Enable Quantum Graph Neural Networks on a Single Qubit with Quantum Walk

Yijie Zhu, Richard Jiang, Qiang Ni, and Ahmed Bouridane

***Abstract*—Quantum computing holds significant potential for advancing machine learning, particularly in handling complex graph-structured data. This paper introduces Single-Qubit Quantum Graph Neural Networks (sQGNNs), a novel model that integrates quantum networks with quantum walk operations to improve generalization in graph learning tasks. By leveraging quantum walks, we demonstrated sQGNNs capture complex relational patterns and enhance network expressiveness beyond classical methods. Our results proved that quantum encoding efficiently represents high-dimensional graph data, preserving dependencies and optimizing memory use. Across benchmark datasets, sQGNNs demonstrate superior generalization and robustness against overfitting, achieving higher accuracy with reduced computational cost. Our results underscore sQGNNs' promise for scalable, quantum-enhanced graph learning, establishing a foundation for future quantum-assisted machine learning applications.**

***Impact Statement*—This research presents a groundbreaking Single-Qubit Quantum Graph Neural Network (sQGNN) that addresses critical NISQ-era limitations through an integrated architecture combining quantum walk embedding with SU(2)-optimized single-qubit unitary operations. Our novel model leverages SU(2) group theory to enhance parameter efficiency in single-qubit encoding while introducing a quantum walk-based compression that reduces parameters compared to classical GNNs. Rigorous hardware validation on IBM Manila demonstrates robust performance under noise, establishing the first experimental proof of accurate graph classification using just one qubit. This unified framework—spanning theoretical innovation, algorithmic design, and NISQ implementation—advances practical quantum machine learning toward real-world applicability.**

***Index Terms*—Quantum Machine Learning, Quantum Walk, Quantum Neural Networks, Quantum Graphs, Qubits.**

I. INTRODUCTION

As information technology advances and data density grows exponentially, innovative approaches to machine learning complexity become imperative [1]. Quantum algorithms show potential for exponential speedups in high-dimensional data processing compared to classical solutions [1]. Combining quantum computing with machine learning has led to quantum machine learning (QML), which could enhance classical algorithms [2, 3]. However, QML faces implementation challenges in today's Noisy Intermediate-Scale Quantum (NISQ) era of quantum hardware [4, 5, 6].

While quantum computers have progressed significantly [1], major NISQ-era limitations persist [4]. These include limited qubit counts, architectural constraints, and high noise levels [7]. These factors, along with the distant prospect of fault-tolerant quantum computers [4], hinder achieving quantum supremacy (real-world computational advantages) [8]. Thus, the key challenge is leveraging current NISQ devices despite limited qubits, connectivity, and error-prone circuits. Variational Quantum Algorithms (VQAs) [9] are popular in QML but face limitations like trainability issues and the "barren plateau" problem (where optimization becomes extremely difficult as systems scale up) [10].

A notable example is the quantum convolutional neural network [11], which adapts classical convolutional concepts to QML. However, current NISQ devices struggle with its high qubit and quantum volume requirements. For instance, processing even the modest MNIST image dataset [12] would require 784 qubits—far beyond current and near-term capabilities.

Current quantum computers remain costly and scarce [13], with limited public access. Given slow resource growth, efficient quantum resource utilization is crucial. Consequently, most QML research uses small or specialized datasets, leaving complex neural networks and real-world benchmark tasks largely unexplored in the quantum domain.

Recent work proposed implementing neural networks on single qubits using data-reuploading techniques [14], later extended to convolutional architectures [15]. Studies confirm the strong expressive power of such single-qubit approaches [16, 17]. Building on this, we apply single-qubit methods to graph neural networks, avoiding complex multi-qubit graph structures. On this basis, we use SU (2) group theory to optimize the single-qubit method and apply it to graph neural networks, avoiding complex multi-qubit graph structures.

This paper introduces a novel Single-Qubit Quantum Graph Neural Network (sQGNN) for graph data analysis on resource-constrained NISQ (Noisy Intermediate-Scale Quantum) Variational Quantum Circuits (VQCs) [14, 15]. Our approach aims to improve generalization on complex data while reducing resource requirements. We introduce a discrete-time quantum walk method for efficient graph embedding and use our innovative single-qubit approach based on SU(2) group theory for efficient encoding and feature learning.

The single-qubit design allows sQGNN to generalize effectively across large-scale graph data, making it ideal for resource-constrained quantum systems. Simulation results demonstrate robust performance on graph data, and successful implementation on real quantum hardware underscores its

Manuscript received September 28, 2024; revised July 21, 2025. This work was supported in part by the UK EPSRC under Grant EP/P009727/2, and the Leverhulme Trust under Grant RF-2019-492. (Correspondent author: Richard Jiang, e-mail: r.jiang2@lancaster.ac.uk).

Yijie Zhu, Richard Jiang and Qiang Ni are with LIRA Center, Lancaster University, Lancaster, Lancashire, LA1 4WA, UK.

Ahmed Bouridane is with the Centre for Data Analytics and Cybersecurity (CDAC), University of Sharjah, Sharjah, United Arab Emirates.

practicality. The model performs well even in the presence of NISQ-era noise, showcasing resilience—essential for near-term quantum applications. In addressing critical resource limitations, sQGNN offers a scalable and effective solution, advancing quantum model generalization within current quantum technology constraints.

This paper is organized as follows: Section 1 is the introduction, Section 2 is the related work, Section 3 is our method, Section 4 is the experimental part, Section 5 is the discussion, and Section 6 is the conclusion.

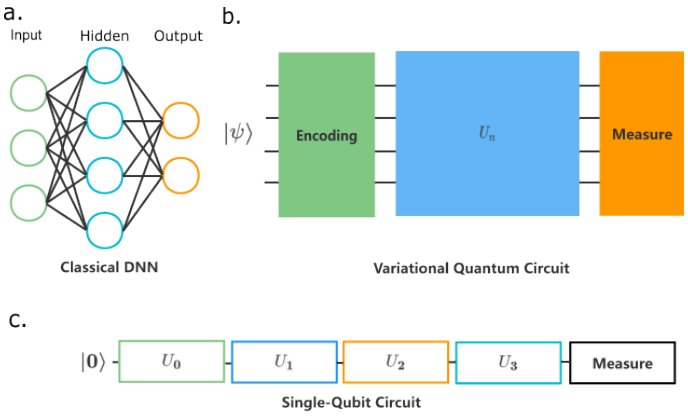


Fig. 1. Schematic representation of the mentioned models.

II. RELATED WORK

Graph-structured data has become a highly active research topic in the field of machine learning. However, classical neural networks are designed to process data with regular structures in Euclidean space, and their efficiency is consequently reduced when dealing with complex graph data. Quantum computing has been suggested as a potential solution to address this computational complexity, as data in quantum machine learning is often represented in a high-dimensional Hilbert space by quantum states, which has been demonstrated to be beneficial for classification tasks [18].

