Quantum tangent kernel

NORIHITO SHIRAI[1,a)]   KENJI KUBO[1,2,b)]   KOSUKE MITARAI[1,3,4,c)]   KEISUKE FUJII[1,3,5,d)]

Abstract: Quantum kernel method is one of the key approaches to quantum machine learning, which has the advantages that it does not require optimization and has theoretical simplicity. By virtue of these properties, several experimental demonstrations and discussions of the potential advantages have been developed so far. However, as is the case in classical machine learning, not all quantum machine learning models could be regarded as kernel methods. In this work, we explore a quantum machine learning model with a deep parameterized quantum circuit and aim to go beyond the conventional quantum kernel method. In this case, the representation power and performance are expected to be enhanced, while the training process might be a bottleneck because of the barren plateaus issue. However, we find that parameters of a deep enough quantum circuit do not move much from its initial values during training, allowing first-order expansion with respect to the parameters. This behavior is similar to the neural tangent kernel in the classical literatures, and such a deep variational quantum machine learning can be described by another emergent kernel, *quantum tangent kernel*. Numerical simulations show that the proposed quantum tangent kernel outperforms the conventional quantum kernel method for an ansatz-generated dataset. This work provides a new direction beyond the conventional quantum kernel method and explores potential power of quantum machine learning with deep parameterized quantum circuits.

1. Introduction

Applying quantum computers to machine learning purposes is an emerging area of research. In particular, motivated by the recent advance of hardware technology [3], techniques to apply so-called noisy intermediate quantum (NISQ) devices [23] are rapidly developed [6], [7], [12], [21], [26]. One direction that is frequently explored is a variational method which use parameterized quantum circuits to construct a model $y(x, \theta)$ which outputs a prediction when fed with an input data x. More specifically, using a parameterized quantum circuit $U(x, \theta)$ with trainable parameters θ, we construct a model $y(x, \theta)$ for an input x by the expectation value of an observable O: $y(x, \theta) = \langle 0| U^\dagger(x, \theta) O U(x, \theta) |0\rangle$. Since there are quantum circuits that are hard to simulate classically and can be implemented on present hardware [3], using such types of circuit for $U(x, \theta)$ we might be able to construct a machine learning model that exceeds the capability of classical computers.

Another promising direction is the so-called quantum kernel method [12], [24], [26]. In this approach, we use quantum computers solely for preparing a classically intractable feature $|\phi(x)\rangle$

and taking the inner products of the feature vectors. The training is performed on a classical computer using the values of inner product between each pair of training dataset. It is advantageous in that we do not have to optimize the quantum circuit, which is often difficult in the aforementioned variational methods. Moreover, it can be shown that the quantum kernel method outperforms the variational ones in a certain sense [24]; if $U(x, \theta)$ takes the form of $U(x, \theta) = V(\theta) U_\phi(x)$, then the quantum kernel method with feature vector $|\phi(x)\rangle = U_\phi(x) |0\rangle$ is guaranteed to achieve better prediction accuracy for training dataset. Its simple framework greatly advanced the construction of quantum machine learning theory, providing insights on potential advantages [13], [20]. Also, owing to its experimental easiness, there have been several experimental demonstrations of the method [5], [10], [16]. However, as is well known in the machine learning field, not all machine learning models can be described by kernel methods, and there must be approaches that go beyond them, such as deep learning.

In this work, we explore how to construct a quantum machine learning model that goes beyond the conventional quantum kernel method. To this end, we investigate the performance of a model constructed by deep quantum circuit in the form of $U(x, \theta) = \prod_i V_i(\theta_i) U_{\phi,i}(x)$. The circuit is now not in the form of $U(x, \theta) = V(\theta) U_\phi(x)$, and hence this model cannot be translated to a kernel framework. Unfortunately, as parameterized quantum circuits become deeper and go beyond the conventional quantum kernel method, it becomes more difficult to train the parameters as was the case with deep neural network in the classical literature. While this issue would be resolved by finding a good ansatz and/or initial parameters [8], [11], [22], we consider an alterna-

