IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 015058 https://doi.org/10.1088/2632-2153/ad35a3

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
5 May 2023

REVISED
5 December 2023

ACCEPTED FOR PUBLICATION
19 March 2024

PUBLISHED
2 April 2024

PAPER

Laziness, barren plateau, and noises in machine learning

Junyu Liu[1,2,3,4,5,*], **Zexi Lin**[1] **and Liang Jiang**[1]

[1] Pritzker School of Molecular Engineering, The University of Chicago, Chicago, IL 60637, United States of America
[2] Department of Computer Science, The University of Chicago, Chicago, IL 60637, United States of America
[3] Kadanoff Center for Theoretical Physics, The University of Chicago, Chicago, IL 60637, United States of America
[4] qBraid Co., Harper Court 5235, Chicago, IL 60615, United States of America
[5] SeQure, Chicago, IL 60615, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: junyuliu@uchicago.edu, zexil@uchicago.edu and liangjiang@uchicago.edu

Keywords: quantum machine learning, machine learning theory, quantum algorithms

Supplementary material for this article is available online

Abstract

We define *laziness* to describe a large suppression of variational parameter updates for neural networks, classical or quantum. In the quantum case, the suppression is exponential in the number of qubits for randomized variational quantum circuits. We discuss the difference between laziness and *barren plateau* in quantum machine learning created by quantum physicists in McClean *et al* (2018 *Nat. Commun.* **9** 1–6) for the flatness of the loss function landscape during gradient descent. We address a novel theoretical understanding of those two phenomena in light of the theory of neural tangent kernels. For noiseless quantum circuits, without the measurement noise, the loss function landscape is complicated in the overparametrized regime with a large number of trainable variational angles. Instead, around a random starting point in optimization, there are large numbers of local minima that are good enough and could minimize the mean square loss function, where we still have quantum laziness, but we do not have barren plateaus. However, the complicated landscape is not visible within a limited number of iterations, and low precision in quantum control and quantum sensing. Moreover, we look at the effect of noises during optimization by assuming intuitive noise models, and show that variational quantum algorithms are noise-resilient in the overparametrization regime. Our work precisely reformulates the quantum barren plateau statement towards a precision statement and justifies the statement in certain noise models, injects new hope toward near-term variational quantum algorithms, and provides theoretical connections toward classical machine learning. Our paper provides conceptual perspectives about quantum barren plateaus, together with discussions about the gradient descent dynamics in Liu *et al* (2023 *Phys. Rev. Lett.* **130** 150601).

1. Barren plateau, laziness and noise

Variational quantum circuits [1–6] can be used to optimize cost function measured on quantum computers. Specifically, these cost functions can be used for machine learning tasks [7–14]. In this case variational quantum circuits are addressed as quantum neural networks.

However, a generically designed variational quantum ansatz may not be applicable to real problems. Specifically, a problem so-called *barren plateau* has been widely discussed in the variational quantum algorithm community, which is believed to be one of the primary problems of quantum machine learning [15]. The argument is given as follows. A typical gradient descent algorithm will look like

$$\theta_\ell(t+1) - \theta_\ell(t) \equiv \delta\theta_\mu = -\eta \frac{\partial \mathcal{L}}{\partial \theta_\ell}, \tag{1}$$

where θ_μ is the variational angle, and t is referring the time step of gradient descent dynamics. η is the learning rate, and $\mathcal{L}$ is the loss function. The observation [15] is that, if our variational ansatz is highly

random, due to the k-design integral formula [16–19], the derivative of the loss function is generically suppressed by the dimension of the Hilbert space N, and we might encounter a situation where the variation of the loss function during gradient descent is very small, namely $\delta\mathcal{L} \equiv \mathcal{L}(t+1) - \mathcal{L}(t) \ll 1$ for the step t. For instance, the second moment formula for Haar ensemble is

$$\int \mathrm{d}U U_{ij} U_{kl}^{\dagger} = \frac{1}{N}\delta_{il}\delta_{jk} \,. \tag{2}$$

Here U is a unitary taken from a 1-design, and δ is the Kronecker delta and i, j, k, l are matrix indexes. For higher moments random integrals [16–20], the factor poly$(1/N)$ will appear. Thus, the difference between the variational angles during iterations will be suppressed by the dimension of the Hilbert space. The work [15] demonstrates this existence of the *barren plateau* (**the statement where** $\delta\mathcal{L} \ll 1$) numerically and understands the result as a primary challenge of variational quantum circuits. It is often considered to be quantum analogs to the *vanishing gradient problem*, but the nature is fundamentally different [21, 22]. A further explanation is given in appendix A.

Although the existence of the barren plateau is verified by numerous works [23–26], the theoretical understanding of the barren plateau problem is unclear. Moreover, the classical machine learning community has been successfully demonstrated its practical usage in science and business for years, and many successful classical neural network algorithms have been run for large scales. For example, `Generative Pre-trained Transformer-3` (`GPT-3`) from `OpenAI` [27] has used 175 billion of training parameters, and it is one of the most successful natural language processing models up to date. Considering the standard LeCun initialization of weights W with the normalization of the variance σ_W^2 [21, 22, 28]

$$\mathbb{E}\left(W_{ij}W_{kl}^{\dagger}\right) = \frac{\sigma_W^2}{\text{width}}\delta_{ik}\delta_{jl} \,, \tag{3}$$

and its formal similarity to equation (2), we might imagine that similar issues will happen for classical neural networks too: they might be highly overparametrized in the large-width limit. Here, σ_W is a number that is independent of the size of the neural networks, and we set the width of the neural network to be the same in each layer for simplicity. In fact, in appendix A, we will show that in the classical large-width neural network, the barren plateau will also happen: the trainable weights do not run that much during gradient descent.

So, why classical overparametrized neural networks are supposed to be practical and good, but the barren plateaus of quantum neural networks are crucial challenges? **In this paper, we define the primary theoretical argument towards the quantum barren plateau, the large suppression of the right hand side of equation (1), as *laziness*.** In the quantum context, the suppression is from the dimension of the Hilbert space, while in the classical case, the suppression is from the width of the classical neural networks. In a more precise language, laziness is referring to small $\delta\theta_\mu$, and barren plateau is referring to small $\delta\mathcal{L}$.

Moreover, we will show that laziness may not imply the quantum barren plateau, from the perspective of overparametrization theory and representation learning theory through quantum neural tangent kernels (QNTKs) [28, 29]. In this paper, for quantum neural networks *overparametrization* is referring to the fact where $L\mathrm{Tr}(O^2)/N^2 \approx \mathcal{O}(1)$, where O is the operator we are optimizing, L is the number of trainable angles, and η is the learning rate as a constant.

Defining quantum analogs of neural tangent kernels (NTKs) from their classical counterparts [22, 30–40], we show that from the first-principle theoretical derivation, random (noiseless) quantum neural networks are still efficient to learn in the large-L limit without barren plateaus, despite their laziness. In fact, although each trainable angle does not move much due to the small magnitude of the gradient, the combined effect of many of them on the loss function will still be significant. In addition, there exist good enough achievable local minima that minimize the training error. See figure 1 for an illustration. The requirements for making this to happen is especially when $L\mathrm{Tr}(O^2)/N^2 \approx \mathcal{O}(1)$, and we have a small learning rate and the mean square loss function. In the case of large Hilbert space dimension without overparametrization, the exponential decay rate during gradient descent might be small, which may not make this phenomenon manifest in the polynomial training iterations. In practice, what we see is a very slow decay of loss functions. Interestingly, in this case quantum noises will not affect us significantly until exponential numbers of iterations. Thus, the averaged QNTK, $\bar{K}$, proportional to $\mathrm{Tr}(O^2)L/N^2$, *explains* the existence of the barren plateau in practice, with or without noises. On the other hand, in the overparametrization regime where $\eta L\mathrm{Tr}(O^2)/N^2 \approx \mathcal{O}(1)$, the exponential decay of gradient descent process is visible.

We note that the large-L expansion is a quantum analog of the classical NTK theory at large width. In fact, we will show in section 3 that we have similar large-width expansion comparing the classical theory, where in our model, *classical width* corresponds to L. The dimension of the Hilbert space plays an important role in the calculation. Moreover, the correspondence between quantum and classical neural networks might

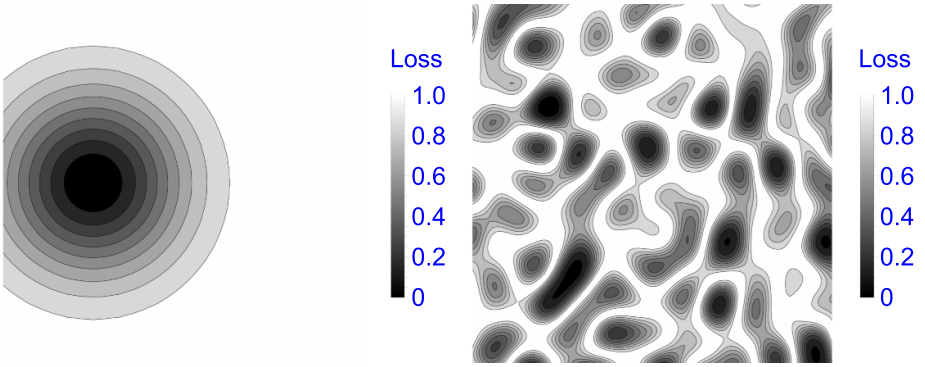

Figure 1. Density plots of the loss function landscape comparing usual and overparametrized variational quantum circuits. We illustrate the landscape by color plots of the loss function for two variational angles. Left: the traditional understanding of *barren plateaus* where we have the a single optimal point. Right: in the overparametrized case, the landscape is not barren, since for a random initial point, we get many good enough local optima that could minimize the loss function. Note that those plots are schematic since it is not possible to directly plot the loss function landscape in very high dimensions, and it might be interesting to visualize the true landscape using some visualization tools in future works (like [41]). Classically, there are discussions about significant numbers of local minima provided in the literature like [42, 43], and it might be interesting to see them in quantum architectures. In order to visualize it in $\mathcal{O}(1)$ numbers of iterations, one might have to have the number of trainable angles L comparable to the dimension of the Hilbert space N.

be explained by some physical heuristics, from the duality between matrix models and quantum field theories. See appendix C for a brief discussion.

