

---

# QFCNN: QUANTUM FOURIER CONVOLUTIONAL NEURAL NETWORK

---

**Feihong Shen**

Jilin University

shenfh5517@mails.jlu.edu.cn

**Jun Liu**

Singapore University of Technology and Design

junliu@sutd.edu.sg

June 22, 2021

## ABSTRACT

The neural network and quantum computing are both significant and appealing fields, with their interactive disciplines promising for large-scale computing tasks that are untackled by conventional computers. However, both developments are restricted by the scope of the hardware development. Nevertheless, many neural network algorithms had been proposed before GPUs become powerful enough for running very deep models. Similarly, quantum algorithms can also be proposed as knowledge reserve before real quantum computers are easily accessible. Specifically, taking advantage of both the neural networks and quantum computation and designing quantum deep neural networks (QDNNs) for acceleration on Noisy Intermediate-Scale Quantum (NISQ) processors is also an important research problem. As one of the most widely used neural network architectures, covolutional neural network (CNN) remains to be accelerated by quantum mechanisms, with only a few attempts have been demonstrated. In this paper, we propose a new hybrid quantum-classical circuit, namely Quantum Fourier Convolutional Network (QFCN). Our model achieves exponential speed-up compared with classical CNN theoretically and improves over the existing best result of quantum CNN. We demonstrate the potential of this architecture by applying it on different deep learning tasks, including traffic prediction and image classification.

## 1 Introduction

Existing works [1, 2] have attempted to incorporate the classical machine learning algorithms and quantum computation techniques, which yields quantum machine learning frameworks. It is also promising for deep convolutional neural network (CNN) to be leveraged with quantum methods towards achieving quantum convolutional models, thanks to the following reasons. First, CNNs manipulate data in the manner of performing matrix operations in high dimensional vector space, while quantum computation also conducts matrix operations in high dimensional vector space [1]. Second, most of the deep CNN models require large datasets for parameter optimization, and often need heavy computation for both the training and testing. Meanwhile, based on special quantum information properties named entanglement and superposition, quantum computers can make exponential number of calculations in parallel. This mechanism, known as quantum speedup, can accelerate some learning algorithms to surpass their classical counterparts based on mathematical proofs [3, 4, 5, 6].

Two decades ago, scientists began to implement quantum algorithms on real quantum machines. With the continuous development of quantum hardware, scientists suggest that human beings will live in Noisy Intermediate-Scale Quantum (NISQ) era in the future, and everyone is expected to have a quantum computer with 50-100 qubits, which could be used to perform some special tasks that surpass the capabilities of today's classical computers [7, 8].

Recently, deep neural networks become quite popular and have been widely applied in computer vision, natural language processing, and many other areas, thanks to the advancement of deep learning techniques [9, 10, 11]. As one of the most important deep learning techniques, the convolution module has been used in many network architectures, and many state-of-art models for handling different types of tasks contain convolutional layers [12, 13, 14, 15].

Given the aforementioned background of quantum computation and flourish CNN fields, it becomes natural that we get interested in integrating the potentials and advantages of these two kinds of techniques, i.e., quantum computing and CNNs, to handle specific convolution-based learning tasks that need continuously increasing computation capabilities. However, the direct convolution of quantum state has been shown to be inaccessible by previous studies [16]. In this paper, we propose a hybrid circuit with quantum Fourier transform to achieve deep convolutional models with quantum.

The major blocks in CNNs are the convolutional operations, i.e., the product of the kernel and the input data that can have various dimensions. This process often accounts for a large proportion of the computation cost and time consumption in the whole network. Another approach that can be used to efficiently implement the convolutional operation is transforming the kernel and data to Fourier basis and then performing inverse transform based on their product. Some of the classical networks like graph convolutional networks (GCNs) have used the approximation of this method to implement convolution [17, 18, 19]. Inspired by this, to achieve quantum convolution, here we propose a novel quantum Fourier convolutional network (QFCN) to speedup the CNN with quantum Fourier transform (QFT), which could be exponentially faster than the classical Fourier transform for convolutions. We circumvent the inaccessibility issue [16] by adopting part quantum circuit of windowed Fourier transform [20]. We also propose a hybrid quantum-classical circuit to avoid some of the quantum noise and also to take advantage of classical computers that are able to perform rapid iterations for model optimization. We demonstrate the potential of our network on the traffic prediction task with spatio-temporal graph convolutional models and the image classification task with 2D convolutional networks.

**Motivations and contributions.** We all know the importance of convolution to the deep learning field. If we want to speed up the training of nowadays' networks which mostly build upon a bunch of convolutional layers, the quantum convolutional layer and its optimization will be inevitable problems. There are only two architectures of convolutional layer being proposed[21, 22]. However, due to the horrific time assumption of quantum stimulation, the results of these two papers are based on toy experiments and their network only contain two quantum convolutional layers. Unfortunately, with the increase of the layer number, the gradient of these two structures may lead to Barren plateaus [23] in training. So, it is essential to proposed another quantum convolution method rather than simply transferring conventional multiply accumulation (MAC) to quantum mechanism like previous works.

The contributions of this paper are summarized below. To the best of our knowledge, we are the first to propose a hybrid quantum-classical circuit to speed up CNN with a parametric quantum circuit (PQC), and exploit quantum Fourier transform for the trainable convolution models. Specifically, our PQC can be inserted into different deep networks to replace the convolutional layers. Besides, we also introduce the quantum optimization algorithm (back-propagation) for our designed network.

## 2 Related Work

**Quantum machine learning.** Quantum machine learning algorithms can be roughly classified into two groups [1, 8]. The first group is the algorithms proposed before the concept ‘‘NISQ’’ have been published, such as quantum PCA [24] and quantum SVM [25], and most of them are still in the developing period. The second group is mainly the deep models with quantum network layers. Previous research works [26, 27, 28] have shown the feasibility of training the networks on NISQ devices. There are several quantum deep learning architectures [21, 29], such as quantum recommendation systems [30], quantum generative adversarial networks [31, 29]. Some models like graph networks [32] and recurrent neural networks [33] have also been inspired by quantum computing.

**Convolution.** As a popular neural network model, CNN usually contains four types of layer: the convolution layer, the activation function, the pooling layer and the fully connected (FC) layer. Recently, there have been a few attempts [21, 22] for designing quantum convolution models, and they mainly focus on using the quantum gate to simulate the layers in traditional CNNs. Differently, in our QFCN model, we exploit discrete quantum Fourier transform for the learnable convolutional modules, which still enables us to control the qubits with the number of kernel size. Besides, our designed quantum circuit makes it convenient to prepare the input signal to be involved in convolution. Also note that besides deep CNN, there are a few attempts focusing on pure quantum convolution without network architecture and learnable parameters [20], and our method is also different from them, since we aim to design a quantum Fourier convolutional neural network with trainable parameters, and the quantum optimization algorithm via back-propagation is also introduced in this paper to learn the parameters for our QFCN.

