Reconstructing Particle Decay Trees with Quantum Graph Neural Networks in High Energy Physics

Melvin Strobl, Eileen Kuehn, Max Fischer, Achim Streit

Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Baden-Wuerttemberg, Germany

E-mail: melvin.strobl@kit.edu, eileen.kuehn@kit.edu, max.fischer@kit.edu, achim.streit@kit.edu

Abstract.

Quantum Computing and Machine Learning are both significant and appealing research fields. In particular, the combination of both has led to the emergence of the research field quantum machine learning which has recently taken enormous popularity. We investigate in the potential advantages of this synergy for the application in high energy physics, more precisely in reconstructing particle decay trees in particle collision experiments. Due to the larger computational space of quantum computers, this highly complex combinatorical problem is well suited for investigating in a potential quantum advantage compared to the classical scenario. However, current quantum devices are subject to noise and provide only a limited number of qubits. We therefore propose the utilization of a variational quantum circuit within a classical graph neural network which has been shown to be feasible for reconstruction of particle decay trees. We evaluate our approach on a quantum simulator and a real quantum computer by IBM Quantum for artificially generated decay trees and compare our results to the purely classical approach. Our proposed approach does not only enable hardware efficient utilization of nowadays quantum devices, but also shows competitive results even in the presence of noise.

1. Introduction

In recent years, there has been much research in the emerging field of Quantum Machine Learning (QML) with a focus on High Energy Physics (HEP) [1, 2]. This interest in the union of quantum computing and Machine Learning (ML) is mainly motivated by having noise-resistant classical models combined with a large computational space covered by even small quantum circuits. Although Noisy Intermediate Scale Quantum (NISQ) computers are not yet able to compete with classical approaches in terms of size, accuracy, and speed, this may change as the quality and scale of these devices rapidly increases [3].

In the case of HEP, the Full Event Interpretation [4] is a highly non-trivial task whose computational complexity seems to be a promising area of research for QML [1]. Classically this task is done using Boosted Decision Trees which, however, require domain-knowledge due to hard-coded specific decay-processes. A more universal and end-to-end trainable solution has been introduced in [5]. The authors propose a Graph Neural Network (GNN) based approach that builds on the encoder part of the Neural Relational Inference model from [6]. The latter proposes an auto-encoder related approach to predict the trajectories of particles based on their interactions, embedded in the latent space. To obtain this latent space representation, the encoder

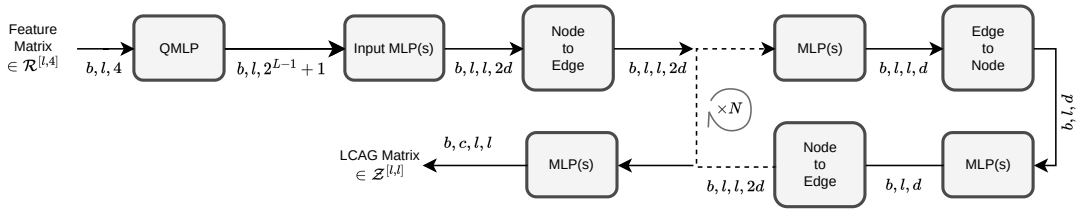


Figure 1: Starting with the input feature matrix, a hybrid-classical QMLP processes the input of b batches of l particles (L in total) with four features each and passes the result to the iterative GNN from [5] to calculate the corresponding LCAG matrix. The latter processes the graph iteratively N times with a hidden dimension of d.

must learn the interactions between individual particles. In [5], however, the GNN reconstructs a Lowest Common Ancestor Generation (LCAG) matrix, which is a dense representation of the structural properties of a graph. Unlike an adjacency matrix, the size of the LCAG does not vary with the number of intermediate nodes in the decay tree, but is fixed by the number of Final State Products (FSPs). Although this representation is ambiguous as to the exact geometry of the graph, the structural properties are deterministic.

In the work from [2], the authors developed a hybrid quantum-classical GNN to classify, whether the particle trajectories between layers of the detector are feasible or not. To this end, they introduced hybrid layers within the GNN that function as node-to-edge and edge-to-node networks. As this approach already showed promising results to classify tracks, we investigate in the options of a hybrid quantum-classical solution for the particle decay tree reconstruction.

