Quantum Graph Neural Networks on a Single Qubit

Author Information

Yijie Zhu and Richard Jiang*

Affiliations

LIRA Center, University of Lancaster, Lancaster, UK

Contributions

Y.Z. contributed the paper drafting, the technical implementation, and the experimental exploration; R.J. contributed the ideas, the experimental design and the paper writing.

Corresponding author

Correspondence to: Richard Jiang, r.jiang2@lancaster.ac.uk

Abstract

Machine learning based on graph data is a hard task due to the complex structure of graph data. When this topic comes to quantum machine learning, it becomes more challenging due to the limited resources in the current NISQ era, in particular when a neural network can have thousands or even millions of parameters. In this paper, to address these challenges, we proposed a novel quantum graph neural network that uses only a single qubit. In our novel work, a graph embedding method was leveraged to waive the needs of hyperparameters, leading to the single-qubit based implementation with the least number of parameters in the network. Our experiments show that our method can be adapted to graph data of different structures and sizes, and can be efficiently deployed on real quantum computers. Our sQGNN quantum graph neural network (sQGNN) exhibits a high capability of fault tolerance towards noisy NISQ devices. In the current NISQ era, our single-qubit-based quantum neural networks can maximize the use of limited qubits and enable the implementation of large quantum graph neural networks on resource-limited VQCs, showing a new promising pathway to overcome the current major shortcomings of VQCs facing real-world applications.

Introduction

Machine learning, particularly bioinspired deep learning, has been widely applied to enable many new applications [1,2]. Meanwhile, with the continuous development of information technology and the exponential increase in information density, the search for innovative approaches to the increasing complexity of machine learning has become imperative [3]. Quantum computing, as a promising candidate with its supremacy in computing power, has attracted the greatest interest of researchers in the field of artificial intelligence. Due to the quantum superposition mechanism, quantum computing has demonstrated its exponential acceleration advantages over classical computing when dealing with high-dimensional data [3]. For example, quantum algorithms can factor numbers [4], simulate quantum systems [5], or solve systems of linear equations [6] with an exponential acceleration compared to classical methods. The cross-disciplinary marriage of quantum computing and machine learning has brought out the celebrated birth of quantum machine learning, with a great hope to leverage the advantages of quantum computing to improve classical machine learning algorithms [7]. Quantum machine learning has emerged as a promising panacea for the challenges in the area of machine learning, despite of the hardware and software implementation challenges in the current noisy intermediate-scale quantum (NISQ) era of quantum computing [8].

Currently, quantum machine learning tends to utilize quantum mechanisms to implement or optimize various classical machine learning algorithms, such as Quantum Boltzmann Machines (QBM) [9], Quantum Principal Component Analysis (QPCA) [10], and Quantum Support Vector Machines (QSVM) [11][12]. Quantum neural networks were proposed initially in 1995 [13], and various possible quantum neural network models have been widely explored, such as the Quantum Perceptron Model [14], Quantum Tensor Neural Network [15] and Quantum Convolutional Neural Network [16]. The researchers hope to solve the problem of high-dimensional space and complexity in classical neural networks by transforming the classical neural network model into a quantum system with a network-like structure in Hilbert space.

Machine learning based on graph-structured data has always become one of the hottest topics in the field of machine learning, with many potential applications in social networks, point clouds, molecular chemistry and other fields [1] [17]. Most of the classical neural network algorithms are designed to process data with regular structures in Euclidean space. Due to the complex structure of graph data, classical neural networks inevitably face a drop in efficiency. Many graph-based problems are NP-hard [18], which is a big challenge for neural network computational performance. Taking advantage of quantum computing over classical methods has become one promising solution to tackle the computational complexity in processing graph data. Unlike classical machine learning, the data in quantum machine learning often presents in a high-dimensional Hilbert space represented by quantum states, while such high-dimensional Hilbert spaces have been shown to be beneficial for classification [19].

Although quantum computers have achieved good results [3], there are several big limitations in the NISQ era [8]. First, the number of qubits is limited in current quantum computers, and the state-of-the-art devices range in size from 50 to 100 qubits, which are expensive and have architectural limitations and limited programmability [8]. Second, the impact of noises is huge [20], and the advent of fault-tolerant quantum computers seems to be years or even decades away [8]. The real promise of quantum computers, the acceleration of real-world applications, often referred to as quantum supremacy [21], has yet to materialize. Therefore, the key technical question is how to fully utilize today's NISQ devices to achieve quantum advantages. Any such strategy must take into account: a limited number of qubits, limited qubit connectivity, and coherent and incoherent errors that limit the depth of quantum circuits. In addition to the limitations of quantum hardware, Variational Quantum Algorithms (VQAs) [22], which are widely used in quantum machine learning, also have many limits, including their trainability, accuracy and efficiency, especially the "barren plateau" can best reflect the limitations currently faced by VQA [23].

In this paper, we offered a novel method by implementing the whole quantum graph neural network on a single qubit [24][25], enabling the tackling of the complex graph data analysis on the limited NISQ

VQCs. First, we use a graph embedding method based on DCH-E [26] to waive the need for any hyperparameters to convert a graph data sample into a vector. Secondly, we utilize a new encoding scheme that can easily encode the vectorized graph data onto quantum circuits, and utilize the properties of quantum computing to drastically reduce the number of parameters that classical neural networks usually need. In our experiments, the number of parameters was reduced to several tens from thousands and beyond. Consequently, the occupancy of quantum volume will be drastically minimized. Thirdly, the uploaded quantum graph data is fed into a novel quantum graph neural network classifier that uses only a single qubit [24][25], which can be trained to fit the objective function through a variety of quantum rotation gates.

Our proposed graph quantum neural networks can be used to encode various graph data of different structures towards the single-qubit-based implementation without the need for hyperparameters. Our above innovations enable the efficient quantum processing of complex graphics data possible in the real world. We verified the feasibility of our model through experiments on both the simulated quantum environments and the real quantum computer. The experimental results successfully validated that our proposed single-qubit-based quantum graph neural networks (sQGNN) can handle the complex real-world graph data with excellent performance on the resource-limited quantum computing platform, demonstrating a novel pathway to offer a panacea for the great challenges in the current NISQ quantum technologies.