**Hybrid quantum-classical algorithm.** While quantum computers still face considerable technological challenges, many quantum algorithms have daunting resource requirements [34]. To handle this problem, some quantum algorithms [35] connected with classical computers are designed, which are often called hybrid quantum-classical algorithms. Besides saving quantum resources like quantum gates, hybrid quantum-classical algorithms also show other advantages. For instance, classical computers are able to perform rapid iteration. Thus some hybrid schemes make use of this

advantage to optimize the parametric quantum circuit [35]. Here the hybrid framework is exploited to design our QFCN model, considering its advantages.

### 3 Preliminaries and Notations

In this section, we discuss the basics of quantum computing and quantum information as the context of this paper, to facilitate the subsequent sections that introduce our proposed quantum Fourier Convolutional Network with the hybrid quantum-classical circuit.

**qubit.** Analogous to the bit in classical computers, qubit is a fundamental concept in quantum computing. Actually, qubits can represent both physical objects and abstract mathematical objects. As a physical object, the most important properties for qubits include the superposition for single qubit and entanglement of multi-qubits. (1) *Superposition*. Generally, a classical bit can only be in either state 1 or 0, while in quantum linear algebra, we analogize these two basic states to  $|1\rangle$  and  $|0\rangle$ , under the Dirac notation. A qubit is expressed by a linear combination of the two quantum states, denoted by:

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

where  $\alpha$  and  $\beta$  are probability amplitudes. The superposition can be viewed intuitively from the Bloch Sphere, as shown in Figure 1a. The positive direction of the z-axis represents  $|0\rangle$ , and the negative direction represents  $|1\rangle$ .  $\theta$  and  $\varphi$  are the phases off the z-axis and x-axis, respectively. Traversing all  $\theta$  and  $\varphi$  values, the qubit arrow forms the Bloch sphere. After the observation of this single qubit, its state will collapse into  $|1\rangle$  with possibility  $|\beta|^2$ , and  $|0\rangle$  with possibility  $|\alpha|^2$ , where  $|\alpha|^2 + |\beta|^2 = 1$ . (2) *Entanglement*. The quantum entanglement is a relation between qubits. Assume we have two qubits with unknown states to be entangled, if we observe the first qubit and get its state, the second qubit, at the same time, will collapse into the same state regardless of the distance. Any entangled state cannot be represented by the production of two single qubit state, for instance the state  $|\phi\rangle = \frac{1}{\sqrt{2}}|aa\rangle + \frac{1}{\sqrt{2}}|bb\rangle$ , cannot be decomposed by:

$$|\phi\rangle = (c_1|a\rangle + c_2|b\rangle) \otimes (c'_1|a\rangle + c'_2|b\rangle) = \frac{1}{\sqrt{2}}|aa\rangle + \frac{1}{\sqrt{2}}|bb\rangle \quad (2)$$

where  $\otimes$  means tensor product operation. It is obvious that we cannot find the solution of  $c_1c'_1 = c_2c'_2 = \frac{1}{\sqrt{2}}$  and  $c_1c'_2 = c_2c'_1 = 0$ . As a mathematical object, we use linear algebra in quantum way to represent the state of the qubit. Usually, we identify the state  $|1\rangle$  with the vector  $[0, 1]^T$  and state  $|0\rangle$  with the vector  $[1, 0]^T$ . Moreover, according to quantum linear algebra, we can represent the state  $|\psi\rangle$  in binary, such as :  $j = j_12^{n-1} + j_22^{n-2} + \dots + j_n2^0$ . We can also represent the binary fraction  $j_l/2 + j_{l+1}/4 + \dots + j_m/2^{m-l+1}$  by  $0.j_lj_{l+1}\dots j_m$ .

**Quantum gate.** Just like logic gates in classical computer, quantum gates are operators that can manipulate the state of qubits. In physics, the quantum gate can be regarded as the remove of qubit arrows on Bloch sphere from one point to another. In mathematics, quantum gates can be regarded as unitary matrices operating on the vectors of quantum states. There are two kinds of quantum gates: the single-qubit gate and the multi-qubit gate. All single-qubit gate can be written as the  $2 \times 2$  unitary matrix. For instance, the Hadamard gate is a frequently used single-qubit quantum gate, that allows to produce superposition of states from the basic states. We can represent the Hadamard gate with the unitary matrix:  $\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ . We can apply this gate on  $|0\rangle$  state:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \quad (3)$$

The above equation explains why quantum gates are unitary matrices, i.e., the sum of amplitudes' square of the output state should be 1.

Another gate in quantum circuits is the multi-qubit gate, also called controlled gate. The input qubits can be divided into the target qubit and the control qubit. Suppose  $U$  is a multi-qubit gate. If the control qubit is set, then  $U$  is applied to the target qubit. Otherwise, the target is left alone. This means:  $|a\rangle|b\rangle \rightarrow |a\rangle U^a|b\rangle$ . The controlled-*NOT* gate is one of the widely-used multi-qubit gates. If the control qubit is  $|1\rangle$ , then the qubit arrow on Bloch sphere will be flipped 180 degree. The controlled-*NOT* gate is illustrated in Figure 1 (b).

**qRAM.** As a fundamental device in the computer, the random access memory (RAM) [36] stores information in its memory cell. The RAM ascribes a unique address to the memory cell. If we want to read out information, we need give the RAM an address and move the information into the register. The quantum computer also has the counterpart equipment, namely, the quantum random access memory (qRAM). A qRAM can be seen as a RAM working in a quantum way. There are many architectures of qRAM, and most of them are based on a promising architecture, called bucket-brigade for qRAM [37, 38, 39]. For simplicity, in this paper we use “qRAM” to denote the “qRAM with the