2. Approach

We extend the existing work from [5] into a hybrid quantum-classical model by preceding a Quantum Multi Layer Perceptron (QMLP), see Figure 1. With this, we aim to infer information from the input feature matrix to achieve a higher accuracy with the subsequent classical GNN.

2.1. Data Generation and Preprocessing

We utilize the Phasespace module [7] to generate artificial decay data that serves as our input data. On an abstract level, the decay events are being parameterised by i) maximum and minimum number of children that result from each particle decay; ii) maximum depth (number of generations) of the decay tree; iii) number of different topologies; and iv) number of events per topology. Intermediate decay particles describe children of the parent particle which decay further either into subsequent intermediate decay particles or into FSPs. Decay trees of the same topology result in the same LCAG, although the masses and momenta of individual particles may differ. In this simplified scenario, missing/virtual particles and incorrect sensor values are neglected.

The resulting input feature matrix is of dimension $4 \times l$, where l depicts the number of FSPs with its four features, i.e. the momentum $p^{(l)}_{[x,y,z]}$ in x, y, and z direction, and the particles energy $E^{(l)}$. To encode this onto a quantum circuit, i.e. the Input Encoding Circuit (IEC), the data needs to be in the range $[0 \dots 1]$ and therefore requires normalisation. Due to the different physical properties of the input features, normalisation for momentum and energy is done separately. The label data consists of the LCAG matrix representation of the decay tree. This matrix describes the structural properties of the decay tree by indicating the first common ancestor of two different FSPs when looking from the leaves to the root. Details of this representation can be found in [5].

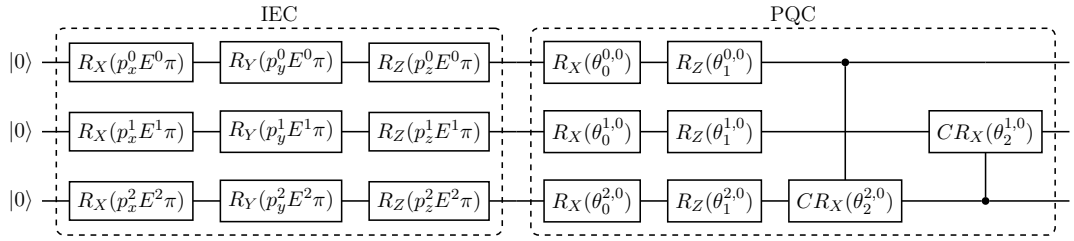


Figure 2: The IEC and PQC of our approach. In the IEC the three moments $p^{(l)}_{[x,y,z]}$ are encoded together with the energy $E^{(l)}$ of a single FSP as parameters of rotational gates. The following PQC is derived from Circuit 19 as introduced in [9]. With the data reuploading [8] used in this approach, the entire circuit in this figure forms a single layer. For simplicity, the qubit indices are omitted.

2.2. Reconstructing the Decay Tree

After generation and preprocessing of the input data, it is passed to the hybrid network to generate the LCAG matrix representation of the decay graph. The schematic process with the different data dimensions is shown in Figure 1. In the QMLP block the IEC, Parameterizable Quantum Circuit (PQC), and measurement interpretation are summarised. Details of the IEC and PQC are illustrated in Figure 2. For data encoding we utilize angle-encoding to map the classical data on the quantum circuit with rotational gates around axes x, y, and z that are parameterised by $p^{(l)}_{[x,y,z]}E^{(l)}\pi$. The normalization ensures that the maximum angle of rotation around each of the three axes is π. In this approach, we consider data reuploading [8], where the IEC is repeated in an alternating sequence with the PQC.

The proposed PQC is similar to the Circuit 19 ansatz from [9]: The PQC of a single layer (of K layers) consists of two rotational gates $R_X^{(l)}(\theta_0^{(l,k)})$, $R_Z^{(l)}(\theta_0^{(l,k)})$, and a controlled rotation gate $CR_X^{(l,l+1)}(\theta^{(l,k)})$, where l denotes the qubit/ FSP and $k \in \{0, \ldots, K-1\}$ the layer inside the ansatz. Considering L FSPs and K layers, the total number of parameters is $(3L-1)K$.

The number of qubits L is not varied during training, as this would lead to a varying number of parameters, which is not feasible in the context of a ML training pipeline, although different events could have FSPs with $l \in [1 \ldots L]$. Therefore, classical post-processing is required to adjust the measurement results so that they represent the actual number of FSPs in the current event.