Existing quantum GNN approaches face significant architectural constraints. The pioneering quantum graph neural network [19] and its derivatives (quantum graph RNNs and CNNs) require prior knowledge of graph structure for circuit design and impose strict size limitations (typically $N < 20$ nodes). For instance, processing standard datasets like MUTAG (average 18 nodes) would demand ∼18 qubits and $O(N^2)$ circuit parameters. Similarly, the quantum graph convolutional network [20], while inspired by classical CNNs, maintains these graph size constraints and requires ∼15 qubits for molecular graphs. Hybrid approaches [21] that append quantum circuits to classical networks fail to achieve quantum advantage, as they incur classical bottlenecks while still requiring $O(N)$ qubits.

Our sQGNN overcomes these constraints through three key innovations: First, quantum walk embedding compresses arbitrary graph structures into fixed-dimensional vectors, eliminating topology-specific circuit design. Second, single-qubit processing reduces qubit requirements from $O(N)$ to $O(1)$. Third, parameter-efficient rotation gates lower parameter counts (e.g., 12 parameters vs. ∼10k in classical GNNs for MUTAG). This architectural simplification enables processing of larger graphs on current NISQ devices while maintaining competitive accuracy, as demonstrated in Section IV. Figure 1 illustrates these architectural differences.

III. SINGLE-QUBIT GRAPH NEURAL NETWORK

Figure 2 illustrates the workflow of our proposed sQGNN method, which consists of three steps. First, the quantum walk graph embedding method is applied to convert raw graph data into a vector form that can be processed. Second, the processed graph representation vector is input into the single-qubit circuit in a predefined order of unitary operations, as depicted in Figure 2. This single-qubit method can encode all data with a single qubit, thus reducing the total number of qubits necessary for the NISQ era and effectively avoiding the ”barren plateau” phenomenon [10] caused by excessive qubits in VQA. Finally, we measure the single-qubit circuit, calculate the fidelity and the loss of the model, and update the parameters of the quantum circuits to perform the training of the neural network. The training process provides us with the desired model. Further details on these three steps can be found in the Methods section. Subsequently, experiments were conducted to evaluate our sQGNN method on several real-world datasets from the chemistry and biology domains. For the specific process, please see Algorithm 1.

In summary, our method has the following advantages: **Save quantum resources**. Our approach requires only one qubit, which is significant in the NISQ era of quantum computing resources. **Anti-noise**. Our model exhibits good noise immunity on simulated depolarization errors as well as real quantum computers. The good robustness of the quantum method itself can save the extra quantum resources required by the Quantum Error Correction (QEC) algorithm. **Avoid the ”barren plateau”**. Our method only needs one qubit, effectively avoiding the ”barren plateau”. **The amount of parameters is small**. Our quantum approach has significantly fewer model parameters than the classical approach. **Extensibility**. Our quantum model can be combined with other quantum or classical structures to form a hybrid neural network. **Easy to deploy**. Our model requires only one qubit and can be easily deployed on real quantum computers.

A. Quantum Walk based Graph Embedding

In order to embed a graph onto a single qubit, classical graph data needs to be processed. Our method is inspired by [22]. When given a group of graphs, the method employs discrete-time quantum walks [23] to analyze them. Compared to classical walks, quantum walks possess remarkable features due to their dependence on unitary matrices instead of random ones. As they evolve, quantum walks create patterns of destructive interference, leading to improved graph representations [23]. Additionally, quantum walks operate in the state space of directed edges rather than vertices, allowing for enhanced detection of the graph's structure. A coordinate wave function can be used to describe the particle position:

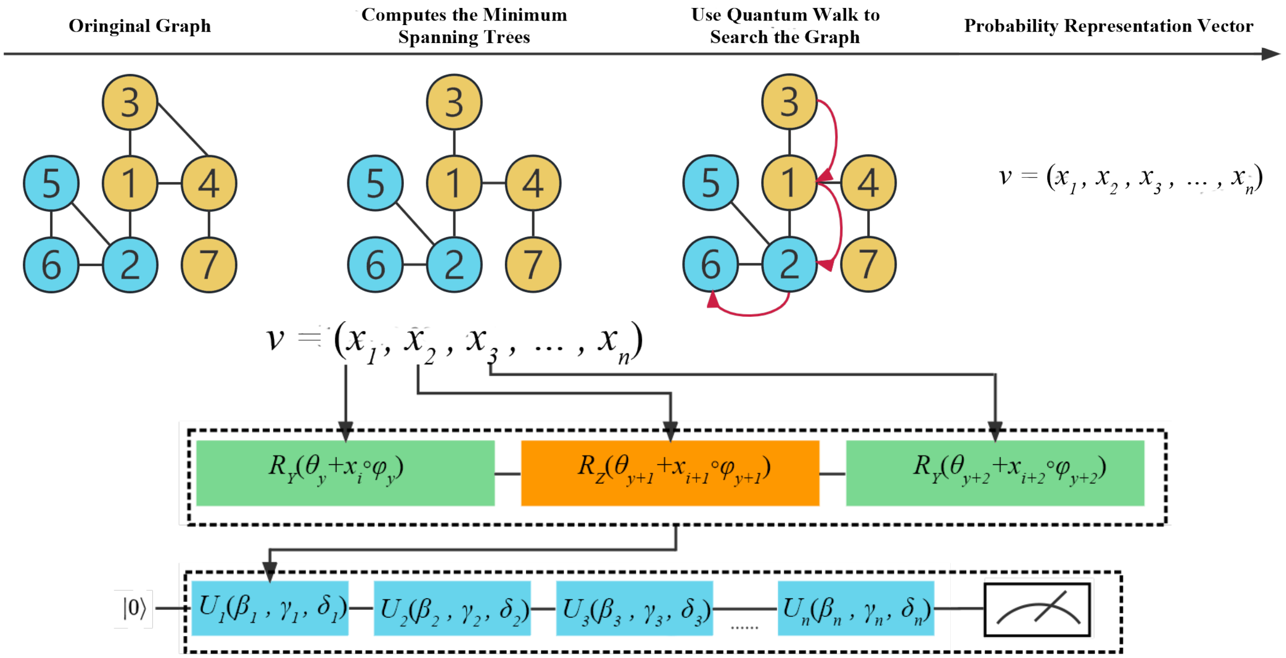


Fig. 2. The diagram of the quantum embedding and the Single-Qubit quantum circuit. The elements in the representation vector of the graph are in units of 3 and are encoded onto the qubits using a quantum rotation gate. Every three quantum rotation gates constitute a unitary operation, and the Single-Qubit method uses this unitary operation as the basic unit of encoding and parameter training.

$$|\psi_p\rangle = \sum_x c_x |x\rangle \tag{1}$$

Where $|x\rangle$ is the wave function of particle in position x and c_x is the complex amplitude. So the probability of finding a particle at x is:

$$p(x) = |c_x|^2 \tag{2}$$

The particle is in a superposition state of various positions during the process of walking. Considering the random walk of particles, we need to introduce the flipping operator C and moving operator S:

$$C = H \otimes I_x \tag{3}$$

$$S = |0\rangle\langle 0| \otimes |x+1\rangle\langle x| + |1\rangle\langle 1| \otimes \sum_x |x-1\rangle\langle x| \tag{4}$$

Where H is the Hadamard gate and I_x is the identity matrix of wave function of x. Apply these two operators to the initial state $|\psi_0\rangle$ where the position is 0 and the state is up spin, the next state is:

$$|\psi_1\rangle = \frac{1+i}{2}|0\rangle|1_x\rangle + \frac{1+i}{2}|0\rangle|-1_x\rangle \tag{5}$$

The state $|\psi_1\rangle$ is in superposition of position $x = -1$ and $x = 1$. This process can obtain a different probability distribution than the classical process.

