Quantum Generative Model with Variable-Depth Circuit

Yiming Huang[1, *], Hang Lei[1], Xiaoyu Li[1, *], Qingsheng Zhu[2], Wanghao Ren[3] and Xusheng Liu[2, 4]

Abstract: In recent years, an increasing number of studies about quantum machine learning not only provide powerful tools for quantum chemistry and quantum physics but also improve the classical learning algorithm. The hybrid quantum-classical framework, which is constructed by a variational quantum circuit (VQC) and an optimizer, plays a key role in the latest quantum machine learning studies. Nevertheless, in these hybrid-framework-based quantum machine learning models, the VQC is mainly constructed with a fixed structure and this structure causes inflexibility problems. There are also few studies focused on comparing the performance of quantum generative models with different loss functions. In this study, we address the inflexibility problem by adopting the variable-depth VQC model to automatically change the structure of the quantum circuit according to the qBAS score. The basic idea behind the variable-depth VQC is to consider the depth of the quantum circuit as a parameter during the training. Meanwhile, we compared the performance of the variable-depth VQC model based on four widely used statistical distances set as the loss functions, including Kullback-Leibler divergence (KL-divergence), Jensen-Shannon divergence (JS-divergence), total variation distance, and maximum mean discrepancy. Our numerical experiment shows a promising result that the variable-depth VQC model works better than the original VQC in the generative learning tasks.

Keywords: Machine learning, quantum information processing, generative model.

1 Introduction

Machine learning techniques have successfully been used in many areas. Not only do they flourish in fields such as computer vision, natural language processing and so on, but they also tremendously boost the other scientific research areas such as quantum physics [Broecker, Carrasquilla, Melko et al. (2017); Qu, Wu, Wang et al. (2017)] and chemistry

[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China.

[2] School of Physics, University of Electronic Science and Technology of China, Chengdu, 610054, China.

[3] School of Information Science and Engineering, University of Jinan, Jinan, 250022, China.

[4] Department of Chemistry and Biochemistry, Utah State University, Logan, 84322, USA.

[*] Corresponding Author: Xiaoyu Li. Email: xiaoyuuestc@uestc.edu.cn.

Received: 01 March 2020; Accepted: 20 May 2020.

[Ramakrishnan, Dral, Rupp et al. (2015)] among others. As quantum computing techniques inherently provide speed-ups in some specific scenario [Harrow, Hassidim and Lloyd (2009)], like in principal component analysis [Chen, Xia, Wang et al. (2018)]. That is the reason why the researchers pay more attention to harness the power of quantum computing to enhance machine learning techniques. However, building a large scale and reliable quantum computer that can be perfectly isolated from the outside is a tremendous challenge. Preskill [Preskill (2018)] predicts that this kind of quantum computer is not likely to be invented in the near future. Fortunately, the intermediate scale quantum computer called NISQ, i.e., Noisy Intermediate-Scale Quantum computer, which can perform hybrid quantum-classical optimization will be released in the next few years [Havlíček, Córcoles, Temme et al. (2019)]. It makes it possible to combine quantum computing with machine learning.

One of the valuable applications on NISQ is the hybrid quantum-classical computing framework. This framework includes two main components, variable quantum circuit and a classical optimizer. The parameterized quantum circuit, the variable quantum circuit, is used to deal with the initial quantum state, which is normally an encoded classical input data. To extract the classical information from the processed quantum state, we perform measurements on all quantum registers. After getting the expectation value of the measurements, we feed them as the input into the classical optimizer. The classical optimizer receives this data and guides the VQC to update the parameters. The whole process of minimizing the loss function is slightly changing the parameters of the VQC which are generated by the optimizer, according to various optimization methods such as a gradient descent. By iteratively running the above optimization steps until convergence, we will get the results that approximate the solution of the target problem. The whole structure of the hybrid quantum-classical framework is shown in Fig. 1. In 2014, Farhi et al. [Farhi, Goldstone and Gutmann (2014)] first proposed this hybrid computing idea, known as the quantum approximate optimization algorithm (QAOA), to solve the combinatorial optimization problem. Various studies based on this idea have successively been developed. Most of these works take advantage of hybrid quantum-classical techniques to solve physics problems, including preparing quantum states and finding low-energy states of a given Hamiltonian, such as in Peruzzo et al. [Peruzzo, McClean, Shadbolt et al. (2014)]. The latter proposed a Variational Quantum Eigensolver (VQE) to find the ground state corresponding to the lowest molecular energy of He-H+.