Figure 1 shows the workflow, which consists of three steps: First, the DHC-E graph embedding method converts raw graph data into a vector form that is easy to process without the need for hyperparameters. Second, the processed graph representation vector is input into the single-qubit circuit in the order of unitary operations, as shown in Figure 1. The single-qubit method can encode all data with a single qubit, saving the total number of qubits in the NISQ era and effectively avoiding the "barren plateau" phenomenon [23] caused by too many qubits in VQCs. Finally, we measure the single-qubit circuit, calculate the fidelity and loss of the model, and update the parameters of the quantum circuits to achieve

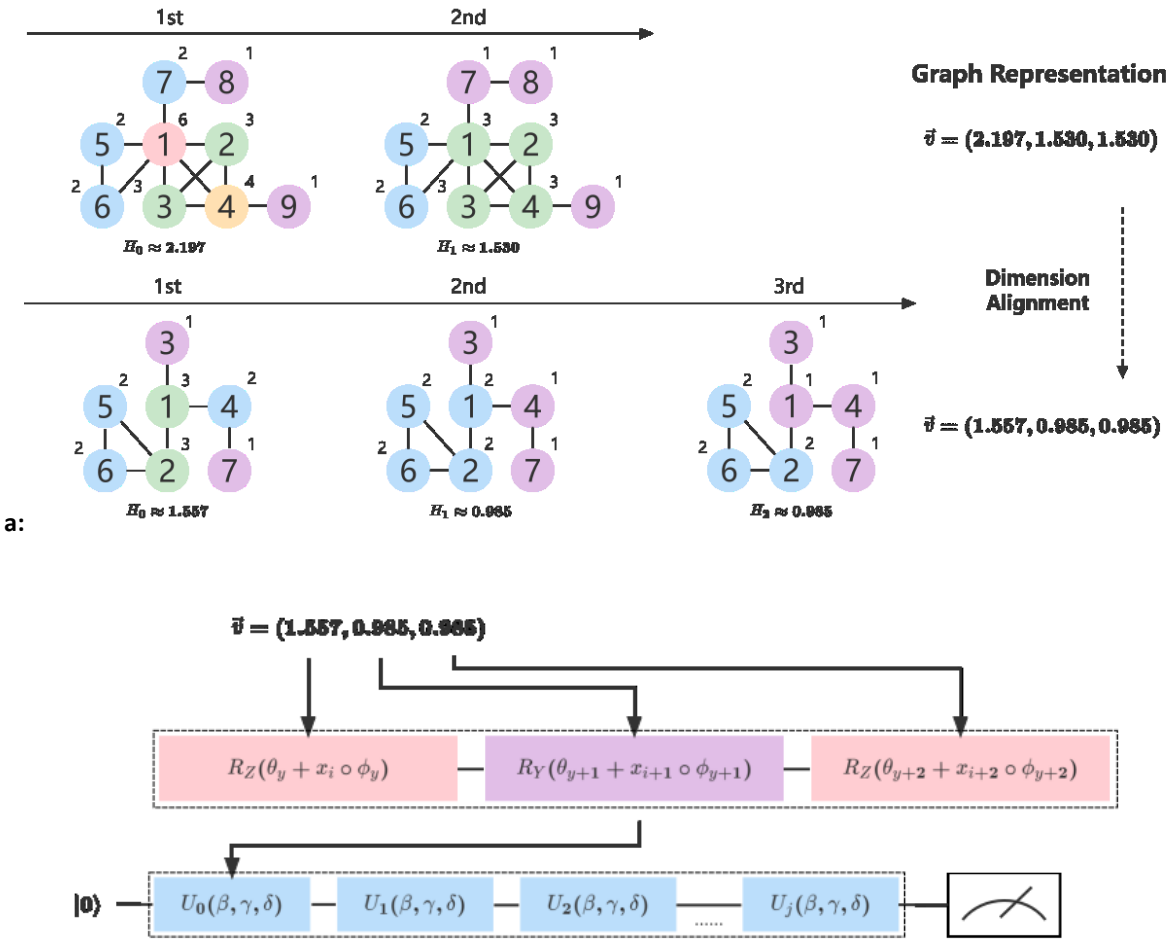

Figure. 1. (a) For the other states generated from the DHC updating process of a graph, the same rules are applied to calculate their Shannon entropy. At the end of the DHC updating process, one obtains two Shannon entropy sequences of the two graphs (2.197, 1.530) and (1.557, 0.985, 0.985), which are two graphs' embeddings. The two embeddings are with unequal dimensions. Based on the highest dimension of the two embeddings (i.e., 3), the last element of (2.197, 1.530) is used to expand (2.197, 1.530) to (2.197, 1.530, 1.530) for dimension alignment. (b) 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 revolving gate. Every three quantum revolving gates constitute a unitary operation, and the Single-Qubit method uses this unitary operation as the basic unit of encoding and parameter training. The initial state of the quantum circuit is |0>, after the state change of the quantum circuit through unitary operation, the expected value of the final state is obtained by measurement at the end of the quantum circuit.

the effects of neural network training. From this training process, we get our trained quantum neural network models. More details on these three steps are in the Methods. Following this, we carried out experiments to evaluate our sQGNN method over several real-world datasets from chemistry and biology domains.

Figure 2 shows a) the schematic diagrams of a classical neural network, b) a common quantum circuit, and c) our single-qubit quantum circuit. As shown in the figure, the quantum circuits and the classical

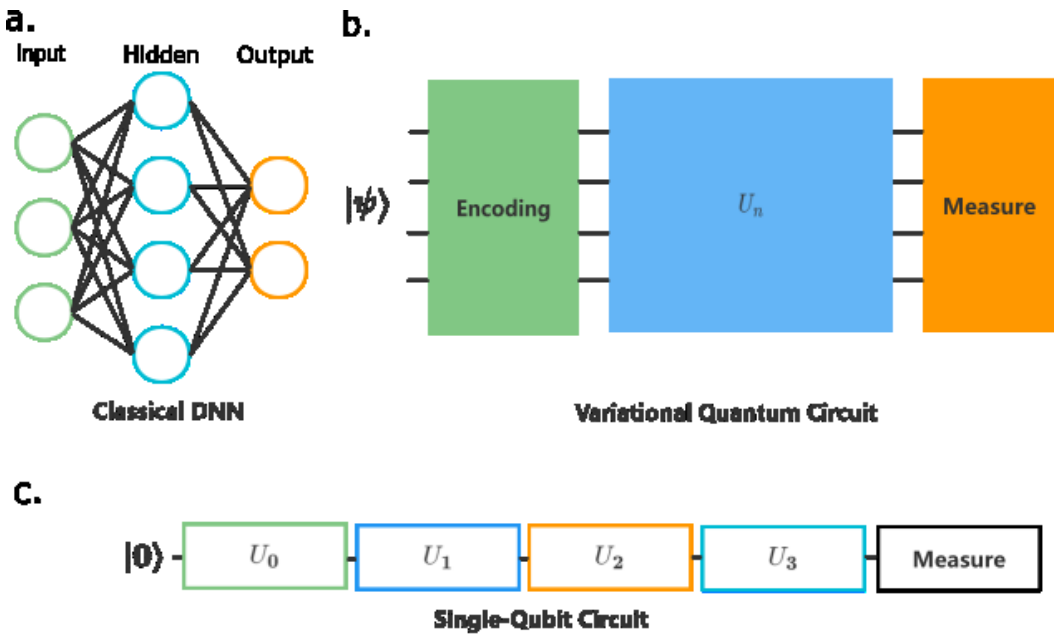

Figure. 2. This figure shows the similarities and differences between classical neural networks, general quantum circuits, and single-qubit circuits.

