EPJ Web of Conferences **295**, 09030 (2024) https://doi.org/10.1051/epjconf/202429509030
CHEP 2023

Pion/Kaon Identification at STCF DTOF Based on Classical/Quantum Convolutional Neural Network

Zhipeng Yao[1,*], *Teng* Li[1,**], and *Xingtao* Huang[1,***]

[1]Shandong University, Qingdao, Shandong, 266237, People's Republic of China

Abstract. Particle identification (PID) is one of the most fundamental tools in various physics research conducted in collider experiments. In recent years, machine learning methods have gradually become one of the mainstream methods in the PID field of high-energy physics experiments, often providing superior performance. The emergence of quantum machine learning may potential arm a powerful new toolbox for machine learning. In this work, targeting at the $\pi^{\pm}/K^{\pm}$ discrimination problem at the STCF experiment, a convolutional neural network (CNN) in the endcap PID system is developed. By combining the hit position and arrival time of each Cherenkov photon at the sensors, a two-dimensional pixel map is constructed as the CNN input. The preliminary results show that the CNN model has a promising performance. In addition, based on the classical CNN, a quantum convolution neural network (QCNN) is developed as well, exploring possible quantum advantages provided by quantum machine learning methods.

1 Introduction

The Super Tau-Charm Facility (STCF) [1] is a future electron-positron collider proposed with a peak luminosity of 0.5×10^{35} cm^{-2} s^{-1} and center-of-mass energy ranging from 2 to 7 GeV. STCF is designed to study the tau, charm physics, hadron physics as well as searching new physics beyond the Standard Model. The particle identification (PID) as one of the most fundamental tools in various physics studies, is critical for the physics potential of STCF. The STCF detector is designed with multiple sub-systems that have PID capabilities, including the dE/dx measurement in the drift chamber, the electromagnetic calorimeter and the muon detector. To boost the hadron identification capabilities, two PID systems, including a Ring-imaging Cherenkov detector (RICH) as well as a DIRC-like time-of-flight detector (DTOF [2]) are equipped in the barrel and endcap region of STCF, respectively.

Over the past two decades, machine learning algorithms, such as boosted decision trees [3] and artificial neural networks [4], have gradually become one of the most performant approaches for PID problems in various collider experiments. One of the most significant merits of machine learning algorithms is their capability to extract effective information from large quantity of data, thus often delivering exceptional performance. In recent years, the emergence of quantum computing has offered opportunities for high-energy physics (HEP)

*e-mail: yaozp@mail.sdu.edu.cn
**e-mail: tengli@sdu.edu.cn
***e-mail: huangxt@sdu.edu.cn

experiments. In particular, the development of quantum machine learning (QML) [5, 6] has connected the domain of quantum computing and machine learning. Some previous studies have shown that QML algorithms have the potential to surpass traditional machine learning algorithms by for instance, accelerating the convergence speed [7, 8]. Therefore, it is valuable and interesting to study the feasibility of employing QML models on various data processing tasks at HEP experiments.

In this work, targeting at the $\pi^{\pm}$ and $K^{\pm}$ discrimination at the STCF experiment, we first developed a CNN model that takes the pixel map constructed from the time and spatial information of Cherenkov photons collected by the sensors in the DTOF detector. Based on the baseline CNN model, we further explored the QCNN on quantum simulators, applying trainable quantum convolutional layers in CNN to explore potential advantages.

2 Methodology

2.1 Convolutional Neural Network

CNN is a type of machine learning model primarily used for processing data with grid-like structures such as images. The origin of CNN can be traced back to the 1980s when Fukushima proposed the Neocognitron model in his work on neural cognitive machines. However, it was the LeNet [9] model proposed by LeCun et al. in 1998 that truly garnered widespread attention, as it demonstrated outstanding performance in handwritten digit recognition tasks. Currently, CNN has become an indispensable tool in the image processing and pattern recognition domain, significantly contributing to the resolution of intricate visual and language tasks.

CNN usually consists of stacking convolutional layers, pooling layers and fully connected layers. Specifically,

1. The convolutional layer extracts features from input data. Each convolutional kernel performs element-wise multiplication with a small region of the input data and sums up the results, generating a feature map.

2. The pooling layer is used to reduce the spatial dimensions of the feature maps to decrease computational complexity and effectively prevent the model from overfitting.