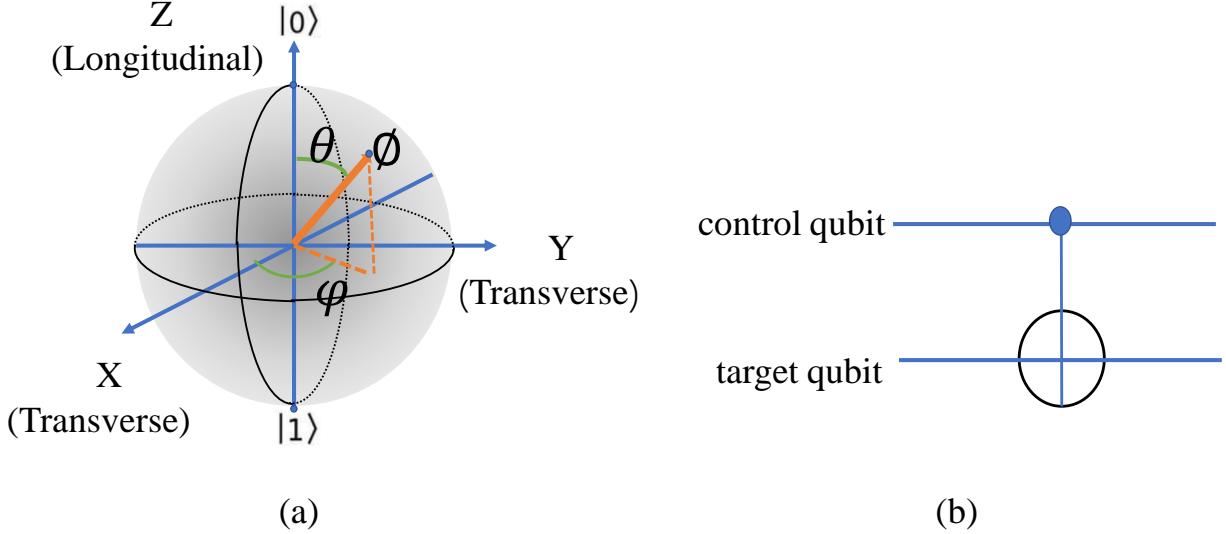


Figure 1: (a) Bloch sphere suggests another representation of a single qubit:  $|\phi\rangle = \cos \frac{\theta}{2}|0\rangle + e^{i\varphi} \sin \frac{\theta}{2}|1\rangle$ , where  $\theta \in [0, \pi]$  , and  $\varphi \in [0, 2\pi]$ . (b) Quantum circuit representation for the controlled not gate.

bucket-brigade architecture”. We input the address of the information in quantum state, and get the data in quantum state as the output [40]:

$$\sum_i \alpha_i |i\rangle_{dir} |0\rangle_{dat} \xrightarrow{\text{qRAM}} \sum_i \alpha_i |i\rangle_{dir} |\psi_i\rangle_{dat} \quad (4)$$

The reason why we introduce qRAM here is that, in this work, it plays an important role in the preparation and observation of the quantum state.

**Preparation and observation of quantum state.** The preparation and the observation of the quantum state are two inevitable problems. They are also important for the design of our quantum Fourier convolutional network, since our hybrid circuit needs to prepare and access the data structure in the quantum network. Thus, in this part, we also briefly introduce how to prepare the input quantum state on the basis of Theorem 3.1 [40], and the pseudo-code about tomography observation in supplementary materials.

**Theorem 3.1** Let  $M \in \mathbb{R}^{n' \times n'}$  be a matrix. Suppose  $w$  is the number of nonzero entries, we can access a quantum data structure with size  $O(w \log^2(n'^2))$ , which takes time  $O(\log(n'^2))$  to store or update an entry. And quantum algorithms with access to this data structure can perform the following maps to precision  $\epsilon^{-1}$  in time  $O(\text{poly log}(n'/\epsilon))$ :

$$\begin{aligned} U_M : |i\rangle|0\rangle &\rightarrow \frac{1}{\|M_{i\cdot}\|} \sum_j M_{ij} |ij\rangle \\ U_N : |0\rangle|j\rangle &\rightarrow \frac{1}{\|M\|_F} \sum_i \|M_{i\cdot}\| |ij\rangle \end{aligned} \quad (5)$$

where  $\|M_{i\cdot}\|$  is the  $l_2$ -norm of the row  $i$  in  $M$ . This means that we can prepare an quantum approximation, i.e,  $1/\|v\|_2 \sum_i v_i |i\rangle$ , which is  $\epsilon$ -close to given classical vector in time  $O(\text{poly log}(n'/\epsilon))$ .

Based on the above theorem and qRAM, we can map the initial quantum state  $|0\rangle^{\otimes N}$  to the quantum state with the amplitudes that we need quickly.

As for observation, suppose the input kernel size is  $n \times n$ , we need to measure the transformed kernel firstly, and then observe the output state with  $|x\rangle = \sum_{i \in [n^2]} x_i |i\rangle$ , where  $\|x\| = 1$ . In our hybrid circuit, we need to transform  $|x\rangle$  to classical vectors for the subsequent calculations and optimization. Here we amend tomography algorithm in work [21], since our result has its independent probability and parameters.

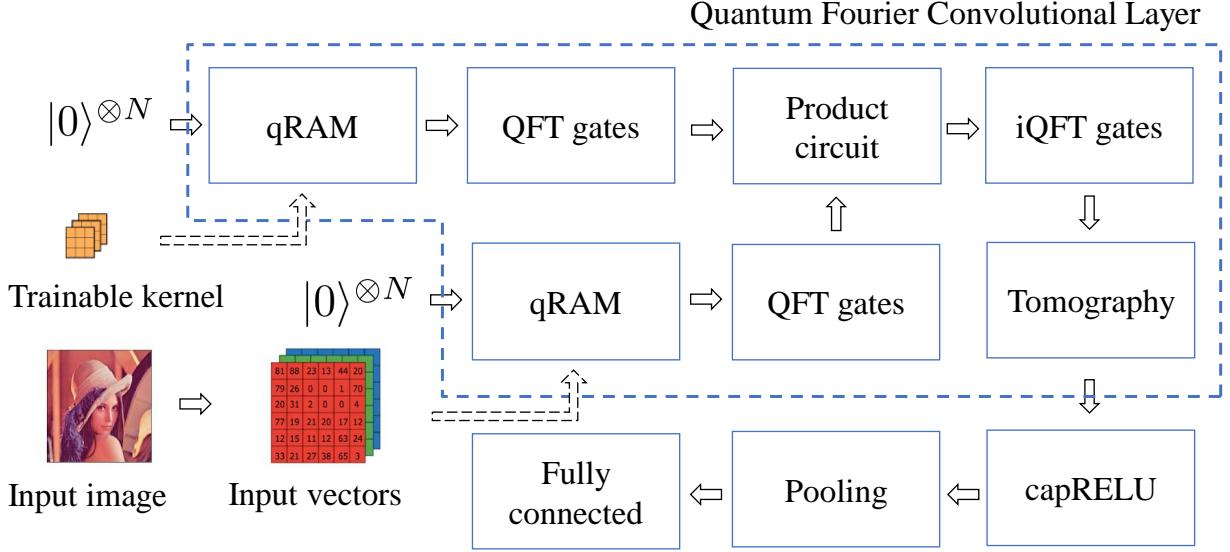


Figure 2: Illustration of our QFCN with a quantum Fourier convolutional layer.

## 4 Quantum Fourier Convolutional Network

To achieve potential quantum speedups of the traditional CNNs, in this section, we propose a quantum Fourier convolutional network (QFCN). The framework of our QFCN is illustrated in Figure 2. We replace the traditional convolutional layers with our Fourier convolutional layers in a quantum way, and utilize the algorithms introduced in section 3 for preparing and observing the quantum input and output. Below we introduce the quantum Fourier convolutional layer and the quantum back-propagation method for our QFCN model in detail.