By measuring the quantum circuit, we obtain 2^L pseudo probabilities. The GNN however, expects an input of size $l \times d$, where d describes a sufficiently large number of hidden elements for each FSP indicated by l. From the 2^L measurement results, all pseudo-probabilities are combined where the qubit with index $\leq l$ is measured in the $|1\rangle$ state which then yields 2^{L-1} values. The $|0\rangle^{\otimes}L$ state is also included, thus resulting in $2^{L-1}+1$ values. By iterating over l FSPs (i.e. qubits), the output size becomes $l \times 2^{L-1}+1$. This approach introduces some redundancy in the output of the quantum layer, but adds structural information to the feature vector for each FSP, which we found to be beneficial for the further GNN. An additional dimension b defines the number of samples per batch.

3. Experiments

The implementation of the GNN is done in PyTorch [10] and the quantum circuits are simulated using Qiskit [11]. All experiments are carried out using the Kedro and MlFlow frameworks. The generated decay data is divided into 75 % training, 17 % validation and 8 % test subsets. The source code is available on Github [1].

[1] `https://doi.org/10.5281/zenodo.7746171`

Table 1: Results and number of parameters in comparison to the classical approach.

Classes	Loss	Accuracy	Logic Acc.	Perfect LCAG	Parameters
GNN A	0.129	0.809	0.796	0.385	37 859
GNN B	0.085	0.852	0.845	0.529	147 395
QGNN (simulation)	0.147	0.824	0.824	0.803	38 317
QGNN (*ibm_perth*)	0.235	0.655	0.655	0.404	38 317

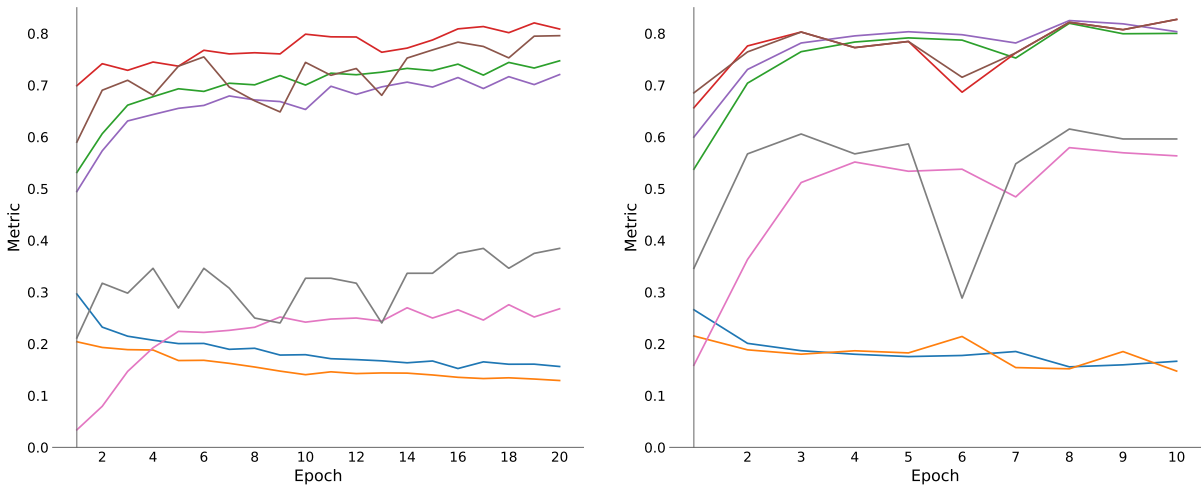
We carried out experiments in both a simulation environment and on a real quantum device from IBM Quantum. Training on real quantum devices requires the application of the parameter shift rule, which yields three circuit evaluations for each parameter [12]. Due to the limited access to these machines and the resulting significant overhead of communicating via cloud access, training on real quantum devices is infeasible for the scale of this application. Thus, we train the Quantum Graph Neural Network (QGNN) in a simulation environment and validate the trained model on a real quantum device, *ibm_perth*. Parameters are tuned using ADAM optimizer with a constant learning rate combined with cross-entropy loss and dropout.

