arXiv:2202.08232v3 [quant-ph] 12 Jun 2022

Quantum Lazy Training

Erfan Abedi, Salman Beigi, Leila Taghavi

QuOne Lab, Phanous Research & Innovation Centre, Tehran, Iran

Abstract

In the training of over-parameterized model functions via gradient descent, sometimes the parameters do not change significantly and remain close to their initial values. This phenomenon is called *lazy training*, and motivates consideration of the linear approximation of the model function around the initial parameters. In the lazy regime, this linear approximation imitates the behavior of the parameterized function whose associated kernel, called the *tangent kernel*, specifies the training performance of the model. Lazy training is known to occur in the case of (classical) neural networks with large widths. In this paper, we show that the training of *geometrically local* parameterized quantum circuits enters the lazy regime for large numbers of qubits. More precisely, we prove bounds on the rate of changes of the parameters of such a geometrically local parameterized quantum circuit in the training process, and on the precision of the linear approximation of the associated quantum model function; both of these bounds tend to zero as the number of qubits grows. We support our analytic results with numerical simulations.

1 Introduction

The goal of achieving near-term quantum advantages have put forward quantum machine learning as one of the main applications of Noisy Intermediate-Scale Quantum (NISQ) devices [16]. A main paradigm for achieving quantum advantage in machine learning is via quantum variational algorithms [4]. In this approach, a quantum circuit consisting of parameterized gates is learned in order to fit some training data. However, this learning process through which optimal parameters of the circuit are found, faces challenges in practice [12, 20], and needs a thorough exploration.

Gradient descent is one of the main methods for solving optimization problems, particularly for training the parameters of a quantum circuit for machine learning. In this method, the parameters are updated by moving in the opposite direction of the gradient of a loss function to be optimized. This updating of the parameters changes not only the value of the loss function, but also the function modeled by the quantum circuit. Thus, studying the evolution of the loss and model functions during the gradient descent algorithm is crucial in understanding variational quantum algorithms.

Approximating the gradient descent algorithm with its continuous version (gradient flow) [1], provides us with analytical tools for the study of the evolution of a function

whose parameters are optimized via gradient descent. Writing down the evolution equation for this continuous approximation, we observe the appearance of a kernel function called *tangent kernel* (see Section 2). In short, letting $f(\Theta, x)$ be our model function with $\Theta = (\theta_1, \ldots, \theta_p)$ as the parameters (weights) of the model and x as a data point on which we evaluate the function, the tangent kernel is defined by

$$K_\Theta(x, x') = \nabla_\Theta f(\Theta, x) \cdot \nabla_\Theta f(\Theta, x'). \tag{1}$$

Here, $\nabla_\Theta f(\Theta, x)$ is the gradient of $f(\Theta, x)$ with respect to Θ and $\nabla_\Theta f(\Theta, x) \cdot \nabla_\Theta f(\Theta, x')$ is the inner product of the gradient vector for two data points x, x'. The tangent kernel at some initial point Θ_0 can be thought of as the kernel associated with the *linear approximation* of the function given by

$$f(\Theta, x) \simeq f\big(\Theta^{(0)}, x\big) + \nabla_\Theta f\big(\Theta^{(0)}, x\big) \cdot \big(\Theta - \Theta^{(0)}\big). \tag{2}$$

The tangent kernel for (classical) neural networks is called the *Neural Tangent Kernel* (NTK). It is shown in [9] that although the NTK depends on Θ which varies during the gradient descent algorithm, when the *width* of the neural network is large compared to its depth, the NTK remains almost unchanged. In fact, for such neural networks, the parameters Θ remain very close to their initial value $\Theta^{(0)}$. This surprising phenomenon is called *lazy training* [6].

In the *lazy regime*, since Θ is close to its initialization $\Theta^{(0)}$, the linear approximation of the function in (2) is accurate. In this case, the behavior of the function under training via gradient descent follows its linear approximation, and is effectively described by the tangent kernel at initialization. We will review these results and related concepts in more detail in Section 2.

Our results: Our main goal in this paper is to develop the theory of lazy training for parameterized quantum circuits as our model function, and to generalize the results of [9] to the quantum case. We prove that when the number of qubits (analogous to the width of a classical neural network) in a quantum parameterized circuit is large compared to its depth, the associated model function can be approximated by a linear model. Moreover, we show that this linear model's behavior is similar to that of the original model under the gradient descent algorithm.

To prove the above results, we need to put some assumptions on the class of parameterized quantum circuits. The results of [9] in the classical case are proven by fixing all layers of a neural network but one, and sending the number of nodes (width) in that layer to infinity. In the quantum case, assuming that we neither introduce fresh qubits nor do we measure/discard qubits in the middle of the circuit, the number of qubits is fixed in all layers. Thus, in the quantum case, unlike [9], we cannot consider layers of the circuit individually and take their width (number of qubits) to infinity independently of other layers. To circumvent this difficulty, we put some restrictions on our quantum circuits:

(i) We assume that the circuit is geometrically local and the entangling gates are performed on neighboring qubits. For example, we assume the qubits are arranged on a 1D or 2D lattice and the two-qubit gates are applied only on pairs of adjacent qubits. More generally, we assume that the qubits are arranged on nodes of a

bounded-degree graph and that the two-qubit gates can be applied only on pair of qubits connected by an edge. We note that this assumption arguably holds in most proposed hardware architectures of realizable quantum computers.

(ii) We also assume that the observable which is measured at the end of the circuit is a *local operator* with its locality being in terms of the underlying bounded-degree graph mentioned above. More precisely, we assume that the observable is a sum of terms, each of which acts only on a constant number of neighboring qubits. We will offer a number of evidences to show that our results do not hold without this assumption.

Given the above assumptions, we prove the followings:

1. To apply the gradient descent algorithm, we usually choose the initial parameters of the circuit at random. In Theorem 1, We show that when choosing the initial parameters *independently* at random, the quantum tangent kernel *concentrates* around its average as the number of qubits tends to infinity. This means that when the number of qubits is large, at first the tangent kernel is essentially independent of the starting parameters and is fixed.

2. We also show, in Theorem 2, that when the number of qubits is large, lazy training occurs; meaning that the parameters of the circuit do not change significantly during the gradient descent algorithm and remain almost constant. This means that the tangent kernel is fixed not only at initialization, but also during the training. As a result and as mentioned above, our model function can be approximated by a linear model which shows a behavior similar to that of the original model during the training via gradient descent.

These results show that in order to analyze the training behaviour of parameterized quantum circuits with the aforementioned assumptions, we may only consider the linearized model. We note that the linearized model is determined by the associated tangent kernel, which assuming that the initial parameters are chosen independently at random, is concentrated around its average. Thus, the eigenvalues of the average tangent kernel determine the training behaviour of such parameterized quantum circuits. Based on this observation, we argue in Remark 3 that if these eigenvalues are far from zero, then the model is trained exponentially fast. We will comment on this result in compared to the no-go results about barren plateaus in Section 6.

We also provide numerical simulations to support the above results.

Related works: The subject of tangent kernels in the quantum case has been previously studied in a few works which we briefly review.

A tangent kernel for *hybrid* classical-quantum networks is considered in [14]. We note, however, that in this work the quantum part of the model is fixed and parameter-free, and only the classical part of the network is trained.

The quantum tangent kernel is considered in [19] for *deep* parameterized quantum circuits. In this work, a *deep circuit* is a circuit with a *multi-layered data encoding* which alternates between data encoding gates and parameterized unitaries. This data encoding

scheme increases the expressive power of the model function. It is shown in [19] that as the number of layers increases, the changes in circuit parameters decrease during the gradient descent algorithm (a signature behavior of lazy training), and the training loss vanishes more quickly. It is also shown that the tangent kernel associated to such deep quantum parameterized circuits can outperform *conventional* quantum kernels, such as those discussed in [17] and [8]. We note that all of these results are based solely on numerical simulations. Moreover, the simulations are performed only for 4-qubit circuits and do not predict the behaviour of the circuits in the large width limit.

Quantum tangent kernel of parameterized quantum circuits (for both optimization and machine learning problems) is also studied in [10]. In this work, *without* exploring conditions under which lazy training occurs, it is shown that in the lazy training regime (or "frozen limit"), the loss function decays exponentially fast.

Finally, tangent kernel for *quantum states* is defined in [11], and based on numerical simulations, it is shown that it can be used in the study of the training dynamics of finite-width *neural network quantum states*.

We emphasize that the missing ingredient shared by these previous works is the absence of explicit conditions on the quantum models under which the training is *provably* enters the lazy regime. This missing part is addressed in our work.

Outline of the paper: The rest of this paper is organized as follows. In Section 2, we review the notions of tangent kernel and lazy training in more detail. In Section 3, we describe quantum parameterized circuits and their training. We also explain in more detail the assumption of geometric locality mentioned above, and give an explicit example of such quantum circuits. Section 4 is devoted to the proof of our main results regarding quantum lazy training. In Section 5, we support our analytic results with numerical simulations. Concluding remarks are discussed in Section 6.

2 Tangent Kernel and Lazy Training

In this section we briefly review the notion of a tangent kernel and explain the results of [9] for classical neural networks.

Let $f(\Theta, x)$ be a model function which for any set of parameters Θ, maps $\mathbb{R}^d$ to $\mathbb{R}$. Having a training dataset $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$, where $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$, our goal is to find the best parameters Θ for which the outputs of our model $f(\Theta, x^{(i)})$ get close to the outputs provided in the dataset $y^{(i)}$ for all $i \in \{1, 2, \ldots, n\}$. To quantify this, we will need a metric to measure our model's ability to match our dataset. On that account, we make use of a *loss function*, which in this paper is chosen to be the commonly used *mean squared error* function:

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \Big(f\big(\Theta, x^{(i)}\big) - y^{(i)} \Big)^2. \tag{3}$$

Then, our goal is to find the optimal parameters that minimize the loss function:

$$\min_{\Theta} L(\Theta). \tag{4}$$

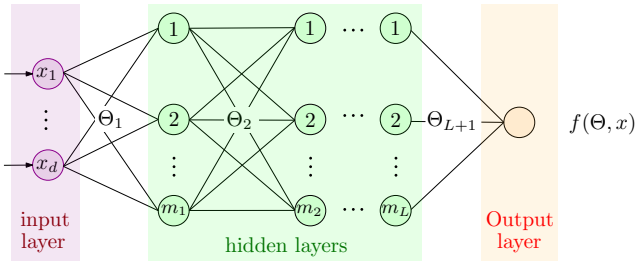


Figure 1: A classical neural network with L hidden layers. The input layer transmits a data-point $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ to the first hidden layer through its outgoing edges like signals. Any edge of the network has a weight that transforms the passing signal. Each node in the hidden layers applies a non-linear *activation function* on its input signals and pass the result to the next layer. The output layer computes the model function $f(\Theta, x)$, where Θ denotes the set of all weights of the network. It is shown in [9] that when $m_1, \ldots, m_L \to \infty$ the model enters the lazy regime.