Moreover, we need to point out that laziness is intrinsically still a precision problem. More precisely, it could be primarily from quantum measurement and quantum control, since the size of classical devices could scale as $\log(1/\epsilon)$ for given precision ϵ, while variational quantum circuits cannot, due to the measurement error and the limitation of quantum control [15]. Thus, it naturally motivates us to think about how to include the effect of noise in the gradient descent calculation. In our work, we introduce a simple and intuitive noise model by adding random variables in the gradient descent dynamics. We show that in the overparametrization regime, our variational quantum algorithms are noise-resilient. More precisely, we find that the residual training error scales as

$$\varepsilon^2(t) \approx (1-\eta K)^{2t}\left(\varepsilon^2(0) - \frac{\sigma_\theta^2}{\eta(2-\eta K)}\right) + \frac{\sigma_\theta^2}{\eta(2-\eta K)}\,, \tag{4}$$

with the NTK K and the standard deviation of the noise introduced in the variational angles σ_θ. Thus, in the late time, we get

$$\mathcal{L}(\infty) = \frac{1}{2}\varepsilon^2(\infty) \approx \frac{\sigma_\theta^2}{2\eta(2-\eta K)}\,. \tag{5}$$

In the late time, we have

$$\mathcal{L}(\infty) = \frac{1}{2}\varepsilon^2(\infty) \approx \frac{\sigma_\theta^2}{2\eta(2-\eta K)}\,. \tag{6}$$

Thus, in the overparametrized regime, we could set $\eta K \approx \mathcal{O}(1)$, so schematically,

$$\mathcal{L}(\infty) \approx \mathcal{O}\left(\frac{\sigma_\theta^2}{\eta}\right)\,, \tag{7}$$

indicating that we could get good predictions at the end as long as we sufficiently control the noises.

We will give more details in the following sections.

This paper is mostly written for audiences in the area of quantum computing and quantum machine learning. However, some of the discussions are also applicable in a general classical machine learning setup. For more general audiences, we give a small introduction on quantum computing, quantum machine learning, k-design theories for random unitaries, the notations and key defintions used in our paper. Combined with comparisons to other works, introduction of the backgrounds is provided in section 2. In section 3, we discuss the theory of QNTK and its relation on laziness. In section 4 we discuss precision of variational parameters and the noise. In section 5, we provide overviews on our findings. Some technical results and numerical experiments are summarized in appendix.

2. Backgrounds, definitions and related works

In this section, we provide background reviews, definitions, and related works. The backgrounds are summarized in sections 2.1–2.3, from quantum computing, quantum machine learning, to k-design theories in quantum information science. Readers could skip those sections if they are familiar with the corresponding theories.

2.1. Quantum mechanics, computing and noises

We start with a short introduction to quantum computing with the language of linear algebra for readers who are not familiar with physics. Further details could be found in standard text books, including [44].

In quantum mechanics, physical states are represented as vectors, where we denote them with the so-called Dirac notation $|a\rangle$. We will only consider the vectors living in the linear space with the complex dimension $N = 2^n$, and n is called the number of qubits. One can expand the vector $|a\rangle$ through the basis expansion, $|a\rangle = \sum_{i=0}^{N-1} a_i |i\rangle$, where $|i\rangle$ is the basis vector and a_i is the coefficients. Here, we only consider finite-dimensional vectors, so we could define an inner product of vectors. One can specify the dual vector space by the space of linear operations on the states, where we write the dual vectors as $\langle a|$. The inner product is defined as $\langle a|b\rangle = \sum_i a_i^* b_i$ where the basis states are orthogonal, and the vector space becomes a Hilbert space. Physical states are, in fact, normalized vectors in the Hilbert space.

In quantum mechanics, observables like energy and momentum are represented by Hermitian operators in the Hilbert space. For a given operator O, like energies, the operator will have the eigenspace expansion with the eigenvector $|o_i\rangle$ and the eigenvalue o_i. One could write the eigenspace expansion as $O = \sum_i o_i |o_i\rangle \langle o_i|$. $|o_i\rangle$ could form a complete basis in the whole Hilbert space, and for an arbitrary state as a normalized vector $|\psi\rangle$, we could expand $|\psi\rangle = \sum_i \psi_i |o_i\rangle$. This expansion has a physical meaning: the number $|\psi_i|^2$ represents the probability of observing the eigenvalue o_i when we observe the operator O in the state $|\psi\rangle$. Since O is Hermitian, all the eigenvalues are real, so we observe real observables. Since the state $|\psi\rangle$ is normalized, we have $\sum_i |\psi_i|^2 = 1$, satisfying the definition of probability (it is called the Born rule). The above rules of quantum mechanics are verified by all experiments in the physical world as far as we know.

Moreover, the quantum dynamics, namely the time (t) evolution of quantum states, is given by the unitary operator $U(t)$ in the Hilbert space, acting on the state $|\psi\rangle$. From linear algebra, we know that $U(t)|\psi\rangle$ is always satisfying the normalization condition of probabilities, and thus the total probability is conserved (always 1). Moreover, the unitary operator $U(t)$ could be exponentiated, and the Schrödinger equation states that $U(t) = \exp(-iHt)$, where H is called the Hamiltonian (a Hermitian operator stands for the energy in a system).

The task of quantum computing is that one could use physical states like $|\psi\rangle$ to encode the information, and a quantum algorithm is a unitary operator U that is made by a sequence of some basic physical operators. The sequence, which is called a quantum circuit, has to be running in polynomial time with respect to the number of qubits n. At the end of a quantum algorithm U, we get $U|\psi\rangle$ starting from a state $|\psi\rangle$, and if needed, we perform physical measurements on $|\psi\rangle$ with respect to an operator O to get the expectation value, $\langle\psi| U^\dagger O U |\psi\rangle$, according to the Born rule. In this case, the output of the quantum algorithm is a classical number.

Quantum computing has some potential to perform better than its classical counterparts in certain problems since $N = 2^n$ is very large and exponential in n. A typical example is Shor's algorithm [45], which factors large numbers exponentially faster than known classical algorithms. However, quantum computing is very challenging to realize with the existing technologies since quantum states are fragile. There are lots of noises that could happen to destroy the programmed unitary operator U, like environmental affections that could *decohere* quantum states to some classical objects. In general, quantum error corrections and error correction codes are needed to perform fault-tolerant quantum computation. Moreover, there are measurement noises due to the probabilistic nature of the Born rule. Thus, in general, quantum systems are hard to control, and we are still working in progress, toward large-scale, fault-tolerant quantum computing.

2.2. Quantum machine learning

It is natural to think about how quantum computing could help solve important machine learning tasks. Currently, one of the leading paradigms is so-called the variational quantum algorithms, that is the closest to the classical machine learning paradigm with algorithms like backpropagation and gradient descent dynamics [7–14].

Fundamental gates are unitary operations defined in quantum computing. In the Hilbert space with dimension $N = 2^n$, a typical type of quantum gate is called the Pauli gate. We define,

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \ , \ \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \ , \ \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} . \tag{8}$$

They are fundamental gates that are physically implementable in quantum computing devices. There are more fundamental gates like CNOT (see [44] for details). For n qubits, one could define local Pauli gates as

$$\sigma_x \otimes I_{2^{n-1}}, I_2 \otimes \sigma_y \otimes I_{2^{n-2}}, I_{2^2} \otimes \sigma_y \otimes I_{2^{n-3}}, \dots . \tag{9}$$

They are made by the Kronecker products between the 2-dimensional Pauli gates $\sigma_{x,y,z}$ and the identity matrix. We denote d-dimensional identity matrix as I_d. Thus, n-qubit local Pauli gates are quantum operations in the whole n-qubit Hilbert space.

Moreover, one could define the variational quantum circuit as,

$$U(\theta) = \left(\prod_{\ell=1}^{L} W_\ell \exp\left(i\theta_\ell X_\ell\right) \right) \equiv \left(\prod_{\ell=1}^{L} W_\ell U_\ell \right) . \tag{10}$$

Here, we have L-dimensional real training parameters (variational angles) $\theta_{\ell=1}^{L}$. X_ℓ represents some implementable quantum gates, like the local Pauli gates introduced before. They are Hermitian operators with $X_\ell^2 = I_N$. Moreover, W_ℓ represents some unitary operators that do not depend on the variational angle, like the CNOT gate. Thus, $U(\theta)$ is a unitary operator parametrized by $\theta = \theta_{\ell=1}^{L}$.