---

1   Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka, Osaka 560-8531, Japan.
2   R4D, Mercari Inc., Roppongi Hills Mori Tower 18F, 6-10-1, Roppongi, Minato-ku, Tokyo 106-6118, Japan
3   Center for Quantum Information and Quantum Biology, Osaka University, Japan.
4   JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan.
5   RIKEN Center for Quantum Computing, Wako Saitama 351-0198, Japan
a)   shirai@qc.ee.es.osaka-u.ac.jp
b)   kenjikun@mercari.com
c)   mitarai@qc.ee.es.osaka-u.ac.jp
d)   fujii@qc.ee.es.osaka-u.ac.jp

tive approach, i.e., overparameterization [15], [17], [27].

More precisely, we find that, when the circuit is deep enough, each of the parameters in θ does not move much and stays close to the initial random guess. This behavior is similar to the classical deep neural network [14], where the amount of change in the parameters is small, and the network is well described by a linear model with the tangent on the parameters as the basis function. Thus optimized parameters on overparameterized networks can be found by another kernel, so-called neural tangent kernel. The observation motivates us to propose quantum tangent kernel (QTK), which is a quantum analog of neural tangent kernel [14].

Specifically, we define QTK by the kernel associated with a feature map $x \to \nabla_\theta y(x, \theta)$ for a deep parameterized quantum circuit. The QTK can be calculated efficiently on a quantum computer by using analytical differentiation of parameterized quantum circuits, so-called parameter-shift rule [21], [25]. Then, QTK combined with the standard kernel methods such as support vector machine (SVM) allows us to learn and infer without any explicit optimization of the parameterized quantum circuit. We compare the performance of QTK and the conventional quantum kernel method for an "ansatz-generated" dataset generated by a deep parameterized quantum circuit, while the parameters are randomly chosen apart from those in the QTK. As a result, we find that QTK outperforms the conventional quantum kernel method. This and the fact that tangent kernel is justified only in the overparameterized limit imply that deep parameterized quantum circuits have a great potential to go beyond the conventional quantum kernel model. This work opens up a new direction beyond the conventional quantum kernel method for deeper quantum machine learning.

2. Background

2.1 Kernel methods

In this subsection, we explain the kernel method and support vector machine (SVM), which is one of the kernel-based classification methods. Throughout this paper, we denote the training dataset by $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ where x_i is input data and y_i is corresponding teacher data.

The kernel method is a technique for mapping features to a higher dimensional feature space in order to introduce nonlinearity to the model. It employs a non-linear map ϕ from the original space to the higher dimensional feature space:

$$\phi : \chi \to \mathcal{H}, \tag{1}$$
$$x_i \to \phi(x_i),$$

where χ is an original space of the data and $\mathcal{H}$ is a higher dimensional feature space. In the kernel method, we only use inner products of ϕ with respect to different input data. For two data x_i and x_j, we define

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product on $\mathcal{H}$. The kernel method has the advantage that it does not require direct computation of the vectors in a high-dimensional feature space as long as their inner product can be calculated efficeintly. This inner product

represents the similarity between the features and is called the kernel function.

As an example of the kernel method, we consider SVM, which is one of the kernel-based classification methods. SVM with kernel $K(x_i, x_j)$ is trained by maximizing the following objective function

$$L_D(\alpha) = -\frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j) + \sum_{i=1}^N \alpha_i, \tag{3}$$

subject to $\sum_{i=0}^N \alpha_i y_i = 0$ and $\alpha_i \geq 0$. L_D depends only on the kernel function, and hence we do not need to explicitly calculate the feature map ϕ. The solution α_i^* to the above optimization problem is used for building the prediction model as follows:

$$y(x) = \text{sign}\left(\sum_{i=1}^N y_i \alpha_i^* K(x_i, x) + b \right), \tag{4}$$

where the bias b is calculated as,

$$b = y_j - \sum_{i=1}^N \alpha_i^* y_i K(x_i, x_j). \tag{5}$$

Although the formula of bias b holds for any j, practically, the average for all j is taken.