We use the gradient descent algorithm to solve (4). To this end, we randomly initialize parameters $\Theta = \Theta^{(0)}$ and in each step update them by moving in the opposite direction of the gradient of the loss function: $\Theta^{(t+1)} = \Theta^{(t)} - \eta \nabla_\Theta L(\Theta^{(t)})$, where η is a fixed scalar called the *learning rate* and $\nabla_\Theta L(\Theta^{(t)})$ denotes the gradient of the loss function with respect to Θ. This updating of parameters is repeated until a *termination condition* is satisfied, e.g., the gradient vector $\nabla_\Theta L(\Theta^{(t)})$ approaches zero, or the number of iterations reaches a maximum limit.

In order to analyze the gradient descent algorithm, we consider its continuous approximation. That is, we assume that the parameters are updated continuously via the gradient flow differential equation:

$$\partial_t \Theta^{(t)} = -\nabla_\Theta L(\Theta^{(t)}).$$

Then, the evolution of the model function computed at a data point x is given by

$$\begin{aligned}\partial_t f\big(\Theta^{(t)}, x\big) &= -\nabla_\Theta L\big(\Theta^{(t)}\big) \cdot \nabla_\Theta f\big(\Theta^{(t)}, x\big) \\ &= -\frac{1}{n} \sum_{i=1}^{n} \Big(f\big(\Theta^{(t)}, x^{(i)}\big) - y^{(i)} \Big) \nabla_\Theta f\big(\Theta^{(t)}, x^{(i)}\big) \cdot \nabla_\Theta f\big(\Theta^{(t)}, x\big).\end{aligned}$$

This computation motivates the definition of the *tangent kernel* as follows:

$$K_\Theta(x, x') = \nabla_\Theta f(\Theta, x) \cdot \nabla_\Theta f(\Theta, x').$$

We note that $K_\Theta(x, x')$ is a valid kernel function, since it is the inner product of two vectors. Then, we have

$$\partial_t f\big(\Theta^{(t)}, x\big) = -\frac{1}{n} \sum_{i=1}^{n} \Big(f\big(\Theta^{(t)}, x^{(i)}\big) - y^{(i)} \Big) K_{\Theta^{(t)}}(x^{(i)}, x), \tag{5}$$

The tangent kernel alone is enough to determine the evolution of the model function in the training process.

Let us consider the case where $f(\Theta, x)$ comes from a neural network as in Figure 1. In this case, for instance, when there is only a *single hidden layer*, the model function is

given by

$$f(\Theta, x) = \frac{1}{\sqrt{m}} \sum_{k=1}^{m} b_k \, \sigma\Big(\sum_{j=1}^{d} a_{kj} x_j \Big). \tag{6}$$

Here, m is the number of nodes in the hidden layer, $\Theta = (a_{kj}, b_k, 1 \leq j \leq d, 1 \leq k, \leq m)$ where a_{kj} is the weight of the edge connecting x_j to the k-th node of the hidden layer, and b_k is the weight of the edge connecting the k-th node of the hidden layer to the output node. Moreover, $\sigma(\cdot)$ is a non-linear *activation function*. Finally, following [9] we introduce the normalization factor $\frac{1}{\sqrt{m}}$ in $f(\Theta, x)$ since we will consider the limit of this model function as m tends to infinity.

When training such a neural network with a large width, i.e., large number of nodes in the hidden layers, it is observed that the initial parameters $\Theta^{(0)}$ do not change significantly, and $\Theta^{(t)}$ remains close to $\Theta^{(0)}$ until the gradient vector $\nabla_\Theta L(\Theta^{(t)})$ approaches zero. This observation motivates the Taylor expansion of the model function at $\Theta^{(0)}$:

$$f(\Theta, x) \simeq f\big(\Theta^{(0)}, x\big) + \nabla_\Theta f\big(\Theta^{(0)}, x\big) \cdot \big(\Theta - \Theta^{(0)}\big). \tag{7}$$

Observe that the right hand side is linear in Θ (but not in x). Indeed, it is a linear transformation after applying the *feature map* $x \mapsto \nabla_\Theta f(\Theta^{(0)}, x)$. Interestingly, the kernel function associated to this feature map is nothing but the tangent kernel $K_{\Theta^{(0)}}(x, x')$ associated to the neural network, and is called the *neural tangent kernel*.

Based on the above observations, it is proven in [9] that when the width of hidden layers in a neural network tends to infinity, it enters the *lazy regime*, meaning that $\Theta^{(t)}$ remains close to $\Theta^{(0)}$ during the gradient descent algorithm. Moreover, it is proven that in this case, linear approximation of the model function as in (7) remains valid not only at initialization, but also during the entire training process. For more details on these results, particularly on the assumptions under which they hold, we refer to the original paper [9]. We also refer to [6] for more details on lazy training.

3 Parameterized Quantum Circuits

Parameterized quantum circuits are considered as the quantum counterpart of classical neural networks [7]. Each parameterized quantum circuit amounts to a model function and similar to neural networks, can be trained to fit some data.

As the name suggests, a parameterized quantum circuit is a circuit with some of its gates non-fixed and dependent on some parameters. Indeed, some gates of the circuit depend on parameters denoted by Θ, and some gates *encode* the input x. A measurement is performed at the end of the circuit which determines the output of computation. The measurement itself could also be parameterized, but in this work, for the sake of simplicity it is assumed to be fixed. See Figure 2 for an example of a parameterized circuit.

Letting $U(\Theta, x)$ be the unitary associated to the circuit, and O be the observable measured at the end, the resulting model function is given by

$$f(\Theta, x) = \langle 0 \cdots 0 | U^\dagger(\Theta, x) O U(\Theta, x) \, | 0 \cdots 0 \rangle \, . \tag{8}$$

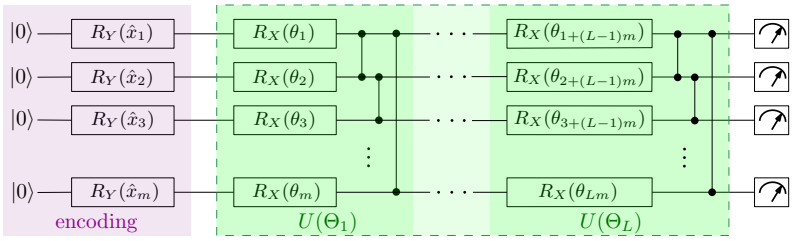


Figure 2: A quantum parameterized circuit with L layers of parameterized gates. Here, all the qubits are initialized at $|0\rangle$, and then a layer of Y-rotations (i.e., $R_Y(\hat{x}_j) = \exp(-i\frac{\hat{x}_j}{2}Y)$) is applied to *encode* the input x, where $\hat{x}_1, \ldots, \hat{x}_m$ are functions (e.g., coordinates) of x. Next, L layers of parameterized gates are applied. We assume that only the single-qubit gates are parameterized and fix the entangling gates to controlled-Z gates. We assume that the qubits are arranged on a cycle, and the controlled-Z in each layer are applied on all pairs of neighboring qubits.

Then, having such a model function and a dataset $D = \big\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\big\}$, with $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$, we may try to find the optimal Θ that minimizes the loss function:

$$L(\Theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\Big(f\big(\Theta, x^{(i)}\big) - y^{(i)}\Big)^2. \tag{9}$$

To this end, as before, we initialize the parameters Θ independently at random and move towards minimizng the value of this loss function by the way of gradient descent.

We usually arrange gates of a parameterized circuit in *layers*. For instance, the circuit of Figure 2 consists of an encoding layer of single-qubit (Y-rotation) gates and L layers, each of which consists of some single-qubit (X-rotation) gates and some two-qubit (controlled-Z) gates. This layer-wise structure of parameterized circuits is crucial for us since in our results, we are going to fix the number of layers L, and consider the limit of large number of qubits ($m \to \infty$).

In this paper, for the stability of the model, we need to assume that the parameterized gates do not change significantly by a slight change in the parameters Θ. To this end, we assume that

$$\Big\|\frac{\partial}{\partial\theta_j}U(\Theta, x)\Big\|, \Big\|\frac{\partial^2}{\partial\theta_i\theta_j}U(\Theta, x)\Big\| \leq c, \qquad \forall i, j, \tag{10}$$

for some constant $c > 0$. We note that this assumption holds in most parameterized circuits in the literature, particularly when the parameterized gates are Pauli rotation (see equation (15) below).

Geometrically local circuits: As mentioned in the introduction, to prove our result we need to restrict the class of circuits to geometrically local ones. To this end, we assume that the qubits are arranged on vertices of a *bounded-degree* graph (e.g., 1D or 2D lattice) and the entangling 2-qubits gates in the circuit are applied only on pairs of neighboring qubits. For instance, in the circuit of Figure 2, we assume that the qubits are arranged on a cycle, and the controlled-Z gates in each layer are applied only on pairs of neighboring qubits.

We also assume that the observable O that is measured at the end of the circuit is a geometrically local one. More precisely, we assume that O is given by

$$O = \frac{1}{\sqrt{m}} \sum_{k=1}^{m} O_k, \tag{11}$$

where m is the number qubits in the circuit, and O_k is an observable acting on the k-th qubit and possibly on a constant number of qubits in its neighborhood, with $\|O_k\| \leq 1$. Moreover, as in the classical case (see, equation (6)), we introduce the normalization factor $1/\sqrt{m}$ in O since we are considering the limit of $m \to \infty$. In this case, the model function (8) can be written as

$$f(\Theta, x) = \frac{1}{\sqrt{m}} \sum_{k=1}^{m} f_k(\Theta, x), \tag{12}$$

where

$$f_k(\Theta, x) = \langle 0 \cdots 0 | \, U^\dagger(\Theta, x) O_k U(\Theta, x) \, | 0 \cdots 0 \rangle \, . \tag{13}$$

We emphasize that the assumption of geometric locality on the quantum circuit described above holds in most quantum hardware architectures. After all, the qubits in the quantum hardware should be arranged on some lattice, and usually the 2-qubits gates can only be applied on neighboring qubits. However, the assumption that the observable is geometrically local is not justified by the hardware architecture. Nevertheless, global observables usually result in *barren plateaus* and a way of avoiding them is to use local observables [5]. Moreover, as our simulations in Section 5 show, our results do not hold for global observables. Thus, we have to somehow restrict the class of observables.