To begin, we apply the commute time spanning tree (CTST) representation of the input graphs. Next, we execute simulations involving the evolution of a discrete-time quantum walk across the CTSTs. With access to the CTST representations of the graphs, we proceed to simulate the evolution of a discrete-time quantum walk on each of the trees, employing the Perron-Frobenius operator [24]. Afterwards, we calculate the associated time-averaged density matrix for each quantum walk. This matrix captures the statistical set of quantum states resulting from the time evolution of the quantum walk. Each element along the main diagonal of this matrix represents the time-averaged probability of the walk occupying an edge in the underlying graph. These probabilities effectively characterize the likelihood of the walk moving along specific arcs throughout the course of its evolution. Based on this, we can obtain a representation vector for each graph. The specific work embedding process is shown in Figure 2.

B. Single-Qubit based Quantum Encoding of Graph Data

Data encoding for many Machine Learning (ML) tasks is often presented as column vectors of classical data. Single-qubit encoding, introduced in [14], is a strategy for encoding a vector of classical data into a characteristic Hilbert space through a series of single operations acting on each input data dimension, applied to a single qubit as shown in Figure 2.

Imagine the quantum state as a point on Earth's surface. Euler angles work like GPS coordinates: β, γ , δ controls rotation at the quantum state. Together they uniquely locate any quantum state on the Bloch sphere.

The unitary operation can be expressed as:

$$U = R_Z(\beta)R_Y(\gamma)R_Z(\delta) = e^{i\beta\sigma_z}e^{i\gamma\sigma_y}e^{i\delta\sigma_z} \tag{6}$$

Where σ are Pauli matrices. The special unitary group SU(2) is defined as the set of all 2×2 complex matrices satisfying:

$$\mathrm{SU}(2) = \{U \in \mathbb{C}^{2\times 2} \mid U^\dagger U = I, \det U = 1\} \tag{7}$$

Algorithm 1 sQGNN Training and Inference Workflow

```
Require: Graph dataset $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$, Target labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_n\}$
Ensure: Trained sQGNN model with optimized parameters $\theta^*, \phi^*$
Initialize quantum circuit parameters $\theta$, $\phi$    ▷ Trainable rotation angles
Define target states: $|\psi_{\text{target}}\rangle = \{|0\rangle$ for class 0, $|1\rangle$ for class 1$\}$
Set hyperparameters: learning rate $\eta$, epochs $E$, batch size $B$
for epoch $= 1$ to $E$ do
    for each batch in Batches$(\mathcal{G}, B)$ do
        Graph Embedding Phase:
        for each graph $G$ in batch do
            $|\psi_{\text{embed}}\rangle \leftarrow$ QuantumWalkEmbedding$(G)$    ▷ Discrete-time quantum walk
            $\mathbf{v} \leftarrow$ Vectorize$(|\psi_{\text{embed}}\rangle)$    ▷ Convert to classical vector
        end for
        Quantum Processing Phase:
        circuit_loss $\leftarrow 0$
        for each embedding vector $\mathbf{v}$ do
            $|\psi\rangle \leftarrow |0\rangle$    ▷ Initialize qubit
            Apply parameterized rotations (Euler angles):
            for $j = 0$ to len$(\mathbf{v})$ step 3 do
                $\beta \leftarrow \theta_j + v[j] \cdot \phi_j$    ▷ Z-rotation angle
                $\gamma \leftarrow \theta_{j+1} + v[j+1] \cdot \phi_{j+1}$    ▷ Y-rotation angle
                $\delta \leftarrow \theta_{j+2} + v[j+2] \cdot \phi_{j+2}$    ▷ Z-rotation angle
                $|\psi\rangle \leftarrow R_z(\beta)R_y(\gamma)R_z(\delta)|\psi\rangle$    ▷ Apply unitary operation
            end for
            Measurement & Loss Calculation:
            $|\psi_{\text{output}}\rangle \leftarrow |\psi\rangle$    ▷ Final quantum state
            $F \leftarrow |\langle\psi_{\text{target}}|\psi_{\text{output}}\rangle|^2$    ▷ Fidelity calculation
            $\mathcal{L} \leftarrow 1 - F$    ▷ Fidelity-based loss
            circuit_loss $\leftarrow$ circuit_loss $+ \mathcal{L}$
        end for
        Classical Optimization:
        Compute gradients: $\nabla_\theta, \nabla_\phi = \partial(\text{circuit\_loss})/\partial(\theta, \phi)$
        Update parameters:
          $\theta \leftarrow \theta - \eta \cdot \nabla_\theta$
          $\phi \leftarrow \phi - \eta \cdot \nabla_\phi$
    end for
end for

function PREDICT$(G_{\text{test}})$
    $\mathbf{v}_{\text{test}} \leftarrow$ Vectorize(QuantumWalkEmbedding$(G_{\text{test}})$)
    $|\psi_{\text{test}}\rangle \leftarrow$ ProcessCircuit$(\mathbf{v}_{\text{test}}, \theta^*, \phi^*)$    ▷ Using trained parameters
    $F_0 \leftarrow |\langle 0|\psi_{\text{test}}\rangle|^2$
    $F_1 \leftarrow |\langle 1|\psi_{\text{test}}\rangle|^2$
    return $\arg\max([F_0, F_1])$    ▷ Class with highest fidelity
end function
```

Each element $U \in \mathrm{SU}(2)$ can be parameterized using the exponential map:

$$U = \exp\left(-i\frac{\theta}{2}\hat{\mathbf{n}} \cdot \boldsymbol{\sigma}\right) \tag{8}$$

where $\theta \in [0, 4\pi)$, $\hat{\mathbf{n}} = (n_x, n_y, n_z)$ is a unit vector in $\mathbb{R}^3$, and $\boldsymbol{\sigma} = (\sigma_x, \sigma_y, \sigma_z)$ are the Pauli matrices. This corresponds to a rotation by angle θ around axis $\hat{\mathbf{n}}$ on the Bloch sphere.

Any SU(2) operation can be decomposed into Euler rotations as:

$$U = \exp\left(-i\frac{\beta}{2}\sigma_z\right)\exp\left(-i\frac{\gamma}{2}\sigma_y\right)\exp\left(-i\frac{\delta}{2}\sigma_z\right) \tag{9}$$

This decomposition is universal for single-qubit operations, with the rotation angles forming a diffeomorphism to SU(2)/U(1).

The Euler angles β, γ, $\delta \in \mathbb{R}$ define rotations around Z-Y-Z axes respectively. These angles are parameterized as:

$$\beta = \theta_i + x_i \cdot \phi_i \tag{10}$$

$$\gamma = \theta_{i+1} + x_{i+1} \cdot \phi_{i+1} \tag{11}$$

$$\delta = \theta_{i+2} + x_{i+2} \cdot \phi_{i+2} \tag{12}$$

Where θ_i and ϕ_i are trainable weights, and x_i is the input value at dimension i.