2.2 Neural Tangent Kernel

Let us denote by $f(x_i, \theta)$ the output of a neural network where θ is parameters in the network, and x_i is the input data. In the large width overparameterized neural network, the values of the parameters change only slightly from the initial values during training even if the initial values of parameters are set randomly [14]. Therefore, the output of such a neural network can be well approximated by the first-order expansion with respect to the parameters around the initial values:

$$f(x_i, \theta) \simeq f(x_i, \theta_0) + \nabla_\theta f(x_i, \theta_0)^T (\theta - \theta_0), \tag{6}$$

where θ_0 is the initial values of parameters of a neural network. This approximation allows us to interpret the neural network as a linear model with a feature map

$$\phi(x) = \nabla_\theta f(x, \theta_0). \tag{7}$$

Using this feature map, we define the following kernel called neural tangent kernel (NTK) [14], [18].

$$K_{\text{ntk}}(x_i, x_j) = \nabla_\theta f(x_i, \theta_0)^T \nabla_\theta f(x_j, \theta_0). \tag{8}$$

By its construction, NTK-based linear models are expected to be equivalent to the large-width neural network as long as the assumption Eq. (6) is valid.

2.3 Quantum machine learning as a kernel method

Conventional variational quantum machine learning models [12], [21], [26] work in the following manner. First, data are encoded into quantum states by $U_\phi(x)$. Then, we apply a trainable parameterized circuit $V(\theta)$. Finally, we measure the expectation value of an observable O, which is used as the model output $y(x, \theta)$. Mathematically, the above process can be written as,

$$y(\boldsymbol{x}, \boldsymbol{\theta}) = \langle 0^n | U_\phi^\dagger(\boldsymbol{x}) V^\dagger(\boldsymbol{\theta}) O V(\boldsymbol{\theta}) U_\phi(\boldsymbol{x}) | 0^n \rangle. \qquad (9)$$

The correspondence of this model to a linear model can be readily seen by rewriting the above expression as,

$$y(\boldsymbol{x}, \boldsymbol{\theta}) = \mathrm{Tr}(O(\boldsymbol{\theta})\rho(\boldsymbol{x})), \qquad (10)$$

where

$$O(\boldsymbol{\theta}) = V^\dagger(\boldsymbol{\theta}) O V(\boldsymbol{\theta}), \qquad (11)$$

$$\rho(\boldsymbol{x}) = U_\phi(\boldsymbol{x})|0^n\rangle \langle 0^n| U_\phi^\dagger(\boldsymbol{x}) \qquad (12)$$

Since $\mathrm{Tr}(A^\dagger B)$ for operators A and B defines an inner product in the operator space, Eq. (10) defines a linear model using the feature vector $\rho(\boldsymbol{x})$ and the weight vector $O(\boldsymbol{\theta})$ [24]. Therefore, if we construct a kernel-based model using the same feature vector $\rho(\boldsymbol{x})$, it is guaranteed to achieve better performance on a training dataset than the model in Eq. (10). In this case, we define the kernel function as,

$$K_q(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathrm{Tr}(\rho(\boldsymbol{x}_i)\rho(\boldsymbol{x}_j)) \qquad (13)$$

$$= |\langle \phi(\boldsymbol{x}_i)|\phi(\boldsymbol{x}_j)\rangle|^2, \qquad (14)$$

where $|\phi(\boldsymbol{x})\rangle$ is defined as follows

$$|\phi(\boldsymbol{x})\rangle = U_\phi(\boldsymbol{x})|0^n\rangle. \qquad (15)$$

We call the kernel methods that are based on $K_q(\boldsymbol{x}_i, \boldsymbol{x}_j)$ the conventional quantum kernel method.

The above argument only holds for a quantum model $y(\boldsymbol{x}, \boldsymbol{\theta}) = \langle 0| U^\dagger(\boldsymbol{x}, \boldsymbol{\theta}) O U(\boldsymbol{x}, \boldsymbol{\theta})|0\rangle$ with $U(\boldsymbol{x}, \boldsymbol{\theta})$ in the form of $V(\boldsymbol{\theta})U_\phi(\boldsymbol{x})$. We propose quantum circuits that cannot be splitted in such a way in Sec. 4. This modification prevents us from rewriting the model into the form of Eq. (10), and thus we cannot construct an equivalent kernel model.