Example: We finish this section by explaining the example of Figure 2 in more detail, since it will be used as our quantum circuit for simulations.

First, we note that our data points $(x^{(i)}, y^{(i)})$ belong to $\mathbb{R}^d \times \mathbb{R}$, so in the circuit we need to encode each input x in an m-qubit circuit. In the circuit of Figure 2 we assume that we first map $x \in \mathbb{R}^d$ to some $\hat{x} \in \mathbb{R}^m$ and then use the coordinates of $\hat{x}$ in the encoding layer of the circuit. The mapping $x \mapsto \hat{x}$ is arbitrary and can even be non-linear. However, for our numerical simulations we use the map:

$$\hat{x}_j = x_{j \bmod d}, \quad 1 \leq j \leq m. \tag{14}$$

Then, the coordinates of $\hat{x}$ are used to encode x in the first layer:

$$U_{\text{enc}}(x) = \prod_{j=1}^{m} R_{Y_j}(\hat{x}_j) = \prod_{j=1}^{m} \exp\Big(-i \frac{\hat{x}_j}{2} Y_j \Big),$$

where Y_j denotes the Pauli-Y matrix acting on the j-th qubit.

Next, we apply L parameterized unitaries $U(\Theta_1), \ldots, U(\Theta_L)$ where

$$U(\Theta_\ell) = \prod_{k=1}^{m} CZ_{k,k+1} \prod_{j=1}^{m} R_{X_j}(\theta_{(\ell-1)m+j}),$$

where

$$R_X(\theta) = \exp\Big(-i\frac{\theta}{2}X\Big), \tag{15}$$

and $CZ_{k,k+1}$ is the controlled-Z gate applied on qubits $k, k+1$. Here, we assume that the qubits are arranged on a cycle, and the indices are modulo m.

With this specific structure for the parameterized circuit, we have

$$U(\Theta, x) = U(\Theta)U_{\text{enc}}(x) = U(\Theta_L)\cdots U(\Theta_1)U_{\text{enc}}(x).$$

Nevertheless, we emphasize that in this paper we do *not* assume that the encoding part of the circuit is only restricted to the first layer; our results are valid even if there are gates in the middle of the circuit that encode x, see [15, 18].

Finally, we assume that the observable is given by

$$O = \frac{1}{\sqrt{m}}(Z_1 + \ldots + Z_m),$$

where Z_k is the Pauli-Z operator acting on the k-th qubit. Hence, the model function associated to this parameterized circuit is equal to

$$f(\Theta, x) = \frac{1}{\sqrt{m}}\sum_{k=1}^{m} \langle 0\cdots 0|\, U_{\text{enc}}^\dagger(x)U^\dagger(\Theta)Z_kU(\Theta)U_{\text{enc}}(x)\,|0\cdots 0\rangle\,. \tag{16}$$

A crucial observation which will be frequently used in our proofs is that each term in the above sum depends only on constantly many parameters (independent of m, the number of qubits). First, note that the last layer of controlled-Z gates does not affect the model function since the controlled-Z gates are diagonal in the Z-basis and commute with the observable. Second, and more importantly, the result of the measurement of the k-th qubit depends only on the *light cone* of this qubit. To clarify this, let us assume that $L = 2$. In this case, the result of the measurement of the k-th qubit depends only on parameters $\theta_{k-1}, \theta_k, \theta_{k+1}, \theta_{m+k}$, see Figure 3. The point is that, when $L = 2$, we have

$$\begin{aligned} U_{\text{enc}}^\dagger(x)U^\dagger(\Theta)Z_kU(\Theta)U_{\text{enc}}(x) = {} & R_{Y_{k-1}}^\dagger(\hat{x}_{k-1})R_{Y_k}^\dagger(\hat{x}_k)R_{Y_{k+1}}^\dagger(\hat{x}_{k+1}) \\ & R_{X_{k-1}}^\dagger(\theta_{k-1})R_{X_k}^\dagger(\theta_k)R_{X_{k+1}}^\dagger(\theta_{k+1}) \\ & CZ_{k-1,k}\,CZ_{k,k+1}R_{X_k}^\dagger(\theta_{m+k}) \\ & Z_k \\ & R_{X_k}(\theta_{m+k})\,CZ_{k,k+1}\,CZ_{k-1,k} \\ & R_{X_{k+1}}(\theta_{k+1})R_{X_k}(\theta_k)R_{X_{k-1}}(\theta_{k-1}) \\ & R_{Y_{k+1}}(\hat{x}_{k+1})R_{Y_k}(\hat{x}_k)R_{Y_{k-1}}(\hat{x}_{k-1}). \end{aligned} \tag{17}$$

Thus, the $f(\Theta, x)$ given by (16) with $L = 2$ is a sum of m terms whose k-th term depends on $\theta_{k-1}, \theta_k, \theta_{k+1}$ and θ_{m+k}, which together make the light cone of the k-th qubit (as depicted in Figure 3).

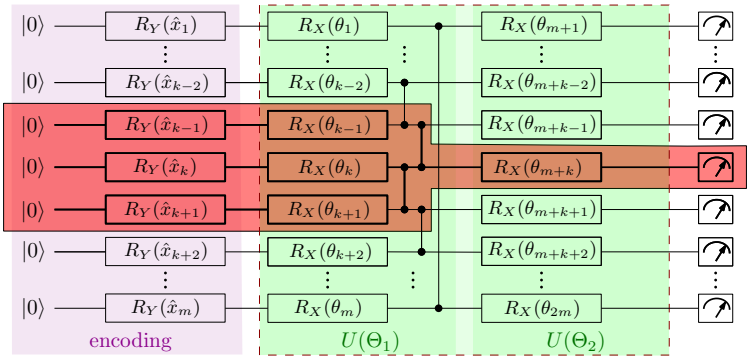


Figure 3: The light cone of the k-th qubit of the parameterized circuit of Figure 2 with $L = 2$ is depicted in red. This means that in order to compute the result of the k-th measurement Z_k, we only need to compute the red part of the circuit and ignore the rest. We note that only the parameters $\theta_{k-1}, \theta_k, \theta_{k+1}$ and θ_{m+k} appear in this light cone.

4 Main results

This section contain the proof of our results. We first show that under certain conditions, when the parameters are initialized independently at random, the tangent kernel is concentrated around its mean.

Theorem 1. *Let $f(\Theta, x)$ be a model function associated to a* geometrically local *parameterized quantum circuit on m qubits as in* (8) *with $\Theta = (\theta_1, \ldots, \theta_p)$ satisfying* (10)*. Suppose that the observable O is also geometrically local given by* (11) *where O_k acts on the k-th qubit and possibly on a constant number qubits in its neighborhood, and satisfies $\|O_k\| \leq 1$. In this case the model function is given by* (12) *and* (13)*. Suppose that $\theta_1, \ldots, \theta_p$ are chosen independently at random. Then, for any $x, x' \in \mathbb{R}^d$ we have*

$$\Pr\Big[\big|K_\Theta(x,x') - \mathbb{E}[K_\Theta(x,x')]\big| \geq \epsilon\Big] \leq \exp\Big(-\Omega\big(\frac{m^2\epsilon^2}{pc^4}\big)\Big). \tag{18}$$

Remark 1. We note that usually, the number of parameters in each layer of a circuit is linear in the number of qubits. Then, assuming that the number of layers L is constant, $p = O(Lm) = O(m)$. In this case, the right hand side of (18) vanishes exponentially fast in m.

As mentioned in the previous section, our main tool in proving this theorem is the geometric locality of the circuit and the observable. Based on this, following similar computations as in (17), we find that each term $f_k(\Theta, x)$ of the model function depends only on constantly many parameters.

In the proof of this theorem of also use McDiarmid's inequality.

Lemma 1 (McDiarmid's Concentration Inequality [13])**.** *Let $X_1 \ldots, X_n$ be independent random variables, each with values in $\mathcal{X}$. Let $f : \mathcal{X}^n \to \mathbb{R}$ be a mapping such that for every $i \in \{1, 2, \ldots, n\}$ and every $(x_1 \ldots, x_n), (x'_1 \ldots, x'_n) \in \mathcal{X}^n$ that differ only in the i-th coordinate (i.e., $x_i \neq x'_i$ and $\forall j \neq i : \; x_j = x'_j$),*

$$|f(x_1, \ldots, x_n) - f(x'_1, \ldots, x'_n)| \leq c_i.$$

For any $\epsilon > 0$

$$\mathbb{P}\Big(f(X_1, \ldots, X_n) - \mathbb{E}[f(X_1 \ldots, X_n)] \geq \epsilon\Big) \leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}\right).$$

□

Proof of Theorem 1. Let $\mathcal{N}_k$ be the set of indices $\{j\}$ with $1 \leq j \leq p$ such that $U^\dagger(\Theta, x) O_k U(\Theta, x)$ depends on θ_j. In other words, $\Theta_{\mathcal{N}_k}$ is the set of θ_j's in the light cone of the k-th observable O_k. Then, we have

$$\begin{aligned} f_k(\Theta, x) &= \langle 0 \cdots 0 |\, U^\dagger(\Theta, x) O_k U(\Theta, x)\, | 0 \cdots 0 \rangle \\ &= \langle 0 \cdots 0 |\, U^\dagger(\Theta_{\mathcal{N}_k}, x) O_k U(\Theta_{\mathcal{N}_k}, x)\, | 0 \cdots 0 \rangle \\ &= f_k(\Theta_{\mathcal{N}_k}, x). \end{aligned}$$

We note that by the assumption of geometric locality, we have $|\mathcal{N}_k| = O(1)$.

Now, by the definition of the tangent kernel we have

$$\begin{aligned} K_\Theta(x, x') &= \nabla_\Theta f(\Theta, x) \cdot \nabla_\Theta f(\Theta, x') \\ &= \frac{1}{m} \sum_{k,k'=1}^{m} \sum_{j=1}^{p} \frac{\partial}{\partial \theta_j} f_k(\Theta_{\mathcal{N}_k}, x) \cdot \frac{\partial}{\partial \theta_j} f_{k'}(\Theta_{\mathcal{N}_{k'}}, x') \\ &= \frac{1}{m} \sum_{k,k'=1}^{m} \sum_{j \in \mathcal{N}_k \cap \mathcal{N}_{k'}} \frac{\partial}{\partial \theta_j} f_k(\Theta_{\mathcal{N}_k}, x) \cdot \frac{\partial}{\partial \theta_j} f_{k'}(\Theta_{\mathcal{N}_{k'}}, x'), \end{aligned} \tag{19}$$

where the last equation follows since $\frac{\partial}{\partial \theta_j} f_k(\Theta_{\mathcal{N}_k}, x) = 0$ for any $j \notin \mathcal{N}_k$.