### 4.1 Quantum Fourier Convolutional Layer

Assume at the layer  $\ell$  of the CNN, the input is the feature maps (or image):  $X^\ell \in \mathbb{R}^{H_\ell \times W_\ell \times C_\ell}$ , where  $H_\ell$ ,  $W_\ell$ , and  $C_\ell$  are the height, width, and channel number, respectively. Let  $K^\ell \in \mathbb{R}^{h_\ell \times w_\ell \times C_\ell \times C_{\ell+1}}$  denote the convolutional kernels. Hence, at layer  $\ell$ , the convolution can be formulated as:

$$X_{i_{\ell+1}, j_{\ell+1}, c_{\ell+1}}^{\ell+1} = \sum_{i=0}^{h_\ell} \sum_{j=0}^{w_\ell} \sum_{c=0}^{C_\ell} X_{i+i_{\ell+1}, j+j_{\ell+1}, c}^\ell K_{i, j, c, c_{\ell+1}}^\ell \quad (6)$$

where the slice  $(i_{\ell+1}, j_{\ell+1}, c_{\ell+1})$  indicates one element of  $X^{\ell+1}$ .

Based on the convolution theorem, the convolutional operation can be accomplished by transforming the kernel and data to Fourier basis, and then performing inverse transform based on their product. Here to achieve quantum speedups for the classical convolutional layer, we need to replace its classical Fourier transform part with quantum Fourier transform and find the process  $P$  to achieve product on combination of quantum state, i.e.,  $X^\ell * K^\ell = iQFT(P(QFT(K^\ell) \otimes QFT(X^\ell)))$ . Unfortunately, there is no such a process  $P$  that is able to complete the convolution (as suggested by [16]), as follows:

$$\sum_{t,j=0}^{N-1} \hat{f}(t) \hat{g}(j) |t\rangle |j\rangle \xrightarrow{P} \lambda \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} \hat{f}(j) \hat{g}(k-j) |k\rangle \quad (7)$$

where  $\lambda$  is a normalization factor.  $\sum_{t,j=0}^{N-1} \hat{f}(t) \hat{g}(j) |t\rangle |j\rangle$  is the combination of the quantum-transformed kernel and quantum-transformed input. Previous works [21] attempt matrix product to circumvent this challenge. Here focusing on Fourier transform, we handle this issue and implement the quantum Fourier convolutional layer as follows:

$$X^{\ell+1} = iQFT \left( M^{QFT(K^\ell)} (QFT(X^\ell)) \right) \quad (8)$$

where  $QFT$  means quantum Fourier transform, and  $iQFT$  means inverse quantum Fourier transform.  $M(\cdot)$  performs the following mapping:

$$\sum_{t,j=0}^{N-1} \hat{f}(t)\hat{g}(j)|t\rangle|j\rangle \xrightarrow{M} \sum_{t,j=0}^{N-1} \hat{f}(t)\hat{g}(t+j)|t\rangle|j\rangle \quad (9)$$

Inspired by [20], after applying inverse quantum Fourier transform on  $\sum_{t,j=0}^{N-1} \hat{f}(t)\hat{g}(t+j)|t\rangle|j\rangle$ , we get the convolution of  $\sum_{k=0}^{N-1} f(k)|k\rangle$  and  $\sum_{k=0}^{N-1} g(k)e^{-2\pi ikj}|k\rangle$  in circuit of input. Since the measurement  $j$  in circuit of kernel can be zero with probability  $\sum_{k=0}^{N-1} |\hat{f}(k)\hat{g}(k)|^2$ , we can get the convolution of  $\sum_{k=0}^{N-1} f(k)|k\rangle$  and  $\sum_{k=0}^{N-1} g(k)|k\rangle$  (quantum states of our kernels and input) when  $j = 0$  is observed [20].

Based on the above mechanism, we can construct the quantum Fourier convolutional layer as illustrated by the dashed box in Figure 2. Below we also present more details of the quantum Fourier convolutional layer, including revisiting the discrete and fast Fourier transform, and introducing the quantum Fourier transform (QFT).

#### 4.1.1 Discrete and Fast Fourier Transform

The Discrete Fourier Transform (DFT) is widely used in digital signal processing and image processing [41]. It can map a vector  $x = (x_0, x_1, \dots, x_{N-1})^T$  to a new vector  $y = (y_0, y_1, \dots, y_{N-1})^T$ , as follows:

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i \frac{kj}{N}} \quad (10)$$

The Fast Fourier transform (FFT) [42] is for fast computation of the DFT, which makes use of the symmetries in the DFT and can reduce the computational cost from  $\mathcal{O}[N^2]$  to  $\mathcal{O}[N \log N]$ .

#### 4.1.2 Quantum Fourier Transform

The quantum Fourier transform (QFT) has the same input and output as the DFT, but QFT is for the quantum state and can achieve exponential speedup. It accepts a quantum state  $|\psi\rangle = \sum_{j=0}^{N-1} x_j |j\rangle$  as input, and performs the following mapping on its quantum circuit:

$$\sum_{j=0}^{N-1} x_j |j\rangle \xrightarrow{QFT} \sum_{k=0}^{N-1} y_k |k\rangle \quad (11)$$

where the amplitude  $y_k$  is the same as Eq. (10). By substituting Eq. (10) into Eq. (11) and offsetting the amplitudes  $x_j$ , it can be written as:

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i \frac{kj}{N}} |k\rangle \quad (12)$$

It is obvious that the data we need is on the amplitudes of the quantum state. Then we describe what happens on the input  $|\psi\rangle$ , and analyse why QFT can be faster than FFT by the circuit for the QFT illustrated in Figure 3. The quantum controlled gate  $R_k$  denotes the unitary operator:

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i / 2^k} \end{bmatrix} \quad (13)$$

After time  $t_1$ , the Hadamard gate is applied on the first qubit of the state, which produces:

$$\frac{1}{2^{1/2}} (|0\rangle + e^{2\pi i 0 \cdot j_1} |1\rangle) |j_2 \dots j_n\rangle \quad (14)$$

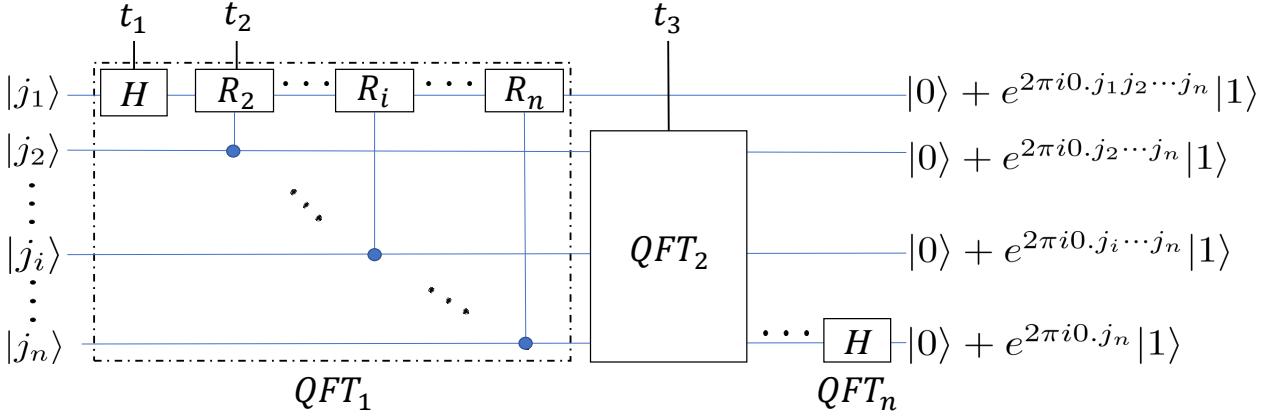


Figure 3: Circuit for QFT gates without swapping gate

According to the Euler formula  $e^{\pi i} = -1$ , we get:

$$e^{2\pi i 0.j_1} = \begin{cases} -1, & j_1 = 1 \\ 1, & \text{otherwise} \end{cases} \quad (15)$$

