loss using the QC approach, there seems to be a trade-off between entanglement and expressibility. Whereas circuit 7 has the lowest entanglement and an intermediate expressibility, circuit 14 is really expressive but at the same time exhibits the highest entanglement within this comparison. In the QFM comparison we observe a proportionality of training success of the classical counterpart to the success in the representing the classical input within the quantum circuit. It is interesting how the difference in best loss value of the 8- and 10-layer version is enhanced in the hybrid model.

## 6 Conclusion

Combining quantum circuits and artificial neural networks to hybrid quantum-classical neural networks is a promising and interesting field of study with possible applications using noisy intermediate-scale quantum technologies [8]. The long simulation times of the quantum circuits that are of  $O(10 \text{ hrs})$  currently restrict the number of qubits to 4 in our approach ( $O(1 \text{ hr})$  training the QC approach for 1 epoch on 4 CPUs and 7-8 hours for training the QFM model for 1 epoch). Training on real quantum hardware is expected to be faster in the future. The current simulation time restrictions lead to a reduced expressiveness of the circuits

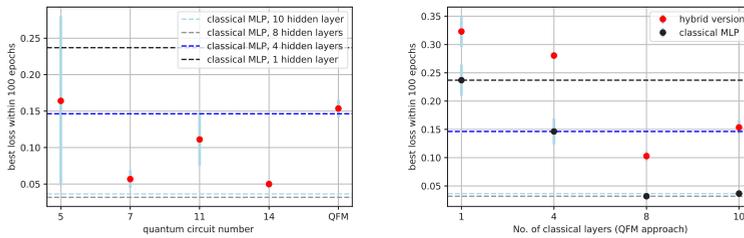


Figure 11: Comparison of best validation loss with different sized classical MLPs trained on 100 epochs. The left figure compares the QC approach circuits and the 10-classical layer QFM version. In the second figure, we vary the number of hidden layers in the classical MLP within the QFM approach. The presented mean and standard deviation is calculated for 3 independent runs each (2 independent runs for QFM 10 layer version). The classical versions are trained using  $lr = 1e^{-3}$  for the 8/10-layer version and  $lr = 1e^{-4}$  for the 1/4-layer version. The dashed lines indicate the respective mean value.

and a low number of parameters. The entanglement and expressibility values of the different tested quantum circuits can be improved by repeating their main structure and thus increase the number of parameters within the network. We observe that the success in representing the classical input within the quantum circuit highly depends on the training success of the classical MLP. More efficient encoding schemes will be tested as next steps. When comparing the best loss, there seems to be a trade-off between entanglement and expressibility. To test more complex circuits efficiently, it is important to speed-up the simulation process using GPUs and parallelization techniques and hence be able to use more qubits and more training data. This would open future possibilities to replace larger parts of classical neural networks with expressive quantum circuits for embedding and classification tasks. It would also allow us to increase the size of the training data sets. To conclude, we perform a successful data embedding step using hybrid quantum-classical neural networks. This is an important step in increasing the accuracy in the particle tracking task using GNNs. We plan to combine both, the embedding and the quantum GNN [5], to obtain this objective.

## Acknowledgements

Part of this work was conducted at QUTE, the CTIC quantum computing simulation platform. We thank Luis Meijueiro from the QUTE team at CTIC as well as Alessandro Roggero for fruitful discussions.

## References

- [1] D. Contardo, M. Klute, J. Mans, L. Silvestris, J. Butler, Tech. Rep. CERN-LHCC-2015-010. LHCC-P-008. CMS-TDR-15-02, Geneva (2015), upgrade Project Leader Deputies: Lucia Silvestris (INFN-Bari), Jeremy Mans (University of Minnesota) Additional contacts: Lucia.Silvestris@cern.ch, Jeremy.Mans@cern.ch, <https://cds.cern.ch/record/2020886>
- [2] S. Amrouche, L. Basara, P. Calafiura, V. Estrade, S. Farrell, D.R. Ferreira, L. Finnie, N. Finnie, C. Germain, V.V. Gligorov et al., The Springer Series on Challenges in Machine Learning p. 231–264 (2019)
- [3] S. Farrell, P. Calafiura, M. Mudigonda, Prabhat, D. Anderson, J.R. Vlimant, S. Zheng, J. Bendavid, M. Spiropulu, G. Cerati et al., *Novel deep learning methods for track reconstruction* (2018), 1810.06111

- [4] N. Choma, D. Murnane, X. Ju, P. Calafiura, S. Conlon, S. Farrell, Prabhat, G. Cerati, L. Gray, T. Klijnsma et al., *Track seeding and labelling with embedded-space graph neural networks* (2020), **2007**.00149
- [5] C. Tüysüz, F. Carminati, B. Demirköz, D. Dobos, F. Fracas, K. Novotny, K. Potamianos, S. Vallecorsa, J.R. Vlimant, EPJ Web of Conferences **245**, 09013 (2020)
- [6] C. Tüysüz, K. Novotny, C. Rieger, F. Carminati, B. Demirköz, D. Dobos, F. Fracas, K. Potamianos, S. Vallecorsa, J.R. Vlimant, *Performance of particle tracking using a quantum graph neural network* (2021), **2012**.01379
- [7] X. Ju, S. Farrell, P. Calafiura, D. Murnane, Prabhat, L. Gray, T. Klijnsma, K. Pedro, G. Cerati, J. Kowalkowski et al., *Graph neural networks for particle reconstruction in high energy physics detectors* (2020), **2003**.11603
- [8] J. Preskill, *Quantum* **2**, 79 (2018)
- [9] M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010)
- [10] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M.S. Alam, S. Ahmed, J.M. Arrazola, C. Blank, A. Delgado, S. Jahangiri et al., *Pennylane: Automatic differentiation of hybrid quantum-classical computations* (2020), **1811**.04968
- [11] D. Rousseau, S. Amrouche, P. Calafiura, V. Estrade, S. Farrell, C. Germain, V. Gligorov, T. Golling, H. Gray, I. Guyon et al., *The TrackML challenge*, in *NIPS 2018 - 32nd Annual Conference on Neural Information Processing Systems* (Montreal, Canada, 2018), pp. 1–23, <https://hal.inria.fr/hal-01745714>
- [12] S. Sim, P.D. Johnson, A. Aspuru-Guzik, *Advanced Quantum Technologies* **2**, 1900070 (2019)
- [13] D.A. Meyer, N.R. Wallach, *Journal of Mathematical Physics* **43**, 4273 (2002), <https://doi.org/10.1063/1.1497700>
- [14] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, S. Woerner, *The power of quantum neural networks* (2020), **2011**.00027
- [15] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, N. Killoran, *Quantum embeddings for machine learning* (2020), **2001**.03622
- [16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein, L. Antiga et al., *Pytorch: An imperative style, high-performance deep learning library* (2019), **1912**.01703
- [17] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F.J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.F. Chen et al. (2019)
- [18] D.P. Kingma, J. Ba, *Adam: A method for stochastic optimization* (2017), **1412**.6980
- [19] M. Cerezo, P.J. Coles, *Impact of barren plateaus on the hessian and higher order derivatives* (2020), **2008**.07454