neural network are logically similar. There are four neurons in the hidden layer of the classical neural network (Fig.2a), which may be embodied as four qubits in the general quantum circuit in Fig.2b to carry the corresponding data. Considering the dimensions of input data, this number may be much higher. In the single qubit circuit, it can be embodied as four unitary operations on one qubit. When processing high-dimensional data, a quantum rotary gate can be added in the unitary operation without adding more qubits. According to the needs of specific input data, the number of unitary operations, and the number and type of rotation gates in unitary operations can be adjusted accordingly. For common quantum circuits, if the data size or dimension is high, it is inevitable to increase the number of qubits. For the single-qubit method, a single qubit can carry as much data as possible by adding unitary operations if the quantum volume allows. This structure cannot only make data easier to encode, but also save limited qubit resources. Besides, fewer qubits can help avoid the problem of difficult model training caused by the "barren plateau" [23], making our sQGNN an ideal candidate for the challenging quantum processing of graph data.

Results

Experiments in the Simulated Quantum Environment

In our experiments, we first used a simulated quantum environment to test the performance of the model. Due to the functional limitations of quantum computers in the NISQ era, the influence of noises is inevitable and unignorable. To fully demonstrate the capabilities of our models, we set up multiple experiments under the simulated environments. We configured two simulation setups for our sQGNN models: one is the ideal quantum environment and another is the noisy environment in the presence of noise interference in the simulated NISQ devices. We hope to test the robustness of the sQGNN model through such an experimental environment. The test models include edGNN[27], R-GCN[28], GIN[29], sQGNN (the ideal quantum setup) in an ideal environment and a quantum environment with depolarizing error for noise testing.

Table 1 shows the test results of the model in an ideal simulated quantum environment. From the results, we found that the sQGNN models had achieved the best performance on the four datasets of small

TABLE 1. The average accuracy of the models over different real world graph datasets

MODEL	MUTAG	PTC_FM	PTC_FR	PTC_MM	PTC_MR
EDGNN[27]	86.9±1.0	59.8±1.5	65.7±1.3	64.4±0.8	56.3±1.9
R-GCN[28]	81.5±2.1	60.7±1.7	65.8±0.6	64.7±1.7	58.2±1.7
GIN[29]	**89.4±5.6**	64.3±10.0	65.3±5.6	**65.8±5.9**	64.6±7.0
SQGNN	**87.3±4.8**	**65.2±7.3**	**66.9±5.5**	**65.9±4.2**	**65.9±5.5**

TABLE 2. The parameters of each model with MUTAG dataset. The parameters of GNN include all the parameters of the weight matrix to be trained, and the parameters of the quantum algorithm are the angles required by all quantum gates.

MODELS	EDGNN[27]	R-GCN[28]	GIN[29]	SQGNN
PARAMETERS	16704	9345	13000	8

molecule compounds, PTC_FM, PTC_FR, PTC_MM and PTC_MR [30]. The average accuracy rates of the sQGNN model on the PTC series datasets are 65.20%, 66.94%, 66.94%, and 65.93%, respectively. For the MUTAG dataset [31], the sQGNN model is slightly inferior to the GIN model.

Our sQGNN models have obvious advantages on the PTC series dataset. According to research [32], as the number of processing layers (i.e. unitary operations) of the single-qubit method increases, the model will perform better, and this effect is most significant when the number of unitary operations is below three. This may also be caused by the relatively few feature elements of the graph representation vector of the DHC-E-processed MUTAG data. The MUATG dataset has only 4 features of each representation vector, while the PTC dataset has 12.

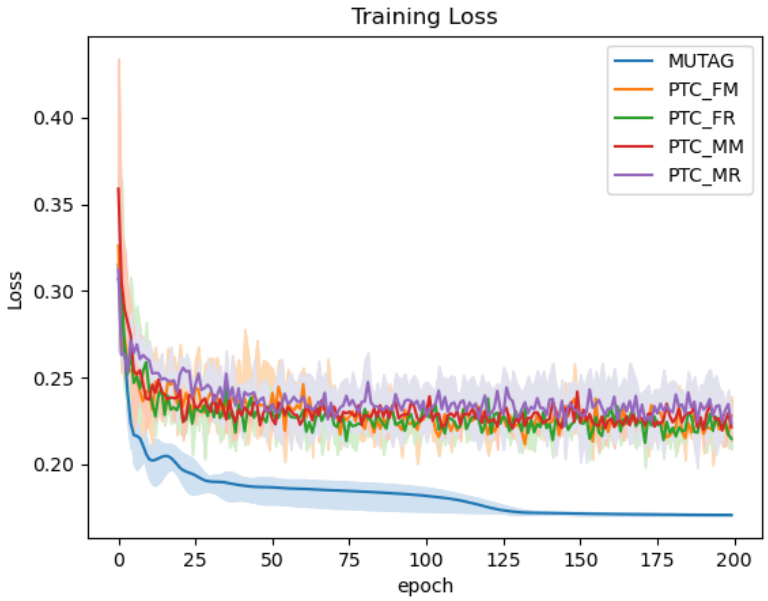

Figure. 3. Training loss of sQGNN model in ideal quantum environment.

TABLE 3. The average accuracy of the models with depolarizing error.

DEPOLARIZATION PROBABILITY	MUTAG	PTC_FM	PTC_FR	PTC_MM	PTC_MR
0.001	87.3±3.3	65.0±7.2	66.5±9.1	65.3±42	65.8±4.4
0.01	86.7±4.9	65.1±8.6	66.1±5.6	65.7±4.5	65.8±5.1
0.1	84.8±6.4	64.0±7.7	65.0±8.4	63.7±4.8	64.5±4.5

From the comparison of the results in the ideal quantum environment, the overall accuracy of our sQGNN model is superior. In addition to accuracy, the advantage of the sQGNN method lies in the compact model structure. Classical graph neural networks are usually complex in terms of the model structure due to the complexity of graph data itself. Due to the graph embedding method without hyperparameters in our model and the use of Single-Qubits QGNN quantum circuits, our model achieves a huge advantage in the total number of parameters. Taking the MUTAG dataset as an example, our model requires only 8 parameters, while a classical neural network such as GIN has 13,000 parameters as shown in Table 2, which fully shows the potential advantage of our model in efficiency.

Figure 3 shows the training loss of sQGNN on five datasets in an ideal quantum environment. Relatively speaking, the PTC series dataset is more difficult to train than the MUTAG dataset because the PTC dataset is more complex. In the PTC dataset, PTC_MM and PTC_MR are relatively difficult to train. This can also be confirmed by the accuracy results. The overall accuracy of these two datasets for various models is relatively low.