Let

$$\Gamma := \{(k, k', j) : j \in \mathcal{N}_k \cap \mathcal{N}_{k'}\}. \tag{20}$$

We note that since O_k acts only on a constant number of qubits in the neighborhood of the k-th qubit, $\mathcal{N}_k$ intersects $\mathcal{N}_{k'}$ only if the qubits k and k' are geometrically close to each other (in the underlying graph). Then, since the underlying graph has a bounded degree, $\mathcal{N}_k$ intersects only a constant number of $\mathcal{N}_{k'}$'s. On the other hand, the size of $\mathcal{N}_k$ is constant. Thus, for each k the number of triples (k, k', j) in Γ is constant, and we have $|\Gamma| = O(m)$.

Next, let

$$T_{k,k',j} = T_{k,k',j}(\Theta) = \frac{\partial}{\partial \theta_j} f_k(\Theta_{\mathcal{N}_k}, x) \cdot \frac{\partial}{\partial \theta_j} f_{k'}(\Theta_{\mathcal{N}_{k'}}, x').$$

Then,

$$K_\Theta(x, x') = \frac{1}{m} \sum_{(k,k',j) \in \Gamma} T_{k,k',j},$$

can be thought of as a normalized sum of $O(m)$ terms. Note that these terms are not independent of each other; each parameter θ_j may appear in more than one term. Nevertheless, again by the assumption of geometric locality, each θ_j appears in at most

constantly many terms. Therefore, by letting Θ, Θ' be two tuples of parameters differing only at the j-th position (i.e., $\theta_i = \theta'_i$ for all $i \neq j$), we get

$$\begin{aligned}|K_\Theta(x,x') - K_{\Theta'}(x,x')| &\leq \frac{1}{m} \sum_{(k,k')\colon (k,k',j)\in\Gamma} |T_{k,k',j}(\Theta) - T_{k,k',j}(\Theta')| \\ &\leq \frac{1}{m} \sum_{(k,k')\colon (k,k',j)\in\Gamma} |T_{k,k',j}(\Theta)| + |T_{k,k',j}(\Theta')| \\ &= O\Big(\frac{c^2}{m}\Big),\end{aligned}$$

where in the last line we use (10) and the fact that for each j, the number of triples (k,k',j) in Γ is constant. Then, by McDiarmid's concentration inequality [13] we have

$$\Pr\Big[\big|K_\Theta(x,x') - \mathbb{E}[K_\Theta(x,x')]\big| \geq \epsilon\Big] \leq 2\exp\Big(-\frac{2\epsilon^2}{pO(c^4/m^2)}\Big) = \exp\Big(-\Omega(\frac{m^2\epsilon^2}{pc^4})\Big).$$

□

The above theorem says that even though the parameters are chosen randomly at initialization, the tangent kernel is essentially fixed. This results in an essentially fixed linearized model via (7).

The following theorem states our second main result, that the training of geometrically local quantum circuits over large number of qubits enters the lazy regime and can be approximated by a linear model.

Theorem 2. *Let $f(\Theta, x)$ be a model function associated with a parameterized quantum circuit satisfying the assumptions of Theorem 1. Suppose that a data set $D = \big\{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\big\}$, with $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$ is given. Assume that at initialization we choose $\Theta^{(0)} = \big(\theta_1^{(0)}, \ldots, \theta_p^{(0)}\big)$ independently at random, and apply the gradient flow to update the parameters in time by $\nabla\Theta^{(t)} = -\nabla_\Theta L(\Theta^{(t)})$, where $L(\Theta)$ is given in (9). Then, the followings hold:*

(i) *For any $1 \leq j \leq p$ we have*

$$\big|\partial_t \theta_j^{(t)}\big| = O\left(\sqrt{\frac{L\big(\Theta^{(0)}\big)}{m}}\right).$$

(ii) *For any x, x' we have*

$$\big|\partial_t K_{\Theta^{(t)}}(x,x')\big| = O\left(\sqrt{\frac{L\big(\Theta^{(0)}\big)}{m}}\right).$$

(iii) *Let $\bar{f}(\bar{\Theta}, x)$ be the function associated to the linearized model, i.e.,*

$$\bar{f}(\bar{\Theta}, x) = f\big(\Theta^{(0)}, x\big) + \nabla_\Theta f\big(\Theta^{(0)}, x\big) \cdot \big(\bar{\Theta} - \Theta^{(0)}\big).$$

Suppose that we start with $\bar{\Theta}^{(0)} = \Theta^{(0)}$*, and train the linearized model with its associated loss function denoted by* $\bar{L}(\bar{\Theta}^{(t)})$ *which results in*

$$\partial_t \bar{f}\big(\bar{\Theta}^{(t)}, x\big) = -\frac{1}{n}\sum_{i=1}^{n}\Big(\bar{f}\big(\bar{\Theta}^{(t)}, x^{(i)}\big) - y^{(i)}\Big)K_{\Theta^{(0)}}(x^{(i)}, x).$$

Let

$$\Delta(t) = \left(\frac{1}{n}\sum_{i=1}^{n}\Big(f(\Theta^{(t)}, x^{(i)}) - \bar{f}(\bar{\Theta}^{(t)}, x^{(i)})\Big)^2\right)^{1/2}$$

Then, for all t *we have*

$$\Delta(t) = O\left(\frac{L(\Theta^{(0)})t^2}{\sqrt{m}}\right). \tag{21}$$

(iv) *With the notation of part (iii), for all* t *we have*

$$\big|L(\Theta^{(t)}) - \bar{L}(\bar{\Theta}^{(t)})\big| = O\left(\frac{L(\Theta^{(0)})^{3/2}t^2}{\sqrt{m}}\right). \tag{22}$$

Part (i) of this theorem says that parameters $\Theta^{(t)}$ do not change significantly during training. Based on this, we expect that the tangent kernel remains close to the initial tangent kernel as well. This is proven in part (ii). Next, since the tangent kernel is almost constant, we expect that our model function behaves like the linearized model in the training process (lazy training). This is formally proven in parts (iii) and (iv).

Remark 2. The bounds of this theorem are effective when the loss function $L(\Theta^{(0)})$ at initialization is a constant independent of m. While we do not explore the conditions under which this holds, since $\Theta^{(0)}$ is chosen at random and $f(\Theta^{(0)}, x)$ approaches a Gaussian process, we expect to have $L(\Theta^{(0)}) = O(1)$ with high probability when we learn a *bounded* function.

Remark 3. Let $\bar{F}(t) = \big(\bar{f}(\bar{\Theta}^{(t)}, x^{(1)}), \ldots, \bar{f}(\bar{\Theta}^{(t)}, x^{(n)})\big)$ and $Y = (y^{(1)}, \ldots, y^{(n)})$. Then, since the kernel associated to the linearized model is time-independent, by (5) we have

$$\bar{F}(t) = \Big(\bar{F}(0) - Y\Big)e^{-\frac{t}{n}K_{\Theta^{(0)}}} + Y.$$

This means that if $K_{\Theta^{(0)}}$ is full-rank and its minimum eigenvalue is far from zero, the training of the linearized model stops exponentially fast. In this case, the stopping time t in the bounds of parts (iii) and (iv) of the theorem is small. Indeed, under the above assumption on the eigenvalues of the tangent kernel, the parameterized quantum circuit is trained exponentially fast since by part (iv) its behaviour is well-approximated by the linearized model.

Proof. (i) We have $\nabla\Theta^{(t)} = -\nabla_\Theta L(\Theta^{(t)})$, and for any j:

$$\begin{aligned}\partial_t\theta_j^{(t)} &= -\frac{1}{n}\sum_{i=1}^{n}\Big(f(\Theta^{(t)},x^{(i)})-y^{(i)}\Big)\cdot\frac{1}{\sqrt{m}}\sum_{k=1}^{m}\frac{\partial}{\partial\theta_j}f_k(\Theta^{(t)},x^{(i)})\\ &= -\frac{1}{n}\sum_{i=1}^{n}\Big(f(\Theta^{(t)},x^{(i)})-y^{(i)}\Big)\cdot\frac{1}{\sqrt{m}}\sum_{k:\ j\in\mathcal{N}_k}\frac{\partial}{\partial\theta_j}f_k(\Theta^{(t)},x^{(i)}).\end{aligned}$$

Thus, using (10) and the fact that there are a constant number of $\mathcal{N}_k$'s containing j, we find that

$$\begin{aligned}\big|\partial_t\theta_j^{(t)}\big| &\le \frac{1}{n}\sum_{i=1}^{n}\Big|f(\Theta^{(t)},x^{(i)})-y^{(i)}\Big|\cdot O\Big(\frac{c}{\sqrt{m}}\Big)\\ &\le \Big(\frac{1}{n}\sum_{i=1}^{n}\Big|f(\Theta^{(t)},x^{(i)})-y^{(i)}\Big|^2\Big)^{1/2}\cdot O\Big(\frac{c}{\sqrt{m}}\Big)\\ &= \Big(2L\big(\Theta^{(t)}\big)\Big)^{1/2}\cdot O\Big(\frac{c}{\sqrt{m}}\Big).\end{aligned}$$

The desired bound follows once we note that we are moving in the opposite direction of the gradient of $L\big(\Theta^{(t)}\big)$ via the gradient flow equationi, so $L\big(\Theta^{(t)}\big)\le L\big(\Theta^{(0)}\big)$.