Applying the *controlled – R<sub>2</sub>* gate on the first qubit, we have the state in time  $t_2$ :

$$\frac{1}{2^{1/2}} (|0\rangle + e^{2\pi i 0.j_1 j_2} |1\rangle) |j_2 \dots j_n\rangle \quad (16)$$

We repeatedly apply *controlled – R<sub>i</sub>* gate on the first qubit. After *controlled – R<sub>n</sub>* gate, we have the state:

$$\frac{1}{2^{1/2}} (|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n} |1\rangle) |j_2 \dots j_n\rangle \quad (17)$$

The second qubit is the same as the first one, and we get the state after  $t_3$ :

$$\frac{1}{2^{2/2}} (|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n} |1\rangle) (|0\rangle + e^{2\pi i 0.j_2 \dots j_n} |1\rangle) |j_3 \dots j_n\rangle \quad (18)$$

Here we can also see the robustness of the QFT circuit. We do not need to prepare a new circuit to adapt the length of the quantum state. The whole circuit can be regarded as the assemble of several separate QFT gates, while the single QFT gate is made by several controlled–R<sub>k</sub> gates and a Hadamard gate to ensure that the QFT gate is unitary.

After applying  $n$  QFT gates on  $n$  qubits, we swap the result and get the state:

$$\begin{aligned} & \frac{1}{2^{n/2}} (|0\rangle + e^{2\pi i 0.j_n} |1\rangle) \\ & (|0\rangle + e^{2\pi i 0.j_{n-1} j_n} |1\rangle) \dots (|0\rangle + e^{2\pi i 0.j_1 j_2 \dots j_n} |1\rangle) \end{aligned} \quad (19)$$

This quantum state is equivalent to the quantum state in Eq. (12), after being simplified by Euler formula.

As for the inverse QFT (iQFT) circuit, it can be achieved just like what it spells: reversing the circuit for the QFT [43].

One of the motivations we exploit QFT in this paper is that the QFT algorithm can achieve exponential speed-up. As we can see, a total of  $n + (n - 1) + \dots + 1 = n(n + 1)/2$  gates are required to perform the QFT on  $n$  qubits. These  $n$  qubits need  $2^n$  bits to be encoded in classical computer and we need  $n2^n$  gates to perform FFT on these bits. Based on the number of gates, the QFT algorithm reduces time complexity from  $\mathcal{O}(n2^n)$  to  $\mathcal{O}(n^2)$ . The best approximation is able to reduce this complexity to  $\mathcal{O}(n \log n)$  [44].

## 4.2 Quantum Back-Propagation

The optimization of the model has always been an essential part of quantum machine learning [45, 46]. Previous works about quantum convolution neural network utilize quantum matrix product process and QEC (a mechanism to detect and correct local quantum errors) respectively to achieve optimization of their models [22, 21]. Differently, in this work, we propose our optimization method using the advantages of the hybrid quantum-classical algorithm mentioned before.

Specifically, this QFCN model can still be trained with back-propagation like the classical CNNs. The quantum Fourier convolutional layer in the hybrid quantum-classical circuit can be seen as a quantum circuit with parametric gates (PQC). This means the gradients of our quantum circuit can be estimated by shifting the parameters and running the same QFT circuit [26]. Once we get the gradients, they can be incorporated into the classical back-propagation process using gradient-based algorithms [47]. Denote  $U(\vec{\theta}) = U_n(\theta_n) \dots U_1(\theta_1)$  as our PQC mathematical expression, and  $H$  as the Hermitian operator representing our goal. Then the expectation quantum value should be  $L(\vec{\theta}) = \langle \psi_0 | U^\dagger(\vec{\theta}) H U(\vec{\theta}) | \psi_0 \rangle$ .

Through a series of calculation and offsetting [26], the gradient of  $j^{th}$  parameter is below:

$$\frac{\partial L}{\partial \theta_j} = \frac{1}{2} [L_{j,+}(\vec{\theta}) - L_{j,-}(\vec{\theta})] \quad (20)$$

where

$$\begin{aligned} L_{j,+}(\vec{\theta}) &= L(\theta_1, \dots, \theta_{j-1}, \theta_j + \frac{\pi}{2}, \theta_{j+1}, \dots, \theta_n) \\ L_{j,-}(\vec{\theta}) &= L(\theta_1, \dots, \theta_{j-1}, \theta_j - \frac{\pi}{2}, \theta_{j+1}, \dots, \theta_n) \end{aligned} \quad (21)$$

which demonstrates that the gradient can be accessed through shifting parameter in qRAM and run the same circuit.

Note that in some scenarios, we may need to stack multiply layers in the model, which may lead gradient to Barren plateaus [23] and vanish, while the robustness of the QFT circuit that we mentioned above can help to achieve the layer-wise algorithm [48] and mitigate Barren plateaus.

## 5 Experiments

In our experiments, we evaluate our proposed QFCN on two tasks. First, we apply QFCN on image classification by following QCNN [21]. Then, we test QFCN on the traffic prediction task by inserting the quantum Fourier convolutional layer into the spatio-temporal graph convolutional network.

### 5.1 Image Classification

In this experiment, our QFCN is designed to accept the 2D image data structure. Specifically, we construct our QFCN using the same baseline CNN as [21].

We apply QFCN on the MNIST dataset including 60,000 training images and 10,000 testing images. Each image presents a hand-written number consisting of  $28 \times 28$  grayscale pixels. Following the work of QCNN [21], we use the PyTorch library to simulate the quantum environment by several parameters of noises. Similar to [21], the cap of the non-linearity function  $C$  is set to 10, and the quantum noise  $\epsilon$  during preparation and observation is set to 0.01. Readers are referred to [21] for more details about the quantum simulation process. In this work, we use the same image samples as [21] to plot the curves. As the input and output values of QFT and FFT will be the same, we perform simulation experiments by applying the aforementioned noise parameters to the process of imitating QFT with FFT, since such simulation settings are also adopted by previous works [21] owing to the inaccessibility of the real quantum computers. Considering the GPU speedups of CNN, we do not compare the computation time of QFCN with the traditional CNN. But we have also conducted a toy experiment to directly compare the FFT with the classical convolution. In our simple test using an i5-6300HQ CPU with 2.30GHz, where the kernel size is  $2 \times 3 \times 5 \times 5$  and the signal size is  $3 \times 1024 \times 1024$ , the FFT-based convolution used  $744ms$  for calculation, which is faster than the classical convolution using  $1.05s$ . This situation is not really uncommon in some recent works [49, 50] because of their large kernel size, and we should also note that theoretically, the QFT algorithm can be exponentially faster than the FFT algorithm.