To verify the ability of the sQGNN method to resist the effects of noise when running on NISQ devices with noise, we train and test the models using a noisy simulated environment. As shown in Table 3, we set the depolarization error as the noise in the environment for the sQGNN model. When a group of qubits undergoes a depolarization error, a random Pauli (example: X, Y, Z) is applied to each qubit. The formula for depolarization error is as follows:

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

Where ε is the depolarizing error param, n is the number of qubits for the error channel and I is the Pauli matrix.

We set 3 levels for the depolarization parameter to test how tolerant sQGNN is to depolarization errors. The test results are shown in Table 3. Under the influence of depolarization error, the sQGNN model is slightly affected, almost unaffected at 0.001 and 0.01, and slightly affected at 0.1. In terms of average accuracy, our sQGNN method shows good error tolerance, thanks to its unique single qubit

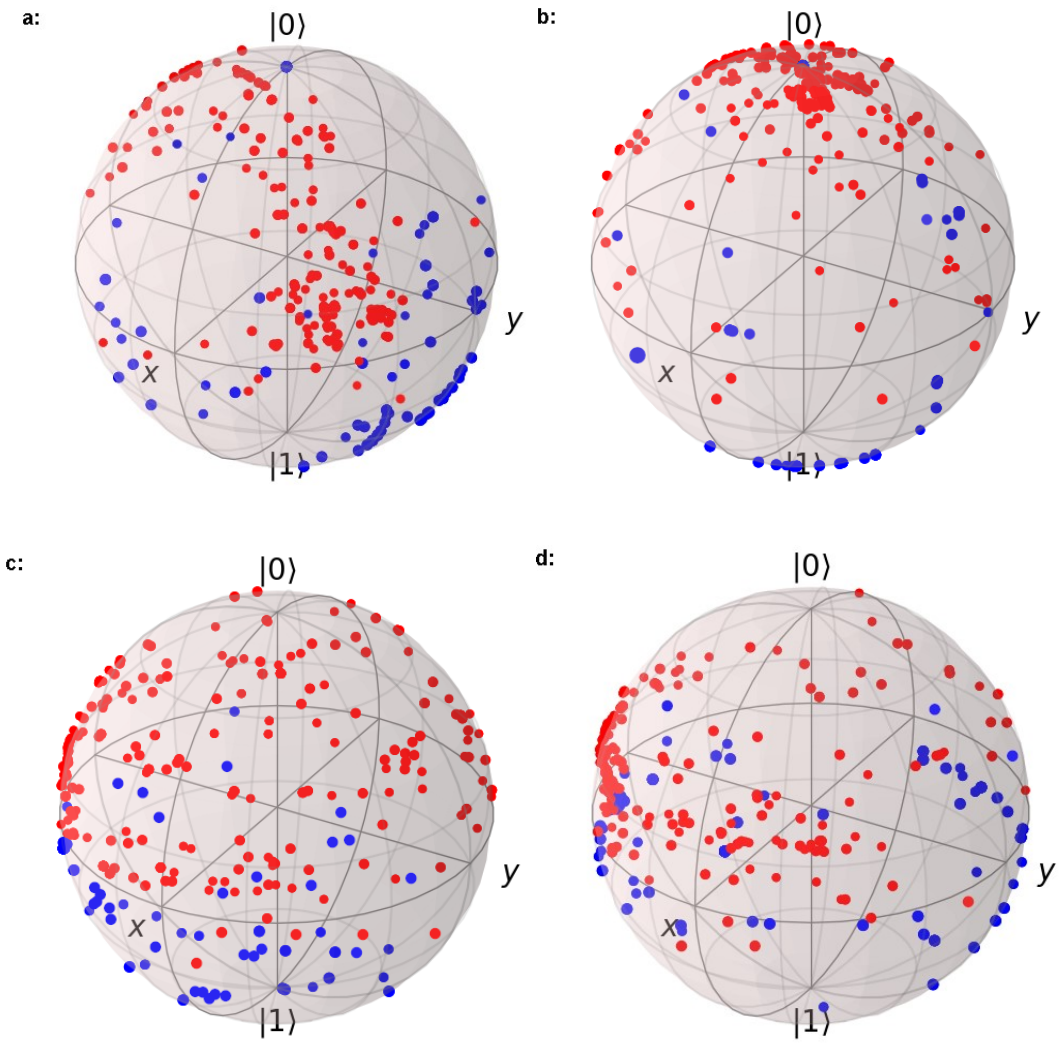

Figure. 4. This figure is a demonstration of all graph data of the PTC_FM dataset in Hilbert space. The red dot is class 0 and the blue dot is class 1. In the figure a, b, c, d are PTC_FM, PTC_FR, PTC_MM, PTC_MR respectively.

implementation. Due to the functional limitations of quantum devices in the NISQ era, the influence of noises is unavoidable, and the resulting degradation is a great problem when coming to real quantum computers. From the results, the sQGNN model can stably maintain good performance under the noise impact.

Figure 4 also shows the distribution of all graphs in the PTC_FM dataset in Hilbert space that are displayed on the surface of Bloch spheres. Among them, the red points belong to Class 0, and the blue points are from Class 1. Overall, Class 0 is concentrated in the upper hemisphere, and Class 1 is concentrated in the lower hemisphere. It can be observed from the figure that the classification ability of

the model for Class 0 is stronger than that of Class 1, and Class 1 has some outlier data distributed in the upper hemisphere, which may be caused by the initial state of the quantum circuit being set to $|0\rangle$. The structure of graph data may also be a cause. Overall, our model can still maintain a good performance under the influence of noise. According to our experimental results, the training difficulty of PTC_MM and PTC_MR is relatively high, which can be confirmed by the accuracy and data distribution.

Real Quantum Device Test Experiment

In addition to the above experiments in the simulated quantum environments, we further designed the experiment to run our sQGNN models on the real quantum computer offered by IBM. We deploy the best-performing model on the MUTAG dataset from previous experiments in a simulated quantum environment to a real quantum computer. Thanks to the online IBM's quantum computer [33] and IBM Quantum Lab, we were able to deploy our single qubit models easily on a quantum computer provided by IBM in relatively simple steps. Currently, IBM online service allows 4 qubits each user, restricted due to the limited resources on their NISQ quantum computer. In addition, the quantum volume of a quantum computer is limited, which limits the total amount of quantum gates that can be deployed in a quantum circuit, and our method of combining parameters and encoding greatly reduces the required quantum gates.