Recently, researchers have also been devoted to enhancing the traditional generative learning methods. Many of the works showing the hybrid quantum-classical based algorithm on NISQ device, may have the potential power for learning tasks, that not being the case for their classical counterparts. For instance, Benedetti et al. [Benedetti, Realpe-Gómez, Biswas et al. (2017); Liu and Wang (2018)] proposed a quantum born machine to learn target probability distribution.

Nonetheless, these are few studies that adopt the variable-depth circuit structure, such as the method proposed for quantum circuit compiling [Khatri, LaRose, Poremba et al. (2019)]. Comparing to other fixed Ansatz models, we employ a more flexible way to build VQC in the hybrid computing framework, and apply it to tackle the generative learning task. Meanwhile, we concentrate on analyzing the performances between the quantum generative models which are based on the variable-depth VQC with different loss functions. We hope that what we found could provide some practical advice for related research, such as in the work of quantum communication [Qu, Li, Xu et al. (2019)].

2 Preliminaries

In this section, it contains some basic concepts of quantum information and computing in our quantum generative model.

2.1 Qubits and quantum states

The qubit is the basic information carrier in quantum information processing. Instead of a classical bit, a qubit has two possible states, including $|0\rangle$ and $|1\rangle$. There are two types of quantum states, including pure quantum states and mixed quantum states. A pure quantum state is represented by a unit complex vector $|\phi\rangle$, where $|\phi\rangle = \sum_i v_i |i\rangle$, $\sum_i |v_i|^2 = 1$. The v_i can be regarded as the coefficients of the linear combination of the computational basis $|i\rangle$. For example, one qubit pure quantum state is in a superposition of the computational basis and represented by a 2-dimensional complex vector $|\varphi\rangle$ in the Hilbert Space $\mathbb{C}^2$, where $|\varphi\rangle = a|0\rangle + b|1\rangle$ and a,b are known as the probability amplitudes, such that $|a|^2 + |b|^2 = 1$. Similarly, an n-qubits quantum system $|\varphi_1 ... \varphi_n\rangle$ can be written as a tensor product of the subsystem $|\varphi_i\rangle$ in 2^n dimensional Hilbert space. A mixed quantum state is an ensemble a mixture of the pure states $|\psi_i\rangle$ with respective probability p_i, such that $\sum_i p_i$=1. We usually describe a mixed state by a density matrix ρ which is in the form of $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, $\rho \succeq 0, Tr[\rho] = 1$.

2.2 Quantum gates

Any operations applied to a quantum state should be reversible and must preserve the ℓ_2 norm. Thus, a quantum gate can be described by a unitary operator, which is a bounded linear operator on a Hilbert space that satisfies $U^\dagger U = UU^\dagger = I$, where $U^\dagger$ is the conjugate transpose of U. A quantum gate for the n-qubits system can be formulated by a unitary matrix in $\mathbb{C}^{2n}$. The widely used single qubit quantum gates are listed as follows.

- Pauli gates

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{1}$$

- Hardmard gates

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2}$$

- Rotation gates

$$R_x(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, R_y(\theta) = \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix}, R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{3}$$

- Entanglement gates

A quantum controlled gate acts on at least two qubits, where some of the qubits are set as the control bits which enable the operation on target qubits. The commonly used controlled gate is controlled NOT gate (CNOT) which flips the target qubit if and only if control qubit is $|1\rangle$.

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4}$$

another entanglement gate which can be easily implemented on trap ion quantum computer is Ising coupling gate, XX gate.

$$XX = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & -ie^{i\phi} \\ 0 & 1 & -i & 0 \\ 0 & -i & 1 & 0 \\ -ie^{-i\phi} & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

2.3 Quantum measurement

Instead of directly outputting the results in classical computing, we need perform a quantum measurement on the processed states to extract classical information. A quantum measurement is always defined by a set of operators. The simplest complete measurement is to measure the output state in the computational basis. For instance, if we measure the state $|\psi\rangle = \sum_{i=1}^{n} \alpha_i |i\rangle$ in the computational basis, it will return i with probability $|\alpha_i|^2$. This class of measurement with orthogonal operators $\{|i\rangle\langle i|\}_{i=1,\ldots,2^n}$ can also be viewed as the projective measurement. A projective measurement is always described by an observable. As an observable can be written as a Hermitian matrix M. As well known to us all, there is a spectral decomposition of M, $M = \sum_m mP_m$, where P_m is the orthogonal projector with eigenvalue m. It is convenient to calculate the expected value of the projective measurement by measuring an observable M , the expected value can be formed as $E\left(m\right) = Tr\left(M\left|\psi\right\rangle\langle\psi\,|\right) = \langle\varphi\,|\,M\left|\varphi\right\rangle$.