(ii) Using (19) we have

$$\begin{aligned}\partial_t K_{\Theta^{(t)}}(x,x') &= \frac{1}{m}\sum_{k,k'=1}^{m}\sum_{j\in\mathcal{N}_k\cap\mathcal{N}_{k'}}\partial_t\bigg(\frac{\partial}{\partial\theta_j}f_k(\Theta^{(t)},x)\cdot\frac{\partial}{\partial\theta_j}f_{k'}(\Theta^{(t)},x')\bigg)\\ &= \frac{1}{m}\sum_{(k,k',j)\in\Gamma}\sum_{i=1}^{p}\partial_t\theta_i\cdot\frac{\partial}{\partial\theta_i}\bigg(\frac{\partial}{\partial\theta_j}f_k(\Theta^{(t)},x)\cdot\frac{\partial}{\partial\theta_j}f_{k'}(\Theta^{(t)},x')\bigg)\\ &= \frac{1}{m}\sum_{(k,k',j)\in\Gamma}\sum_{i\in\mathcal{N}_k\cup\mathcal{N}_{k'}}\partial_t\theta_i\cdot\frac{\partial}{\partial\theta_i}\bigg(\frac{\partial}{\partial\theta_j}f_k(\Theta^{(t)},x)\cdot\frac{\partial}{\partial\theta_j}f_{k'}(\Theta^{(t)},x')\bigg).\end{aligned}$$

By (10), for any i,j,k,k' we have

$$\bigg|\frac{\partial}{\partial\theta_i}\bigg(\frac{\partial}{\partial\theta_j}f_k(\Theta^{(t)},x)\cdot\frac{\partial}{\partial\theta_j}f_{k'}(\Theta^{(t)},x')\bigg)\bigg| = O(c^2) = O(1).$$

$$\big|\partial_t K_{\Theta^{(t)}}(x,x')\big| = O\bigg(\frac{1}{m}\sum_{(k,k',j)\in\Gamma}\sum_{i\in\mathcal{N}_k\cup\mathcal{N}_{k'}}\big|\partial_t\theta_i\big|\bigg).$$

Next, recall that $|\Gamma| = O(m)$, and for any k,k' the size of $\mathcal{N}_k\cup\mathcal{N}_{k'}$ is a constant. Thus, the desired bound follows from part (i).

(iii) To prove this part we borrow ideas from [6]. Using (5) we compute

$$\begin{aligned}\frac{1}{2}\partial_t\Delta^2(t) &= \frac{1}{n}\sum_{j=1}^{n}\big(\partial_t f(\Theta^{(t)},x^{(j)})-\partial_t\bar{f}(\bar{\Theta}^{(t)},x^{(j)})\big)\cdot\big(f(\Theta^{(t)},x^{(j)})-\bar{f}(\bar{\Theta}^{(t)},x^{(j)})\big)\\ &= -\frac{1}{n^2}\sum_{i,j=1}^{n}\Big(\big(f(\Theta^{(t)},x^{(i)})-y^{(i)}\big)K_{\Theta^{(t)}}(x^{(i)},x^{(j)})\big(f(\Theta^{(t)},x^{(j)})-\bar{f}(\bar{\Theta}^{(t)},x^{(j)})\big)\\ &\qquad -\big(\bar{f}(\bar{\Theta}^{(t)},x^{(i)})-y^{(i)}\big)K_{\Theta^{(0)}}(x^{(i)},x^{(j)})\big(f(\Theta^{(t)},x^{(j)})-\bar{f}(\bar{\Theta}^{(t)},x^{(j)})\big)\Big).\end{aligned}$$

Next, the fact that $K_{\Theta^{(0)}}$ is positive semidefinite and

$$\sum_{i,j=1}^{n}\big(f(\Theta^{(t)},x^{(i)})-\bar{f}(\bar{\Theta}^{(t)},x^{(i)})\big)K_{\Theta^{(0)}}(x^{(i)},x^{(j)})\big(f(\Theta^{(t)},x^{(j)})-\bar{f}(\bar{\Theta}^{(t)},x^{(j)})\big)\geq 0,$$

yields,

$$\begin{aligned}&\frac{1}{2}\partial_t\Delta^2(t)\\ &\leq -\frac{1}{n^2}\sum_{i,j=1}^{n}\Big(\big(f(\Theta^{(t)},x^{(i)})-y^{(i)}\big)K_{\Theta^{(t)}}(x^{(i)},x^{(j)})\big(f(\Theta^{(t)},x^{(j)})-\bar{f}(\bar{\Theta}^{(t)},x^{(j)})\big)\\ &\qquad -\big(f(\Theta^{(t)},x^{(i)})-y^{(i)}\big)K_{\Theta^{(0)}}(x^{(i)},x^{(j)})\big(f(\Theta^{(t)},x^{(j)})-\bar{f}(\bar{\Theta}^{(t)},x^{(j)})\big)\Big)\\ &= -\frac{1}{n^2}\sum_{i,j=1}^{n}\big(f(\Theta^{(t)},x^{(i)})-y^{(i)}\big)\cdot\big(K_{\Theta^{(t)}}(x^{(i)},x^{(j)})-K_{\Theta^{(0)}}(x^{(i)},x^{(j)})\big)\big(f(\Theta^{(t)},x^{(j)})-\bar{f}(\bar{\Theta}^{(t)},x^{(j)})\big)\\ &\leq \frac{1}{n^2}\big\|K_{\Theta^{(t)}}-K_{\Theta^{(0)}}\big\|\cdot\left(\sum_{i=1}^{n}\big(f(\Theta^{(t)},x^{(i)})-y^{(i)}\big)^2\right)^{1/2}\cdot\left(\sum_{j=1}^{n}\big(f(\Theta^{(t)},x^{(j)})-\bar{f}(\bar{\Theta}^{(t)},x^{(j)})\big)^2\right)^{1/2}\\ &= \frac{\sqrt{2}}{n}\big\|K_{\Theta^{(t)}}-K_{\Theta^{(0)}}\big\|\cdot L\big(\Theta^{(t)}\big)^{1/2}\cdot\Delta(t).\end{aligned}$$

We also note that $\frac{1}{2}\partial_t\Delta^2(t)=\Delta(t)\cdot\partial_t\Delta(t)$. Therefore,

$$\big|\partial_t\Delta(t)\big|\leq\frac{\sqrt{2}}{n}\big\|K_{\Theta^{(t)}}-K_{\Theta^{(0)}}\big\|\cdot L\big(\Theta^{(t)}\big)^{1/2}\leq\frac{\sqrt{2}}{n}\big\|K_{\Theta^{(t)}}-K_{\Theta^{(0)}}\big\|\cdot L\big(\Theta^{(0)}\big)^{1/2}.$$

Now, using part (ii) and the fact that $K_{\Theta^{(t)}}$ is an $n\times n$ matrix, we have

$$\frac{1}{n}\big\|K_{\Theta^{(t)}}-K_{\Theta^{(0)}}\big\|=O\left(\sqrt{\frac{L\big(\Theta^{(0)}\big)}{m}}\,t\right).$$

Therefore,

$$\big|\partial_t\Delta(t)\big|=O\big(L(\Theta^{(0)})t/\sqrt{m}\big),$$

which gives the desired result by integration.

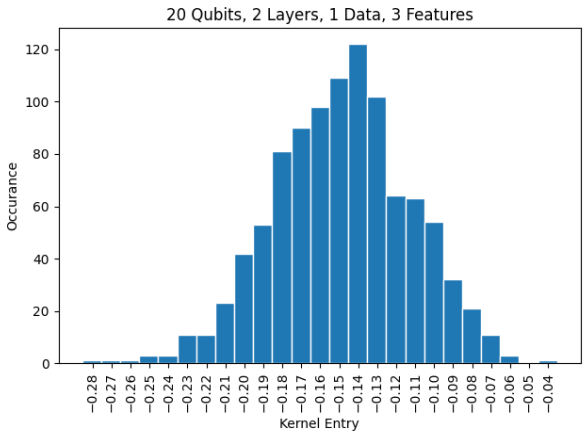


Figure 4: Histogram of $K_\Theta(x, x')$ for two fixed inputs x, x', where θ_j's are chosen independently and uniformly at random in $[-2\pi, 2\pi]$. This histogram corresponds to the quantum circuit of Figure 2 for $L = 2$ and $m = 20$. For this experiment, the analytical mean was found to be $\mathbb{E}[K_\Theta(x, x')] = -0.14842$ and the empirical mean was found to be $\bar{K}_\Theta(x, x') = -0.14854$. Both numbers are rounded up to the 5-th decimal point.

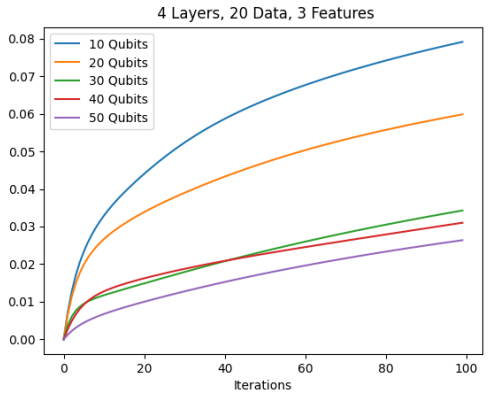


Figure 5: Evolution of $\frac{\|\Theta^{(t)} - \Theta^{(0)}\|}{\|\Theta^{(0)}\|}$ as a function of the number of iterations of the gradient descent algorithm. We observe that as the number of qubits m increases, the relative change of parameters decrease. This means that as the number of qubits increase, training enters the lazy regime.

(iv) Using the triangle inequality for the 2-norm, we have

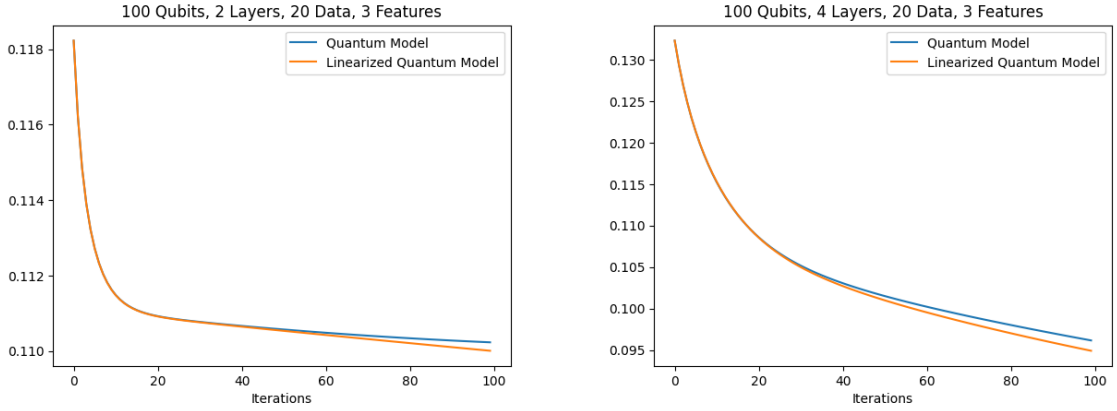
$$\begin{aligned}
\left|L(\Theta^{(t)}) - \bar{L}(\bar{\Theta}^{(t)})\right| &= \left(\sqrt{L(\Theta^{(t)})} + \sqrt{\bar{L}(\bar{\Theta}^{(t)})}\right) \cdot \left|\sqrt{L(\Theta^{(t)})} - \sqrt{\bar{L}(\bar{\Theta}^{(t)})}\right| \\
&\leq 2\sqrt{L(\Theta^{(0)})} \cdot \left(\frac{1}{2n} \sum_{i=1}^{n} \left(f(\Theta^{(t)}, x^{(i)}) - \bar{f}(\bar{\Theta}^{(t)}, x^{(i)})\right)^2\right)^{1/2} \\
&= \sqrt{2L(\Theta^{(0)})} \cdot \Delta(t) \\
&= O\left(\frac{L(\Theta^{(0)})^{3/2} t^2}{\sqrt{m}}\right).
\end{aligned}$$