The combined unitary operation is:

$$U\left(\vec{\omega}\right) = e^{i\vec{\omega}\cdot\vec{\sigma}} \tag{13}$$

Where $\vec{\omega} = (\omega(\beta), \omega(\gamma), \omega(\delta))$ components convert angular rotations to exponential form. Combining three rotation gates into a unitary operation, there is:

$$\omega(\beta) = c\left(\sqrt{1 - cos^2 c}\right)^{-1}\sin\left(\frac{\gamma - \delta}{2}\right)\sin\left(\frac{\beta}{2}\right) \tag{14}$$

$$\omega(\gamma) = c\left(\sqrt{1 - cos^2 c}\right)^{-1}\cos\left(\frac{\gamma - \delta}{2}\right)\sin\left(\frac{\beta}{2}\right) \tag{15}$$

$$\omega(\delta) = c\left(\sqrt{1 - cos^2 c}\right)^{-1}\sin\left(\frac{\gamma + \delta}{2}\right)\cos\left(\frac{\beta}{2}\right) \tag{16}$$

Where $\cos c = \cos\left(\frac{\gamma+\delta}{2}\right)\cos\left(\frac{\beta}{2}\right)$. The single-qubit encoding method can be employed to encode up to three input dimensions per unitary operation. The input vector is thus cycled through in order to encode three-dimensional values until the entire input has been encoded. This method can be flexibly implemented on quantum circuits that process data of different structures and can increase the data capacity per qubit.with the scaling factor $c(\sqrt{1 - \cos^2 c})^{-1}$ ensuring proper normalization on the Bloch sphere.

For graph data, our quantum walk embedding first converts topological features to a classical vector (e.g., [bond-count, node-degree, centrality]). This vector is then encoded via the above Euler parameterization - each graph ”fingerprint” maps to unique Bloch sphere coordinates. For example, after quantum walk embedding, we get the vector $\vec{v} = \{x_0, x_1, x_2\}$, then x_0, x_1, x_2 will correspond to β, γ, δ respectively, as shown in Eq. (7-9). If the vector has more elements, more unitary operators are added according to the situation.

For classic models, there is Universal Approximation Theorem (UAT) [25] to support its approximation capabilities. Similarly, for quantum models, UAT can be used to demonstrate approximation capabilities. According to [26], a quantum analogue can be constructed on the basis of UAT. Let f and ϱ be a pair of functions, with $f \in \mathbb{R}^m \to [0,1]$ and $\varrho \in \mathbb{R}^m \to [0, 2\pi)$, there is:

$$\left| f\left(\vec{x}\right) e^{i\varrho(\vec{x})} - \left\langle 1 \left| \prod_{i=1}^{N} U\left(\vec{x}, \vec{\theta}_i, \vec{\phi}_i\right) \right| 0 \right\rangle \right| < \epsilon \tag{17}$$

Where $\epsilon > 0$. Based on this quantum UAT, it can be considered that the Single-Qubit method is able to approximate the functions.

C. Measurement and Loss Calculation in Our Single-Qubit GNN

In the measurement phase, we observe the quantum circuit to obtain its final state. Due to the relatively special step of measurement in quantum circuits compared with classical methods, we can use the characteristics of measurements to construct an evaluation method similar to entropy. As opposed to traditional direct measurement of the qubit state, we adopt a fidelity-based measurement method for classification tasks.

Fidelity measures how ”similar” two quantum states are, analogous to comparing two arrows in 3D space. If two arrows point in exactly the same direction, their fidelity is 1 (perfect match). If they point in opposite directions, fidelity is 0 (complete mismatch).

In quantum information theory, fidelity formally quantifies this ”closeness” between two quantum states. For two states ρ and σ, it’s defined as:

$$F\left(\rho, \sigma\right) = \left(tr\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)^2 \tag{18}$$

This represents the probability that state ρ would be mistaken for state σ during measurement.

For binary classification, we assign target states $|0\rangle$ (north pole of Bloch sphere) for Class 0 and $|1\rangle$ (south pole) for Class 1. Each graph’s quantum state is compared to both targets, and the higher fidelity determines classification:

$$F\left(\vec{x}, \vec{\theta}, \vec{\phi}\right) = \left| \left\langle \vec{\psi}_l \middle| \psi_{output}\left(\vec{x}_i, \vec{\theta}, \vec{\phi}\right) \right\rangle \right|^2 \tag{19}$$

Consider classifying a mutagenic compound (Class 1). After quantum processing: Fidelity with $|0\rangle$: 0.2 → unlikely Class 0; Fidelity with $|1\rangle$: 0.8 → high probability Class 1.

The loss function quantifies total classification error as the sum of ”mismatches” across all samples:

$$\mathcal{L}\left(\vec{x}, \vec{\theta}, \vec{\phi}\right) = \sum_{i=1}^{M} \left(1 - F\left(\vec{x}, \vec{\theta}, \vec{\phi}\right)\right) \tag{20}$$

Where $\vec{\psi}_l$ is the correct label state.

Having obtained the loss, we can use the optimizer to maximize the sum of the fidelity of all data points and find the best weight for classification, the parameters θ_i and ϕ_i in the unitary operation above. The optimization strategy we use is Quantum Natural Gradient, based on [27]. A distinctive feature of the quantum state space is its possession of an intrinsic metric tensor known as the Fubini-Study metric tensor. By capitalizing on this property, we can develop quantum natural gradient descent:

$$\omega_{t+1} = \omega_t - \eta g^{+}\left(\omega_t\right) \nabla\mathcal{L}(\omega) \tag{21}$$

Where g^{+} is the pseudo-inverse of the Fubini-Study metric tensor. The following is a variational quantum circuit:

$$U(\omega)|\psi_0\rangle = V_l\left(\omega_l\right) V_{l-1}\left(\omega_{l-1}\right) \cdots V_0\left(\omega_0\right)|\psi_0\rangle \tag{22}$$

Where $V_l\left(\omega_l\right)$ are layers of quantum gates with parameters. ω represents the parameters, including the above mentioned β, γ and δ. Considering that only rotation gates are used in the Single-Qubit method, these quantum gates can be transformed into the following form:

$$X\left(\omega_i^{(l)}\right) = e^{i\omega_i^{(l)} K_i^{(l)}} K_i^{(l)} \tag{23}$$

The block-diagonal submatrix of the Fubini-Study tensor is:

$$g_{ij}^{(l)} = \langle\psi_{l-1}|K_i K_j|\psi_{l-1}\rangle - \langle\psi_{l-1}|K_i|\psi_{l-1}\rangle\,\langle\psi_{l-1}|K_j|\psi_{l-1}\rangle \tag{24}$$

$$|\psi_{l-1}\rangle = V_{l-1}\left(\omega_{l-1}\right) \cdots V_0\left(\omega_0\right)|\psi_0\rangle \tag{25}$$

Based on this, through quantum backpropagation, we can train the quantum model at a faster speed.

