Hybrid Quantum Kolmogorov-Arnold Networks for High Energy Physics Analysis at the LHC

Siddhant Dutta[1] and Sadok Ben Yahia[2]

[1] Nanyang Technological University, Singapore
siddhant010@e.ntu.edu.sg
[2] University of Southern Denmark (SDU)
say@mmmi.sdu.dk

Abstract. The advent of the High-Luminosity Large Hadron Collider (HL-LHC) presents formidable computational challenges, necessitating innovative machine learning architectures for High Energy Physics (HEP) data analysis. Kolmogorov-Arnold Networks (KANs), inspired by the Kolmogorov-Arnold representation theorem, deviate from traditional Multi-Layer Perceptrons (MLPs) by employing learnable activation functions on network edges rather than fixed activations on nodes. This work introduces Hybrid Quantum KANs (QKANs), a hybrid quantum-classical architecture wherein the base linear transformation component of a KAN layer is realized by a variational quantum circuit (VQC). We detail the architectural distinctions between KANs and QKANs & present a comparative performance analysis on a benchmark HEP task: the classification of high-pT jets using simulated LHC proton-proton collision data via the Pennylane framework. Preliminary findings suggest QKANs exhibit validation performance comparable to classical KANs, warranting further investigation into their potential for enhanced generalization & applicability in complex scientific domains.[3]

Keywords: KANs · HEP · Jet Classification · QML · Hybrid Quantum-Classical Models · Learnable Activation Functions.

1 Introduction

The High-Luminosity Large Hadron Collider (HL-LHC) is projected to significantly increase data output, demanding novel computational approaches for High Energy Physics (HEP) [1][2]. Machine learning (ML), particularly deep learning, is integral to HEP data analysis. Kolmogorov-Arnold Networks (KANs) [3], based on the Kolmogorov-Arnold representation theorem, offer an alternative to MLPs by locating learnable activation functions on network edges. KANs replace fixed linear weights with learnable univariate spline functions, potentially offering better accuracy & interpretability.

[3] Code available at: https://github.com/elucidator8918/QKAN-ML4SCI/tree/main/Task-IX

This work proposes Hybrid Quantum KANs (QKANs), a hybrid architecture that integrates variational quantum circuits (VQCs) into KAN layers. Specifically, the classical base linear transformation within a KAN layer is substituted by a PQC, while the spline-based activations remain classical. This research aims to evaluate the feasibility & performance of QKANs for HEP tasks, implemented using Pennylane [4], against classical KANs.

2 Architectural Frameworks

Kolmogorov-Arnold Networks (KANs). Constructed from layers where each edge, connecting a neuron from layer l to layer $l+1$, embodies a learnable univariate activation function $\phi_{ji}^{(l)}(x_i^{(l)})$. The activation of neuron j in layer $l+1$ is the sum of these functions applied to the activations $x_i^{(l)}$ from layer l:

$$x_j^{(l+1)} = \sum_{i=1}^{n_l} \phi_{ji}^{(l)}(x_i^{(l)}) \tag{1}$$

Each $\phi_{ji}^{(l)}$ is parameterized as a sum of a base function — in our case, the SiLU activation applied to a linear transformation of $x_i^{(l)}$ — & a B-spline:

$$\phi_{ji}^{(l)}(x_i^{(l)}) = w_{ji}^{base} \cdot \text{base\_act}(x_i^{(l)}) + \sum_k c_{jik}^{spline} B_k(x_i^{(l)}; \mathbf{g}_{ji}^{(l)}) \tag{2}$$

where w_{ji}^{base} & c_{jik}^{spline} are learnable coefficients, & B_k are B-spline basis functions defined over a learnable grid $\mathbf{g}_{ji}^{(l)}$. The grid size & spline order are key hyperparameters. A KAN network is a composition of such layers: $\text{KAN}(\mathbf{x}) = (\mathbf{\Phi}^{(L-1)} \circ \cdots \circ \mathbf{\Phi}^{(0)})(\mathbf{x})$.

Quantum Kolmogorov-Arnold Networks (QKANs). Constructed by replacing the base linear transformation component in Equation 2 (or its layer-wise matrix equivalent) with a VQC. The ‘QKANLinear‘ layer thus integrates quantum processing. For an input $\mathbf{x}^{(l)}$, first processed by a classical base activation $\mathbf{x}'^{(l)} = \text{base\_act}(\mathbf{x}^{(l)})$, the quantum operations are:

1. **Quantum Data Encoding**: $\mathbf{x}'^{(l)}$ is encoded into a quantum state $|\psi(\mathbf{x}'^{(l)})\rangle$ using the `qml.AngleEmbedding` method.
2. **Variational Quantum Circuit (VQC)**: A parameterized quantum circuit $U(\boldsymbol{\theta})$ — specifically, the `qml.BasicEntanglerLayers` with learnable parameters $\boldsymbol{\theta}$ — is applied to $|\psi(\mathbf{x}'^{(l)})\rangle$.
3. **Quantum Measurement**: Expectation values of observables, such as $\langle \sigma_z \rangle$ on output qubits, yield a classical vector, $\mathbf{o}_{qbase}$.