**Result.** We plot the convergence curves of the classical CNN, QCNN [21], and QFCN in Figure 4. We do not show many more training curves for QCNN [21], because QCNN [21] does not perform well under other parameters, as shown by the paper of [21]. We can also observe that our QFCN achieves a faster and easier convergence on the MINST dataset, compared to the QCNN.

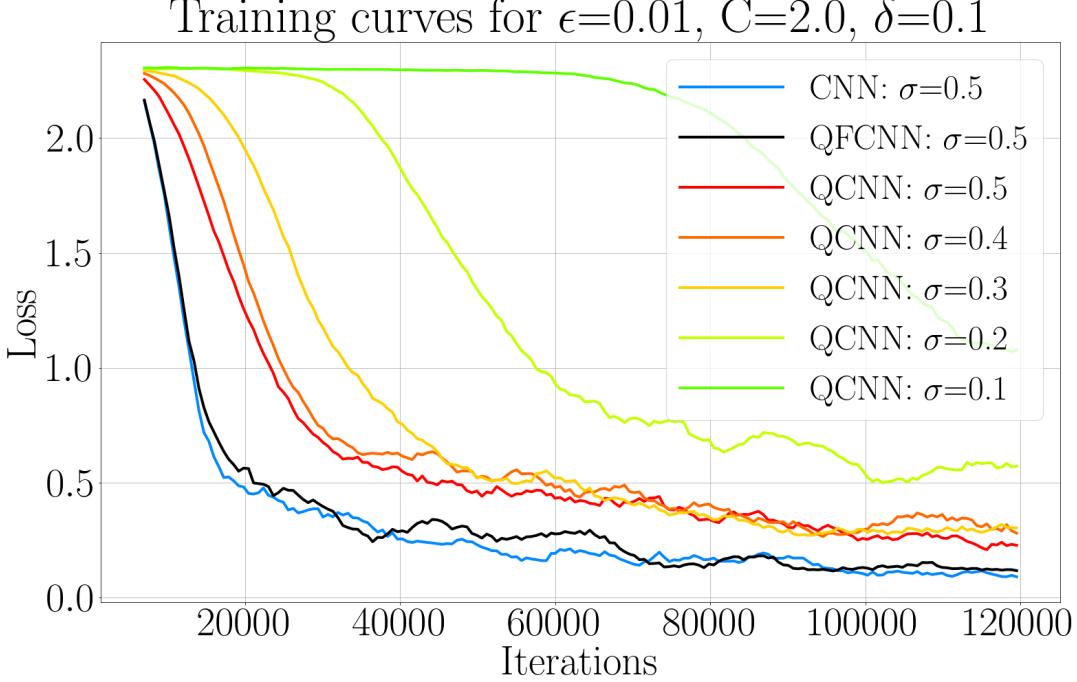


Figure 4: Numerical simulations of the training of the QCNN [21] and our QFCN.

Table 1: Loss and accuracy of QFCN and QCNN.

MNIST	Loss	Accuracy
QCNN ( $\sigma = 0.1$ )	1.07	86.1
QCNN ( $\sigma = 0.2$ )	0.552	92.8
QCNN ( $\sigma = 0.3$ )	0.391	94.3
QCNN ( $\sigma = 0.4$ )	0.327	94.4
QCNN ( $\sigma = 0.5$ )	0.163	95.9
QFCN	<b>0.113</b>	<b>96.3%</b>

In Table 1, we show the performance of our model, and compare it with the QCNN model. We can observe that our model achieves higher accuracy than QCNN. The performance disparity can be partially due to that our hybrid quantum-classical circuit avoids quantum sampling and quantum pooling, which are simulated by adding quantum noises in QCNN.

## 5.2 Traffic Prediction

Many existing works [51, 52] use the models constructed with both temporal convolution and spatial graph convolution. This kind of networks is generally called Spatio-Temporal Graph Convolutional Network (STGCN) [53, 54, 55]. We apply our quantum Fourier convolution for the STGCN [56] and use the designed quantum Fourier convolutional layer to handle the convolution for the time series information. In this experiment, we mainly aim to demonstrate the ability of our quantum Fourier convolutional layer for replacing the temporal convolution over large-scale series input in the baseline STGCN [56]. We name this model as quantum Fourier Spatio-Temporal Graph Convolutional Network (qf-STGCN).

The qf-STGCN uses the quantum mechanism to predict time series on the basis of previous time series and graph-structured data. We adopt most of the structure parts of the previous STGCN [56]. Specifically, the spatio-temporal convolutional blocks in STGCN contain two gated-convolutional blocks for handling temporal properties, and one graph convolutional block to handle spatial properties of the graph. We utilize our quantum Fourier convolutional layer to replace the gated convolutional layer, and add simulation of tomography on the quantum machine. The framework of qf-STGCN is illustrated in Figure 5.