3. The fully connected layer flattens the feature maps into a vector and processes it through a series of fully connected neurons. Finally, the output is mapped to the desired range through an activation function to obtain the prediction result.

With the advancement of computer hardware and the availability of large-scale annotated datasets, CNN has achieved tremendous success in the field of computer vision and is widely applied in tasks such as image classification, object detection, and semantic segmentation.

2.2 Quantum Convolutional Neural Network

Although CNN has achieved great success in image processing tasks, it still faces certain difficulties and challenges in certain cases. As the dimension of the input data increases, the number of parameters in CNN grows fast, resulting in significant computational and storage costs. This limits the efficiency and scalability of CNN in handling large-scale and high-dimensional data.

Quantum computing algorithms usually take advantage of superposition and entanglement of quantum states, leading to highly parallel computations. Based on the idea of

processing high-dimensional complex data with relatively small number of parameters, the QCNN is proposed to explore potential quantum advantages against classical CNN [10]. The QCNN model is a quantum-classical hybrid algorithm, which is a feasible choice in the current noisy intermediate-scale quantum device (NISQ) era. Specifically, QCNN introduces quantum convolutional layers into the classical CNN architecture, which transforms local sub-regions of the input data by designing variational quantum circuits to extract new hidden features. In this way, by using only a small number of qubits, the quantum circuits can scan the local pixel values of the image multiple times, enabling large-scale image processing.

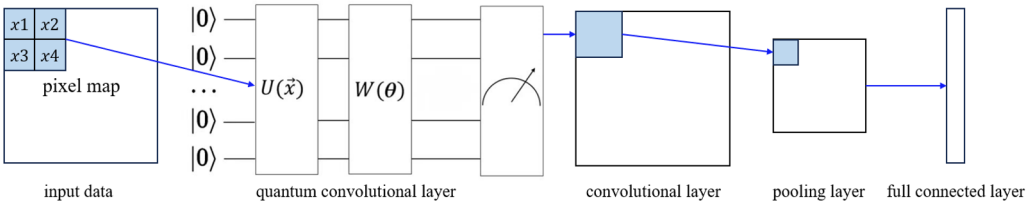


Figure 1: The structure of Quantum Convolutional Neural Network

The structure of QCNN is shown in Figure 1, where the implementation of the quantum convolutional kernels plays a crucial role. The quantum convolutional kernel can be divided into three parts as introduced in the following sub-sections.

2.2.1 Encoding Circuit

The initial step of the quantum convolutional kernel is the data encoding that transforms the input pixels into quantum states. This step is performed by the encoding circuit that projects local sub-regions of the input data $\vec{x}$ onto a quantum state $|\psi_{in}(\vec{x})\rangle$ described in a high-dimensional Hilbert space. By using one encoding circuit, we can encode the pixel values of an $N \times N$ region of an image onto a quantum state using N qubits. Similar to how the convolution is performed by classical convolution kernels, by iterating the same process on different regions, the entire input image can be encoded as quantum states.

2.2.2 Variational Circuit

The variational circuit $W(\theta)$ transforms the quantum state $|\psi_{in}(\vec{x})\rangle\rangle$ prepared earlier and generates output states $|\psi_{\text{out}}(\vec{x})\rangle$. The variational circuit could contain free parameters to be trained by an optimization algorithm (i.e. gradient descent), which is defined as:

$$|\psi_{\text{out}}(\vec{x})\rangle = W(\theta)\,|\psi_{\text{in}}(\vec{x})\rangle = U_{\text{rot}}(\theta)U_{\text{ent}} \ldots U_{\text{rot}}(\theta)U_{\text{ent}}\,|\psi_{in}(\vec{x})\rangle \tag{1}$$

The parameters of the circuit are mainly concentrated in $U_{\text{rot}}(\theta)$ which usually consists of a series of rotations on the qubits. It is worth noting that θ, as the free parameters of the circuit, is not directly associated with the input data but depends on the training of the variational circuit, which is different from the encoding circuit. In addition, U_{ent} refers to the entanglement between different qubits. By embedding a series of entanglement gates such as the Controlled-NOT (CNOT) gate in the variational circuit, it is possible to effectively create and manipulate entangled states between qubits, thereby enhancing the expressiveness of the circuit.