3 Quantum generative model

Similar to the variational auto-encoder and Boltzmann machine, the goal of the quantum generative model is to learn a distribution by a given dataset $D = \left\{\mathbf{x}_i\right\}_{i=1}^{m}$, in which the samples are independent and identically distributed from the target distribution. Instead of modeling by neural networks, the quantum generative model applies a hybrid framework to reconstruct the target distribution. It is built by a variational quantum circuit (VQC) and a classical optimizer. The VQC plays a similar role, such as a neural network in classical generative tasks. The process of quantum part in the hybrid framework includes using VQC to model the target distribution in the Hilbert space, measuring the processed states in the computational basis, and outputting the expected value of the observable. The classical optimizer receives the expected value, then optimizes the loss function, finally feeds back and guides the VQC to adjust the parameters until the model converges. The main differences between VQC-based models are the structure of quantum circuits, the loss function, and the optimization methods.

In this work, we use a flexible circuit structure for quantum generative model and make comparison of performance between different models based on widely used loss functions.

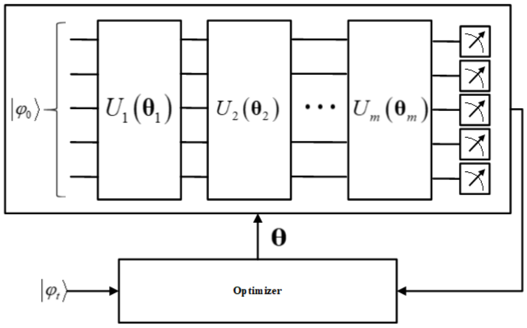

$|\varphi_0\rangle$
$U_1(\boldsymbol{\theta}_1)$
$U_2(\boldsymbol{\theta}_2)$
$\cdots$
$U_m(\boldsymbol{\theta}_m)$
$\boldsymbol{\theta}$
$|\varphi_t\rangle$
Optimizer

Figure 1: framework of VQC

3.1 Variable-depth quantum circuit structure

Instead of the previous studies which are based on fixed circuit structure, we use a more flexible way to construct the quantum circuit, variable-depth variational quantum circuit (vVQC). The basic idea is that we regard the number of circuit layers l as a parameter rather than a fixed number. We will introduce the following two ways to set up the layers of the circuit.

We call the quantum circuit block $U_l\left(\theta_l\right)$ as a layer, which is constructed by parameterized circuit block and entanglement circuit block shown in Fig. 1. The parameterized circuit block is built by rotation gates $e^{-i\theta\sigma}$, where σ is chosen from

Pauli gates set $\{X, Y, Z\}$, which are applied on each qubit. The degree θ of the rotation is viewed as the parameter of the circuit which to be tuned. The single qubit layer can be written as,

$$U_s = \bigotimes_{i=1}^{n} \prod_{j=1}^{3} e^{-i\theta_{ij}\sigma_j} \tag{6}$$

The entanglement block is constructed by entanglement gates such as CNOT gates and Ising coupling gates, and can be written as the following form,

$$U_{ent} = \prod_{k<l\in[n]} U(k,l) \tag{7}$$

where $U(k,l)$ represents a entanglement block which acts on k^{th} and l^{th} qubits. We choose a fully-connected pattern as circuit structure of entanglement block, that is applying entanglement gates on any two qubits pair as shown in Fig. 2. One variational circuit layer is combining the parameterized block and fully-connected entanglement block. Here we note it as U_l, which is a unit composed of vVQC.

$$U_l = U_{ent} U_s = \prod_{j<k\in[n]} U(j,k) \bigotimes_{i=1}^{n} e^{-i\theta_i\sigma} \tag{8}$$

This kind of structure can also be implemented on trap-ion quantum device.

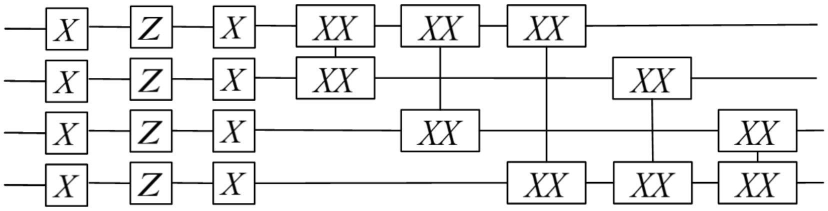


Figure 2: The circuit can be implemented on trap-ion quantum device, where X, Z represent $e^{-u\theta\sigma_x}$ and $e^{-u\theta\sigma_z}$ respectively. XX represents Ising coupling gate

Another choice of building the circuit is employing universal 2-qubits circuit block [Sousa and Ramos (2007)], which require 15 single rotation gates and 3 CNOT gates, as shown in Fig. 3, where $U_i = e^{-i\alpha_{z1}^{(i)}\sigma_z/2} e^{-i\alpha_y^{(i)}\sigma_y/2} e^{-i\alpha_{z2}^{(i)}\sigma_z/2}$. We apply these universal 2-qubits circuit to the near-neighbor qubits, so the single mixed layer requires $m-1$ the block to construct shown in Fig. 4.