Based on the operator $U(\theta)$, we could define the loss function,

$$\mathcal{L}(\theta) = \frac{1}{2}\left(\left\langle \Psi_0 \right| U^\dagger(\theta) O U(\theta) \left| \Psi_0 \right\rangle - O_0 \right)^2 , \tag{11}$$

similar to the mean square loss used in classical machine learning. Here, O_0 is a constant, $|\Psi_0\rangle$ is a quantum state that is prepared in a quantum computer, and O is an observable (a Hermitian operator like energy). The above loss function is evaluated using quantum measurements in the quantum computer, and it is parametrized by classical variational parameters θ. One could perform gradient descent algorithms to minimize $\mathcal{L}(\theta)$, which is,

$$\delta\theta_\ell(t) \equiv \theta_\ell(t+1) - \theta_\ell(t) = -\eta \frac{\partial \mathcal{L}}{\partial \theta_\ell}(t) \ , \tag{12}$$

with the number of iterations t, and the learning rate η. Note that here, the derivative of the loss function is evaluated using quantum measurements, and we use the measurement result to update the classical parameter θ that parametrize the quantum circuit. Thus, variational quantum algorithms are called hybrid quantum–classical algorithms.

The process of optimizing the loss function $\mathcal{L}$ is similar to unsupervised learning in the classical machine learning literature. It is easy to extend the task towards supervised learning, if we encode classical or quantum data sets into the state $|\Psi_0\rangle$. Moreover, the construction of $U(\theta)$, also called the quantum neural network, is analogous to classical neural networks. Mathematically, $U(\theta)$ could be interpreted as a single-layer classical neural network where L is the width, but the possible entanglement structure in $U(\theta)$ makes it different to the data structure in classical machine learning.

2.3. *k*-designs

Another background knowledge in this paper is the so-called k-design theory. See [16] for a more detailed introduction.

Classical weights and biases can be initialized using Gaussian distributions, like Kaiming initialization in classical machine learning. In quantum machine learning, one could randomly initialize variational angle θ. An architecture-independent treatment is to study the uniform distribution of the unitary group $\mathrm{U}(N)$. The uniform measure in $\mathrm{U}(N)$ is unique, and it is called the Haar measure.

k-designs are approximations towards the Haar measure in the unitary group, where ks are integers. More precisely, k-designs are ensembles of unitary operators that could reproduce the matrix element correlation functions up to $2k$ moments. Thus, a $(k+1)$-design is always a k-design. The larger k is, the closer the ensemble is to the Haar measure.

One of the equivalent definitions of k-designs is the following. An ensemble (collection of unitaries) $\mathcal{E}$ is a k-design of the unitary group $\mathrm{U}(N)$, if and only if,

$$\int_{U\in\mathrm{Haar}} \mathrm{d}U U^{\dagger\otimes k}\rho U^{\otimes k} = \int_{U\in\mathcal{E}} \mathrm{d}U U^{\dagger\otimes k}\rho U^{\otimes k}\,, \tag{13}$$

for all ρ, where $U^{\otimes k}$ is the k-fold Kronecker product of the same U, and ρ is an arbitrary density matrix (a positive definite Hermitian matrix with trace 1 in the Hilbert space with the dimension N^k). If $\mathcal{E}$ is a k-design, one could show that,

$$\int_{U\in\mathcal{E}} \mathrm{d}U \left(U_{i_1j_1}U^{\dagger}_{i_2j_2}\dots U_{i_{2k-1}j_{2k-1}}U^{\dagger}_{i_{2k}j_{2k}}\right) = \mathcal{O}\left(\frac{1}{\mathrm{poly}\,(N)}\right)\,, \tag{14}$$

where $\mathrm{poly}(N)$ is a fixed, computable polynomial with the degree k. This formula is very useful when we study random averaging properties of variational quantum circuits, just like the Gaussian process properties of classical large-width neural networks. Assuming 2-designs, one could obtain the so-called barren plateau problem [15]. Moreover, k-designs are implementable in practical quantum circuits. For instance, the Pauli group will form 1-design, and the Clifford group will form 2-design [16]. One could allow some errors and define approximate k-designs, and it is shown that for local random circuits, it will converge towards k-designs approximately with polynomial time [46].

2.4. Symbols

In this section, we provide a list of definitions for readers.

Definitions	Notations
Dimension of Hilbert space	N
Number of qubits	$n = \log_2 N$
Variational angles	θ
index of variational angles	ℓ
Number of trainable angles	L
Loss function	$\mathcal{L}$
Quantum observables	O
Quantum states	$\lvert\cdots\rangle, \langle\cdots\rvert$
Variational circuits	$U(\theta)$
The ℓth trainable gate	X_ℓ
The ℓth fixed gate	W_ℓ
Residual training error	ε
Initial states	$\lvert\Psi_0\rangle$
Noise standard deviation	σ_θ
QNTK	K
dQNTK	μ
Total training step	T
Generic training step	t
Change of a quantity o between $t+1$ and t	δo
Learning rate	η
Relative training error at the time T	$\varepsilon_r = \varepsilon(T)/\varepsilon(0)$
Average of o	$\mathbb{E}(o)$ or $\bar{o}$
Standard deviation of o	Δo

2.5. Related works

In this section, we briefly summarize some related works and our contributions related to those works.

The study of barren plateaus starts [15] by quantum physicists, where 2-design assumptions are used for variational quantum algorithms. Our contribution provides an alternative understanding of the barren plateau problem from the vision of the QNTK theory, and a more refined definition, *laziness*, in order to clarify the relation from traditional barren plateaus. The papers [28, 29] initialize the study of QNTK theory. Compared to those works, in this paper, we clearly clarify the relationship between barren plateau and laziness, and technically we discuss how precision and noises will affect the calculation of QNTK.

Anschuetz and Kiani [47] proposes a related result from different theoretical backgrounds. The paper claims that for shallow quantum circuits, there will be lots of traps in the loss function landscape: there are only a small fraction of local minima that are good enough. On the other hand, our paper shows that when the model goes deeper and deeper (when we have larger and larger L), those local minima become better and better, such that they are good enough to minimize the loss function. Abedi *et al* [48] expresses some similar ideas to our work. The paper shows that for local variational quantum circuits, one could obtain a lazy phase with exponentially converging training dynamics. Note that geometric locality makes the system away from barren plateaus, and the laziness, defined according to our paper, is polynomial instead of exponential. However, geometric locality makes the variational circuits have lower expressibility, and the convergence of the loss function will have different behaviors at the late time. Thus, [48] is not focusing on discussions about relationships between laziness and barren plateaus. Both of the papers are complementary to our results. Moreover, [49] introduces the term *lazy training* and discuss its relation to the scale of neural networks. Our work, based on [49], discusses how it is related to quantum machine learning, barren plateaus and noises.

2.6. Refined definition about laziness and barren plateau

Although partially introduced in the beginning of the paper (see section 1), in this part we clearly clarify the two core definitions in our paper, *barren plateau* and *laziness*.

Say that we start from a loss function $\mathcal{L}(\theta)$ with the variational angles (training parameters) $\theta = (\theta_\mu)$. We define:

Definition 2.1 (Barren plateau). We say there is a **barren plateau** when $\delta\mathcal{L} \equiv \mathcal{L}(t+1) - \mathcal{L}(t)$ satisfies $|\delta\mathcal{L}| \ll 1$ for some numbers of iterations t.

Definition 2.2 (Laziness). We say there is **laziness** when the loss gradient satisfies $|\frac{\partial\mathcal{L}}{\partial\theta_\mu}| \ll 1$ for some numbers of iterations t and some variational angle components θ_μ.

The clarification of those two definitions will be the primary point of our paper. We point out that laziness may not necessarily indicate barren plateaus, and the QNTK theory will be a natural formulation for the barren plateau problems.

3. The loss function landscape and the QNTK theory

In order to provide a complete theoretical understanding about the barren plateau problem and laziness, we start to introduce a powerful theory, called QNTK, on the analytic study of variational quantum algorithms from the first principle. We begin by considering a variational quantum circuit ansatz, on a Hilbert space of size N with $\log_2 N$ qubits, as follows,

$$U(\theta) = \left(\prod_{\ell=1}^{L} W_\ell \exp\left(i\theta_\ell X_\ell\right)\right) \equiv \left(\prod_{\ell=1}^{L} W_\ell U_\ell\right), \tag{15}$$

with some trainable angles θ_ℓ, constant unitary operators W_ℓ, and Pauli operators X_ℓ. Following [28], we consider the mean square loss function

$$\mathcal{L}(\theta) = \frac{1}{2}\left(\left\langle \Psi_0 \right| U^\dagger(\theta) O U(\theta) \left| \Psi_0 \right\rangle - O_0\right)^2 \equiv \frac{1}{2}\varepsilon^2, \tag{16}$$

and train the expectation value $\left\langle \Psi_0 \right| U^\dagger(\theta) O U(\theta) \left| \Psi_0 \right\rangle$ on an initial state $|\Psi_0\rangle$ towards a value O_0. We define the residual training error $\varepsilon = \left\langle \Psi_0 \right| U^\dagger(\theta) O U(\theta) \left| \Psi_0 \right\rangle - O_0$. We use the gradient descent algorithm equation (1) with the learn ing rate η and an initial variational angle $\theta(0)$. We look now at the difference of the residual training error

$$\delta\varepsilon \equiv \varepsilon(t+1) - \varepsilon(t)\ . \tag{17}$$

When the learning rate of equation (1) η is small, we can perform a Taylor expansion,

$$\delta\varepsilon \approx \sum_\ell \frac{\partial\varepsilon}{\partial\theta_\ell}\delta\theta_\ell = -\eta\sum_\ell \frac{\partial\varepsilon}{\partial\theta_\ell}\frac{\partial\varepsilon}{\partial\theta_\ell}\varepsilon = -\eta K\varepsilon\ . \tag{18}$$

The quantity K here is called the QNTK [28], $K = \sum_\ell \frac{\partial\varepsilon}{\partial\theta_\ell}\frac{\partial\varepsilon}{\partial\theta_\ell}$. Note that in a general supervised learning setup where one has a labeled dataset instead of just one expected value O_0, K is a positive-semidefinite and symmetric matrix instead of a non-negative number. Here we focus on the optimization problem

equation (16): this example will demonstrate the validity of our theory, that can be readily generalized to a full supervised quantum machine learning setup.