Figure 5 shows the test results of our model on the real quantum computer. Our model achieved an accuracy rate of 88.89% on the MUTAG dataset on the real IBM quantum computer, with no degradation in accuracy compared to the simulated quantum environment. As shown in Figure 5a, the classification performance of the model on quantum computers is the same as in the simulation environment. Figure 5b shows the measured results on the quantum computer. The number of results obtained in the measurements of $|0\rangle$ and $|1\rangle$ is similar, that is, Class 0 is better than Class 1. The reasons for this phenomenon are the same as in the simulation environment, namely the initial state is $|0\rangle$ and the difference in the graph data structure of the two classes.

198 Figure 5c shows the resulting visual image of the Wigner quasi-probabilistic equations [34]. This figure
199 relates the wave function to a probability distribution in phase space. We can see that the model clearly
200 distinguishes between the two classes and is less affected by NISQ device noise. Mostly the initial state of
201 available quantum computers is $|0\rangle$. If the initial state is set to $|1\rangle$, the result may be the opposite. Due to
202 the limitations of the physical system itself, the initial state of $|0\rangle$ may be the normal state of quantum
203 computers for a period of time, and therefore the algorithm needs to be optimized for this problem.
204 Although the phenomenon in the simulated environment still exists, it also illustrates the robustness of
205 our model. The performance of our sQGNN model on quantum computers does not degrade and
206 maintains high accuracy and interpretability in the simulated environments. The experience validates the
207 real-world feasibility of our model with the huge benefits of our single-qubit approach.

208 ## Discussion

209 In the previous section, we presented our experimental results and analyzed them in details. In general,
210 considering the research purpose and experimental environment, our work results are good and can
211 provide some valuable reference methods for the development of related fields. We use the single-qubit
212 method to construct quantum circuits and realize the efficient use of finite qubits to encode graph
213 representation vectors. In the absence of quantum computing resources in the NISQ era, our method can
214 save a lot of quantum resources. Theoretically, we only need a single qubit to implement any neural
215 networks with thousands or even millions of parameters.

216 In the NISQ era, the noise of quantum devices is an unavoidable problem. The existence of noise makes
217 some methods that perform well in ideal simulation environments but difficult to put into practical use in
218 NISQ devices. The experimental results show that our single-qubit models achieved good performance in
219 our tests and exhibited a good anti-noise ability in the simulation environment and quantum computers, as
220 shown in Table 1 and 3. Against the depolarizing error (Table 3), our method shows steady performance
221 over different levels of noises.

Due to the efficient use of the sQGNN method for a single qubit, we can keep the total number of qubits to a very low level, and thus avoid the training difficulties on the "barren plateau" issues widely existing in VQCs. Although some studies [23, 40 & 41] have proposed some methods to alleviate the "barren plateau", this problem has not been solved fundamentally. In the current state of quantum machine learning methods applying VQA, the "barren plateau" may accompany quantum machine learning for a while. However, our method provides a fundamental way to avoid it at all.

In our work, we explored a hyperparameter-free graph embedding method, which greatly reduces the required computing resources in the graph embedding stage. The total number of parameters of our quantum neural networks can be several orders of magnitude smaller than classical methods, as shown in Table 2. This makes our model much more efficient. Moreover, our quantum circuit structure can be combined with other quantum or classical models to suit different tasks. In this paper, the sQGNN is combined with the preprocessing steps on graph data using the DHC algorithm.

We deployed our models to IBM's quantum computer to validate, and the test results confirmed there is no degradation on its performance from the simulated ones. Our single-qubit model uses only one qubit and its quantum circuits can be easily deployed in quantum computers as they require the least number for qubits, enabling the practical implementation on real variational quantum circuits with limited resources. Predictably, our method will enable a wide range of practical use towards various real world data, besides the biological molecular datasets in this paper.

At this stage, learning and reasoning on quantum simulators are computationally expensive processes compared to classical neural network models. Although the purpose of quantum machine learning is to utilize quantum mechanisms to improve the performance of neural networks, the limitation of physical conditions still exists as the simulation of multiple qubits are compute-intensive. However, the sQGNN method provides an efficient option for using fewer qubits in the NISQ era, and the sQGNN method can maximize the use of the finite qubits on VQCs and provides a novel practical way towards the practical application of quantum neural networks in the NISQ era. With its simple structure and high utilization,

the single-qubit method can be applied to more challenging tasks and bring out a revolution towards the real world applications of NISQ devices.

Conclusion

Graph learning has been extensively studied in the field of deep learning, but little work has been done exploring it in the context of quantum machine learning. Due to limited quantum resources as well as some shortcomings of VQA, quantum machine learning has difficulties in generalizing to more complex data types, and our single-qubit method can help to solve this problem. In this work, we proposed a novel quantum graph classification method, namely sQGNN, which leverages a hyperparameter-free graph embedding method to minimize the size of quantum graph neural networks. We conduct multiple experiments on different datasets to explore various experimental results towards in-depth analysis. Experimental results show that our model achieved excellent capability in the classification of biological molecular structures, while the number of parameters is several orders of magnitude lower than that of classical graph neural networks. We also shows our model achieved robust performance on real quantum computers. Our concise model structure enables the model to be deployed on quantum devices in the NISQ era and is easy to use, and has a good ability to resist noises. Our work clearly shows that our new work enlightens a pathway to implement many very complicated neural networks on a single qubit, and offered a panacea solution to the bottleneck of realizing various machine learning algorithms over resource-limited VQC devices in the NISQ era.

Methods

In this paper, we develop a novel quantum-classical hybrid machine learning algorithm for small-molecule compound graph structure data. The idea is to replace Euclidean space-based transformation matrices in classical deep learning with unitary matrices for quantum computing and implement the quantum neural networks with the least number of parameters over a single qubit. In this way, we incorporate the theoretical ideas of quantum machine learning into deep learning in the graph domain.

Graph Embedding

DHC-E is a simple and hyperparameter-free graph embedding method that combines the generalized DHC theorem and Shannon entropy (E) for network compression and representation. The DHC theorem [35] reveals the relationship between degree, H-index, and centrality: the H-index of a node is computed according to the degree centrality of its neighbours, and ~~its H-index can be calculated according to its T~~the previous H-index of the neighbour is updated. A node's H-index (short for Hirsch index) [36] is the maximum value h such that it has at least h neighbours with a degree no less than h [35]. A node's coreness further takes its location in the graph into account, measuring its influence based on the k-core decomposition process [37], where a larger coreness indicates that a node locates more centrally in the graph. Such an update process continues iteratively, yielding a sequence of H-indexes of nodes. DHC-E utilizes H-index sequences for graph embedding. DHC-E first calculates the probability distribution of the node's iterative H-index and then calculates its Shannon entropy. Shannon entropy's formula is:

$$H = -\sum_{i=1}^{n} p_i \, log_2 \, p_i \tag{2}$$

Since the iterations of the above DHC update process on different graphs are usually dimensionally unequal, the generated whole graph embeddings are not compatible for some downstream machine learning tasks (graph classification). To align them, DHC-E further converts all generated graphs into the maximum dimensions of all embeddings as the baseline, and any other elementa with fewer dimensions will be supplemented with its last element to the max ones.