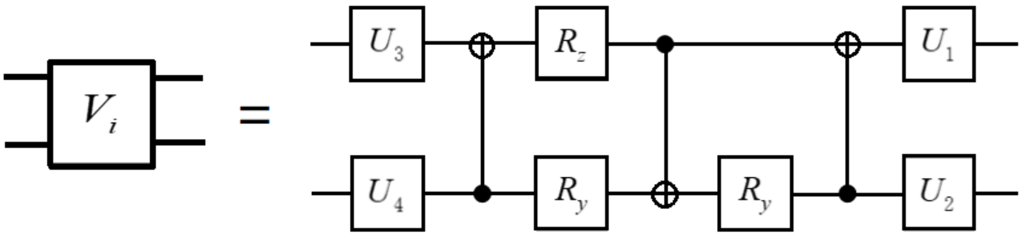


Figure 3: The universal two-qubit circuit

Similar to the work based on a variable-depth circuit in compiling quantum circuit tasks [Khatri, LaRose, Poremba et al. (2019)], the variational-depth quantum circuit is applied in this work for the quantum generative learning task. With the increasing of the number of the layers, the quantum circuit will enrich its expressive power in general. Our goal is to automatically extend the layers of the circuit according to the qBAS score. We set the number of the layers as l_0 and check the qBAS score every ep iterations [Benedetti, Garcia-Pintos, Perdomo et al. (2019)]. When the qBAS score is over the preset threshold value after ep iterations, we will reset the parameters, increase l_0, and randomly choose some parts of the old parameters to update instead of initializing all the parameters. By repeating the above process, we will get a model with reasonable performance.

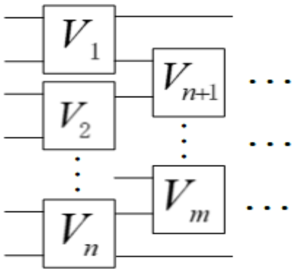


Figure 4: The quantum circuit block of vVQC

3.2 Quantum generative model with vVQC

Our quantum generative model is similar to quantum circuit learning and also consists of two parts. One is a variable-depth quantum circuit which is used to model the target distribution, another one is the post-processing module which contains a classical optimizer that is used to optimize the loss function, that is $\min_\theta L\left(D_t, D_g\right)$ where D_t is the target distribution and D_g is the generated distribution, by the gradient-based methods. Initially, we set the zero state $\left|\varphi_0\right\rangle = \left|00...0\right\rangle$ as the input state, then feed it to the vVQC. The state travels forward through the vVQC like classical feedforward neural network. However, the variational circuit generates a quantum state instead of producing classical data as output. Since we are unable to use the quantum state which encodes the processed data directly, it is necessary to measure the output state with a specific observable and get the element of the generated distribution $D_g = Tr\left[MU\left|\varphi_0\right\rangle\langle\varphi_0\,|U^\dagger\right]$. Post-processing is feeding the D_g to a classical optimizer that finds the of the optimal solution of the loss function and returns the new parameters θ to the vVQC.

We set the initial number of the vVQC layers as same as the number of system size. To avoid trapping in local minima, the training process of vVQC will extend the size of the circuit and reset some parts of the parameters when the qBAS score is greater than the threshold. Similar to the classical neural network, with an appropriately increasing of the depth of the quantum circuit, the expression ability of the quantum variational circuit will also be enhanced [Niu, Lu and Chuang (2019)]. However, increasing the number of layers is not always beneficial for training. Not only it increases the parameter space to optimize, but it also increases the complexity of the variational quantum circuit to evaluate.

Algorithm 1: training process of quantum generative model with variable-depth VQC

Input: Initial quantum state $\left|0\right\rangle^{\otimes n}$, target distribution Q, learning rate η, threshold value ε_{qBAS}, iteration threshold value ep, type of loss function k, iteration $iter$.