□

5 Numerical simulations

In this section we present numerical simulations to support our results. To this end, we simulate the parameterized circuit of Figure 2, explained in detail in Section 3. To classically simulate this circuit for a large number of qubits (large m), we again use the idea of light cones (see Figure 3). To this end, we evaluate the model function term by term, knowing that each term can be computed by a sub-circuit of constant size (when L is constant). We use PennyLane [3] for our simulations.[1]

[1]Code accessible at https://github.com/phanous/quantum-lazy-training



(a) $L = 2$ (b) $L = 4$

Figure 6: Evolution of the loss function as a function of the number of iterations of gradient descent. These plots correspond to the quantum circuit of Figure 2 and its linearized version. We observe that for both cases of number of layers $L = 2$ and $L = 4$, the losses of the original quantum model and its linearized version are almost identical.

We choose the data set $D = \{(x^{(i)}, y^{(i)}) : i = 1, \dots, n\}$ randomly, where $x^{(i)}$'s are in $[-2\pi, 2\pi]$, and $y^{(i)}$'s are in $[-1, 1]$. We apply the gradient descent algorithm with a *learning rate* of $\eta = 1$ to train the circuit.

We first verify Theorem 1. We let $L = 2$, pick two random inputs x, x' and compute $K_\Theta(x, x')$ for random choices of θ_j in $[-2\pi, 2\pi]$. Figure 4 shows the histogram of these values. This histogram confirms that $K_\Theta(x, x')$ is concentrated around its average. This average is analytically computed in Appendix A, which shows

$$\mathbb{E}\left[K_\Theta(x, x')\right] = \frac{1}{4m} \sum_{k=1}^{m} 2\cos(\hat{x}_k)\cos(\hat{x}'_k) + \cos(\hat{x}_{k-1})\cos(\hat{x}_k)\cos(\hat{x}_{k+1})\cos(\hat{x}'_{k-1})\cos(\hat{x}'_k)\cos(\hat{x}'_{k+1}).$$

Next, in order to verify Theorem 2, we plot the relative change of the parameters Θ in the training process. That is, we plot

$$\frac{\|\Theta^{(0)} - \Theta^{(t)}\|}{\|\Theta^{(0)}\|},$$

where t denotes the number of gradient descent iterations. As Figure 5 shows, this relative change decreases by increasing the number of qubits m. This is an indicator of the occurrence of lazy training.

We also plot the loss functions $L(\Theta^{(t)}), \bar{L}(\Theta^{(t)})$ of both the original quantum model and its linearized version as functions of the number of iterations in Figure 6. We observe that for large numbers of qubits (e.g., $m = 100$), these two loss functions have almost the same values in every step of the learning process. This confirms our results in Theorem 2. Moreover, we observe that as suggested in Remark 3, these models converge very quickly.

Remark 4. We note that in the plots of Figure 6 the loss functions do not vanish as we increase the number of iterations. This is because, as mentioned above, the label $y^{(i)}$ for each data point $x^{(i)}$ is chosen randomly, and the quantum parameterized circuit chosen

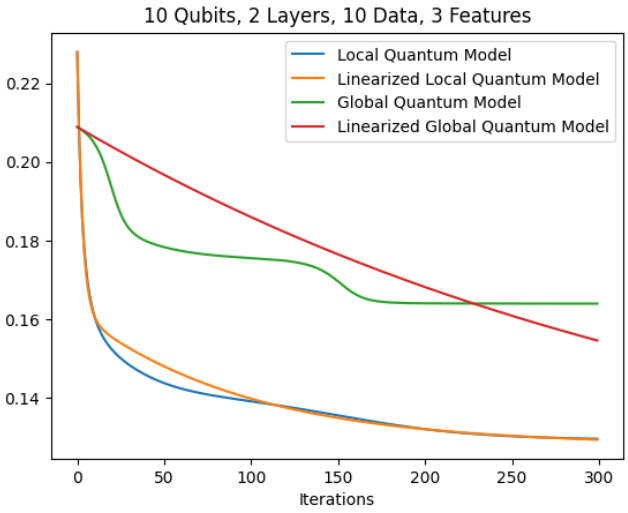


Figure 7: The evolution of the loss function in training for the quantum circuit of Figure 2 and its linearized version with local and global observables. We observe that while the model with a local observable enters the lazy regime, the original and the linearized model for the global observable become separated from each other as the original global model gets trapped in a barren plateau.

for our simulations is not expressive enough to fit such a random dataset. Alternatively, we can choose our dataset's inputs to be random $x^{(i)}$'s as before, and this time to fix the labels, pick random parameters Θ', feed the input $x^{(i)}$ to the parameterized circuit with parameters Θ', and let the outputs $y^{(i)}$ be the labels.[2] In this case, we make sure that our model is expressive enough to fit the dataset, and our simulations show that the loss function converges to zero as the number of iterations increase. Nevertheless, no matter how we choose the dataset, the behaviour of the loss functions of the original quantum and the linearized models remain the same and they decrease with an exponential rate with the number of iterations.

In order to justify our assumption that the observable is geometrically local, we also consider the circuit of Figure 2 with a *global observable*. We observe in Figure 7 that the quantum model with the global observable $O = Z_1 Z_2 \dots Z_m$ is separated from its linearized version. This shows that the assumption of the locality of observable is necessary for lazy training. Interestingly, we also observe that the linearized version of the quantum model with a global observable doesn't learn and remains almost constant. This is because, as can be verified by direct computations, the associated tangent kernel is a low-rank matrix, in which case the model function has a low expressive power.

6 Conclusion

In this paper, we proved that the training of parameterized quantum circuits that are geometrically local enters the lazy regime. This means that if the associated model

[2]We of course forget Θ' after fixing the labels.

function is rich enough, in which case the tangent kernel is full-rank and its eigenvalues are far from zero, training converges quickly.

We emphasize that although in our explicit example of parameterized quantum circuit the encoding is performed only in the first layer, our results hold for general forms of data encoding including parallel and sequential ones [18].

We proved our results under the assumptions that first, the circuit is geometrically local and second, the observable is a local operator. The first assumption is motivated by common hardware architectures, and numerical simulations suggest that the second assumption is necessary. Nevertheless, it is interesting to investigate other settings in which lazy training occurs in quantum machine learning. In particular, it would be interesting to study lazy training for quantum parameterized circuits whose number of qubits varies in different layers, i.e., fresh qubits are introduced and qubits are measured/discarded in the middle of the circuit [2].

Our results show that as long as the tangent kernel associated to a parameterized quantum circuit satisfying the above assumptions is full-rank and its minimum eigenvalue is far from zero, the quantum model is trained exponentially fast (see Remark 3). This is in opposite direction to barren plateaus occurring in the training of certain quantum parameterized circuits [12]. The point is that the circuits considered in our work are *not* random, and are geometrically local. Moreover, we consider only local observables, which remedies barren plateaus [5].

In this paper, we fixed the loss function to be the mean squared error, yet most of the results hold for more general loss functions as well. Indeed, for a general loss function we should only modify the proof of parts (iii) and (iv) of Theorem 2. Modifying these parts with weaker bounds, this can be done based on ideas in [6].

In the appendix, we explicitly compute the model function as well as the associated tangent kernel corresponding to a two-layer quantum circuit. We believe that such computations are insightful in understanding the expressive power of quantum parameterized circuits and their training properties.

References

[1] F. Bach. Effortless optimization through gradient flows. `https://francisbach.com/gradient-flows/`. Accessed: 2022-02-15.

[2] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf. Training deep quantum neural networks. *Nature communications*, 11(1):1–6, 2020.

[3] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

[4] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

[5] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. 12(1):1791, 2021.

[6] L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

[7] E. Farhi and H. Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

[8] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, Mar 2019.

[9] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

[10] J. Liu, F. Tacchino, J. R. Glick, L. Jiang, and A. Mezzacapo. Representation learning via quantum neural tangent kernels. *arXiv preprint arXiv:2111.04225*, 2021.

[11] D. Luo and J. Halverson. Infinite neural network quantum states. *arXiv preprint arXiv:2112.00723*, 2021.

[12] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):1–6, 2018.

[13] C. McDiarmid. On the method of bounded differences. In *Surveys in combinatorics, 1989 (Norwich, 1989)*, volume 141 of *London Math. Soc. Lecture Note Ser.*, pages 148–188. Cambridge Univ. Press, Cambridge, 1989.

[14] K. Nakaji, H. Tezuka, and N. Yamamoto. Quantum-enhanced neural networks in the neural tangent kernel framework. *arXiv preprint arXiv:2109.03786*, 2021.

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

[16] J. Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, Aug. 2018.

[17] M. Schuld and N. Killoran. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504, Feb 2019.

[18] M. Schuld, R. Sweke, and J. J. Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A*, 103:032430, Mar 2021.

[19] N. Shirai, K. Kubo, K. Mitarai, and K. Fujii. Quantum tangent kernel. *arXiv preprint arXiv:2111.02951*, 2021.

[20] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature communications*, 12(1):1–11, 2021.

A Explicit computation of $\mathbb{E}[K_\Theta(x,x')]$

In this Appendix, we explicitly compute $\mathbb{E}[K_\Theta(x,x')]$ for the parameterized circuit of Figure 2 with $L=2$ when θ_j's are chosen uniformly at random in $[-2\pi,2\pi]$. To this end, we first explicitly compute the model function, and then compute its associated tangent kernel.