Figure 3b shows the performance (metric over epoch) of our approach while the performance of the original approach of [5] on the same dataset is shown in Figure 3a. In addition to the common accuracy score, we also evaluate the Perfect LCAG and Logic Accuracy. The former indicates how well the model reconstructs the exact LCAG matrix as provided in the label dataset. The Logic Accuracy indicates how useful the model output is after post-processing. This post-processing involves pruning structural properties that are not feasible within the constraints imposed on the original decay dataset. Although the classical GNN achieves a higher validation accuracy of 0.852 compared to 0.824 from our approach, the proposed approach significantly reduces the number of parameters that need to be trained to 25.996 % (147 395 (GNN B) vs. 38 317 (QGNN))

GNN A reflects a classical approach where the number of parameters (37 859) has been reduced to a similar number as in the case of QGNN (38 317). Here the QGNN model outperforms the classical approach in almost all metrics except loss. Furthermore, the perfect LCAG score of our approach is significantly higher even when compared to the much larger GNN B with 147 395 parameters. On the real quantum device, the model achieves a validation accuracy of 0.655 and a perfect lcag score of 0.404. This drop in accuracy can most likely be explained by the noisiness of the device. A detailed comparison of the obtained results is shown in Table 1.

4. Discussion

We propose to extend a Graph Neural Network (GNN) by a Quantum Multi Layer Perceptron (QMLP) and show that it is able to outperform the purely classical experiment on the perfect Lowest Common Ancestor Generation (LCAG) score. Furthermore, the accuracy and logical accuracy scores as well as the loss can be considered as very competitive. Especially in terms of parameters, the quantum part seems to significantly improve the overall performance. This can also be seen when comparing the hybrid approach with a classical one, where the number of parameters is similar. In this scenario, all metrics are significantly outperformed by the hybrid approach. However, validation on a real quantum computer reduces, for example, the perfect LCAG score by almost 20 %. This motivates further research towards noise-resistant approaches, which might also improve accuracy on noisy input data. We want to note that the much longer training time, the limited access to real quantum devices and the limited scalability of this approach does not make real-world applications feasible in the near future, but rather helps to build an understanding and intuition of quantum technologies and their application.



(a) Performance of the classical GNN approach from [5].

(b) Performance of the proposed hybrid quantum-classical approach.

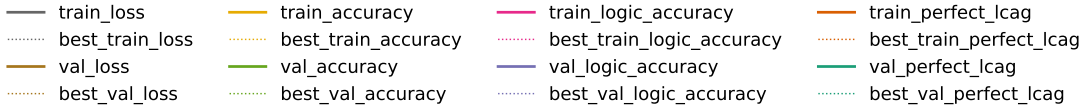


Figure 3: Validation scores of the classical in comparison with the hybrid quantum-classical approach trained on the same dataset. Besides the common validation score, the practical usability of the results (logic accuracy) and the direct similarity with the label (perfect LCAG) metrics are plotted.

References

[1] Wen Guan, Gabriel Perdue, Arthur Pesah, et al. Quantum machine learning in high energy physics. 2020.

[2] Cenk Tüysüz, Carla Rieger, Kristiane Novotny, et al. Hybrid quantum classical graph neural networks for particle track reconstruction. Technical Report University,, 2021.

[3] J. M. Gambetta. IBM Quantum roadmap to build quantum-centric supercomputers, February 2021.

[4] T. Keck, F. Abudinén, Florian U. Bernlochner, et al. The Full Event Interpretation. *Computing and Software for Big Science*, 3(1):6, February 2019.

[5] James Kahn, Ilias Tsaklidis, Oskar Taubert, et al. Learning tree structures from leaves for particle decay reconstruction. *Machine Learning: Science and Technology*, 3(3):035012, September 2022.

[6] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, et al. Neural relational inference for interacting systems. 2018.

[7] Albert Navarro and Jonas Eschle. phasespace: n-body phase space generation in python. *Journal of Open Source Software*, 4(42):1570, October 2019.

[8] Adriá n Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, feb 2020.

[9] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

[10] Adam Paszke, Sam Gross, Francisco Massa, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[11] A. N. I. S. MD SAJID, Abby-Mitchell, Héctor Abraham, et al. Qiskit: An open-source framework for quantum computing, 2021.

[12] Maria Schuld, Ville Bergholm, Christian Gogolin, et al. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, March 2019. arXiv:1811.11184 [quant-ph].