IV. EXPERIMENTS

We chose the MUTAG, PTC series, PROTEINS, and NCI1 datasets as experimental data. The MUTAG dataset contains 188 nitro compounds represented as isomer graphs. The full name of PTC is Predictive Toxicology Challenge, which is used to develop advanced SAR technology predictive toxicology models. According to the experimental rodent species, there are a total of 4 datasets: PTC_FM, PTC_FR, PTC_MM, and PTC_MR. The PROTEINS dataset consists of 1,113 protein structures where nodes represent secondary structure elements and edges indicate neighborhood relationships, with classification tasks identifying enzyme/non-enzyme proteins. The NCI1 dataset contains 4,110 chemical compounds screened for activity against non-small cell lung cancer, represented as atom-bond graphs with classification of active/inactive compounds. We use PennyLane [37] and PyTorch [38] to perform experiments. The quantum computer we use is the ibm_manila node provided by IBM [39]. This quantum computer features 5 qubits, a Quantum Volume of 32, capability to perform 2800 circuit layer operations per second, and employs the Falcon r5.11L processor model.

TABLE I

THE AVERAGE ACCURACY OF THE MODELS OVER DIFFERENT REAL-WORLD GRAPH DATASETS.

Model		**MUTAG**	**PTC_FM**	**PTC_FR**	**PTC_MM**	**PTC_MR**	**PROTEINS**	**NCI1**
Classical	**edGNN[28]**	86.9±1.0	59.8±1.5	65.7±1.3	64.4±0.8	56.3±1.9	74.1 ± 0.6	74.2 ± 0.3
	R-GCN[29]	81.5±2.1	60.7±1.7	**65.8±0.6**	64.7±1.7	58.2±1.7	72.6±1.5	69.1±1.2
	GIN[30]	85.4±3.5	64.4±6.7	65.1±5.3	**64.8±5.4**	**64.6±7.0**	**76.2±2.8**	**77.0±3.3**
	RW-GNN[31]	**88.3±4.1**	60.9±2.7	63.1±1.3	63.2±1.4	57.1±1.4	74.7±3.3	75.3±2.6
	TOGL[32]	87.2±3.8	**64.9±4.6**	64.8±3.8	63.1±4.1	60.3±4.7	75.2±3.9	76.0±3.9
Quantum	**GBS[33]**	86.4±0.3	53.8±1.0	-	-	-	66.9 ±0.2	63.6±0.2
	QJSK[34]	83.4±0.5	-	-	-	58.2±0.8	-	67.4±0.2
	QJSKT[34]	81.6±0.5	-	-	-	57.4±0.4	-	67.0±0.2
	QSVM[35]	81.8±10.3	60.7±5.4	62.5±10.2	57.5±8.8	60.9±7.3	72.5±9.4	77.1±6.6
	QMLP[36]	81.5±9.7	60.3±6.2	62.7±9.6	58.2±8.4	61.2±6.9	73.1±9.1	77.0±6.2
	QCNN[11]	81.0±14.1	59.9±10.9	63.2±11.2	59.0±9.9	61.5±8.6	74.1±10.6	77.1±10.1
Ours	**sQGNN**	**87.3±4.8**	**66.2±7.3**	**66.9±5.5**	**65.9±4.2**	**65.9±5.5**	**76.8±6.6**	**78.2±6.7**
	sQGNN-Dual	87.2±4.2	66.2±6.9	66.4±4.8	66.0±4.1	65.9±6.1	75.6±3.2	77.3±6.5

TABLE II

THE AVERAGE ACCURACY OF THE MODELS WITH DEPOLARIZING ERROR.

Model	**Depolarization Probability**	**MUTAG**	**PTC_FM**	**PTC_FR**	**PTC_MM**	**PTC_MR**	**PROTEINS**	**NCI1**
QSVM[35]	**0.001**	81.8±10.3	60.7±5.4	62.5±10.2	57.5±8.8	60.9±7.3	72.5±9.4	77.1±6.6
	0.01	78.3±9.8	58.1±5.6	60.2±9.6	54.2±8.5	58.3±6.9	69.8±8.9	74.3±6.1
	0.1	73.6±9.1	54.7±6.2	56.8±9.1	50.8±7.8	54.9±6.5	65.4±8.1	70.0±5.6
QMLP[36]	**0.001**	81.5±9.7	60.3±6.2	62.7±9.6	58.2±8.4	61.2±6.9	73.1±9.1	77.0±6.2
	0.01	77.8±9.2	57.2±6.0	59.3±8.9	54.9±7.7	57.8±6.5	69.6±8.4	73.7±6.0
	0.1	72.1±8.6	53.8±6.4	55.7±8.4	50.2±7.2	53.9±6.1	65.1±7.7	69.5±5.5
QCNN[11]	**0.001**	81.0±14.1	59.9±10.9	63.2±11.2	59.0±9.9	61.5±8.6	74.1±10.6	77.1±10.1
	0.01	76.4±13.1	56.5±10.1	60.0±10.5	55.6±9.1	58.2±8.1	69.3±9.9	72.6±9.4
	0.1	70.2±12.0	52.0±9.5	55.3±9.7	50.8±8.6	53.6±7.5	63.7±8.7	67.2±8.3
sQGNN	**0.001**	87.3±3.3	65.0±7.2	66.5±9.1	65.3±4.2	65.8±4.4	76.5±6.2	77.8±6.3
	0.01	86.7±4.9	65.1±8.6	66.1±5.6	65.7±4.5	65.8±5.1	76.2±6.3	77.3±6.5
	0.1	84.8±6.4	64.0±7.7	65.0±8.4	63.7±4.8	64.5±4.5	74.9±6.6	75.4±6.7

A. Experiments in the Simulated Quantum Environment

We configured two simulation setups for our sQGNN models: an ideal quantum environment and a noisy environment involving noise interference in the simulated NISQ devices. To evaluate the robustness of the sQGNN model, we conducted multiple experiments under both simulated environments and tested the classic models with edGNN[28], R-GCN[29], GIN[30], RW-GNN[31], TOGL[32] and sQGNN. The tested quantum models include GBS[33], QJSK, QJSKT[34], QSVM[35], QMLP[36], QCNN[11]. To verify the Single-Qubit method, we set up a control model composed of two quantum bits. The two qubits of the sQGNN-Dual model are the same. The input data is encoded once in the two qubits, and there is a quantum entanglement between the two qubits. We set the learning rate to 0.1, epoch to 200, dataset ratio to 8:2, and repeated experiments 10 times for each model and dataset.

Ideal simulated environment. Table 1 presents the test results of the sQGNN model in an ideal simulated quantum environment. The best result is underlined and the second result is bolded. The results indicated that the sQGNN model achieved the best performance on the most of datasets. As a research [**?**], increasing the number of processing layers (i.e., unitary operations) of the single-qubit method improves the model's performance, especially when the number of unitary operations is below three. This could be attributed to the relatively low number of feature elements in the graph representation vector of MUTAG data.