Data Encoding

For many ML tasks, data is often presented as column vectors. Traditionally, this D-dimensional vector of classical data can be encoded either by initializing the $2D$ qubit quantum states to their binary string equivalents (basic encoding), or by transforming the data dimensions into their corresponding superposition state probability magnitudes (amplitude encoding). Although these data encoding schemes have been used in other work [3], their implementation is often very expensive or impractical and

susceptible to error-prone quantum operations. Therefore, these encoding schemes may not always be an efficient means of minimizing qubit usage. Single-qubit encoding, developed in [24], is a strategy to encode a vector of classical data into a characteristic Hilbert space using a series of single operations acting on each input data dimension, using only one single qubit. The unitary operation of single-qubit encoding can be expressed by the following formula:

$$U = e^{i\alpha} R_Z(\beta) R_Y(\gamma) R_Z(\delta) \tag{3}$$

With a global phase factor α, Euler angles $\beta, \gamma, \delta, \in \mathbb{R}$ that define the extent of each rotation (R) around the Z, Y and Z axes respectively. Within this method of encoding, these Euler angles are parameterized further and defined as:

$$\beta = \theta_i + x_i \cdot \phi_i \tag{4}$$
$$\gamma = \theta_{i+1} + x_{i+1} \cdot \phi_{i+1} \tag{5}$$
$$\delta = \theta_{i+2} + x_{i+2} \cdot \phi_{i+2} \tag{6}$$

Where θi and ϕi are trainable weight parameters assigned to xi, the value of the input vector x at dimension i. Therefore, the extent of rotation β, γ, δ is with respect to the weighted value of the input. Using the above parameter definitions, a maximum of three input dimensions can be encoded per unitary operation applied. Here, the input vector will be continually cycled through, encoding a series of three-dimensional values at a time, until the entirety of the input has been encoded. This is known as a full 'upload layer' of the input data. The single-qubit encoding method can be flexibly deployed on quantum circuits that process graph data of different structures and can increase the amount of data that each qubit can carry. The graph data processed by DHC-E is a high-dimensional vector, and the number of qubits required for basic encoding and amplitude encoding is large. Using single-qubit encoding, the number of qubits in the quantum circuits can be easily reduced to less than 10, thereby avoiding the "barren plateau" problem [23] in VQA training.

Quantum Circuit Design

Quantum circuits are the core part of quantum computing, and most functions are realized by the combination of qubits and quantum gates in quantum circuits. In our single-qubit approach, we limit the number of qubits to one. The single-qubit method can theoretically use quantum circuits with multiple qubits. In this case, CNOT gates are used to create quantum entanglement between qubits, and each qubit is equivalent to a hidden layer of a neural network. We stick to using one qubit in this model in order to better demonstrate the efficiency of the single-qubit approach. When the graph data processed by the DHC-E method is embedded in the quantum circuit according to the above method, we treat each combination of RZ, RY and RZ gates as a unitary operation. Taking this unitary operation as the basic unit, how many unitary operations are set on a unique qubit is determined according to the size of the input data. In our method, each unitary operation is able to encode 3 inputs with 6 parameters. Taking data of size 8 as an example, in order for the data to be encoded into a quantum circuit, 3 unitary operations will be set on a single qubit. In this case, after the encoding of the graph data is complete, there is still a gap. At this time, this vacancy will be replaced by 0 to ensure that this vacancy will not affect the quantum state in the final measurement. In this way, the single-bit method can be scaled according to the size of the input data and can be widely used.

Measurement

In the measurement part, we will observe the quantum circuit to obtain the final state of the circuit. Unlike the general direct measurement of the state of the qubit, for application to classification tasks, we adopt a fidelity-based measurement method. The overall goal of this measurement method is to minimize the fidelity between a set of data encodings and their respective target states. For binary classification tasks, given a set size D of images with corresponding class values in $\{0,1\}$, assign each image a corresponding target state $|0\rangle$ or $|1\rangle$. Any number of classes can be merged using this method, provided that the target states are at the greatest distance from each other. Fidelity F is a measure of the similarity or proximity between two quantum states, where $0 \leq F \leq 1$. The higher the fidelity of two quantum states, the more

similar they are in direction. The highest class fidelity value given is then considered the classification result. For example, for a graph with a label of 0, its corresponding target state is $|0\rangle$, when we measure, we get the result $\{\alpha|0\rangle, \beta|1\rangle\}$ where $\alpha + \beta = 1$ and α is the fidelity of class 0. The formula of the fidelity is shown as follow:

$$F(\vec{x}, \vec{\theta}, \vec{\phi}) = \left|\langle \vec{\psi}_l | \psi_{output}(\vec{x}_i, \vec{\theta}, \vec{\phi})\rangle\right|^2 \quad (7)$$

Loss calculation and model training

The role of the loss function is to describe the gap between the predicted value of the model and the true value. The loss function can find a criterion to help the training mechanism optimize the parameters at any time in order to find the parameters at the highest accuracy of the network. From the measurement section, we use fidelity to measure the similarity between results and labels. Therefore, the loss function of our model is also designed based on fidelity. The formula of the loss function is as follows:

$$\mathcal{L}(\vec{x}, \vec{\theta}, \vec{\phi}) = \sum_{i=1}^{M} \left(1 - \left|\langle \vec{\psi}_l | \psi_{output}(\vec{x}_i, \vec{\theta}, \vec{\phi})\rangle\right|^2\right) \quad (8)$$

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

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 and ϕ in the unitary operation above. With the optimization of parameters, the operation of the quantum revolving gate on qubits is also optimized, so that the entire model can be trained.

Quantum Barren Plateau

A typical gradient descent algorithm of quantum circuits is shown:

$$\theta(t+1) - \theta(t) \equiv \delta\theta_\mu = -\eta \frac{\partial \mathcal{L}}{\partial \theta} \quad (9)$$

Where $\mathcal{L}$ is the loss function and θ_μ is the variational angle of quantum circuits.

For large variational quantum circuits, the gradient of the objective function has an average value of zero, and the probability that any given instance of such a quantum circuit deviates from this average value by a small constant decreases exponentially with the number of qubits [23]. When the measure of the space is concentrated in this way, the value of any reasonably smooth function tends to its mean with exponential probability, which means that the gradient is zero over a wide range of quantum space, like the situation in Figure 6. This is the plateau of quantum barrenness (Barren plateaus in quantum neural network training landscapes).