2.2.3 Measurement

The output of the quantum convolutional kernel depends on the result of the measurement of one or more qubits that define an observable $\widehat{O}$. According to the basic principle of quantum mechanics, the expectation value for measuring the observable $\widehat{O}$ is deterministic. The general form of the expectation value, as the output is given by

$$f(\theta) = \left\langle \psi | W^{\dagger}(\theta) \widehat{O} W(\theta) | \psi \right\rangle \tag{2}$$

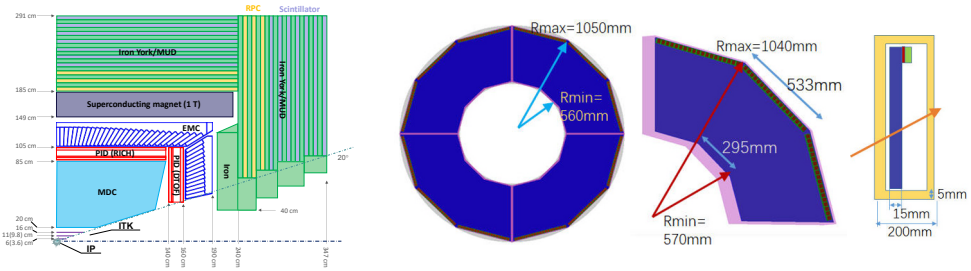
The probability distribution of the state of the measured qubit can be estimated after repeating the experiment for a reasonable number of times. Similar to classical convolutional layers, each expectation value corresponds to the convolutional features obtained by quantum convolutional processing on the original pixels.

Since the extracted new features are classical quantities, we can directly connect classical layers after the quantum convolutional layer. This allows us to couple the quantum convolutional layer with classical CNN layers, enabling joint training and optimization of classical and quantum layers.

3 Pion and Kaon Discrimination at the STCF DTOF detector

3.1 DTOF Detector

Figure 2(a) shows the STCF detector that is composed of a tracking system (an μWELL based inner tracker and a main drift chamber), a PID system composed of a Ring-imaging Cherenkov detector (RICH) and a DIRC-like time-of-flight detector (DTOF), an electromagnetic calorimeter (EMC), a superconducting solenoid (SCS) and a muon detector (MUD), from the interaction point outward. The DTOF detector is designed to enhance the PID of hadrons in the endcap region that employs a technology based on the detection of internal total-reflected Cherenkov light.



(a) (b)

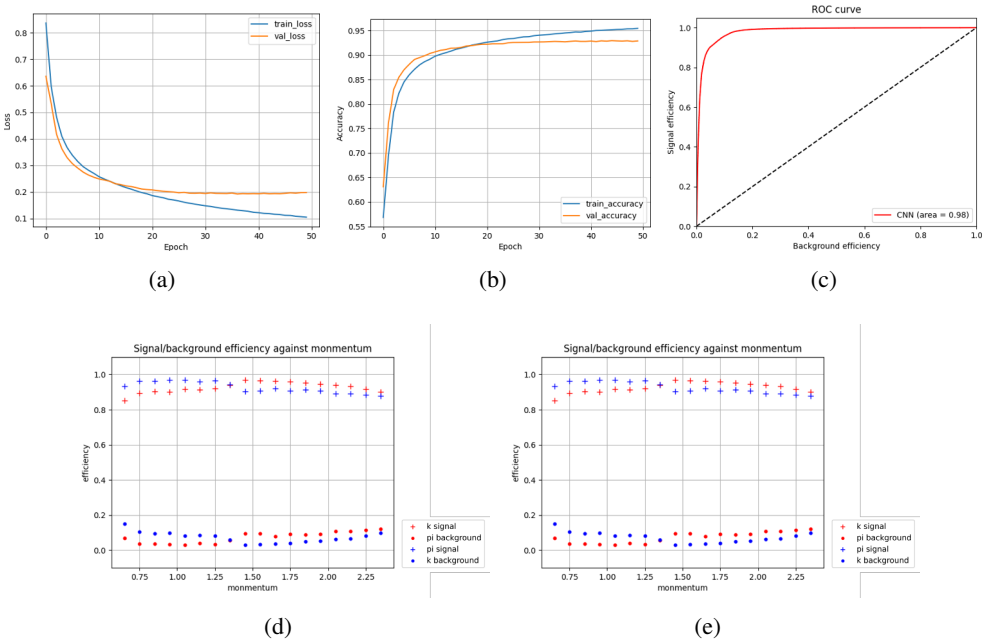
Figure 2: (a) Schematic layout of the STCF detector. (b) Geometry of the DTOF detector.