3. Quantum Tangent Kernel

In this section, we apply the formulation of the NTK described in Sec.2.2 to parametrized quantum circuits. We consider a model whose output is given as the expectation value of an operator O as,

$$y(\boldsymbol{x}, \boldsymbol{\theta}) = \langle 0^n | U^\dagger(\boldsymbol{x}, \boldsymbol{\theta}) O U(\boldsymbol{x}, \boldsymbol{\theta})|0^n\rangle, \qquad (16)$$

where $\boldsymbol{x}$ is the input data and $\boldsymbol{\theta}$ are parameters of a quantum circuit. We define the following kernel by using the output of a quantum circuit analogous to the NTK.

$$K_{\mathrm{qtk}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \nabla_\theta y(\boldsymbol{x}_i, \boldsymbol{\theta}_0)^T \nabla_\theta y(\boldsymbol{x}_j, \boldsymbol{\theta}_0). \qquad (17)$$

We call the kernel $K_{\mathrm{qtk}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ quantum tangent kernel (QTK). As in NTK, QTK is valid as long as the parameters do not change much from their initial random guess $\boldsymbol{\theta}_0$ when, for example, updating them by gradient descent based with respect to some suitable cost function.

We can numerically check that such a phenomena occurs for quantum circuits with a large number of layers. To this end, we train quantum circuits with varying numbers of layers and look at the changes of their parameters. Using a 4-qubit circuit as shown
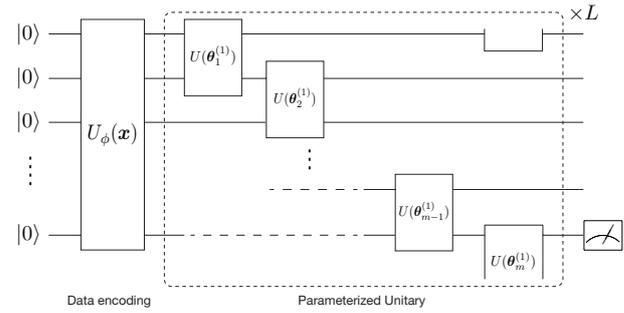

Fig. 1 The m-qubit ansatz used for numerical simulations. $U_\phi(\boldsymbol{x})$ is the quantum feature map for encoding classical data. $U(\boldsymbol{\theta}_j^{(i)}) \in SU(4)$ is the parameterized unitary of the i-th layer. At the output, we measure the Pauli Z expectation value of the final qubit.

in Fig. 1, we train them on the iris dataset [1], [9]. In this numerical experiment, the quantum feature map to encode the data is given by $U_\phi(\boldsymbol{x}) = \bigotimes_{i=1}^{4} \exp(ix_i Y_i)$. The initial values of the parameters of these quantum circuits are set randomly using a uniform distribution between 0 and 2π. We use the mean squared error as the loss function and the standard gradient descent to train the quantum circuits. Fig. 2 (a) shows the relative norm changes in the parameters of the quantum circuits with layers $L = 1, 3, 10$ and 20. We can observe that when a quantum circuit has more layers, the changes of the parameters during training becomes small. As shown in Fig. 2 (b), the training loss for each quantum circuits decreases which indicates the success of training. This result implies that it is possible to linearly approximate $y(\boldsymbol{x}, \boldsymbol{\theta})$ with respect to its parameters when the circuit is sufficiently deep. Hence, we can expect QTK to provide a machine learning model that is approximately equivalent to $y(\boldsymbol{x}, \boldsymbol{\theta})$.

For a given ansatz, the QTK can be calculated on a quantum computer with parameter-shift rules [21], [25] and its generalizations [4], [28]. This provides us an alternative quantum machine learning model other than the conventional variational methods and quantum kernel methods. Note that, as opposed to the case of the NTK [2], [14], currently we do not know how to calculate the QTK analytically for a given form of ansatz in the infinite size limit. We leave this direction as an interesting future direction to explore.