A *frozen* QNTK will remain constant during a gradient descent flow will lead to gradient flow equations which can be solved exactly [28], showing that the error will decay exponentially at the gradient descent iteration t as

$$\varepsilon(t) = (1-\eta K)^t \varepsilon(0) \ . \tag{19}$$

For sufficient random variational ansätze, we could compute the value of K based on the same assumption of the barren plateau problem [15]. After computing 2-design random average $\mathbb{E}$ (see [29] for more details)

$$\mathbb{E}(O) = \int_{U\in 2\text{-design}} \mathrm{d}U O(U) \ . \tag{20}$$

More precisely, we define

$$\begin{aligned} &U_{-,\ell} \equiv \prod_{\ell'=1}^{\ell-1} W_{\ell'} U_{\ell'}, U_{+,\ell} \equiv \prod_{\ell'=\ell+1}^{L} W_{\ell'} U_{\ell'}, \\ &V_{-,\ell} = U_{-,\ell} W_\ell U_\ell, V_{+,\ell} = U_{+,\ell} \ . \end{aligned} \tag{21}$$

And we assume that $V_{-,\ell}$ and $V_{+,\ell}$ form 2-designs independently in all ℓs. We get the following expression of the averaged QNTK,

$$\bar{K} = \mathbb{E}(K) = L\left(N\mathrm{Tr}\left(O^2\right) - \mathrm{Tr}^2(O)\right) \frac{2}{N+1}\left(\frac{1}{N^2-1}\right) \approx \frac{2L\mathrm{Tr}\left(O^2\right)}{N^2} \ . \tag{22}$$

This simple equation combined with equation (19) reveals how, on average, the residual training error of a gradient descent dynamics will decay exponentially. Moreover, one should also check the standard deviation ΔK. If $\Delta K \ll \bar{K}$, we get a distribution of K which is concentrated at $\bar{K}$. In fact, one could show that from k-design assumptions,

$$\Delta K \approx \frac{\sqrt{L}}{N^2}\sqrt{\left(8\mathrm{Tr}^2\left(O^2\right) + 12\mathrm{Tr}\left(O^4\right)\right)} \ . \tag{23}$$

Thus, we have $\Delta K/\bar{K} = \mathcal{O}(1/\sqrt{L})$. In the limit where $L \gg 1$, the NTK is concentrated around a fixed value $\bar{K}$.

The precise proof of the derivation on $\bar{K} \sim L/N^2$ and $\Delta K \sim \sqrt{L}/N^2$ is provided in [29]. Here, we give a brief overview of the techniques used. Giving $\bar{K}$ as an example. The first step is to derive an explicit formula of K for arbitrary unitary circuits analytically, which is,

$$K = -\sum_{\ell} \left\langle \Psi_0 \middle| V_{+,\ell}^\dagger \left[X_\ell, V_{-,\ell}^\dagger O V_{-,\ell}\right] V_{+,\ell} \middle| \Psi_0 \right\rangle^2 \ , \tag{24}$$

where,

$$\begin{aligned} &U_{-,\ell} \equiv \prod_{\ell'=1}^{\ell-1} W_{\ell'} U_{\ell'}, \quad U_{+,\ell} \equiv \prod_{\ell'=\ell+1}^{L} W_{\ell'} U_{\ell'} \ , \\ &V_{-,\ell} = U_{-,\ell} W_\ell U_\ell, \quad V_{+,\ell} = U_{+,\ell}. \end{aligned} \tag{25}$$

Now, we average over all $V_{-,\ell}$ and $V_{+,\ell}$ assuming that they are independent k-designs. See section 2.3 for a more comprehensive summary. Thus, we obtain $\bar{K} = \mathbb{E}(K)$. Using a similar method, we could obtain the scaling of $\Delta K = \sqrt{\mathbb{E}(K^2) - (\mathbb{E}(K))^2}$ from higher-order k-designs.

A more precise constraint will also include a time-dependent statement including the perturbations of higher-order Taylor expansion of the residual training error, which is characterized by the so-called quantum meta-kernel or dQNTK. See appendix B for more details.

4. Precision and noise

Now we give some physical interpretations about equation (22). We see in section 3 that the theory should work in the regime where $L \gg 1$, and also the overparametrization regime where $\eta K \approx \mathcal{O}(1)$. From equation (19), we know that $\bar{K}$ would serve as an exponent of exponential decay: the larger $\bar{K}$ is, the faster the algorithm will converge. This qualitative description has been formulated in [28], with numerical evidence in [50] around the same time.

Moreover, a statement about precision could be made by combining equations (19) and (22). We have

$$\log \frac{1}{\varepsilon_r} \approx -T \log\left(1-\eta\bar{K}\right) \approx \eta \bar{K} T \,. \tag{26}$$

Here, T is the total training steps, and ε_r is the relative residual training error around the end of training $\varepsilon_r = \varepsilon(T)/\varepsilon(0)$. The relative error ε_r could be as small as the precision of the quantum device. Using equation (22), we get

$$\log \frac{1}{\varepsilon_r} \approx \frac{2\eta L \mathrm{Tr}\left(O^2\right) T}{N^2} \,. \tag{27}$$

Equation (27) makes the barren plateau problem manifestly as a precision problem. If we want to see the convergence within $T \approx \mathcal{O}(1)$, we want $\eta\bar{K} \approx 1$. The smaller $\bar{K}$ is, the smaller decaying exponent we have, and more likely we will experience a barren plateau in practice. Otherwise, there will be good enough local optima around the small random fluctuations of variational angles. The more overparametrized the quantum neural networks are, the faster convergence they could have. In this case, we do not have a barren plateau if we assume that we do not have the measurement noise and the quantum hardware noise, although we have laziness.

Originally, a relation between the barren plateau problem and the precision has also been stated in [15], while we make it more clear by showing that the barren plateau is not algorithmic. In fact, in appendix A, we show that classical overparametrized neural networks have laziness as well. Many useful, practical machine learning algorithms have to be in this case [22]. Thus, variational quantum algorithms here have no algorithmic issue, and the origin of the problem comes from measurement and control (see also [51]).

Let us take a look at equation (1) again. To implement variational algorithms, we need to perform measurements to evaluate the loss function or its derivatives (involving quantum measurements), and update the trainable angles through equation (1) (involving quantum control). On the measurement side, classical computations could handle the precision-ε computation with the resource scaling as $\log 1/\epsilon$, while measurement errors will be produced in the quantum setup, making the scaling $1/\epsilon^{\alpha}$ for positive α [15]. There is no known way to date to avoid it because of limitations of metrology [52]. On the control side, it is also challenging to update the variational angles with exponential precision. In a sense, our theory makes the statement from [15] more precise.

The discussion naturally motivates us to introduce the noise model. In figure 2, we show an example about how quantum noise models could affect the training dynamics in the overparametrized regime.

Heuristically, we will expect that during the gradient descent process, the effective noise term will also be exponentially decaying because of the original recurrence relation and its solution. To verify this, we could add a random fluctuation term $\Delta\theta_\ell$ to model the uncertainty of measuring the expectation value. One could also assume that the random variable $\Delta\theta_\ell$ is Markovian. Namely, it is independent for the time step t. Moreover, we assume that $\Delta\theta_\ell$s are distributed with Gaussian distributions $\mathcal{N}(0,\sigma_\theta^2)$. Note that σ_θ could come from the measurement noise during estimations of quantum observables used for the gradient descent, which scales as $1/\sqrt{n}$, where n is the number of measurements. And the Gaussian assumptions come from the central limit theorem in the large-n limit. Furthermore, σ_θ could also come from the hardware noises. On the other hand, the physical implementation of rotation angle will also have limited precision. One could note that robust quantum control techniques can suppress errors of rotation angles to higher orders, see [54].