Figure 2(b) shows the geometry of the DTOF detector. It consists of two identical endcap discs located at $\sim \pm 1400$ mm from the collision point along the beam direction. Each disc is made up of a few sectors, covering in polar angles of $\sim 22^{\circ} - 36^{\circ}$. In each sector, a synthetic fused silica plate is used as a Cherenkov radiator to generate Cherenkov photons. The outer side of the radiator is optically coupled to an array of multianode micro-channel plate photomultipliers (MCP-PMTs). When charged particles hit the Cherenkov radiator,

Cherenkov photons are emitted, and these photons are collected by the externally coupled MCP-PMTs after undergoing multiple internal reflections within the radiator. Due to the different topologies of Cherenkov photons for various particle species, the DTOF detector is capable of discriminating between particles.

3.2 Data Sample

The $\pi^{\pm}$ and $K^{\pm}$ data sample is simulated using the STCF offline data processing software OSCAR [11]. By combining the hit positions and arrival times of the Cherenkov photons collected by MCP-PMTs, a two-dimensional pixel map of channel and arrival time of photons can be constructed. Based on the distribution of photon arrival time across different readout channels, we constructed an image of size 217×200 as the input classical CNN. The training, validation, and testing sets for the CNN contain 200k, 100k, and 100k tracks, respectively. Additionally, due to the limited qubits and to further accelerate computation speed, the image size prepared for the QCNN was reduced to 32×32, and the dataset size was set at 10% of the classical CNN dataset. By comparing the performance of QCNN and CNN using the same small dataset, we aim to explore the feasibility of QCNN.



(a) (b) (c)

(d) (e)

Figure 3: (a) Loss value versus number of epochs. (b) PID accuracy versus number of epochs. (c) ROC curve and AUC value of the classical CNN model. (d) Signal efficiency and background efficiency as a function of track momentum. (e) Signal efficiency and background efficiency as a function of incident polar angle.

3.3 Classical CNN performance

The classical CNN used to discriminate $\pi^{\pm}$ and $K^{\pm}$ consists of three convolution layers and three pooling layers, as well as two fully connected layers before the softmax output layer.

In the fully connected layer, additional input neurons for track momentum and incident angle are added since the photon topology is also relevant to the momentum and flying angle. The ADAM optimizer [12] is employed to optimize the cross-entropy loss.

Figures 3(a) and 3(b) show the learning curves of the CNN model. As the optimizer iterates, the model's loss value on the validation set continuously decreases, while the accuracy steadily improves. Eventually the model converges, achieving an accuracy of approximately 92% and an AUC value of 0.98. This demonstrates CNN's excellent ability to extract features from the original images. The signal efficiency and background efficiency of $\pi^{\pm}$ and $K^{\pm}$ as a function of momentum and polar angle in the test set, as shown in Figures 3(d) and 3(e). Currently, the CNN model exhibits sub-optimal performance in discriminating $\pi^{\pm}$ and $K^{\pm}$ with higher or lower momentum, as well as at larger incident polar angles. Further optimization of the model is required to address these limitations.

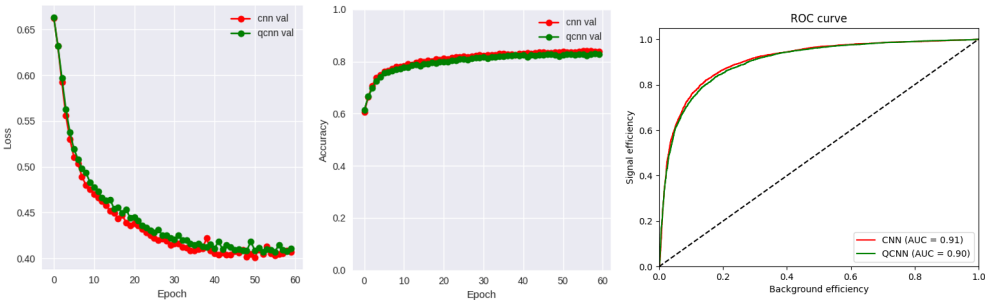
3.4 QCNN performance

Due to the current limitations of quantum simulator, the original image size is too large to process, therefore images for QCNN are scaled down to 32×32. As a comparison with the CNN model in the previous section, we developed a simpler QCNN model using the TensorFlow Quantum simulator platform [13].