4. Numerical Experiment

We numerically demonstrate the expressive power of quantum tangent kernel. Here, we consider two types of quantum tangent kernel according to how the data is encoded into quantum states. First one is a circuit where $\boldsymbol{x}$ is encoded only at the first layer as in Fig. 1:

$$U_{\mathrm{shallow}}(\boldsymbol{x}, \boldsymbol{\theta}) = V(\boldsymbol{\theta}) U_\phi(\boldsymbol{x}), \qquad (18)$$

where $V(\boldsymbol{\theta})$ is a parameterized unitary and $U_\phi(\boldsymbol{x})$ is a quantum feature map to encode data. This type of quantum circuit can be interpreted as the conventional quantum kernel method. We call the QTK associated with this type of ansatz shallow QTK. Next, in order to increase the non-linearity, we consider a multi-layered circuit that alternates between data encoding and a parameterized unitary as shown in Fig. 3. More concretely, we consider the following unitary:

$$U_{\text{deep}}(\boldsymbol{x}, \boldsymbol{\theta}) = \prod_{i=1}^{L} \left[V(\boldsymbol{\theta}_i) U_\phi(\boldsymbol{x}) \right] . \quad (19)$$

We call the QTK associated with this type of ansatz deep QTK.

Since U_{deep} has higher non-linearity, it is expected to show higher expressive power than shallow QTK. In order to demonstrate the expressive power, we generate a "ansatz-generated" dataset by using a quantum circuit for U_{deep} and classify the data using SVM with the three types of kernels: shallow QTK, deep QTK and conventional quantum kernel defined by the feature map $U_\phi(\boldsymbol{x})$. If this data is classified efficiently by SVM with deep QTK, it has higher expressive power than the other two.

In this numerical experiment, we use $U_\phi(\boldsymbol{x})$ in the form of,

$$U_\phi(\boldsymbol{x}) = \exp\left(i \sum_{S \subseteq [m]} \phi_S(\boldsymbol{x}) \bigotimes_{i \in S} Y_i \right), \quad (20)$$

where Y_i is the Pauli Y operator acting on the i-th qubit. In our experiment, interaction up to 2-qubit $|S| \le 2$ is given and acts on the initial state $|0^n\rangle$. This type of quantum feature map has been proposed in Ref. [12]. The functions $\phi_i(\boldsymbol{x})$ and $\phi_{ij}(\boldsymbol{x})$ are given by,

$$\phi_i(\boldsymbol{x}) = \arcsin(x_i), \quad (21)$$

$$\phi_{ij}(\boldsymbol{x}) = \arcsin(x_i x_j). \quad (22)$$

The "ansatz-generated" dataset is generated in the following manner. Four dimensional random value data $\{\boldsymbol{x}_i\}$ are inputted into $U_{\text{deep}}(\boldsymbol{x}, \boldsymbol{\theta})$ consisting of $n = 4$ qubits and $L = 10$ layers. Then, we evaluate the expectation value

$$l(\boldsymbol{x}_i, \boldsymbol{\theta}) = \langle 0^n | U_{\text{deep}}^\dagger(\boldsymbol{x}_i, \boldsymbol{\theta}) Z_4 U_{\text{deep}}(\boldsymbol{x}_i, \boldsymbol{\theta}) | 0^n \rangle \quad (23)$$

with a randomly chosen $\boldsymbol{\theta}$. We label each $\boldsymbol{x}_i$ as 1 and -1 if $l(\boldsymbol{x}, \boldsymbol{\theta}) \ge 0$ and $l(\boldsymbol{x}, \boldsymbol{\theta}) < 0$, respectively. We generate the 15,000 samples of the four dimensional input data $\boldsymbol{x}_i$ and its label y_i. They are splitted into 10,000 training and 5000 test data.