Recall that

$$f(\Theta,x)=\frac{1}{\sqrt{m}}\sum_{k=1}^{m} f_k(\Theta,x),$$

where

$$f_k(\Theta,x)=\langle 0\dots 0|U_{\text{enc}}^\dagger(x)U^\dagger(\Theta)Z_kU(\Theta)U_{\text{enc}}(x)|0\dots 0\rangle.$$

We use (17) to compute $f_k(\Theta,x)$.

Lemma 2. *We have*

$$\begin{aligned}f_k(\Theta,x)=&\cos(\hat{x}_k)\cos(\theta_{m+k})\cos(\theta_k)\\&-\cos(\hat{x}_{k-1})\cos(\hat{x}_k)\cos(\hat{x}_{k+1})\cos(\theta_{k-1})\cos(\theta_{k+1})\sin(\theta_k)\sin(\theta_{m+k}).\end{aligned}$$

Proof. Using (17) we find that $f_k(\Theta,x)$ is the outcome of a sub-circuit acting on qubits $k-1,k,k+1$, where indices are modulo m. On the other hand, we have

$$Z_k=\sum_{b_{k-1},b_k,b_{k+1}\in\{0,1\}}(-1)^{b_k}\left|b_{k-1}b_kb_{k+1}\right\rangle\left\langle b_{k-1}b_kb_{k+1}\right|.$$

Therefore, by (17) we have

$$\begin{aligned}f_k(\Theta,x)=\sum_{b_{k-1},b_k,b_{k+1}\in\{0,1\}}(-1)^{b_k}\Big|&\left\langle b_{k-1}b_kb_{k+1}\right| R_{X_k}(\theta_{m+k})CZ_{k,k+1}CZ_{k-1,k}\\&R_X(\theta_{k-1},\theta_k,\theta_{k+1})R_Y(\hat{x}_{k-1},\hat{x}_k,\hat{x}_{k+1})\left|000\right\rangle\Big|^2,\end{aligned}$$

where

$$R_X(\theta_{k-1},\theta_k,\theta_{k+1})=R_{X_{k+1}}(\theta_{k+1})R_{X_k}(\theta_k)R_{X_{k-1}}(\theta_{k-1}),$$

and $R_Y(\hat{x}_{k-1},\hat{x}_k,\hat{x}_{k+1})$ is defined similarly. Using the notation

$$\left|(\theta,x)\right\rangle=R_X(\theta)R_Y(x)\left|0\right\rangle,$$

we have

$$R_X(\theta_{k-1},\theta_k,\theta_{k+1})R_Y(\hat{x}_{k-1},\hat{x}_k,\hat{x}_{k+1})\left|000\right\rangle=\left|(\theta_{k-1},\hat{x}_{k-1})\right\rangle\left|(\theta_k,\hat{x}_k)\right\rangle\left|(\theta_{k+1},\hat{x}_{k+1})\right\rangle.$$

We also have

$$\begin{aligned}&\left\langle b_{k-1}b_kb_{k+1}\right| R_{X_k}(\theta_{m+k})CZ_{k,k+1}CZ_{k-1,k}\\=&\left\langle b_{k-1}\right|\otimes\Big(\cos(\theta_{m+k}/2)\left\langle b_k\right|-i\sin(\theta_{m+k}/2)\left\langle b_k+1\right|\Big)\otimes\left\langle b_{k+1}\right| CZ_{k,k+1}CZ_{k-1,k}\\=&(-1)^{b_k(b_{k+1}+b_{k-1})}\left\langle b_{k-1}\right|\otimes\Big(\cos(\theta_{m+k}/2)\left\langle b_k\right|-i(-1)^{(b_{k+1}+b_{k-1})}\sin(\theta_{m+k}/2)\left\langle b_k+1\right|\Big)\otimes\left\langle b_{k+1}\right|\\=&(-1)^{b_k(b_{k+1}+b_{k-1})}\left\langle b_{k-1}\right|\otimes\left\langle(\theta_{m+k},b_k,b_{k+1}+b_{k-1})\right|\otimes\left\langle b_{k+1}\right|,\end{aligned}$$

where $|(\theta, b, s)\rangle$ is defined by

$$|(\theta, b, s)\rangle = \cos(\theta/2)\,|b\rangle + i(-1)^s \sin(\theta/2)\,|b+1\rangle\,.$$

Therefore,

$$f_k(\Theta, x) = \sum_{b_{k-1}, b_k, b_{k+1}} (-1)^{b_k} \Big| \langle (\theta_{m+k}, b_k, b_{k+1} + b_{k-1})|\,(\theta_k, \hat{x}_k)\rangle \Big|^2 \prod_{p \in \{k-1, k+1\}} \Big| \langle b_p\,|(\theta_p, \hat{x}_p)\rangle \Big|^2$$

Now, using part (ii) of Lemma 3, we obtain

$$\begin{aligned} &f_k(\Theta, x) \\ &= \sum_{b_{k-1}, b_{k+1}} \prod_{p \in \{k-1, k+1\}} \Big| \langle b_p\,|(\theta_p, \hat{x}_p)\rangle \Big|^2 \cos(\hat{x}_k) \Big(\cos(\theta_{m+k}) \cos(\theta_k) - (-1)^{b_{k-1} + b_{k+1}} \sin(\theta_{m+k}) \sin(\theta_k) \Big) \end{aligned}$$

Next, using the fact that $|(\theta_p, \hat{x}_p)\rangle$ is a normal vector and $\sum_{b_p} \big|\, \langle b_p|\,(\theta_p, \hat{x}_p)\big|^2 = 1$, as well as part (i) of Lemma 3 we find that

$$\begin{aligned} f_k(\Theta, x) = \cos(\hat{x}_k) \cos(\theta_{m+k}) \cos(\theta_k) \\ - \cos(\hat{x}_{k-1}) \cos(\hat{x}_k) \cos(\hat{x}_{k+1}) \cos(\theta_{k-1}) \cos(\theta_{k+1}) \sin(\theta_k) \sin(\theta_{m+k}). \end{aligned}$$

□

Now recall that the tangent kernel is given by

$$\begin{aligned} K_\Theta(x, x') &= \sum_{j=1}^{2m} \frac{\partial}{\partial \theta_j} f(\Theta, x) . \frac{\partial}{\partial \theta_j} f(\Theta, x') \\ &= \frac{1}{m} \sum_{j=1}^{2m} \sum_{k,k'=1}^{m} \frac{\partial}{\partial \theta_j} f_k(\Theta, x) . \frac{\partial}{\partial \theta_j} f_{k'}(\Theta, x'). \end{aligned}$$

Suppose that we choose $\theta_1, \ldots, \theta_{2m}$ independently and uniformly at random in $[-2\pi, 2\pi]$. We note that for such a random θ we have $\mathbb{E}[\cos(\theta)] = \mathbb{E}[\sin(\theta)] = 0$. Based on this and using Lemma 2, it is not hard to verify that

$$\mathbb{E}\left[\frac{\partial}{\partial \theta_j} f_k(\Theta, x) \cdot \frac{\partial}{\partial \theta_j} f_{k'}(\Theta, x') \right] = 0, \quad \forall k \neq k'.$$

Next, using the fact that $f_k(\Theta, x)$ depends only on parameters $\theta_{k-1}, \theta_k, \theta_{k+1}$ and θ_{m+k} we have

$$\mathbb{E}\left[K_\Theta(x, x')\right] = \frac{1}{m} \sum_{k=1}^{m} \sum_{j \in \{k-1, k, k+1, m+k\}} \mathbb{E}\Big[\frac{\partial}{\partial \theta_j} f_k(\Theta, x) \cdot \frac{\partial}{\partial \theta_j} f_k(\Theta, x') \Big].$$

Then, by Lemma 4 we find that

$$\begin{aligned} &\mathbb{E}\Big[K_\Theta(x, x')\Big] \\ &= \frac{1}{m} \sum_{k=1}^{m} \frac{2}{4} \cos(\hat{x}_k) \cos(\hat{x}'_k) + \frac{4}{16} \cos(\hat{x}_{k-1}) \cos(\hat{x}_k) \cos(\hat{x}_{k+1}) \cos(\hat{x}'_{k-1}) \cos(\hat{x}'_k) \cos(\hat{x}'_{k+1}) \\ &= \frac{1}{4m} \sum_{k=1}^{m} 2 \cos(\hat{x}_k) \cos(\hat{x}'_k) + \cos(\hat{x}_{k-1}) \cos(\hat{x}_k) \cos(\hat{x}_{k+1}) \cos(\hat{x}'_{k-1}) \cos(\hat{x}'_k) \cos(\hat{x}'_{k+1}). \end{aligned}$$

Auxiliary lemmas: Here we present some auxiliary lemmas needed in the above proofs in this appendix.

Lemma 3. *Let* $|(\theta,x)\rangle = R_X(\theta)R_Y(x)\,|0\rangle$ *and* $|(\theta,b,s)\rangle = \cos(\theta/2)\,|b\rangle + i(-1)^s\sin(\theta/2)\,|b+1\rangle$. *Then, the followings hold:*

(i) $\sum_{b\in\{0,1\}}(-1)^b\big|\langle b\,|(\theta,x)\rangle\,\big|^2 = \cos(\theta)\cos(x)$.

(ii) $\sum_{b\in\{0,1\}}(-1)^b\big|\,\langle(\theta,b,s)|\,(\theta',x)\rangle\big|^2 = \cos(x)\Big(\cos(\theta)\cos(\theta') - (-1)^s\sin(\theta)\sin(\theta')\Big)$.