Thus, one could show that the residual training error has the recursion relation in the linear order of the Taylor expansion,

$$\delta\varepsilon = -\eta\varepsilon K + \sum_\ell \frac{\partial\varepsilon}{\partial\theta_\ell}\Delta\theta_\ell \,. \tag{28}$$

Now, let us assume that K is still a constant, $K \approx \bar{K}$. Since $\Delta\theta_\ell \sim \mathcal{N}(0,\sigma_\theta^2)$, we get

$$\sum_\ell \frac{\partial\varepsilon}{\partial\theta_\ell}\Delta\theta_\ell \sim \mathcal{N}\left(0, K\sigma_\theta^2\right) \,. \tag{29}$$

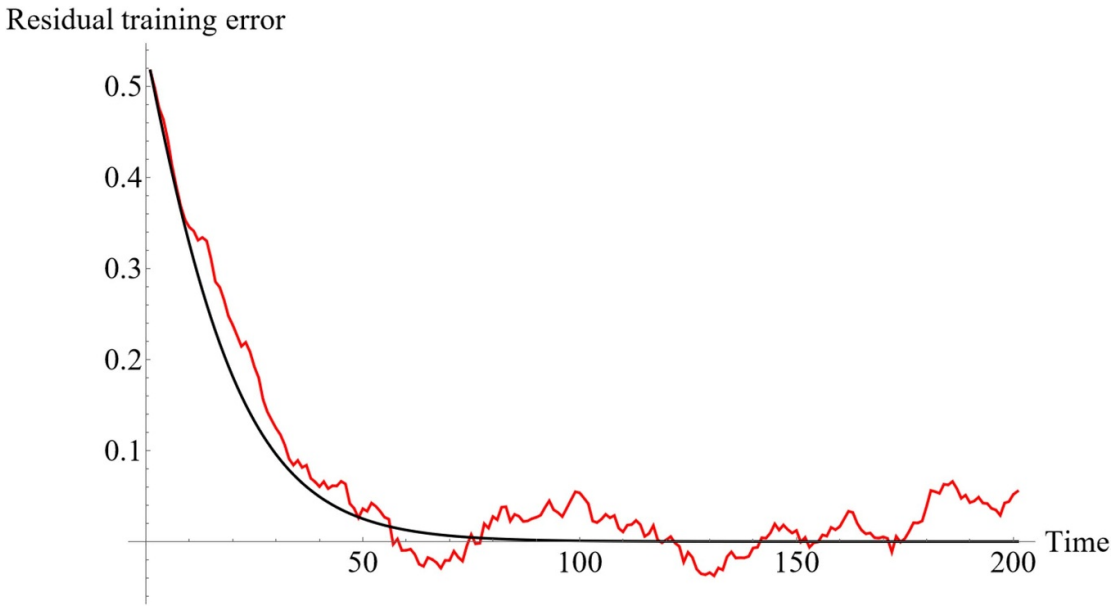

Figure 2. Noise-resilience in the overparametrized quantum machine learning models from the time dependence of the residual training error. We set the number of qubits $n = 2$ with $L = 64$ trainable parameters, where the variational circuits are chosen from a randomized hardware-efficient ansatz introduced in [28, 29]. We set the learning rate as $\eta = 10^{-3}$. Moreover, we set the standard deviation as $\sigma_\theta = 10^{-3}$ in the Gaussian random noise model (see discussions in this section and the paper [53] for more details). The noisy model has the training dynamics with the red line, while the noiseless model has the exponential converging training dynamics with the blue line. This figure shows our statement from noisy quantum neural network models: in such a simple noise model, overparametrized quantum neural networks are resilient against noise. They are still exponentially converging towards a plateau induced by noise, and the convergence will take exponentially long time.

Including the noise term into the recursion relation, one could show that averaging over the random distribution of the noise, we have

$$\overline{\varepsilon^2}(t) = (1-\eta K)^{2t}\left(\varepsilon^2(0) - \frac{\sigma_\theta^2}{\eta(2-\eta K)}\right) + \frac{\sigma_\theta^2}{\eta(2-\eta K)}\,. \tag{30}$$

Note that the first term is decaying when the time t is increasing. At the late time, we have

$$\overline{\varepsilon^2}(\infty) = \frac{\sigma_\theta^2}{\eta(2-\eta K)} \approx \mathcal{O}\left(\frac{\sigma_\theta^2}{\eta}\right)\,, \tag{31}$$

where we assume the overparametrization $\eta K \approx \mathcal{O}(1)$. Thus, at the late time, the loss function will arrive at a constant plateau at $\mathcal{O}(\sigma_\theta^2/\eta)$. One could improve σ_θ to make the constant plateau controllable and do not increase significantly with N, indicating that our algorithm could be noise-resilient. See appendix D for a more detailed discussion, and see figure 1 for an illustration. Some numerical results are also obtained in figures 3 and 4.

5. Conclusion and outlook

In this paper, we point out that for variational circuits with sufficiently large numbers of trainable angles, the gradient descent dynamics could still be efficiently performed, despite the existence of the exponential suppression of the variational angle updates (laziness). We point out that laziness is not uniquely happening in quantum machine learning, but also for overparametrized classical neural networks with large widths. The efficiency of large-width neural networks is justified by the NTK theory, so do their quantum counterparts. A solid and simple theory has been established based on the above ideas, and the relation between the number of training steps, the quantum device error, the trainable depth, the dimension of the Hilbert space, and the norm of operators appearing in the loss function has been explicitly derived. Moreover, we have justified that for simple and natural noise models, we could make the variational quantum circuits noise-resilient in the overparametrized regime, with solid theoretical and numerical evidence.

Our results also indicate a more well-defined path to designing quantum neural networks from the first principle. If we are sampling unitary operators uniformly in the whole unitary group, it is hard to avoid polynomial factors of N, the dimension of the Hilbert space, into the expression of the number of iterations

in order to obtain the visible laziness (see parallel efforts in [48, 55]). One idea is to reduce the space of searching, and reduce the space of variational circuits to some subspaces, where people observe some evidence for setups in quantum convolutional neural networks [24, 56] and local loss function [23], and the barren plateau phenomena are less drastic in those cases. However, since the subspace we are searching is reduced, the decreased expressibility will lead to a lower performance for the final convergence of the loss function on the training set [48]: around the end of the training, drastic corrections towards fixed NTKs will stop the exponential decay, and we get a local minimum which may not be good enough. The design of variational circuits will be a trade-off between barren plateaus and performance [57], which could be manifest in the presence of laziness. Despite generalizations to full learning setups with multiple output dimensions, other interesting directions include detailed discussions about the quantum noise in the real machines during quantum representation learning to understand how the noise will affect laziness and the barren plateau, a justification of our theory with large-scale classical and quantum simulation, and possible theoretical understandings beyond the limit $L \gg 1$. Finally, it will be useful to explore how our QNTK theory is able to make practical guidance to improve the performance of variational algorithms. For instance, since we know that higher QNTKs will lead to faster convergences, one can make plots at initialization about the value of the QNTK eigenvalues for different variational angles. Thus, one could choose larger QNTK initializations at the beginning of training, and in practice, it might lead to better convergence. Moreover, there are relationships between the NTK eigenvalues, generalization error, and alignments [58], where our results might be helpful in improving the generalization properties of quantum machine learning models. We look forward to further analysis and research along our path.

Data availability statement

The data cannot be made publicly available upon publication because they are not available in a format that is sufficiently accessible or reusable by other researchers. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

We thank Jens Eisert, Keisuke Fujii, Isaac Kim, Risi Kondor, Kenji Kubo, Antonio Mezzacapo, Kosuke Mitarai, Khadijeh Najafi, Sam Pallister, John Preskill, Dan A Roberts, Norihito Shira, Eva Silverstein, Francesco Tacchino, Shengtao Wang, Xiaodi Wu, Yi-Zhuang You, Han Zheng, and Quntao Zhuang for useful discussions. J L is supported in part by International Business Machines (IBM) Quantum through the Chicago Quantum Exchange, and the Pritzker School of Molecular Engineering at the University of Chicago through AFOSR MURI (FA9550-21-1-0209), L J acknowledges supports from the ARO(W911NF-23-1-0077), ARO MURI (W911NF- 21-1-0325), AFOSR MURI (FA9550-19-1-0399, FA9550-21-1-0209), NSF (OMA-1936118, ERC-1941583, OMA-2137642), NTT Research, Packard Foundation (2020-71479).

Appendix A. Comments on the barren plateau in the *classical* machine learning

Now we consider a classical neural network, the multilayer perceptron (MLP) model (see [22]). The definition is

$$\begin{aligned} & z_i^{(1)}(x_\alpha) \equiv b_i^{(1)} + \sum_{j=1}^{n_0} W_{ij}^{(1)} x_{j;\alpha}, \\ & \text{for} \quad i = 1, \ldots, n_1, \\ & z_i^{(\ell+1)}(x_\alpha) \equiv b_i^{(\ell+1)} + \sum_{j=1}^{n_\ell} W_{ij}^{(\ell+1)} \sigma\left(z_j^{(\ell)}(x_\alpha)\right), \\ & \text{for} \quad i = 1, \ldots, n_{\ell+1}; \ell = 1, \ldots, L-1. \end{aligned} \tag{32}$$

Here, σ is a non-linear activation function, and we have widths $n_{1,2,\cdots,L}$ in layers $\ell = 1, 2, \cdots L$. The input dimension is n_0 and the output dimension is n_L. Weights and biases at layer ℓ are denoted as $W^{(\ell)}$ and $b^{(\ell)}$. $z^{(\ell)}$ is called the *preactivation*. $x_{j,\alpha}$ will denote the data where j is the vector index, and α is the data sample index. At the beginning, we initialize the neural network by

$$\mathbb{E}\left[b_{i_1}^{(\ell)} b_{i_2}^{(\ell)}\right] = \delta_{i_1 i_2} C_b^{(\ell)} ,$$
$$\mathbb{E}\left[W_{i_1 j_1}^{(\ell)} W_{i_2 j_2}^{(\ell)}\right] = \delta_{i_1 i_2}\delta_{j_1 j_2}\frac{C_W^{(\ell)}}{n_{\ell-1}} . \tag{33}$$

Here, C_b and C_W will set the variance of biases and weights (we use the notation $C_W = \sigma_W^2$ in the main text). And we train the neural networks by gradient descent algorithms. We could consider the simplest version of the gradient descent algorithm,

$$\theta_\mu(t+1) = \theta_\mu(t) - \eta \frac{\mathrm{d}\mathcal{L}_\mathcal{A}}{\mathrm{d}\theta_\mu}\bigg|_{\theta(t)} . \tag{34}$$

The loss function is

$$\mathcal{L}_\mathcal{A} \equiv \frac{1}{2}\sum_{i,\tilde{\alpha}\in\mathcal{A}} \left(z_i(x_{\tilde{\alpha}};\theta) - y_{i,\tilde{\alpha}}\right)^2 = \frac{1}{2}\sum_{i,\tilde{\alpha}\in\mathcal{A}} \varepsilon_{i,\tilde{\alpha}}^2 , \tag{35}$$

where $\tilde{\alpha} \in \mathcal{A}$ form a training set $\mathcal{A}$, and we have a supervised learning task with the data label y. z_i is the final prediction from the MLP model, $z_i^{(L)}$, η is the training rate. θ_μ is a vector combining all Ws and bs. ε here is the residual training error,

$$\varepsilon_{i,\tilde{\alpha}} = z_i(x_{\tilde{\alpha}}) - y_{i,\tilde{\alpha}} . \tag{36}$$

A.1. The fundamental difference between barren plateau and vanishing gradient

Firstly, we wish to comment on the fact that there is a fundamental difference between the barren plateau problem and the vanishing gradient problem.