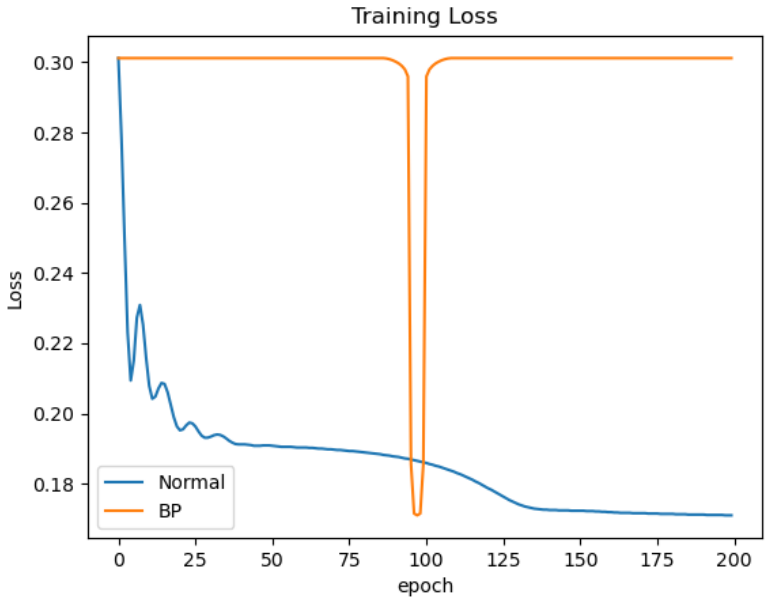

Figure. 6. Schematic diagram of a barren plateau.

In VQA, the expected value of the quantum circuit needs to be obtained. Consider an objective function E(θ) expressed as the expectation value over some Hermitian operator H:

$$E(\vec{\theta}) = \langle 0|U(\vec{\theta})^{\dagger} H U(\vec{\theta})|0\rangle \tag{10}$$

The gradient of the objective function:

$$\partial_k E \equiv \frac{\partial E(\vec{\theta})}{\partial \theta_k} = i\langle 0|U_{-}^{\dagger}\left[V_k, U_{+}^{\dagger} H U_{+}\right]U_{-}|0\rangle \tag{11}$$

V is a a Hermitian operator. U_+ and U_- are two circuits, both match the Haar distribution up to the second moment, and the circuits are independent. The average value of the gradient can then be expressed as:

$$\langle \partial_k E \rangle = \int dU p(U)\, \partial_k \langle 0|U(\vec{\theta})^\dagger H U(\vec{\theta})|0\rangle \tag{12}$$

Where $p(U)$ is the probability distribution function of U. In different cases, its variance is:

$$\mathrm{Var}[\partial_k E] \approx \begin{cases} -\dfrac{\mathrm{Tr}(\rho^2)}{(2^{2n}-1)} \mathrm{Tr}\langle [V, u^\dagger H u]^2 \rangle_{U_+} \\ -\dfrac{\mathrm{Tr}(H^2)}{(2^{2n}-1)} \mathrm{Tr}\langle [V, u^\dagger \rho u]^2 \rangle_{U_-} \\ \dfrac{1}{2^{(3n-1)}} \mathrm{Tr}(H^2)\mathrm{Tr}(\rho^2)\mathrm{Tr}(V^2) \end{cases} \tag{13}$$

Among them, the number of qubits is n. This means that when the number of qubits is large, in most cases, the gradient of the cost function approaches 0, that is, any training method based on VQA will not be able to make the cost function converge.

Experimental Setup

We choose MUTAG dataset and PTC series dataset as experimental data. The MUTAG dataset contains 188 nitro compounds. The labels are used to determine whether the compound is aromatic or heteroaromatic. The graph data belongs to the isomer graph. The full name of PTC is Predictive Toxicology Challenge, which is used to develop advanced SAR technology predictive toxicology models. This dataset contains carcinogenicity-labelled compounds in rodents. According to the experimental rodent species, there are a total of 4 datasets: PTC_FM (female mice), PTC_FR (female rats), PTC_MM (male mice), and PTC_MR (male rats). The specific parameters of the dataset are shown in the table below.

In this experiment, the experimental flow of our method is as follows: the graph data is embedded using the DHC-E method, and then the graph data is encoded into a single-qubit circuit. According to the size of the embedded graph data, a corresponding number of unitary operations will be set on the qubits. In the measurement part, the expected values of the target states $|0\rangle$ and $|1\rangle$ are measured to calculate the fidelity of the corresponding data. The training of the neural network is achieved by evaluating the effect of the above loss function and updating the parameters at the quantum revolving gate. The model with the best parameters is obtained after many iterations and evaluated with test data.

We use PennyLane [38] and PyTorch [39] to perform experiments. PennyLane is an open-source python-based framework that enables automatic differentiation of hybrid quantum-classical computations. It is compatible with mainstream machine learning frameworks like PyTorch and has a huge plugin ecosystem. The quantum computer we use is the ibm_manila node provided by IBM [33]. The number of qubits available to this quantum computer is 5, the Quantum volume is 32, it can perform 2800 Circuit layer operations per second, and the processor model is Falcon r5.11L.

References

1. Min, S., Lee, B., & Yoon, S. Deep learning in bioinformatics. *Briefings in bioinformatics*, **18**(5), 851-869 (2017).

2. Gligorijević, V., et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, **12**(1), 1-14 (2021).

3. Schuld, M., Sinayskiy, I., & Petruccione, F. An introduction to quantum machine learning. *Contemporary Physics*, **56**(2), 172-185 (2015).

4. Shor, P. W. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science* (pp. 124-134). IEEE (1994).

5. Lloyd, S. Universal quantum simulators. *Science*, **273**(5278), 1073-1078 (1996).

6. Harrow, A. W., Hassidim, A., & Lloyd, S. Quantum algorithm for linear systems of equations. *Physical review letters*, **103**(15), 150502 (2009).

7. Huang, H. Y., et al. Power of data in quantum machine learning. *Nature communications*, **12**(1), 1-9 (2021).

8. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum*, **2**, 79 (2018).

9. Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B., & Melko, R. Quantum boltzmann machine. *Physical Review X*, **8**(2), 021050 (2018).

10. Lloyd, S., Mohseni, M., & Rebentrost, P. Quantum principal component analysis. *Nature Physics*, **10**(9), 631-633 (2014).

11. Rebentrost, P., Mohseni, M., & Lloyd, S. Quantum support vector machine for big data classification. *Physical review letters*, **113**(13), 130503 (2014).

12. Havlíček, V., et al. Supervised learning with quantum-enhanced feature spaces. *Nature*, **567**(7747), 209-212 (2019).

13. Kak, S. C. Quantum neural computing. *Advances in imaging and electron physics*, **94**, 259-313 (1995).

14. Kapoor, A., Wiebe, N., & Svore, K. Quantum perceptron models. *Advances in neural information processing systems*, 29 (2016).

15. Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R., & Tkatchenko, A. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, **8**(1), 1-8 (2017).