Output: the trained vVQC

```
Initialize the parameters of variational quantum circuit $qc$, and set initial layer number $l_0 = 1$, $v = 100$.
While $v > \varepsilon_{qBAS}$ do
    generate fake distribution $P$ by $qc$.
    $\theta \leftarrow \theta - \eta \cdot \partial_\theta L_k\left(qc, Q\right)$
    calculate qBAS score, $v = qBAS(P, Q)$
    If $iter$ mod $ep = 0$ then
        add a variational quantum layer. $qc = \mathrm{add\_layer(qc)}$
        $l \leftarrow l + 1$
        randomly choose parameters of new circuit to initialize
$iter \leftarrow iter + 1$
```

In the generative learning task, the loss function is always based on the probability distance. Since the performance of the model will be affected by the different metric distances between the generated distribution and target distributions, we estimate the following metrics and set them as the loss functions for the generative learning task. Here Q is the target distribution, and P is the generated distribution which can be written as

$$P\left(x\right) = \langle 0\,|U^\dagger\left(\theta\right)\left|x\right\rangle\langle x\,|U\left(\theta\right)\left|0\right\rangle \tag{9}$$

and the partial derivative of $P(x)$ with respected θ_i is

$$\begin{aligned}\frac{\partial P(x)}{\partial \theta_i} &= \langle 0 | \left(U_{n:i+1} \frac{\partial U_i}{\partial \theta_i} U_{i-1:1} \right)^{\dagger} |x\rangle\langle x | U_n ... U_1 |0\rangle \\ &+ \langle 0 | \left(U_n ... U_1 \right)^{\dagger} |x\rangle\langle x | \left(U_{n:i+1} \frac{\partial U_i}{\partial \theta_i} U_{i-1:1} \right) |0\rangle\end{aligned} \tag{10}$$

where $U_{n:1} = U_n U_{n-1} ... U_1$.

• Kullback-Leibler divergence

The Kullback-Leibler divergence between discrete probability distribution P and Q is define as

$$D_{KL}\left(P \| Q\right) = \sum_{x\in X} P(x)\log\left(P(x)\right) - \sum_{x\in X} P(x)\log\left(Q(x)\right) \tag{11}$$

We compute the gradient with following way respected to the parameters θ_i of the vVQC.

$$\frac{\partial D_{KL}}{\partial \theta_i} = \sum_x \frac{\partial P(x)}{\partial \theta_i}\left[\log\left(P(x)\right) - \log\left(Q(x)\right)\right] + \sum_x \frac{\partial P(x)}{\partial \theta_i} \tag{12}$$

• Jensen-Shannon divergence

The Jensen-Shannon divergence is another statistic metric based on KL divergence, defined as

$$D_{JS}\left(P \| Q\right) = \frac{1}{2} D_{KL}\left(P \middle\| \frac{P+Q}{2}\right) + \frac{1}{2} D_{KL}\left(Q \middle\| \frac{P+Q}{2}\right) \tag{13}$$

Then the gradient of JS loss function with respected to parameter θ_i is

$$\frac{\partial D_{JS}}{\partial \theta_i} = \frac{1}{2}\left[\sum_x \frac{\partial P(x)}{\partial \theta_i}\left(\log\left(P(x)\right) - \log\left(P(x)+Q(x)\right)\right) - \log\frac{1}{2}\sum_x \frac{\partial P(x)}{\partial \theta_i}\right] \tag{14}$$

• Total variation distance

As our set is discrete and countable, the total variation distance is

$$D_{TV}\left(P \| Q\right) = \frac{1}{2}\sum_x \left|P(x) - Q(x)\right| \tag{15}$$

then we can get the gradient,

$$\frac{\partial D_{TV}}{\partial \theta_i} = \frac{1}{2}\sum_x \frac{V_x}{|V_x|}\frac{\partial P(x)}{\partial \theta_i} \tag{16}$$

where $V_x = \langle 0 | \left(U_n ... U_1\right)^{\dagger} |x\rangle\langle x | U_n ... U_1 |0\rangle - Q(x)$

• Maximum mean discrepancy

We can also use kernel maximum mean discrepancy (MMD) to distinguish the two distribution by finite samples. The square of MMD can be formulated as

$$D_{MMD}\left(P\|Q\right)=E_{x,x'\sim P}[k(x,x')]-2E_{x,y\sim P,Q}[k(x,y)]+E_{y,y'\sim Q}[k(y,y')] \tag{17}$$

where $x,y\in\{0,1\}^n$ are the permutations. Here we choose Gaussian function $e^{-\frac{1}{2\sigma}\|x-y\|^2}$ as our kernel. It is easy to use chain rule to calculate the gradient of the parameter θ_i,

$$\begin{aligned}\frac{\partial D_{MMD}}{\partial\theta_i}=&\sum_{x,x'\sim P}k\left(x,x'\right)\left[\frac{\partial P\left(x\right)}{\partial\theta_i}P\left(x'\right)+P\left(x\right)\frac{\partial P\left(x'\right)}{\partial\theta_i}\right]\\&-2\sum_{x,y\sim P,Q}k\left(x,y\right)\frac{\partial P\left(x\right)}{\partial\theta_i}Q\left(y\right)\end{aligned} \tag{18}$$