The vanishing gradient problem is claimed to be a challenge of machine learning algorithms, where the gradient is vanishing for some neural network constructions, and it will be challenging to train the network [59, 60]. A standard and traditional explanation of the vanishing gradient problem is due to multiplicatively large number of layers in a deep neural network. The loss will have exponential behavior against some multiplicative factors during gradient descent, which will cause either exploding or vanishing of the loss function if there is no fine tuning. A resolution of the vanishing gradient problem is associated with the idea of *He initialization* or *Kaiming initialization*, which fine-tunes the neural network towards its critical point [61] (see also [22]).

The *barren plateau problem* is a term invented from the quantum community since [15]. As far as we know, there is no such term in classical machine learning instead of geography. One of the theoretical argumentw supporting the barren plateau problem in [15] is the following, where we define the argument as *laziness*. If we consider the gradient descent process of the variational angles,

$$\theta_\mu(t+1) = \theta_\mu(t) - \eta \frac{\mathrm{d}\mathcal{L}_\mathcal{A}}{\mathrm{d}\theta_\mu}\bigg|_{\theta(t)} . \tag{37}$$

and if we make a sufficiently random variational ansatz, the factor $\mathrm{poly}(\dim\mathcal{H})$ where $\dim\mathcal{H}$ is the dimension of the Hilbert space, will appear in the formula of $\mathrm{d}\mathcal{L}_A/\mathrm{d}\theta_\mu$. Thus, the change of the variational angle will always suppressed by the dimension of the Hilbert space. A simple example of the Haar random factor $\mathrm{poly}(\dim\mathcal{H})$ will be the integration formula over a 2-design,

$$\int \mathrm{d}U U_{ij} U_{kl}^\dagger = \frac{\delta_{il}\delta_{jk}}{\dim\mathcal{H}} , \tag{38}$$

where the matrix U forms a 2-design. The higher k is in a k-design, the higher factor of $\dim \mathcal{H}$ will appear if we consider higher moments of U. Thus, one claim that the variational angles almost cannot run in the randomized variational quantum architectures.

We could notice that the argument of the barren plateau problem using laziness is fundamentally different from the vanishing gradient problem: the vanishing gradient problem is *dynamical* when going to deeper and deeper neural networks, while the laziness is *static* and appears everywhere. Thus they are two intrinsically different problems. Moreover, from the similarity between the 2-design integral formula (38) and the LeCun parametrization (33), we could expect that the large-width neural networks will have similar behaviors: their weights and biases will also almost not run. Considering that classical overparametrized neural networks are proven to be practically useful (see, for instance, a comparison [62]), and large-scale neural networks could be implemented commonly nowadays, laziness may not always be bad in the actual machine learning tasks.

A.2. Classical large-width neural network has laziness as well

Now we prove that in the above setup, the large-width classical neural network will also have laziness. We have

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}_{\mathcal{A}}}{\mathrm{d}\theta_\mu} &= \sum_{i,\tilde{\alpha}} \varepsilon_{i,\tilde{\alpha}} \frac{\mathrm{d}\varepsilon_{i,\tilde{\alpha}}}{\mathrm{d}\theta_\mu} = \sum_{i,\tilde{\alpha}} \varepsilon_{i,\tilde{\alpha}} \frac{\mathrm{d}z_{i,\tilde{\alpha}}}{\mathrm{d}\theta_\mu} \\ &= \sum_{i,\tilde{\alpha}} y_{i,\tilde{\alpha}} \frac{\mathrm{d}z_{i,\tilde{\alpha}}}{\mathrm{d}\theta_\mu} + \sum_{i,\tilde{\alpha}} z_{i,\tilde{\alpha}} \frac{\mathrm{d}z_{i,\tilde{\alpha}}}{\mathrm{d}\theta_\mu} \,. \end{aligned} \tag{39}$$

We wish to represent the derivatives over W and b by the derivatives of early-layer preactivation $z^{(\ell)}$,

$$\begin{aligned}\frac{\mathrm{d}z_{i;\alpha}^{(L)}}{db_j^{(\ell)}} &= \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{j;\alpha}^{(\ell)}} \,, \\ \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}W_{jk}^{(\ell)}} &= \sum_m \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{m;\alpha}^{(\ell)}} \frac{\mathrm{d}z_{m;\alpha}^{(\ell)}}{\mathrm{d}W_{jk}^{(\ell)}} = \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{j;\alpha}^{(\ell)}} \sigma_{k;\alpha}^{(\ell-1)} \,. \end{aligned} \tag{40}$$

Here, $\sigma^{(\ell)}$ is a short-hand notation of $\sigma(z^{(\ell)})$, and we introduce $\sigma_{j;\alpha}^{(\ell)}$ as $\sigma(z_{j;\alpha}^{(\ell)})$. Finally, we have,

$$\begin{aligned}&\frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{j;\alpha}^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{k;\alpha}^{(\ell+1)}} \frac{\mathrm{d}z_{k;\alpha}^{(\ell+1)}}{\mathrm{d}z_{j;\alpha}^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{k;\alpha}^{(\ell+1)}} W_{kj}^{(\ell+1)} \sigma_{j;\alpha}^{(\ell)\prime} \\ &\text{for} \quad \ell < L \,, \\ &\frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{j;\alpha}^{(L)}} = \delta_{ij} \,. \end{aligned} \tag{41}$$

This is a back-propagation iterative formula, giving the recurrence relation from the end of the neural networks to the beginning. Moreover, we use σ' to denote derivatives of σ. So we get

$$\begin{aligned}\frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{j;\alpha}^{(\ell)}} &= \sum_{k=1}^{n_{\ell+1}} \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{k;\alpha}^{(\ell+1)}} \frac{\mathrm{d}z_{k;\alpha}^{(\ell+1)}}{\mathrm{d}z_{j;\alpha}^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{k;\alpha}^{(\ell+1)}} W_{kj}^{(\ell+1)} \sigma_{j;\alpha}^{(\ell)\prime} \\ &= \sum_{k_{\ell+1},k_{\ell+2}}^{n_{\ell+1},n_{\ell+2}} \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{k_{\ell+2};\alpha}^{(\ell+2)}} W_{k_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} \sigma_{j;\alpha}^{(\ell+1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \\ &= \sum_{k_{\ell+1},k_{\ell+2},\ldots,k_L}^{n_{\ell+1},n_{\ell+2},\ldots,n_L} \frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{k_L;\alpha}^{(L)}} W_{k_Lj}^{(L)} W_{k_{L-1}j}^{(L-1)} \ldots W_{k_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} \\ &\quad \times \sigma_{j;\alpha}^{(L-1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \ldots \sigma_{j;\alpha}^{(\ell+1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \\ &= \sum_{k_{\ell+1},k_{\ell+2},\ldots,k_{L-1}}^{n_{\ell+1},n_{\ell+2},\ldots,n_{L-1}} W_{i,j}^{(L)} W_{k_{L-1}j}^{(L-1)} \ldots W_{k_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} \\ &\quad \sigma_{j;\alpha}^{(L-1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \ldots \sigma_{j;\alpha}^{(\ell+1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} . \end{aligned} \tag{42}$$

We find the expectation value will vanish directly (which is exactly similar to the quantum case). Thus, we could estimate the norm by computing the variance of the gradients from,

$$\begin{aligned}&\mathbb{E}\left(\left(\frac{\mathrm{d}z_{i;\alpha}^{(L)}}{\mathrm{d}z_{j;\alpha}^{(\ell)}}\right)^2\right) \\ &= \sum_{k_{\ell+1},k_{\ell+2},\ldots,k_{L-1},\bar{k}_{\ell+1},\bar{k}_{\ell+2},\ldots,\bar{k}_{L-1}}^{n_{\ell+1},n_{\ell+2},\ldots,n_{L-1},n_{\ell+1},n_{\ell+2},\ldots,n_{L-1}} \mathbb{E}\left(\begin{array}{c} W_{i,j}^{(L)} W_{i,j}^{(L)} W_{k_{L-1}j}^{(L-1)} W_{\bar{k}_{L-1}j}^{(L-1)} \cdots \\ W_{k_{\ell+2}j}^{(\ell+2)} W_{\bar{k}_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} W_{\bar{k}_{\ell+1}j}^{(\ell+1)} \end{array} \right) \mathbb{E}\left(\left(\Sigma_{j;\alpha}^{(\ell);(L-1)}\right)^2\right) \\ &= \sum_{k_{\ell+1},k_{\ell+2},\ldots,k_{L-1},\bar{k}_{\ell+1},\bar{k}_{\ell+2},\ldots,\bar{k}_{L-1}}^{n_{\ell+1},n_{\ell+2},\ldots,n_{L-1},n_{\ell+1},n_{\ell+2},\ldots,n_{L-1}} \mathbb{E}\left(\begin{array}{c} W_{i,j}^{(L)} W_{i,j}^{(L)} W_{k_{L-1}j}^{(L-1)} W_{\bar{k}_{L-1}j}^{(L-1)} \cdots \\ W_{k_{\ell+2}j}^{(\ell+2)} W_{\bar{k}_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} W_{\bar{k}_{\ell+1}j}^{(\ell+1)} \end{array} \right) \mathbb{E}\left(\left(\Sigma_{j;\alpha}^{(\ell);(L-1)}\right)^2\right) \\ &= \frac{1}{n_L} C_W^{(L)} C_W^{(L-1)} \ldots C_W^{(\ell+1)} \mathbb{E}\left(\left(\Sigma_{j;\alpha}^{(\ell);(L-1)}\right)^2\right) , \end{aligned} \tag{43}$$

where

$$\Sigma_{j;\alpha}^{(\ell);(L-1)} = \sigma_{j;\alpha}^{(L-1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \ldots \sigma_{j;\alpha}^{(\ell+2)\prime} \sigma_{j;\alpha}^{(\ell+1)\prime} . \tag{44}$$