16. Cong, I., Choi, S., & Lukin, M. D. Quantum convolutional neural networks. *Nature Physics*, **15**(12), 1273-1278 (2019).

17. Xia, F., et al. Graph learning: A survey. *IEEE Transactions on Artificial Intelligence*, **2**(2), 109-127 (2021).

18. Caetano, T. S., McAuley, J. J., Cheng, L., Le, Q. V., & Smola, A. J. Learning graph matching. *IEEE transactions on pattern analysis and machine intelligence*, **31**(6), 1048-1058 (2009).

19. Schuld M. Machine learning in quantum spaces. *Nature*, **567**(7747), 179–181 (2019).

20. Clerk, A. A., et al. Introduction to quantum noise, measurement, and amplification. *Reviews of Modern Physics*, **82**(2), 1155 (2010).

21. Preskill, J. Quantum computing and the entanglement frontier. Preprint at https://doi.org/10.48550/arXiv.1203.5813 (2012).

22. Cerezo, M., et al. Variational quantum algorithms. *Nature Reviews Physics*, **3**(9), 625-644 (2021).

23. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nature communications*, **9**(1), 1-6 (2018).

24. Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., & Latorre, J. I. Data re-uploading for a universal quantum classifier. *Quantum*, **4**, 226 (2020).

25. Easom-McCaldin, P., Bouridane, A., Belatreche, A., Jiang, R., & Al-Maadeed, S. Efficient Quantum Image Classification Using Single Qubit Encoding. IEEE Transactions on Neural Networks and Learning Systems (2022).

26. Wang, H., Deng, Y., Lü, L., & Chen, G. Hyperparameter-free and explainable whole graph embedding. Preprint at https://doi.org/10.48550/arXiv.2108.02113 (2021).

27. Jaume, G., Nguyen, A. P., Martínez, M. R., Thiran, J. P., & Gabrani, M. edGNN: a Simple and Powerful GNN for Directed Labeled Graphs. Preprint at https://doi.org/10.48550/arXiv.1904.08745 (2019).

28. Schlichtkrull, M., et al. Modeling relational data with graph convolutional networks. In *European semantic web conference*. Springer, Cham. 593-607 (2018).

29. Xu, K., Hu, W., Leskovec, J., & Jegelka, S. How powerful are graph neural networks?. Preprint at https://doi.org/10.48550/arXiv.1810.00826 (2018).

30. Helma, C., King, R. D., Kramer, S., & Srinivasan, A. The predictive toxicology challenge 2000–2001. *Bioinformatics*, **17**(1), 107-108 (2001).

31. Debnath, A. K., Lopez de Compadre, R. L., Debnath, G., Shusterman, A. J., & Hansch, C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, **34**(2), 786-797 (1991).

32. Easom-Mccaldin, P., Bouridane, A., Belatreche, A., & Jiang, R. On Depth, Robustness and Performance Using the Data Re-Uploading Single-Qubit Classifier. *IEEE Access*, **9**, 65127-65139 (2021).

33. IBM Quantum. estimator primitive (ibm_manila) [computer software]. https://quantum-computing.ibm.com/ (2022).

34. Hiley, B. J. Phase space descriptions of quantum phenomena. In Proc. Int. Conf. Quantum Theory: Reconsideration of Foundations, **2**, 267-86 (2004).

35. Lü, L., Zhou, T., Zhang, Q. M., & Stanley, H. E. The H-index of a network node and its relation to degree and coreness. *Nature communications*, **7**(1), 1-7 (2016).

36. Hirsch, J. E. An index to quantify an individual's scientific research output. *Proceedings of the National academy of Sciences*, **102**(46), 16569-16572 (2005).

37. Dorogovtsev, S. N., Goltsev, A. V., & Mendes, J. F. F. K-core organization of complex networks. *Physical review letters*, **96**(4), 040601 (2006).

38. Bergholm, V., et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. Preprint at https://doi.org/10.48550/arXiv.1811.04968 (2018).

39. Paszke, A., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, **32** (2019).

40. Grant, E., et al. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, **3**, 214 (2019).

41. Stokes, J., et al. Quantum natural gradient. *Quantum*, **4**, 269 (2020).

Acknowledgements (optional)

Keep acknowledgements brief and do not include thanks to anonymous referees or editors, or effusive comments. Grant or contribution numbers may be acknowledged.

Ethics declarations

Competing interests

Submission of a competing interests statement is required for all content of the journal.

Supplementary Information

The basic concept and architecture of the Single-Qubit method were first proposed in "Data re-uploading for a universal quantum classifier". They propose a hybrid classical-quantum algorithm based on the angle of reuploading classical data to a single qubit multiple times along a circuit. Along with the data points, other parameters are introduced into the circuit and tuned by classically minimizing the cost function. The quantum classifier structure in this paper is shown in the formula:

$$\tilde{U}(\vec{\phi},\vec{x}) = U(\vec{\phi}_N)U(\vec{x})\ldots U(\vec{\phi}_1)U(\vec{x}) \tag{1}$$

which acts as:

$$|\psi\rangle = \tilde{U}(\vec{\phi},\vec{x})|0\rangle \tag{2}$$

where ψ is the final state, U is the unitary operation, ϕ is the parameter, and x is the input classical data. The final classification of the pattern will come from the measurement of $|\psi\rangle$. We can introduce the concept of processing layers as compositions to combine unitary operations:

$$L(i) = U(\vec{\phi})U(\vec{x}) \tag{3}$$

so the formula of the whole quantum classifier can be:

$$\tilde{U}(\vec{\phi},\vec{x}) = L(N)\ldots L(1) \tag{4}$$

The depth of the quantum circuit is 2N. The higher the number of layers, the stronger the representation capability of the quantum circuit and the more powerful the classifier will be, analogous to the hidden layers of a classical neural network.

In this paper, we take the approach of combining parameters and input data into the same unitary operation with the following formula:

$$L(i) = U(\vec{\theta}_i + \vec{\omega}_i \circ \vec{x}) \tag{5}$$

where θ and ω are parameters. This method is more efficient for quantum gates, especially on real quantum computers with limited quantum volume.

Depth settings for single-qubit methods

The single-qubit circuit we apply sets the dimension of each unitary operation to three, that is, each unitary unit consists of three quantum revolving gates that encode three data. As the depth of the quantum circuit increases, that is, the processing layers increase, the dimension of the input data also increases accordingly. According to the experimental results of "On Depth, Robustness and Performance Using the Data Re-Uploading Single-Qubit Classifier", the deeper the quantum circuit depth, usually improves the average classification performance, and the largest improvement usually occurs in layers 1, 2 and 3. between depth increments. A clear advantage of increasing depth is the ability to produce more complex data maps by visualizing the difference between depth increments. Likewise, higher depths consistently exhibit higher stability as depth increases. Any advantage in robustness during model training appears to level off as depth increases.