4 Experimental results

We focus on training our model on a widely used Bars-and-Stripes dataset, which is a set of binary images with $n\times n$ pixels for the generative task [Gao, Zhang and Duan (2017); Benedetti, Garcia-Pintos, Perdomo et al. (2019)]. The probability of each sample in Bars and Stripes dataset is uniform and equals $1/(2^{(n+1)}-2)$. For example, there are six different Bar-and-Stripe patterns in the 2×2 pixels case shown in Fig. 5. To capture the patterns, we represent a pixel by one qubit, so the system size should be $2^{n\times n}$. In fact, the system size will exponentially increase as the number of pixel increases. Because there are limits of the computational power of our device for classical simulation, our following numerical experiments are mainly based on the 2×2 Bar-and-Strips dataset and we built the models on a four-qubits quantum system. The target distribution Q is encoded into the probability amplitudes of a quantum state, i.e., $\left|\varphi_t\right\rangle=\sum_i\sqrt{Q\left(i\right)}\left|i\right\rangle$.

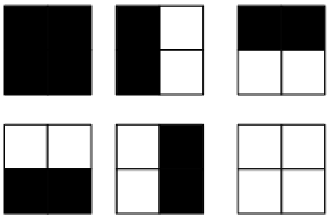


Figure 5: Example of bar-and-strip dataset

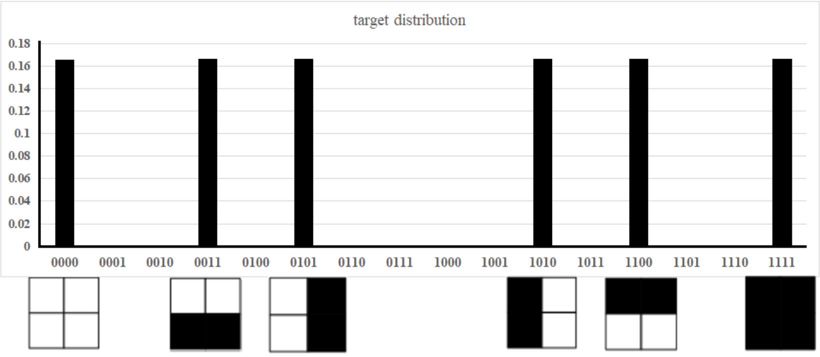


Figure 6: The Bar-and-Stripe pattern and the target distribution of 2×2 case

We tested the models based on four different loss functions including KL-divergence, JS-divergence, total variation distance, and maximum mean discrepancy for the generative learning task on Bar-and-Strips dataset respectively. We choose the gradient descent as the optimization method with the learning rate 0.1. The loss curves of the experiments are shown in Fig. 7, and the performances of quantum generative models based on VQC and vVQC are shown in Fig. 8. As shown in Figs. 7 and 8, the vVQC-based model has better performance than the original VQC-based model.