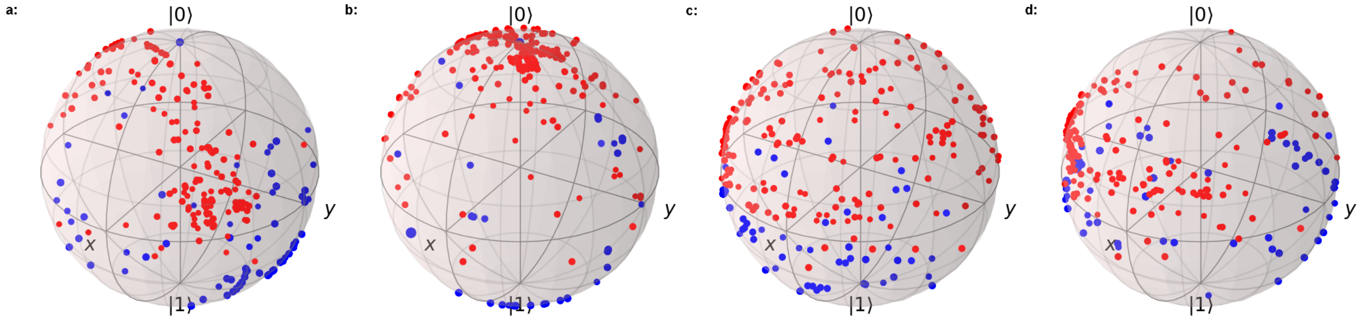
The comparison of the results in the ideal quantum environment revealed that the overall accuracy of our sQGNN model was superior. Moreover, the sQGNN method has the advantage of a compact model structure. Classical graph neural networks tend to be complex in terms of the model structure due to the complexity of graph data itself. By contrast, our model, which employs a graph embedding method without hyperparameters and Single-Qubits QGNN quantum circuits, obtains a huge advantage in the total number of parameters. For instance, in the MUTAG dataset, our model requires only 12 parameters, whereas a classical neural network such as GIN has 13,000 parameters.

The performance of sQGNN and sQGNN-Dual models did not show a significant difference. The standard deviation of the sQGNN-Dual model is relatively small, but this is foreseeable, because the input data of the sQGNN-Dual model has been entered once more once, and the number of parameters increases. In the task set in the experiment, the number of qubits increased significantly, but the quantum resources used were doubled.

Noisy simulated environment. To verify the robustness of sQGNN against noise when running on NISQ devices, we train and test the models in a noisy simulated environment. As shown in Table 3, the depolarization error was set as the noise in the environment for the sQGNN model. The depolarization error applies a random Pauli (e.g. X, Y, Z) to each qubit in a group of qubits and is described by the following formula:

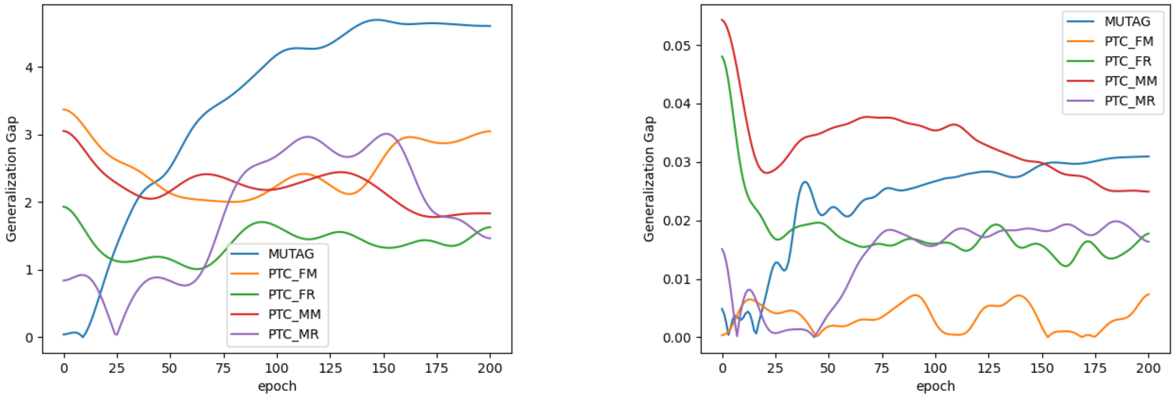
$$E(\rho) = (1 - \varepsilon)\,\rho + \varepsilon Tr[\rho]\frac{I}{2^n} \tag{26}$$

Where ε is the depolarizing error param, n is the number of



(a) (b) (c) (d)

Fig. 3. This figure is a demonstration of all graph data of the PTC dataset in Hilbert space. The red dot is class 0 and the blue dot is class 1. Red points correspond to Class 0 and blue points to Class 1. This distribution indicates stronger classification ability for Class 0, as these states remain closer to the circuit's initial state $|0\rangle$ and require less transformation.



(a) (b)

Fig. 4. (a) Results of Generalization Gap between training error and testing error. (b) Results of Generalization Gap between training loss and testing loss.

qubits for the error channel and I is the Pauli matrix.

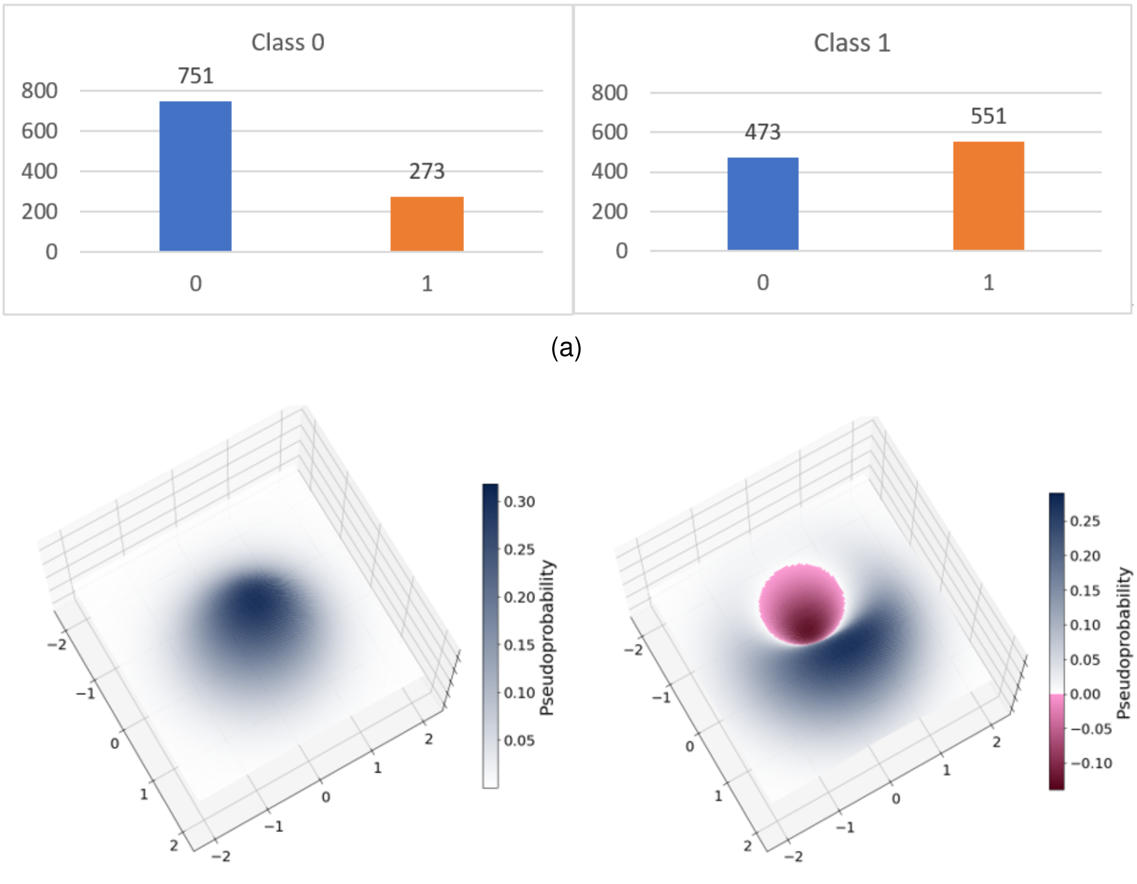
We set three levels of the depolarization parameter to assess the tolerance of sQGNN to depolarization errors. The test results are shown in Table 2 and reveal that sQGNN is almost unaffected by depolarization errors at 0.001 and 0.01, and slightly affected at 0.1, compared to other multi-qubits quantum circuit model. Despite the functional limitations of quantum devices in the NISQ era, the influence of noise is unavoidable, resulting in significant degradation of performance on real quantum computers. However, the results of this experiment show that the sQGNN model can stably maintain good performance in the presence of noise.