We have used the Wick contraction rule and the LeCun parametrization (33) according to [22]. Plug equation (43) back to equation (40), we see that this $1/n_L$ factor appears. This is the classical barren plateau in the large-width classical neural networks.

A.3. Classical large-width neural network could still learn efficiently

Here we show that the classical NTK will not vanish in classical MLPs, despite its laziness. This indicates that there are many good enough local minima around the point of initialization, so even the variational angles run slowly (the barren plateau problem), it will not matter for our practical purpose. On the other hand, more variational parameters will make us converge faster.

This part is a review of existing results, presented in the language of [22]. In classical MLPs, similar to the quantum cases we have discussed in the whole paper, the residual training error ε will decay exponentially at large width. We define the NTK as

$$H_{i_1 i_2;\alpha_1\alpha_2} \equiv \sum_{\mu} \frac{\mathrm{d}z_{i_1;\alpha_1}}{\mathrm{d}\theta_\mu} \frac{\mathrm{d}z_{i_2;\alpha_2}}{\mathrm{d}\theta_\mu} . \tag{45}$$

The gradient descent rule will imply,

$$\delta\varepsilon_{i;\delta} = -\eta \sum_{i_1,\tilde{\alpha}\in\mathcal{A}} H_{ii_1;\delta\tilde{\alpha}} \varepsilon_{i_1,\tilde{\alpha}} . \tag{46}$$

One could compute the average of the NTK. One could define the frozen NTK and the fluctuating NTK as

$$H_{i_1 i_2;\alpha_1\alpha_2} = \bar{H}_{i_1 i_2;\alpha_1\alpha_2} + \Delta H_{i_1 i_2;\alpha_1\alpha_2} , \tag{47}$$

and we have

$$\mathbb{E}\left(\Delta H_{i_1 i_2;\alpha_1\alpha_2}\Delta H_{i_3 i_4;\alpha_3\alpha_4}\right)=\frac{1}{n_{L-1}}\left[\delta_{i_1 i_2}\delta_{i_3 i_4}A_{(\alpha_1\alpha_2)(\alpha_3\alpha_4)}+\delta_{i_1 i_3}\delta_{i_2 i_4}B_{\alpha_1\alpha_3\alpha_2\alpha_4}+\delta_{i_1 i_4}\delta_{i_2 i_3}B_{\alpha_1\alpha_4\alpha_2\alpha_3}\right].$$

The full expressions of A, B are given in chapter 8 of [22]. Similarly, in the statistics language, one could check [31]. The suppression of ΔH in the large width indicates that the large-width neural networks will learn efficiently through non-trivial $\bar{H}_{i_1 i_2;\alpha_1\alpha_2}$, which is guaranteed to converge exponentially. In the large-width limit, the gradient descent algorithm is theoretically equivalent to the kernel method, where the kernel is defined effectively by NTKs. In chapter 11 of [22], it is shown that dNTK, the higher-order corrections to the exponential decay, will vanish on its own, averaging over the Gaussian distribution of weights and bias. Moreover, the correlations between dNTK and other operators, which cause even numbers of Ws in total, will be suppressed by the large width polynomially. Those theoretical results are classical analogs of random unitary calculations done in our work.

Appendix B. Some further details about concentration conditions

For concentration conditions including the quantum meta-kernel, one could see [29] for further details. Here we provide a simple review.

Now, we would like to ask when the QNTK approximation is valid. When the learning rate is small, the error of the prediction in equation (22) could possibly come from two sources: the fluctuation of K about $\bar{K}$ during the gradient descent, and the higher-order corrections comparing the leading order Taylor expansion in equation (18). The fluctuation ΔK could come from higher-order statistical calculations over the k-design assumption, similar to the analysis of higher-order effects in the barren plateau setup [25],

$$\Delta K=\sqrt{\mathbb{E}\left((K-\bar{K})^2\right)}\approx\frac{\sqrt{L}}{N^2}\sqrt{\left(8\mathrm{Tr}^2\left(O^2\right)+12\mathrm{Tr}\left(O^4\right)\right)}\,, \tag{49}$$

in the large-N limit, and we present a detailed calculation in [29] with formulas up to 4-design. Moreover, we could look at higher order corrections to the Taylor expansion by the quantum meta-kernel (dQNTK) [28],

$$\begin{aligned}\delta\varepsilon&=-\eta\sum_{\ell}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\theta_\ell}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\theta_\ell}\varepsilon+\frac{1}{2}\eta^2\varepsilon^2\sum_{\ell_1,\ell_2}\frac{\mathrm{d}^2\varepsilon}{\mathrm{d}\theta_{\ell_1}\mathrm{d}\theta_{\ell_2}}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\theta_{\ell_1}}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\theta_{\ell_2}}\\&\equiv-\eta K\varepsilon+\frac{1}{2}\eta^2\varepsilon^2\mu\,.\end{aligned} \tag{50}$$

Here $\mu=\sum_{\ell_1,\ell_2}\frac{\mathrm{d}^2\varepsilon}{\mathrm{d}\theta_{\ell_1}\mathrm{d}\theta_{\ell_2}}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\theta_{\ell_1}}\frac{\mathrm{d}\varepsilon}{\mathrm{d}\theta_{\ell_2}}$ could be computed statistically using k-design formulas again. One can show that $\mathbb{E}(\mu)=0$ (which is the same as its classical counterpart [22]), and we have

$$\Delta\mu=\sqrt{\mathbb{E}\left(\mu^2\right)}\approx\frac{\sqrt{32}L}{N^3}\mathrm{Tr}^{3/2}\left(O^2\right)\,, \tag{51}$$

in the large-N limit. The condition where the QNTK estimation in equation (22) is valid when

$$\Delta K\ll K\Leftrightarrow L\gg 1\,, \tag{52}$$

$$\begin{aligned}&\frac{1}{2}\eta^2\varepsilon^2\Delta\mu\ll\eta\bar{K}\varepsilon\Leftrightarrow\eta\varepsilon\left(0\right)\frac{L}{N^3}\mathrm{Tr}^{3/2}\left(O^2\right)\ll\frac{L\mathrm{Tr}\left(O^2\right)}{N^2}\\&\Leftrightarrow\frac{\eta\Omega_O}{N}\varepsilon\left(0\right)\ll 1\,.\end{aligned} \tag{53}$$

We call the conditions (52) and (53) as the *concentration conditions*. Here, we denote $\varepsilon(0)=\varepsilon(t=0)$, and we assume that $\mathrm{Tr}(O^2)\equiv\Omega_O^2>\mathrm{Tr}^2(O)$. This is correct, for instance, if O is a Pauli operator, where we have $\mathrm{Tr}(O^2)=N$ but $\mathrm{Tr}^2(O)=0$.

Note that the condition equation (53) is a weak condition. It only tells that how small η is needed to make sure the nearly expansion is valid. In practice, we often assume that $\eta<\mathcal{O}(1)$ and $\Omega_O\geqslant\mathcal{O}(N)$, so equation (53) is automatically satisfied. The condition that usually matters is equation (52), which is the definition of overparametrization here $L\gg 1$. Thus, if L is large, the prediction will be correct, no matter how large N is. But if N is large, the decay rate itself $\bar{K}$ will be small. So this is exactly the definition of the barren plateau!

Furthermore, we wish to mention that if we only count for powers of N and L, we have

$$\frac{\Delta K}{\bar{K}} = \mathcal{O}\left(\frac{1}{\sqrt{L}}\right), \frac{\Delta\mu}{\bar{K}} = \mathcal{O}\left(\frac{1}{N}\right). \tag{54}$$

If we demand $\bar{K} = \mathcal{O}(1)$ and ignore η, we get $L = \mathcal{O}(N)$, so we get $\frac{\Delta K}{\bar{K}} = \mathcal{O}\left(\frac{1}{N}\right)$ as well. The $1/N$ or $1/\text{width}$ expansion is exactly observed in the classical neural networks [22]. The origin of this equivalence comes from the similarity between equations (2) and (55), while a higher level (but heuristic) understanding comes from a connection between quantum field theory and the large-width expansion [22, 37, 38] and a similarity between Feynman rules in quantum field theory and matrix models [63], which we will briefly explain in appendix C for readers who are interested in how observations about this paper might be discovered from another perspective.