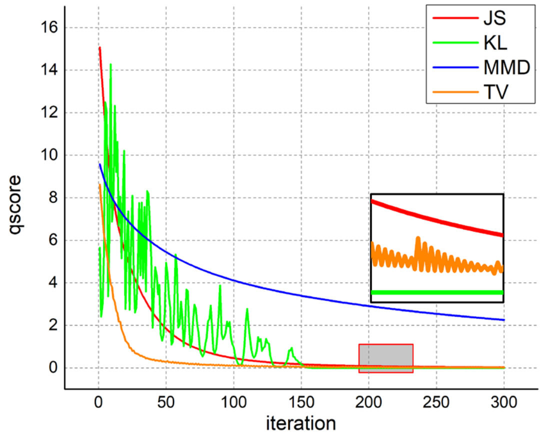
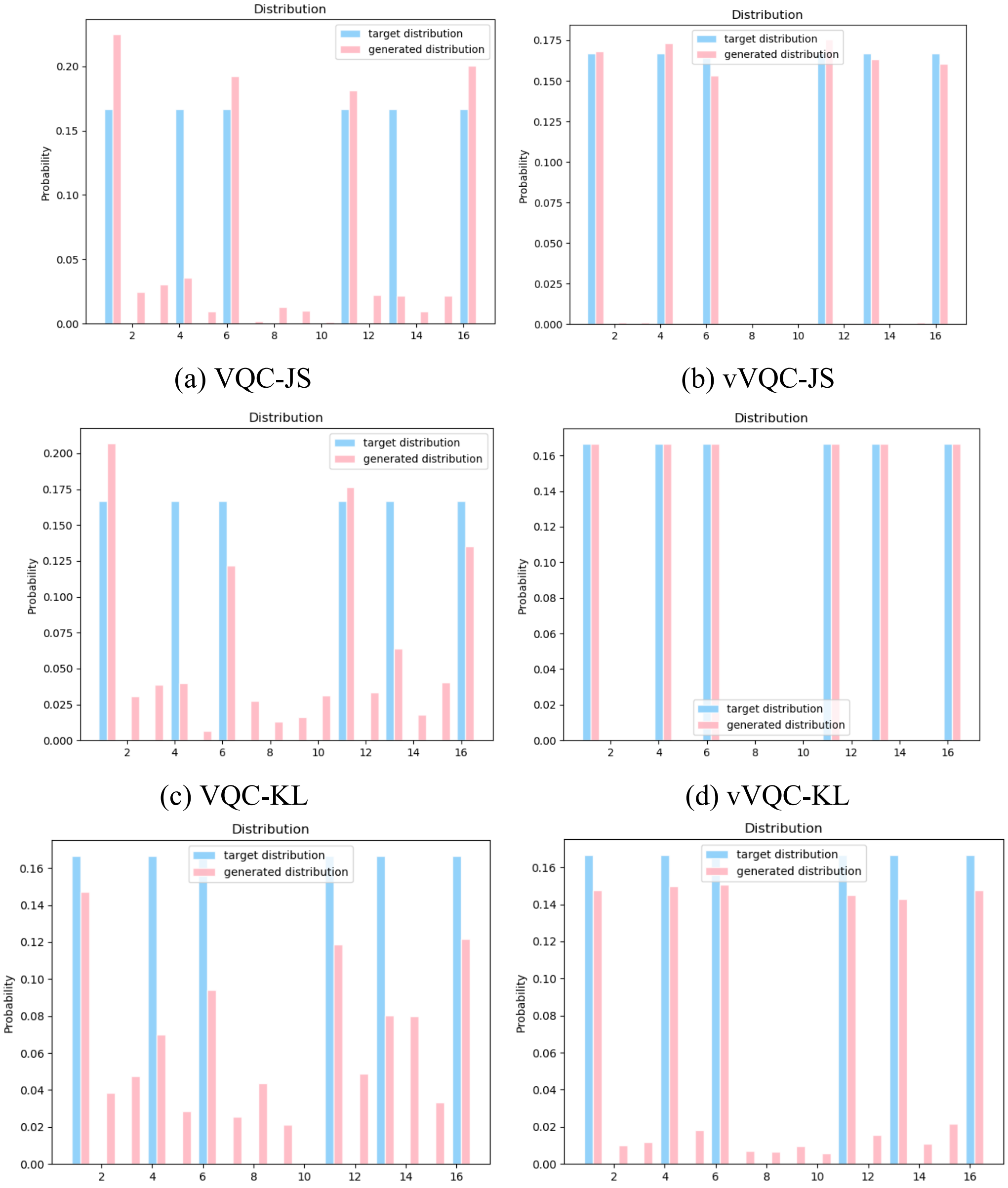


Figure 7: qScroe of KL-divergence, JS-divergence, total variation distance and MMD

From Fig. 8, the qBAS score curves of JS, KL, and TV can reach 0.01 around 150 iterations, and the performance of KL-divergence is the best and reach 0.001 after around 200 iterations. The reason why the performance of MMD does not provide a good result is that the parameter σ of the Gaussian kernel function is fixed and not optimal.



(a) VQC-JS

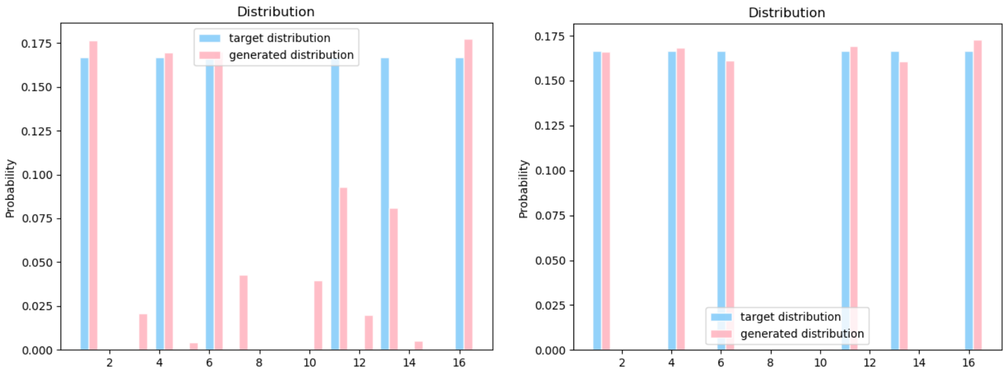
(b) vVQC-JS

(c) VQC-KL

(d) vVQC-KL

(e) VQC-MMD

(f) vVQC-MMD



(g) VQC-TV

(h) vVQC-TV

Figure 8: The comparison between generated distribution by normal VQC and vVQC with MMD and total variation distance as the loss function and target distribution