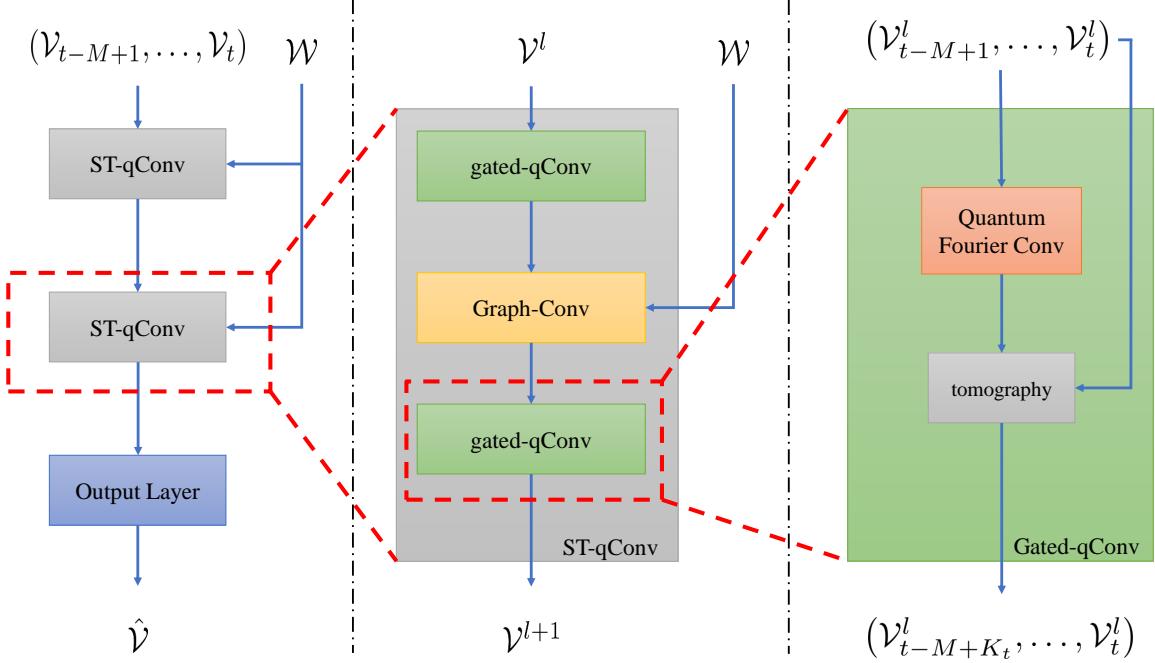


Figure 5: The architecture of the qf-STGCN. The input  $\mathcal{W}$  denotes the weighted adjacency matrix of the graph. One single  $\mathcal{V}_t$  represents the data captured at time  $t$ .  $\hat{\mathcal{V}}$  denotes the final prediction, and  $\mathcal{V}^{l+1}$  denotes the input data of layer  $l + 1$ .  $K_t$  is the width of the kernel.  $M$  is the length of the input series. The qf-STGCN is adapted from the classical STGCN in [56]. Refer to [56] for more details of the original STGCN.

We assess the performance of the model with two metrics, namely, Mean Absolute Error (MAE) and Mean Squared Error (MSE), following the previous works on traffic prediction [54, 57], where  $\text{MAE} = \frac{1}{n} \sum_{t=1}^n |\mathcal{V}_t - \hat{\mathcal{V}}_t|$ , and  $\text{MSE} = \frac{1}{n} \sum_{t=1}^n (\mathcal{V}_t - \hat{\mathcal{V}}_t)^2$ .  $\mathcal{V}_t$  and  $\hat{\mathcal{V}}_t$  are the ground truth and prediction of traffic condition in time  $t$  respectively.

We adopt the METR-LA dataset [54] that is widely used in traffic prediction works. The METR-LA is a dataset about information collected by loop sensors in the highway of Los Angeles, which contains 207 nodes, 1515 edges and 34272 time steps. The distribution of the sensors can be found in Li [54].

In this experiment, we add a baseline which puts our qf-STGCN on the pure quantum circuit with numerical simulations of quantum sampling, quantum pooling, and quantum back-propagation, as done by [21].

**Result.** We can observe that in Figure 6, both of our qf-STGCN models can still converge fast and correctly. However, the data structure of the spatial-temporal graph in this task does not have many parts of redundant information like images. Besides, some previous works have also found that GCN is vulnerable to attacks and noise [58, 59], and then when we change the attribution on the node, the performance of the following graph convolution will also be influenced. This implies that there could be a need of more robust quantum graph convolution [60]. Nevertheless, despite of the influences of quantum noise on the graph data, our model can still be used for quantum speedups. This experiment also shows the ability of our model for handling the large-scale time series dataset.

## 6 Conclusion

Quantum computation and CNN have become two popular research fields recently. In this work, we have presented a new quantum Fourier convolutional network. Our QFCN could be able to run on both hybrid quantum-classical circuits and pure quantum computers. Our experiments show our model's ability of embedding within the classical CNNs and handling large datasets. This means our model can also be inserted into many other neural networks with convolutional layers to achieve quantum speedups. There are also some shortcomings: like many other quantum machine learning work, we still lack of a real programmable quantum computer to prove our research and we believe that day is coming soon.

## References

- [1] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.
- [2] Vedran Dunjko and Hans J Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018.
- [3] Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: A quantum algorithm for unsupervised machine learning. In *Advances in Neural Information Processing Systems*, pages 4134–4144, 2019.
- [4] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411*, 2013.
- [5] Iordanis Kerenidis and Anupam Prakash. Quantum gradient descent for linear systems and least squares. *Physical Review A*, 101(2):022316, 2020.
- [6] Nathan Wiebe, Ashish Kapoor, and Krysta Svore. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *arXiv preprint arXiv:1401.2142*, 2014.
- [7] Swamit S Tannu and Moinuddin K Qureshi. A case for variability-aware policies for nisq-era quantum computers. *arXiv preprint arXiv:1805.10224*, 2018.
- [8] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
- [9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4401–4410, 2019.
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [12] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [14] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
- [15] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018.
- [16] Chris Lomont. Quantum convolution and quantum correlation algorithms are physically impossible. *arXiv preprint quant-ph/0309070*, 2003.
- [17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.
- [18] Harry Pratt, Bryan Williams, Frans Coenen, and Yalin Zheng. Fcnn: Fourier convolutional neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 786–798. Springer, 2017.
- [19] Oren Rippel, Jasper Snoek, and Ryan P Adams. Spectral representations for convolutional neural networks. In *Advances in neural information processing systems*, pages 2449–2457, 2015.
- [20] Guangsheng Ma, Hongbo Li, and Jiman Zhao. Windowed fourier transform and general wavelet algorithms in quantum computation. *Quantum Inf. Comput.*, 19(3&4):237–251, 2019.
- [21] Iordanis Kerenidis, Jonas Landman, and Anupam Prakash. Quantum algorithms for deep convolutional neural networks. In *International Conference on Learning Representations*, 2019.
- [22] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.
- [23] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):1–6, 2018.
- [24] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.
- [25] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.
- [26] Chen Zhao and Xiao-Shan Gao. Qdnn: Dnn with quantum neural network layers. *arXiv preprint arXiv:1912.12660*, 2019.
- [27] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