Figure 4: The schematic diagram of the quantum convolutional kernel. "RX", "RY", and "RZ" represent single-qubit rotation gates around the X, Y, and Z axes, respectively, while "⊕" represents the CNOT gate with entanglement operation.

This QCNN model first employs a quantum convolutional layer, constructed as shown in Figure 4, to process the original images. It is then followed by a classical CNN model consists of classical convolution and pooling layers. In Figure 4, the left side of the red dashed line is the encoding circuit which maps the pixel values to quantum states utilizes a set of Pauli rotation gates. On the right side, the variational circuit is constructed by introducing trainable parameters based on Pauli rotation gates as well as some entanglement structures using CNOT gates. In terms of standard CNN terminology, the quantum convolution kernel corresponds to a 2×2 convolution kernel with a stride of one. Finally, the input of the following classical CNN are obtained by measuring the first qubit of the circuit.

Figures 5(a) and 5(b) show the comparison of the learning curves of the QCNN model and the classical CNN model on the same dataset during the training process. The CNN model has the same structure and comparable number of free parameters as the QCNN model, with the exception of the first convolution layer. We can see clearly that the QCNN and classical CNN models exhibit similar trends during the training process. Both models show a gradual convergence of accuracy and loss values during the training process, eventually reaching a similar level. This indicates that the QCNN and classical CNN models have similar learning capabilities. Figure 5(c) further demonstrates this by comparing the ROC curves and AUC values on the test set.



(a) (b) (c)

Figure 5: Comparison of learning curve and ROC curve between the QCNN and classical CNN.

4 Conclusion

In this work, targeting the $\pi^{\pm}/K^{\pm}$ discrimination problem at the STCF DTOF detector, we first constructed a two-dimensional channel-time pixel map as the input for the classical CNN. We then studied the learning capability of the CNN in extracting features. The results showed that the CNN model performed well in the $\pi^{\pm}/K^{\pm}$ discrimination problem. Additionally, we attempted to construct a QCNN, which utilized encoding circuits and variational circuits to build trainable quantum convolution layers coupled with classical layers. Compared to the corresponding CNN structure, the QCNN achieved similar performance on the same small dataset. Although the QCNN did not demonstrate significant quantum advantages, it still shows the feasibility of applying quantum machine learning techniques to HEP experiments. With the future development of quantum computing, more applications can be anticipated.

5 Acknowledgements

This work was supported by National Natural Science Foundation of China (NSFC) under Contracts Nos. 12025502, 12105158, 12188102.

References

[1] Q. Luo, D. Xu et al., *Progress on preliminary conceptual study of HIEPA, a super tau-charm factory in China*, in *9th International Particle Accelerator Conference* (2018), Vol. 6

[2] B. Qi, Z. Li, M. Shao, J. Liu, Z. Fang, H. Shi, X. Li, Journal of Instrumentation **16**, P08021 (2021)

[3] B.P. Roe, H.J. Yang, J. Zhu, Y. Liu, I. Stancu, G. McGregor, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **543**, 577 (2005)

[4] C.M. Bishop, *Neural networks for pattern recognition* (Oxford university press, 1995)

[5] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, Nature **549**, 195 (2017)

[6] M. Schuld, F. Petruccione, *Supervised learning with quantum computers*, Vol. 17 (Springer, 2018)

[7] S.L. Wu, S. Sun, W. Guan, C. Zhou, J. Chan, C.L. Cheng, T. Pham, Y. Qian, A.Z. Wang, R. Zhang et al., Physical Review Research **3**, 033221 (2021)
[8] A. Blance, M. Spannowsky, Journal of High Energy Physics **2021**, 1 (2021)
[9] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Proceedings of the IEEE **86**, 2278 (1998)
[10] M. Henderson, S. Shakya, S. Pradhan, T. Cook, Quantum Machine Intelligence **2**, 2 (2020)
[11] W. Huang, H. Li, H. Zhou, T. Li, Q. Li, X. Huang, Journal of Instrumentation **18**, P03004 (2023)
[12] D.P. Kingma, J. Ba, arXiv:1412.6980 (2014)
[13] M. Broughton, G. Verdon, T. McCourt, A.J. Martinez, J.H. Yoo, S.V. Isakov, P. Massey, R. Halavati, M.Y. Niu, A. Zlokapa et al., arXiv:2003.02989 (2020)