Figure 3 displays the distribution of graphs from the PTC_FM dataset in Hilbert space on the surface of Bloch spheres. Red points correspond to Class 0 and blue points to Class 1. The spatial separation shows Class 0 primarily concentrated in the upper hemisphere (near the $|0\rangle$ state at the north pole), while Class 1 predominantly occupies the lower hemisphere (near the $|1\rangle$ state at the south pole). This distribution indicates stronger classification ability for Class 0, as these states remain closer to the circuit's initial state $|0\rangle$ and require less transformation. The presence of Class 1 outliers in the upper hemisphere suggests either: (1) residual influence of the initial $|0\rangle$ state on harder-to-classify samples, or (2) inherent structural properties in certain graph data that resist full state transformation.

Figure 4 shows our analysis of the generalization ability of sQGNN on various data sets. The Generalization Gap in Figure 4(a) is derived from calculating the difference between training error and testing error, and in 4(b) is between training loss and testing loss. It can be noticed from the two figures that although the model gradually becomes stable as the training process proceeds, there are some fluctuations in this overall trend due to the existence of quantum noise. The generalization ability of the model on the PTC data set is stronger than that on the MUTAG data set. The representation vector obtained by the MUTAG data set after quantum walking processing only has 4 elements. Compared with 24 PTC data, quantum noise will have a greater impact on some MUTAG data.

B. Real Quantum Device Test Experiment

We evaluated the performance of our single-qubit sQGNN model on the MUTAG dataset using a real quantum computer provided by IBM. The model was deployed in a simulated



(a)

(b)

Fig. 5. (a) Results of testing the effects of the model on IBM's quantum computer, ibm_manila. Here are the measurements of two graphs with classes 0 and 1 in the test data. The ordinate represents the number of occurrences of a particular state in the measurement. (b) Visualization of the Wigner quasi-probability function for Class 0 and 1. Our model discriminates between the two classes clearly on a quantum computer.

quantum environment to an IBM online quantum computer, facilitated by IBM Quantum Lab.

Figure 5 demonstrates the test results of two samples on the real IBM quantum computer and the corresponding simulated quantum environment. The accuracy obtained on the MUTAG dataset was 88.89%, with no degradation in accuracy compared to the simulation environment. Furthermore, the measurement results of $|0\rangle$ and $|1\rangle$ are shown in Figure 5(a), leading to the observation that Class 0 is better classified than Class 1. This phenomenon can be attributed to the fact that the initial state is $|0\rangle$ and the difference in the graph data structure of the two classes, which is consistent with the results obtained in the simulated environment in Figure 3.

Figure 5(b) visualizes the Wigner quasi-probability distribution [40] of the final quantum state for a representative sample, providing phase-space insight into classification performance. The distinct separation between the two classes' distributions demonstrates the model's ability to map different graph structures to well-separated quantum states - a critical factor for classification accuracy. The preserved separation despite noise artifacts confirms the algorithm's robustness on NISQ devices, as overlapping distributions would indicate degraded class distinguishability. Since current quantum computers initialize to $|0\rangle$ (represented at the sphere's north pole), the clear differentiation from the $|1\rangle$ region (south pole) in Class 1 samples validates our state transformation effectiveness. This initialization constraint will persist due to hardware limitations, making the demonstrated separation crucial for practical classification tasks.

Despite promising results, several failure cases and hardware limitations emerged during real-device deployment on IBM Manila. The most significant constraint was readout error (avg. 2.594e-2), which caused misclassification of samples near decision boundaries. For example, when processing graph structures with ambiguous topological features, flipping classification outcomes. This was particularly pronounced in graphs with betweenness centrality values in the 0.4-0.6 range, where quantum state measurements became inherently ambiguous.

V. DISCUSSION

In the Noisy Intermediate-Scale Quantum (NISQ) era, quantum algorithm implementation is constrained by unavoidable device noise. The single-qubit method efficiently utilizes limited quantum resources, suggesting our sQGNNs could be extended to other tasks. This approach economizes scarce quantum computing resources.

Experimental evidence indicates our single-qubit method exhibits noise resilience in both simulated and actual quantum environments. While Quantum Error Correction (QEC)

TABLE III
ERROR PROFILE COMPARISON ON REAL QUANTUM HARDWARE

	Our model
Qubits	5
Quantum Volume	32
Circuit layer operations per second	2800
Processor model	Falcon r5.11L
Layout	Linear
Coherence times average T1	155.49µs
Coherence times average T2	88.69µs
Frequency(GHz)	4.963
Single-qubit gate AVG. error	3.093e-4
Two-qubit gate AVG. error	8.256e-3
Average readout error	2.594e-2

algorithms mitigate device noise errors, they require substantial additional resources. Our algorithm's inherent robustness conserves valuable resources by reducing reliance on costly QEC mechanisms.

Implementing complex neural networks on Variational Quantum Circuits (VQCs) presents significant challenges. Our graph embedding technique based on discrete-time quantum walks effectively extracts structural information while balancing computational resources. Compared to classical methods, our quantum circuit achieves comparable performance with substantially fewer parameters, enabling efficient integration with both quantum and classical models.

The single-qubit approach inherently reduces susceptibility to quantum decoherence. However, recent theoretical analysis [17] questions whether single-qubit implementations limit expressivity for multivariate functions. To address this, we compared sQGNN with a dual-qubit version (sQGNN-Dual). As Table I shows, sQGNN achieved slightly better accuracy, indicating no significant performance difference and alleviating concerns about single-qubit efficacy in our tested tasks.

Nevertheless, we acknowledge that single-qubit circuits may face limitations in approximating complex multivariate dependencies or exploiting entanglement, potentially affecting scalability for larger or more intricate problems. While increasing to two qubits slightly enhanced robustness, it did not significantly improve performance in our experiments. This suggests our tasks may not fully capture the benefits of increased entanglement capacity, which remains an important direction for future exploration.

Prior work [17] demonstrated single-qubit methods' approximation power, while [16] established their ability to approximate univariate functions. Our work provides practical validation of single-qubit quantum neural networks on complex graph data, but further investigation is warranted to assess scalability and generalization to higher-dimensional cases.

By using only one qubit, sQGNN minimizes resource requirements and avoids the "barren plateau" problem in VQCs. While studies [10, 27, 41] have proposed mitigation strategies, a fundamental solution remains elusive. The single-qubit method offers an effective near-term approach given barren plateaus' persistence in VQA-based QML.