- [28] Hongxiang Chen, Leonard Wossnig, Simone Severini, Hartmut Neven, and Masoud Mohseni. Universal discriminative quantum neural networks. *arXiv preprint arXiv:1805.08654*, 2018.
- [29] Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. *Physical review letters*, 121(4):040502, 2018.
- [30] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. *arXiv preprint arXiv:1603.08675*, 2016.
- [31] Pierre-Luc Dallaire-Demers and Nathan Killoran. Quantum generative adversarial networks. *Physical Review A*, 98(1):012324, 2018.
- [32] Alexey A Melnikov, Leonid E Fedichkin, and Alexander Alodjants. Predicting quantum advantage by quantum walk with convolutional neural networks. *New Journal of Physics*, 21(12):125002, 2019.
- [33] Mohamed Hibat-Allah, Martin Ganahl, Lauren E Hayward, Roger G Melko, and Juan Carrasquilla. Recurrent neural network wave functions. *Physical Review Research*, 2(2):023358, 2020.
- [34] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, 2016.
- [35] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O’brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.
- [36] MA Reed, J Chen, AM Rawlett, DW Price, and JM Tour. Molecular random access memory cell. *Applied physics letters*, 78(23):3735–3737, 2001.
- [37] Daniel K Park, Francesco Petruccione, and June-Koo Kevin Rhee. Circuit-based quantum random access memory for classical data. *Scientific reports*, 9(1):1–8, 2019.
- [38] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.
- [39] Srinivasan Arunachalam, Vlad Gheorghiu, Tomas Jochym-O’Connor, Michele Mosca, and Priyaa Varshinee Srinivasan. On the robustness of bucket brigade quantum ram. *New Journal of Physics*, 17(12):123010, 2015.
- [40] Pablo Antonio Moreno Casares. Circuit implementation of bucket brigade qram for quantum state preparation. *arXiv e-prints*, pages arXiv–2006, 2020.
- [41] Shmuel Winograd. On computing the discrete fourier transform. *Mathematics of computation*, 32(141):175–199, 1978.
- [42] Henri J Nussbaumer. The fast fourier transform. In *Fast Fourier Transform and Convolution Algorithms*, pages 80–111. Springer, 1981.
- [43] Michael A Nielsen and Isaac Chuang. Quantum computation and quantum information, 2002.
- [44] Lisa Hales and Sean Hallgren. An improved quantum fourier transform algorithm and applications. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 515–525. IEEE, 2000.
- [45] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.
- [46] Stuart Hadfield, Zhihui Wang, Bryan O’Gorman, Eleanor G Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, 2019.
- [47] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.
- [48] Andrea Skolik, Jarrod R McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. Layerwise learning for quantum neural networks. *arXiv preprint arXiv:2006.14904*, 2020.
- [49] Kamran Chitsaz, Mohsen Hajabdollahi, Nader Karimi, Shadrokh Samavi, and Shahram Shirani. Acceleration of convolutional neural network using fft-based split convolutions. *arXiv preprint arXiv:2003.12621*, 2020.
- [50] Shaohua Li, Kaiping Xue, Bin Zhu, Chenkai Ding, Xindi Gao, David Wei, and Tao Wan. Falcon: A fourier transform based approach for fast and secure convolutional neural network predictions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8705–8714, 2020.
- [51] Pengpeng Zhao, Anjing Luo, Yanchi Liu, Fuzhen Zhuang, Jiajie Xu, Zhixu Li, Victor S Sheng, and Xiaofang Zhou. Where to go next: A spatio-temporal gated network for next poi recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2020.
- [52] Pallabi Ghosh, Yi Yao, Larry Davis, and Ajay Divakaran. Stacked spatio-temporal graph convolutional networks for action segmentation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 576–585, 2020.
- [53] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [54] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.
- [55] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 922–929, 2019.

- [56] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017.
- [57] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing*, pages 362–373. Springer, 2018.
- [58] Hanjun Dai, Hui Li, Tian Tian, Xin Huang, Lin Wang, Jun Zhu, and Le Song. Adversarial attack on graph structured data. *arXiv preprint arXiv:1806.02371*, 2018.
- [59] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2847–2856, 2018.
- [60] Guillaume Verdon, Trevor McCourt, Enxhell Luzhnica, Vikash Singh, Stefan Leichenauer, and Jack Hidary. Quantum graph neural networks. *arXiv preprint arXiv:1909.12264*, 2019.

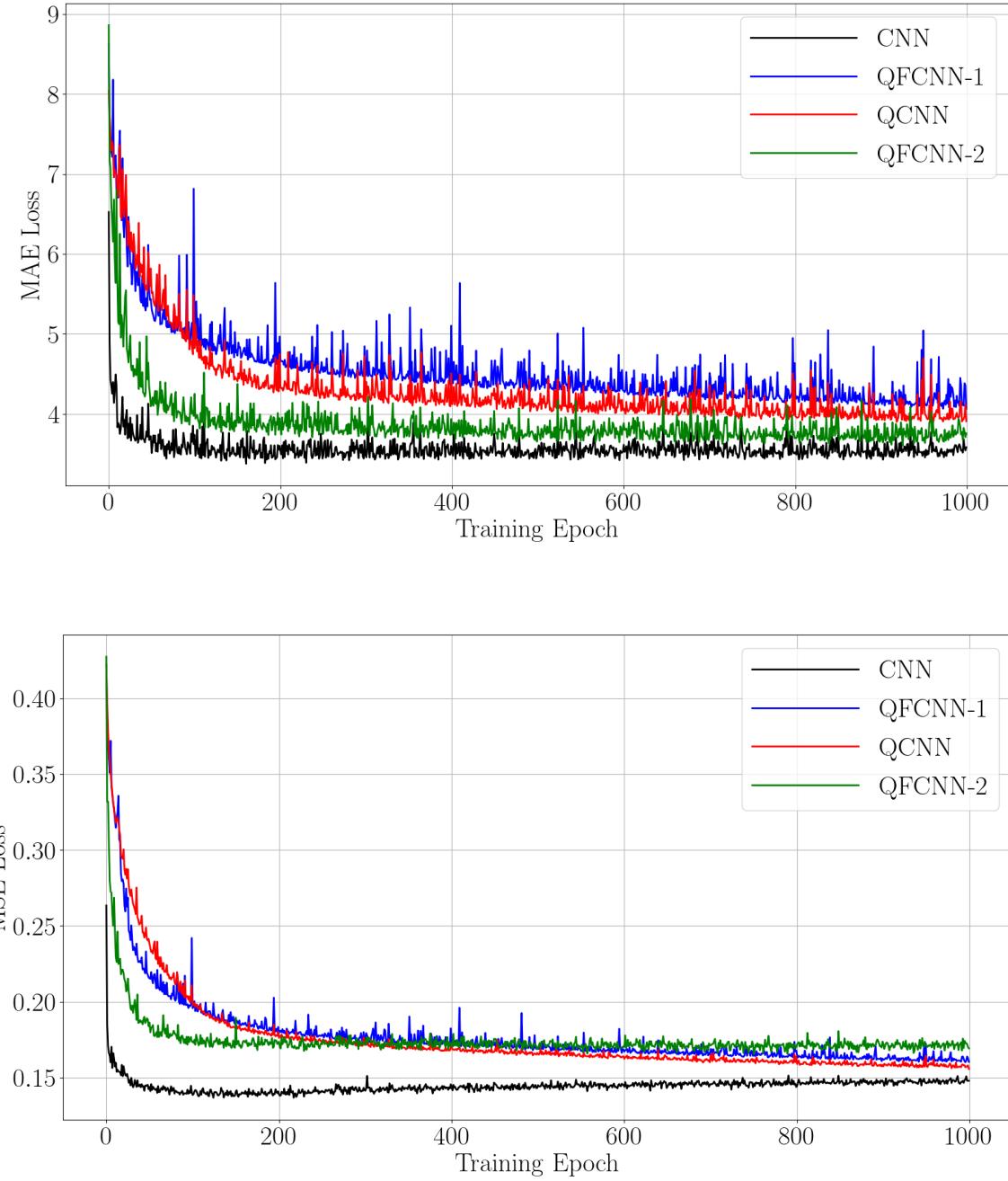


Figure 6: The MAE and MSE curves of STGCN under different settings. The blue curve represents the qf-STGCN on the hybrid circuit, named q-STGCN-1. The red curve indicates the qf-STGCN in the pure quantum environment, named q-STGCN-2. The black curve represents the performance of the origin STGCN on the same dataset.