The output of the j-th QKANLinear neuron is then:

$$x_j^{(l+1)} = (\mathbf{o}_{qbase})_j + \sum_{i=1}^{n_l} \sum_k c_{jik}^{spline} B_k(x_i^{(l)}; \mathbf{g}_{ji}^{(l)}) \tag{3}$$

The VQC parameters $\boldsymbol{\theta}$ are co-optimized with classical spline & grid parameters.

3 Experiments & Results

We use the *High-pT Jets Dataset* from HLS4ML, consisting of 53 high-level features per jet for classifying jets into five categories (g, q, W, Z, t). Features were standardized using `StandardScaler`, & labels were one-hot encoded.

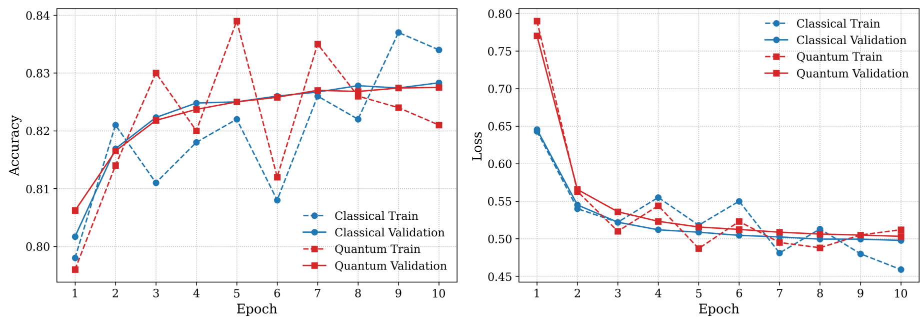

Fig. 1. Training & validation performance comparison between classical KAN & hybrid QKAN models. Left: Accuracy over epochs. Right: Loss over epochs.

Benchmark Model Setup. The classical KAN & hybrid quantum QKAN architectures were configured with identical input dimensionality of 53 features, a grid size of 7, spline order of 13, & a fixed random seed of 42. The classical model consists of hidden layers with sizes [53, 32, 16, 5, 5], while the quantum variant uses [53, 32, 16, 5], followed by a QKANLinear layer with 5 input & 5 output units. The QKAN employs 5 qubits & a variational quantum circuit (VQC) composed of 5 layers. Both models were trained using the AdamW optimizer with a learning rate of 1e−3, weight decay of 1e−4, & a CrossEntropy loss function, over 10 epochs with a batch size of 2048.

Table 1. Performance Metrics

Model	Training		Validation	
	Accuracy (%)	Loss	Accuracy (%)	Loss
KAN	83.37	0.459	82.83	0.498
QKAN	82.05	0.512	82.75	0.503

Analysis. The performance metrics for both models are in Table 1. The classical KAN had slightly higher training accuracy (83.37% vs. 82.05%) and lower training loss (0.459 vs. 0.512) than the QKAN. On the validation set, QKAN and KAN performed similarly, with validation accuracies of 82.75% and 82.83%, and validation losses of 0.503 and 0.498. Figure 1 shows training and validation accuracy and loss over epochs. While KAN slightly outperformed QKAN in training, QKAN matched validation performance, indicating the quantum layer may aid regularization or optimization and improve explainability.

References

1. Javier Duarte, Song Han, Philip Harris, Sergo Jindariani, Edward Kreinar, Benjamin Kreis, Jennifer Ngadiuba, Maurizio Pierini, Ryan Rivera, Nhan Tran, et al. Fast inference of deep neural networks in fpgas for particle physics. *Journal of instrumentation*, 13(07):P07027, 2018.
2. Eric A Moreno, Olmo Cerri, Javier M Duarte, Harvey B Newman, Thong Q Nguyen, Avikar Periwal, Maurizio Pierini, Aidana Serikova, Maria Spiropulu, and Jean-Roch Vlimant. Jedi-net: a jet identification algorithm based on interaction networks. *The European Physical Journal C*, 80:1–15, 2020.
3. Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024.
4. Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M Sohaib Alam, Guillermo Alonso-Linaje, B AkashNarayanan, Ali Asadi, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.