5 Discussion

The VQC plays an important role in the widely used hybrid quantum-classical computing framework, especially in quantum machine learning. However, most studies based on the VQC are using the fixed circuit structure, which is not flexible enough. In this work, we explored a flexible way to build the VQC and applied it for the generative learning task. It is called variable-depth VQC, which automatically increases the number of the quantum circuit layers and randomly chooses parts of parameters to update according to the qBAS score. As the layers of the quantum circuit are added dynamically, the expressive power of VQC is enhanced, especially when the size of the quantum circuit is not easy to determine. Besides, we designed the quantum generative models based on the variable-depth VQC with the different loss functions and tested them on learning the given probability distribution. The results show that the performances of variable-depth VQC are better than that of the normal VQC models. There are still open questions, including whether there is an optimal circuit structure with fewer gates and finding the optimal parameters in MMD distance.

Funding Statement: This work has received support from the National Key Research & Development Plan of China under Grant No. 2018YFA0306703.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Benedetti, M.; Garcia-Pintos, D.; Perdomo, O.; Leyton-Ortega, V.; Nam, Y. et al. (2019): A generative modeling approach for benchmarking and training shallow quantum circuits. *NJP Quantum Information*, vol. 5, no. 1, pp. 1-9.

Benedetti, M.; Realpe-Gómez, J.; Biswas, R; Perdomo-Ortiz, A. (2017): Quantum-assisted learning of hardware-embedded probabilistic graphical models. *Physical Review X*, vol. 7, no. 4, 041052.

Broecker, P.; Carrasquilla, J.; Melko, R. G.; Trebst, S. (2017): Machine learning quantum phases of matter beyond the fermion sign problem. *Scientific Reports*, vol. 7, no. 1, pp. 1-10.

Chen, Y. T.; Xia, R. L.; Wang, Z.; Zhang, J. M.; Yang, K. et al. (2019): The visual saliency detection algorithm research based on hierarchical principle component analysis method. *Multimedia Tools and Applications*, vol. 75, pp. 16943-16958.

Farhi, E.; Goldstone, J.; Gutmann, S. (2014): A quantum approximate optimization algorithm. *arXiv preprint. arXiv*:1411.4028.

Gao, X.; Zhang, Z.; Duan, L. (2017): An efficient quantum algorithm for generative machine learning. *arXiv preprint. arXiv*:1711.02038.

Harrow, A. W.; Hassidim, A.; Lloyd, S. (2009): Quantum algorithm for linear systems of equations. *Physical Review Letters*, vol. 103, no. 15, pp. 150502.

Havlíček, V.; Córcoles, A. D.; Temme, K.; Harrow, A. W.; Kandala, A. et al. (2019): Supervised learning with quantum-enhanced feature spaces. *Nature*, vol. 567, no. 7747, pp. 209-212.

Khatri, S.; LaRose, R.; Poremba, A.; Cincio, L.; Sornborger, A. T. et al. (2019): Quantum-assisted quantum compiling. *Quantum*, vol. 3, pp. 140.

Liu, J. G.; Wang, L. (2018): Differentiable learning of quantum circuit born machines. *Physical Review A*, vol. 98, no. 6, pp. 062324.

Niu, M. Y.; Lu, S.; Chuang, I. L. (2019): Optimizing qaoa: success probability and runtime dependence on circuit depth. *arXiv preprint. arXiv*:1905.12134.

Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M. H.; Zhou, X. Q. et al. (2014): A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, vol. 5, pp. 4213.

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

Qu, Z.; Li Z.; Xu, G.; Wu, S. Y.; Wang, X. J. et al. (2019): Quantum image steganography protocol based on quantum image expansion and Grover search algorithm. *IEEE Access*, vol. 7, pp. 50849-50857.

Qu, Z.; Wu, S.; Wang, M.; Sun, L.; Wang, X. J. (2017): Effect of quantum noise on deterministic remote state preparation of an arbitrary two-particle state via various quantum entangled channels. *Quantum Information Processing*, vol. 16, no. 12, pp. 306.

Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Lilienfld, A. (2015): Big data meets quantum chemistry approximations: the Δ-machine learning approach. *Journal of Chemical Theory and Computation*, vol. 11, no. 5, pp. 2087-2096.

Sousa, P. B. M.; Ramos, R. V. (2007): Universal quantum circuit for N-qubit quantum gate: a programmable quantum gate. *Quantum Information & Computation*, vol. 7, no. 3, pp. 228-242.