We classify the above dataset with SVM using three different kernels: shallow QTK (Eq. (17) with $U(\boldsymbol{x}, \boldsymbol{\theta}) = U_{\text{shallow}}(\boldsymbol{x}, \boldsymbol{\theta})$ and $L = 10$ layers), deep QTK (Eq. (17) with $U(\boldsymbol{x}, \boldsymbol{\theta}) = U_{\text{deep}}(\boldsymbol{x}, \boldsymbol{\theta})$ and $L = 10$ layers) and conventional quantum kernel (Eq. (13)). In order to calculate QTK and deep QTK, we randomly set parameter values of a quantum circuit using a uniform distribution between 0 and 2π. Note that the parameters used in this learning phase is different from the one used for data generation. We apply $U(\boldsymbol{\theta}_i^{(j)}) \in SU(4)$ on each neighboring qubits. $SU(4)$ is parameterized according to Cartan decomposition as follows

$$U(\boldsymbol{\theta}_i^{(j)}) = \quad (24)$$
$$k_1 \exp\left[\frac{i}{2} (\theta_{xx} X_i X_{i+1} + \theta_{yy} Y_i Y_{i+1} + \theta_{zz} Z_i Z_{i+1}) \right] k_2 .$$

where $k_1, k_2 \in SU(2) \otimes SU(2)$ and X_i, Y_i, Z_i are the Pauli operators. QTKs are calculated by the parameter-shift rule [21], [25]. The regularization strength is determined via cross-validation for each kernel.

The results of the classification task are listed in Table 1. Among three kernels, deep QTK outperforms other kernels. This
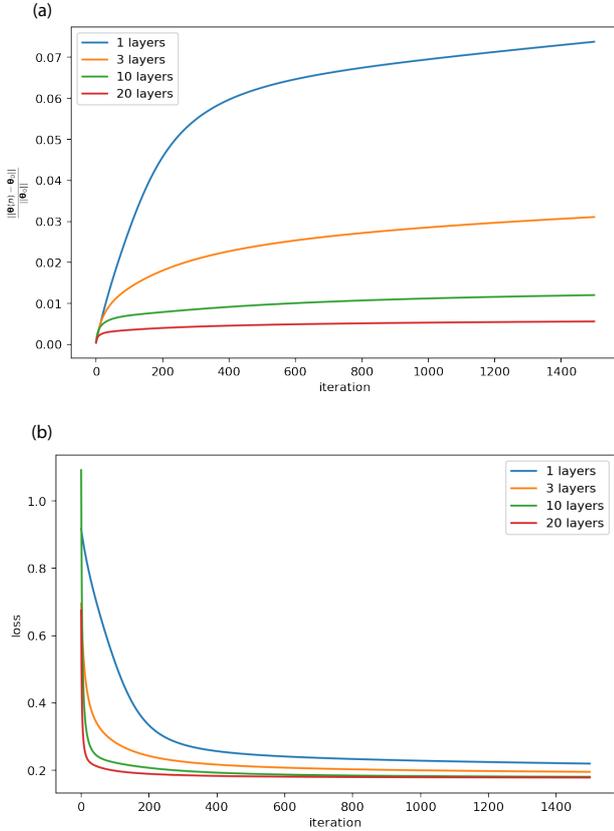

Fig. 2 (a) Relative norm change in the parameters of the quantum circuit from initial values during training by gradient descent. $\boldsymbol{\theta}(n)$ is the parameter at the n-th iteration. (b) Behavior of training losses for different number of layers during training.
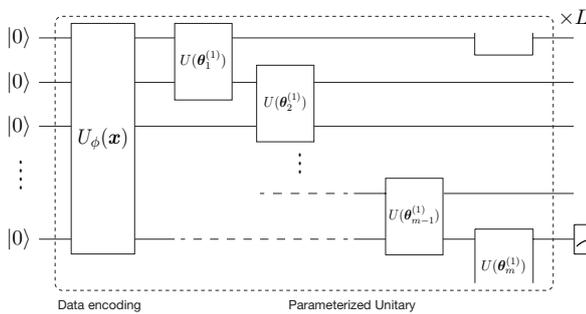

Fig. 3 Quantum circuit that defines deep quantum tangent kernel. $U_\phi(\boldsymbol{x})$ is the feature map and $U(\boldsymbol{\theta}_j^{(i)}) \in SU(4)$ is the parameterized unitary of the i-th layer. To increase the non-linearity of the kernel, the feature map $U_\phi(\boldsymbol{x})$ and the parameterized unitary are iteratively applied. At the output, we measure the Pauli Z expectation value of the final qubit.