VI. CONCLUSION

In conclusion, to address the challenges on the constraints of limited quantum resources facing the current VQCs of NISQ era, we successfully leveraged a single qubit strategy and developed a novel quantum graph neural network architect on a single qubit by integrating a quantum walk graph embedding method with sQGNN quantum circuits that can successfully reduce the required parameters from over 10k to only 12 parameters (for MUTAG dataset). Our new method forms a concise structure suitable for deployment on quantum devices during the NISQ era. Our experimental results underscore the model's robustness on both quantum computing simulators and real quantum computers, showing its great potential on overcoming resource bottlenecks, achieving robust performance, withstanding quantum noises, and mitigating the "barren plateau" issues. However, we recognize that the single-qubit approach may face challenges in modeling complex multivariate dependencies or leveraging entanglement. While it serves as a practical and noise-resilient method for near-term quantum devices, future work will be needed to explore scalability and expressive power as quantum hardware advances. Our results represent a promising step toward harnessing quantum machine learning for real-world applications within current resource constraints.

REFERENCES

[1] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[2] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, "Power of data in quantum machine learning," *Nature communications*, vol. 12, no. 1, p. 2631, 2021.

[3] Y. Zhu, A. Bouridane, M. E. Celebi, D. Konar, P. Angelov, Q. Ni, and R. Jiang, "Quantum face recognition with multi-gate quantum convolutional neural network," *IEEE Transactions on Artificial Intelligence*, 2024.

[4] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[5] M. Fellous-Asiani, J. H. Chai, R. S. Whitney, A. Auffèves, and H. K. Ng, "Limitations in quantum computing from resource constraints," *PRX Quantum*, vol. 2, no. 4, p. 040335, 2021.

[6] S. Chen, J. Cotler, H.-Y. Huang, and J. Li, "The complexity of nisq," *Nature Communications*, vol. 14, no. 1, p. 6001, 2023.

[7] A. A. Clerk, M. H. Devoret, S. M. Girvin, F. Marquardt, and R. J. Schoelkopf, "Introduction to quantum noise, measurement, and amplification," *Reviews of Modern Physics*, vol. 82, no. 2, p. 1155, 2010.

[8] J. Preskill, "Quantum computing and the entanglement frontier," *arXiv preprint arXiv:1203.5813*, 2012.

[9] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[10] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature communications*, vol. 9, no. 1, p. 4812, 2018.

[11] I. Cong, S. Choi, and M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.

[12] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[13] S. Krinner, S. Storz, P. Kurpiers, P. Magnard, J. Heinsoo, R. Keller, J. Luetolf, C. Eichler, and A. Wallraff, "Engineering cryogenic setups for 100-qubit scale superconducting circuit systems," *EPJ Quantum Technology*, vol. 6, no. 1, p. 2, 2019.

[14] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, 2020.

[15] P. Easom-McCaldin, A. Bouridane, A. Belatreche, R. Jiang, and S. Al-Maadeed, "Efficient quantum image classification using single qubit encoding," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[16] A. Pérez-Salinas, D. López-Núñez, A. García-Sáez, P. Forn-Díaz, and J. I. Latorre, "One qubit as a universal approximant," *Phys. Rev. A*, vol. 104, p. 012405, Jul 2021.

[17] Z. Yu, H. Yao, M. Li, and X. Wang, "Power and limitations of single-qubit native quantum neural networks," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 27 810–27 823.

[18] M. Schuld, "Machine learning in quantum spaces," 2019.

[19] G. Verdon, T. McCourt, E. Luzhnica, V. Singh, S. Leichenauer, and J. Hidary, "Quantum graph neural networks," *arXiv preprint arXiv:1909.12264*, 2019.

[20] J. Zheng, Q. Gao, and Y. Lü, "Quantum graph convolutional neural networks," in *2021 40th Chinese Control Conference (CCC)*. IEEE, 2021, pp. 6335–6340.

[21] C. Tüysüz, C. Rieger, K. Novotny, B. Demirköz, D. Dobos, K. Potamianos, S. Vallecorsa, J.-R. Vlimant, and R. Forster, "Hybrid quantum classical graph neural networks for particle track reconstruction," *Quantum Machine Intelligence*, vol. 3, pp. 1–20, 2021.

[22] L. Bai, L. Rossi, L. Cui, J. Cheng, and E. R. Hancock, "A quantum-inspired similarity measure for the analysis of complete weighted graphs," *IEEE Transactions on Cybernetics*, vol. 50, no. 3, pp. 1264–1277, 2020.

[23] D. Emms, S. Severini, R. C. Wilson, and E. R. Hancock, "Coined quantum walks lift the cospectrality of graphs and trees," *Pattern Recognition*, vol. 42, no. 9, pp. 1988–2002, 2009.

[24] P. Ren, T. Aleksić, D. Emms, R. C. Wilson, and E. R. Hancock, "Quantum walks, ihara zeta functions and cospectrality in regular graphs," *Quantum Information Processing*, vol. 10, pp. 405–417, 2011.

[25] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[26] A. Pérez-Salinas, D. López-Núñez, A. García-Sáez, P. Forn-Díaz, and J. I. Latorre, "One qubit as a universal approximant," *Physical Review A*, vol. 104, no. 1, p. 012405, 2021.

[27] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, "Quantum natural gradient," *Quantum*, vol. 4, p. 269, 2020.

[28] G. Jaume, A.-p. Nguyen, M. R. Martínez, J.-P. Thiran, and M. Gabrani, "edgnn: a simple and powerful gnn for directed labeled graphs," *arXiv preprint arXiv:1904.08745*, 2019.

[29] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*. Springer, 2018, pp. 593–607.

[30] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[31] G. Nikolentzos and M. Vazirgiannis, "Random walk graph neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 16 211–16 222, 2020.

[32] M. Horn, E. De Brouwer, M. Moor, Y. Moreau, B. Rieck, and K. Borgwardt, "Topological graph neural networks," in *International Conference on Learning Representations (ICLR)*, 2022.

[33] M. Schuld, K. Brádler, R. Israel, D. Su, and B. Gupt, "Measuring the similarity of graphs with a gaussian boson sampler," *Phys. Rev. A*, vol. 101, p. 032314, Mar 2020.

[34] L. Bai, L. Rossi, L. Cui, Z. Zhang, P. Ren, X. Bai, and E. Hancock, "Quantum kernels for unattributed graphs using discrete-time quantum walks," *Pattern Recognition Letters*, vol. 87, pp. 96–103, 2017.

[35] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Physical review letters*, vol. 113, no. 13, p. 130503, 2014.

[36] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, no. 3, p. 032308, 2020.

[37] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. Akash-Narayanan, A. Asadi *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.

[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[39] IBM, "Ibm quantum," https://quantum-computing.ibm.com/, 2023, [Online; accessed 1-March-2023].

[40] B. Hiley, "Phase space descriptions of quantum phenomena," in *Proc. Int. Conf. Quantum Theory: Reconsideration of Foundations*, vol. 2, 2004, pp. 267–86.

[41] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, "An initialization strategy for addressing barren plateaus in parametrized quantum circuits," *Quantum*, vol. 3, p. 214, 2019.