Proof. (i) We have

$$\begin{aligned}
|(\theta,x)\rangle =& R_X(\theta)R_Y(x)\,|0\rangle \\
=& \Big(\cos(\theta/2)I - i\sin(\theta/2)X\Big)\Big(\cos(x/2)I - i\sin(x/2)Y\Big)\,|0\rangle \\
=& \Big(\cos(\theta/2)\cos(x/2) - i\sin(\theta/2)\sin(x/2)\Big)\,|0\rangle \\
&+ \Big(\cos(\theta/2)\sin(x/2) - i\sin(\theta/2)\cos(x/2)\Big)\,|1\rangle\,. \qquad (23)
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\sum_{b\in\{0,1\}}(-1)^b\big|\langle b\,|(\theta,x)\rangle\,\big|^2 \\
&= \cos^2\left(\frac{\theta}{2}\right)\cos^2\left(\frac{x}{2}\right) + \sin^2\left(\frac{\theta}{2}\right)\sin^2\left(\frac{x}{2}\right) - \cos^2\left(\frac{\theta}{2}\right)\sin^2\left(\frac{x}{2}\right) - \sin^2\left(\frac{\theta}{2}\right)\cos^2\left(\frac{x}{2}\right) \\
&= \left(\cos^2\left(\frac{\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right)\right)\left(\cos^2\left(\frac{x}{2}\right) - \sin^2\left(\frac{x}{2}\right)\right) \\
&= \cos(\theta)\cos(x).
\end{aligned}$$

(ii) Using (23), we have

$$\begin{aligned}
&\big|\,\langle(\theta,0,s)|\,(\theta',x)\rangle\big|^2 \\
&= \Big|\cos(\theta/2)\left(\cos(\theta'/2)\cos(x/2) - i\sin(\theta'/2)\sin(x/2)\right) \\
&\qquad - i(-1)^s\sin(\theta/2)\left(\cos(\theta'/2)\sin(x/2) - i\sin(\theta'/2)\cos(x/2)\right)\Big|^2 \\
&= \Big|\cos(\theta/2)\cos(\theta'/2)\cos(x/2) - (-1)^s\sin(\theta/2)\sin(\theta'/2)\cos(x/2) \\
&\qquad - i\left(\cos(\theta/2)\sin(\theta'/2)\sin(x/2) + (-1)^s\sin(\theta/2)\cos(\theta'/2)\sin(x/2)\right)\Big|^2 \\
&= \cos^2(\theta/2)\cos^2(\theta'/2)\cos^2(x/2) + \sin^2(\theta/2)\sin^2(\theta'/2)\cos^2(x/2) - \frac{(-1)^s}{2}\sin(\theta)\sin(\theta')\cos^2(x/2) \\
&\quad + \cos^2(\theta/2)\sin^2(\theta'/2)\sin^2(x/2) + \sin^2(\theta/2)\cos^2(\theta'/2)\sin^2(x/2) + \frac{(-1)^s}{2}\sin(\theta)\sin(\theta')\sin^2(x/2),
\end{aligned}$$

and

$$
\begin{aligned}
&\big|\,\langle(\theta,1,s)|\,(\theta',x)\rangle\big|^2\\
&\quad=\Big|\cos(\theta/2)\,(\cos(\theta'/2)\sin(x/2)-i\sin(\theta'/2)\cos(x/2))\\
&\qquad\qquad -i(-1)^s\sin(\theta/2)\,(\cos(\theta'/2)\cos(x/2)-i\sin(\theta'/2)\sin(x/2))\,\Big|^2\\
&\quad=\Big|\cos(\theta/2)\cos(\theta'/2)\sin(x/2)-(-1)^s\sin(\theta/2)\sin(\theta'/2)\sin(x/2)\\
&\qquad\qquad -i\,(\cos(\theta/2)\sin(\theta'/2)\cos(x/2)+(-1)^s\sin(\theta/2)\cos(\theta'/2)\cos(x/2))\,\Big|^2\\
&\quad=\cos^2(\theta/2)\cos^2(\theta'/2)\sin^2(x/2)+\sin^2(\theta/2)\sin^2(\theta'/2)\sin^2(x/2)-\frac{(-1)^s}{2}\sin(\theta)\sin(\theta')\sin^2(x/2)\\
&\qquad+\cos^2(\theta/2)\sin^2(\theta'/2)\cos^2(x/2)+\sin^2(\theta/2)\cos^2(\theta'/2)\cos^2(x/2)+\frac{(-1)^s}{2}\sin(\theta)\sin(\theta')\cos^2(x/2).
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
&\big|\,\langle(\theta,0,s)|\,(\theta',x)\rangle\big|^2-\big|\,\langle(\theta,1,s)|\,(\theta',x)\rangle\big|^2\\
&\quad=\Big(\cos^2(x/2)-\sin^2(x/2)\Big)\Big(\cos^2(\theta/2)\cos^2(\theta'/2)+\sin^2(\theta/2)\sin^2(\theta'/2)\\
&\qquad\qquad -\frac{(-1)^s}{2}\sin(\theta)\sin(\theta')-\cos^2(\theta/2)\sin^2(\theta'/2)\\
&\qquad\qquad -\sin^2(\theta/2)\cos^2(\theta'/2)-\frac{(-1)^s}{2}\sin(\theta)\sin(\theta')\Big)\\
&\quad=\Big(\cos^2(x/2)-\sin^2(x/2)\Big)\Big(\cos^2(\theta/2)\Big(\cos^2(\theta'/2)-\sin^2(\theta'/2)\Big)\\
&\qquad\qquad -\sin^2(\theta/2)\Big(\cos^2(\theta'/2)-\sin^2(\theta'/2)\Big)-(-1)^s\sin(\theta)\sin(\theta')\Big)\\
&\quad=\cos(x)\Big(\cos(\theta)\cos(\theta')-(-1)^s\sin(\theta)\sin(\theta')\Big).
\end{aligned}
$$

□

Lemma 4. *The followings hold:*

$$
\begin{aligned}
\mathbb{E}\Big[\frac{\partial}{\partial\theta_{k-1}}f_k(\Theta,x)\cdot\frac{\partial}{\partial\theta_{k-1}}f_k(\Theta,x')\Big]&=\mathbb{E}\left[\frac{\partial}{\partial\theta_{k+1}}f_k(\Theta,x)\cdot\frac{\partial}{\partial\theta_{k+1}}f_k(\Theta,x')\right]\\
&=\frac{1}{16}\cos(\hat{x}_{k-1})\cos(\hat{x}_k)\cos(\hat{x}_{k+1})\cos(\hat{x}'_{k-1})\cos(\hat{x}'_k)\cos(\hat{x}'_{k+1}),
\end{aligned}
$$

and

$$
\begin{aligned}
\mathbb{E}\Big[\frac{\partial}{\partial\theta_k}f_k(\Theta,x)\cdot\frac{\partial}{\partial\theta_k}f_k(\Theta,x')\Big]&=\mathbb{E}\left[\frac{\partial}{\partial\theta_{m+k}}f_k(\Theta,x)\cdot\frac{\partial}{\partial\theta_{m+k}}f_k(\Theta,x')\right]\\
&=\frac{1}{4}\cos(\hat{x}_k)\cos(\hat{x}'_k)+\frac{1}{16}\cos(\hat{x}_{k-1})\cos(\hat{x}_k)\cos(\hat{x}_{k+1})\cos(\hat{x}'_{k-1})\cos(\hat{x}'_k)\cos(\hat{x}'_{k+1}).
\end{aligned}
$$

Proof. We note that

$$\mathbb{E}\left[\cos^2(\theta)\right] = \mathbb{E}\left[\sin^2(\theta)\right] = \mathbb{E}\Big[\Big(\frac{\partial}{\partial\theta}\cos(\theta)\Big)^2\Big] = \mathbb{E}\Big[\Big(\frac{\partial}{\partial\theta}\sin(\theta)\Big)^2\Big] = \frac{1}{2},$$

and

$$\mathbb{E}\left[\sin(\theta)\cos(\theta)\right] = \mathbb{E}\left[\frac{\partial}{\partial\theta}\sin(\theta)\cdot\frac{\partial}{\partial\theta}\cos(\theta)\right] = 0.$$

Then, using Lemma 2 we compute:

$$\begin{aligned}
&\mathbb{E}\Big[\frac{\partial}{\partial\theta_{k-1}}f_k(\Theta,x)\cdot\frac{\partial}{\partial\theta_{k-1}}f_k(\Theta,x')\Big] \\
&\qquad = \cos(\hat{x}_{k-1})\cos(\hat{x}_k)\cos(\hat{x}_{k+1})\cos(\hat{x}'_{k-1})\cos(\hat{x}'_k)\cos(\hat{x}'_{k+1}) \\
&\qquad\qquad \cdot \mathbb{E}\left[\Big(\frac{\partial}{\partial\theta_{k-1}}\cos(\theta_{k-1})\Big)^2\cdot\cos^2(\theta_{k+1})\cdot\sin^2(\theta_k)\cdot\sin^2(\theta_{m+k})\right] \\
&\qquad = \frac{1}{16}\cos(\hat{x}_{k-1})\cos(\hat{x}_k)\cos(\hat{x}_{k+1})\cos(\hat{x}'_{k-1})\cos(\hat{x}'_k)\cos(\hat{x}'_{k+1}).
\end{aligned}$$

Next, we have

$$\begin{aligned}
\frac{\partial}{\partial\theta_k}f_k(\Theta,x) &= \cos(\hat{x}_k)\cos(\theta_{m+k})\frac{\partial}{\partial\theta_k}\cos(\theta_k) \\
&\quad - \cos(\hat{x}_{k-1})\cos(\hat{x}_k)\cos(\hat{x}_{k+1})\cos(\theta_{k-1})\cos(\theta_{k+1})\frac{\partial}{\partial\theta_k}\sin(\theta_k)\sin(\theta_{m+k}).
\end{aligned}$$

Therefore, since

$$\mathbb{E}\Big[\frac{\partial}{\partial\theta_k}\cos(\theta_k)\cos(\theta_{k-1})\cos(\theta_{k+1})\frac{\partial}{\partial\theta_k}\sin(\theta_k)\sin(\theta_{m+k})\Big] = 0,$$

we have

$$\begin{aligned}
&\mathbb{E}\Big[\frac{\partial}{\partial\theta_k}f_k(\Theta,x)\cdot\frac{\partial}{\partial\theta_k}f_k(\Theta,x')\Big] \\
&\qquad = \cos(\hat{x}_k)\cos(\hat{x}'_k)\mathbb{E}\Big[\cos^2(\theta_{m+k})\Big(\frac{\partial}{\partial\theta_k}\cos(\theta_k)\Big)^2\Big] \\
&\qquad\quad + \cos(\hat{x}_{k-1})\cos(\hat{x}_k)\cos(\hat{x}_{k+1})\cos(\hat{x}'_{k-1})\cos(\hat{x}'_k)\cos(\hat{x}'_{k+1}) \\
&\qquad\qquad \cdot \mathbb{E}\Big[\cos^2(\theta_{k-1})\cos^2(\theta_{k+1})\Big(\frac{\partial}{\partial\theta_k}\sin(\theta_k)\Big)^2\sin^2(\theta_{m+k})\Big] \\
&\qquad = \frac{1}{4}\cos(\hat{x}_k)\cos(\hat{x}'_k) + \frac{1}{16}\cos(\hat{x}_{k-1})\cos(\hat{x}_k)\cos(\hat{x}_{k+1})\cos(\hat{x}'_{k-1})\cos(\hat{x}'_k)\cos(\hat{x}'_{k+1}).
\end{aligned}$$

The other two equations hold by symmetry. □