Kernel	Accuracy
Quantum kernel	0.7842
Shallow quantum tangent kernel	0.7484
Deep quantum tangent kernel	0.812

Table 1 Classification accuracy for SVM with three types of kernels. Three SVMs classify the ansatz-generated dataset generated by a quantum circuit for deep QTK as shown in Fig.3.

result indicates that deep QTK employs a feature map that cannot be expressed very accurately with other kernels. In contrast, shallow QTK is essentially just a quantum kernel method using the feature map in Eq. (18), so its performance is not improved compared to conventional quantum kernel as expected.

The expressive power of kernels can be illustrated more directly by visualizing the feature map of each kernel. Fig. 4 shows the distribution of the ansatz-generated dataset generated by quantum circuits for $U_{\text{shallow}}(\boldsymbol{x}, \boldsymbol{\theta})$ and $U_{\text{deep}}(\boldsymbol{x}, \boldsymbol{\theta})$. In order to visualize the distribution, we generate two dimensional data using the two-qubit quantum circuits. These distributions show that $U_{\text{deep}}(\boldsymbol{x}, \boldsymbol{\theta})$ expresses higher non-linearity than $U_{\text{shallow}}(\boldsymbol{x}, \boldsymbol{\theta})$ does.
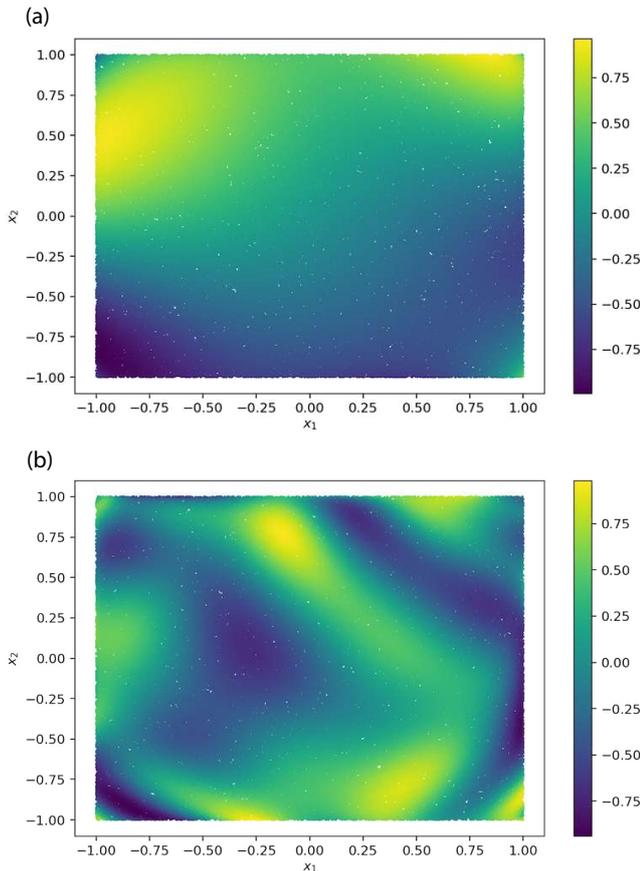

Fig. 4 (a) The distribution of outputs of the quantum circuit with $L = 10$ layers (Fig. 1) which can be interpreted as the conventional quantum kernel method. (b) The distribution of outputs of the quantum circuit with $L = 10$ layers (Fig. 3) beyond the conventional quantum kernel.

5. Conclusion

We proposed quantum tangent kernel (QTK) and deep quantum tangent kernel which cannot be interpreted as conventional quantum kernel methods described in Ref. [24]. QTK is defined

by applying the formulation of NTK to parametrized quantum circuits. We find that parameters of an overparameterized quantum circuit change only slightly from its initial values during training. This indicates that the output of an overparameterized quantum circuit can be linearly approximated, which validates the formulation of QTK. By using this overparameterization, we can avoid the gradient descent for quantum circuits with a large number of parameters and easily optimize the parameters. Then, in order to increase the non-linearity of a feature map, we introduced a multilayered data encoding that alternates between data encoding and a parameterized unitary. This encoding method increases non-linearity of the feature map and improves the expressive power of kernels.

We demonstrated the performance of shallow QTK and deep QTK for the classification task. Using the ansatz-generated dataset generated by the quantum circuit for deep QTK (Eq. (19)), we evaluate the performance by using SVM with three kernels: shallow QTK, deep QTK and conventional quantum kernel. We showed that SVM with deep QTK outperforms SVM with other kernels and deep QTK has a feature map with high non-linearity.

Our results imply that deep parameterized quantum circuits with repetitive data encoding unitary have a higher representation power and better performance for quantum machine learning than the conventional quantum kernel method. While we here employed an overparameterization limit and hence a deeper quantum circuit to take the neural tangent kernel approach, sophisticatedly trained neural networks, such as deep neural networks, provide a better performance in general than neural tangent kernels as known in the classical literatures [19]. It gives us hope that the real fruit of quantum machine learning is not in the shallow or deep limit, but in the mild depth, which could be modeled by neither conventional nor tangent kernel methods. Therefore, better ansatz constructions and parameter optimization methods are crucially important.

References

[1] Edgar Anderson. The species problem in iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509, 1936.

[2] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *CoRR*, abs/1904.11955, 2019.

[3] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew

Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, Oct 2019.

[4]   Leonardo Banchi and Gavin E. Crooks. Measuring Analytic Gradients of General Quantum Evolution with the Stochastic Parameter Shift Rule. *Quantum*, 5:386, January 2021.

[5]   Karol Bartkiewicz, Clemens Gneiting, Antonín Černoch, Kateřina Jiráková, Karel Lemr, and Franco Nori. Experimental kernel-based quantum machine learning in finite feature space. *Scientific Reports*, 10(1):12356, Jul 2020.

[6]   Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, Nov 2019.

[7]   M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, Aug 2021.

[8]   Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, Aug 2019.

[9]   R. A. FISHER. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[10]   Jennifer R. Glick, Tanvi P. Gujarati, Antonio D. Corcoles, Youngseok Kim, Abhinav Kandala, Jay M. Gambetta, and Kristan Temme. Covariant quantum kernels for data with group structure, 2021.

[11]   Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, 3:214, December 2019.

[12]   Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.

[13]   Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):1–9, 2021.

[14]   Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *CoRR*, abs/1806.07572, 2018.

[15]   Joonho Kim, Jaedeok Kim, and Dario Rosa. Universal effectiveness of high-depth circuits in variational eigenproblems. *Physical Review Research*, 3(2), Jun 2021.

[16]   Takeru Kusumoto, Kosuke Mitarai, Keisuke Fujii, Masahiro Kitagawa, and Makoto Negoro. Experimental quantum kernel trick with nuclear spins in a solid. *npj Quantum Information*, 7(1):94, Jun 2021.

[17]   Martin Larocca, Nathan Ju, Diego García-Martín, Patrick J. Coles, and M. Cerezo. Theory of overparametrization in quantum neural networks, 2021.

[18]   Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32:8572–8583, 2019.

[19]   Zhiyuan Li, Ruosong Wang, Dingli Yu, Simon S. Du, Wei Hu, Ruslan Salakhutdinov, and Sanjeev Arora. Enhanced convolutional neural tangent kernels, 2019.

[20]   Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9):1013–1017, 2021.

[21]   Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

[22]   Arthur Pesah, M. Cerezo, Samson Wang, Tyler Volkoff, Andrew T. Sornborger, and Patrick J. Coles. Absence of barren plateaus in quantum convolutional neural networks. *Phys. Rev. X*, 11:041011, Oct 2021.

[23]   John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, August 2018.

[24]   Maria Schuld. Supervised quantum machine learning models are kernel methods. 2021.

[25]   Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

[26]   Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504, Feb 2019.

[27]   David Wierichs, Christian Gogolin, and Michael Kastoryano. Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer. *Physical Review Research*, 2(4), Nov 2020.

[28]   David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients, 2021.