Appendix C. A physical interpretation

Here we make some comments about possible, heuristic, physical interpretations of the agreement between classical and quantum neural networks. There is a duality, pointed out in [22, 37–39] where the large-width classical neural networks could be understood in the quantum field theory language. In the large-width limit, the output of neural networks will follow a Gaussian process, averaging with respect to Gaussian distribution over weights and bias according to the LeCun parametrization,

$$\mathbb{E}\left(W_{ij}W_{kl}\right) = \frac{\sigma_W^2}{\text{width}}\delta_{ik}\delta_{jl}, \tag{55}$$

or more generally,

$$\mathbb{E}\left(W_{i_1j_1}W_{i_2j_2}\dots W_{i_{2k-1}j_{2k-1}}W_{i_{2k}j_{2k}}\right) = \mathcal{O}\left(\frac{1}{\text{poly}\left(\text{width}\right)}\right), \tag{56}$$

for all positive integer k. Here, we are considering the MLP model with weights W, and the width is defined as the number of neurons in each layer. The limit is mathematically similar to the large-N limit of gauge theories, which becomes almost generalized free theories. We could understand the ratio between the depth, the number of layers, and the width, the number of neurons, as perturbative corrections against the Gaussian process, which is similar to what we have done in the large-N expansion of gauge theories.

This physical interpretation will be helpful also when we consider its quantum generalization. If classical MLPs are similar to quantum field theories, quantum neural networks will be similar to matrix models [64, 65]. Matrix models have been studied for a long time, around and after the second string theory revolution [63], and they have deep connections to the holographic principle [66] and the AdS/CFT correspondence [67, 68]. Haar ensembles are toy versions of matrix models, which have been widely studied as toy models of chaotic quantum black holes [16, 69]. The similarity between the LeCun parametrization (55) and the 1-design Haar integral formula

$$\mathbb{E}\left(U_{ij}U_{kl}^\dagger\right) = \frac{1}{\dim\mathcal{H}}\delta_{il}\delta_{jk}, \tag{57}$$

or more generally,

$$\mathbb{E}\left(U_{i_1j_1}U_{i_2j_2}^\dagger\dots U_{i_{2k-1}j_{2k-1}}U_{i_{2k}j_{2k}}^\dagger\right) = \mathcal{O}\left(\frac{1}{\text{poly}\left(\dim\mathcal{H}\right)}\right), \tag{58}$$

where $\dim\mathcal{H}$ is the dimension of the Hilbert space, might be potentially related to the similarity of Feynman rules between matrix models and quantum field theories. Thus, the similarity between quantum and classical neural networks might have a physical interpretation between matrix models and their effective field theory descriptions.

The above analogy is heuristic. We should point out that machine learning and physical systems are very different. Some mathematical similarities could provide guidance towards new discoveries and better insights, but we have to be careful that they are intrinsically different phenomena.

Appendix D. Noises

Now let us add the affection of the noise. From the original gradient descent equation,

$$\theta_\ell(t+1)-\theta_\ell(t)\equiv\delta\theta_\mu=-\eta\frac{\partial\mathcal{L}}{\partial\theta_\ell}=i\eta\left\langle\Psi_0\left|V_{+,\ell}^\dagger\left[X_\ell,V_{-,\ell}^\dagger OV_{-,\ell}\right]V_{+,\ell}\right|\Psi_0\right\rangle\,, \tag{59}$$

we add a random fluctuation term $\Delta\theta_\ell$ to model the uncertainty of measuring the expectation value. We assume that the random variable $\Delta\theta_\ell$ is Markovian. Namely, it is independent for the time step t. Moreover, we assume that $\Delta\theta_\ell$s are distributed with Gaussian distributions $\mathcal{N}(0,\sigma_\theta^2)$.

Thus, the residual training error has the recursion relation in the linear order of the Taylor expansion,

$$\delta\varepsilon=-\eta\varepsilon K+\sum_\ell\frac{\partial\varepsilon}{\partial\theta_\ell}\Delta\theta_\ell\,. \tag{60}$$

Now, let us assume that K is still a constant. Since $\Delta\theta_\ell\sim\mathcal{N}(0,\sigma_\theta^2)$, we get

$$\sum_\ell\frac{\partial\varepsilon}{\partial\theta_\ell}\Delta\theta_\ell\sim\mathcal{N}\left(0,K\sigma_\theta^2\right)\,. \tag{61}$$

Thus, we could write the recursion relation as

$$\delta\varepsilon=-\eta\varepsilon K+\sqrt{K}\Delta\theta\,. \tag{62}$$

Here, $\Delta\theta\approx\mathcal{N}(0,\sigma_\theta^2)$. One can solve the difference equation iteratively. The answer is

$$\varepsilon(t)=(1-\eta K)^t\varepsilon(0)+\sqrt{K}\sum_{i=0}^{t-1}(1-\eta K)^i\Delta\theta(t-1-i)\,. \tag{63}$$

Now, we have

$$\sqrt{K}\sum_{i=0}^{t-1}(1-\eta K)^i\Delta\theta(t-1-i)\sim\mathcal{N}\left(0,K\sigma_\theta^2\sum_{i=0}^{t-1}(1-\eta K)^{2i}\right)=\mathcal{N}\left(0,\sigma_\theta^2\frac{1-(1-\eta K)^{2t}}{\eta(2-\eta K)}\right)\,. \tag{64}$$

At the initial time $t=0$, there is no effect of noise. The relative size of the error will grow during time compared to the exponential decay term without noises. Based on the distribution, we could compute the average ε^2 against the noises, $\overline{\varepsilon^2}$, as

$$\overline{\varepsilon^2}(t)=(1-\eta K)^{2t}\left(\varepsilon^2(0)-\frac{\sigma_\theta^2}{\eta(2-\eta K)}\right)+\frac{\sigma_\theta^2}{\eta(2-\eta K)}\,. \tag{65}$$

Note that the first term is decaying when the time t is increasing. At the late time, we have

$$\overline{\varepsilon^2}(\infty)=\frac{\sigma_\theta^2}{\eta(2-\eta K)}\approx\mathcal{O}\left(\frac{\sigma_\theta^2}{\eta}\right)\,, \tag{66}$$

where we assume the overparametrization $\eta K\approx\mathcal{O}(1)$. Thus, at the late time, the loss function will arrive at a constant plateau at $\mathcal{O}(\sigma_\theta^2/\eta)$. One could improve σ_θ to make the constant plateau controllable and do not increase significantly with N, indicating that our algorithm could be noise-resilient.

One could also estimate the time scale where the contribution of the noise could emerge. We could define the time scale, T_{noise}, as,

$$(1-\eta K)^{T_{\text{noise}}}\varepsilon(0)\approx\sigma_\theta\sqrt{\frac{1-(1-\eta K)^{2T_{\text{noise}}}}{\eta(2-\eta K)}}\,. \tag{67}$$

It means that at T_{noise}, the noise contribution is comparable to the noiseless part in the residual training error. We have,

$$T_{\text{noise}}\approx\frac{\log\left(\frac{\sigma_\theta}{\sqrt{2\varepsilon^2(0)\eta-\varepsilon^2(0)\eta^2K+\sigma_\theta^2}}\right)}{\log(1-\eta K)}\,,\varepsilon(T_{\text{noise}})=2(1-\eta K)^{T_{\text{noise}}}\varepsilon(0)=\frac{2\sigma_\theta^2}{\sqrt{\varepsilon(0)^2(2\eta-\eta^2K)+\sigma_\theta^2}}\varepsilon(0)\,. \tag{68}$$

We find that choosing $\eta \approx \mathcal{O}(1/K)$ will minimize $\varepsilon(T_{\text{noise}})$. It is exactly the overparametrization condition we use in this paper.

To be self-consistent, we need to check if the choice $\eta \approx \mathcal{O}(1/K)$ is consistent with the concentration condition about dQNTK. In fact, we find that $\eta \approx \mathcal{O}(1/K)$ will naturally satisfy the dQNTK concentration condition if $\varepsilon(0) < \mathcal{O}(L\sqrt{N})$. This is naturally satisfied in generic situations in variational quantum algorithms since we will usually not have an exponential amount of residual training error initially.

Appendix E. Numerical results

In this part, we show some simple numerical evidences based on the analysis done in [29]. We will use the randomized version of the hardware-efficient variational ansatz defined in [29]. In figure 3, for each σ_θ value, we run 10 experiments of 100 steps using the same setup of the ansatz $U(\theta)$, the operator O and the input state θ_0 as in [29]. After that, we get the residual error of the last step and take the average value over 10 experiments to get the mean ε value, shown with black dots in the figure. The red line in the figure is the theoretical prediction. In these experiments, $L = 64$, and we have 4 qubits. We can further get the analytic result of the mean value of $\bar{\varepsilon}$ after a long time as

$$\bar{\varepsilon} = \sqrt{\frac{2}{\pi}} \cdot \frac{\sigma_\theta}{\sqrt{2\eta - \eta^2 K}}, \tag{69}$$

where the K value is taken from the value of the last step, as it fluctuates a lot in the early time.

We run multiple experiments to approach the theoretical value as much as possible, where 10 experiments are done for each σ_θ value. To verify that the numerical result lies in a reasonable regime, we calculated the 90% confidence interval of ε theoretically.

To compensate for the effect of large K on our numerical simulations, since in every experiment setup, due to randomness, the training will lead the parameters to different regimes of different Ks, we choose those experiments which fulfill our theoretical restrictions for small K. The numerical results above are with $K \approx \mathcal{O}(10)$, which still shows great agreement with our theoretical formalism.

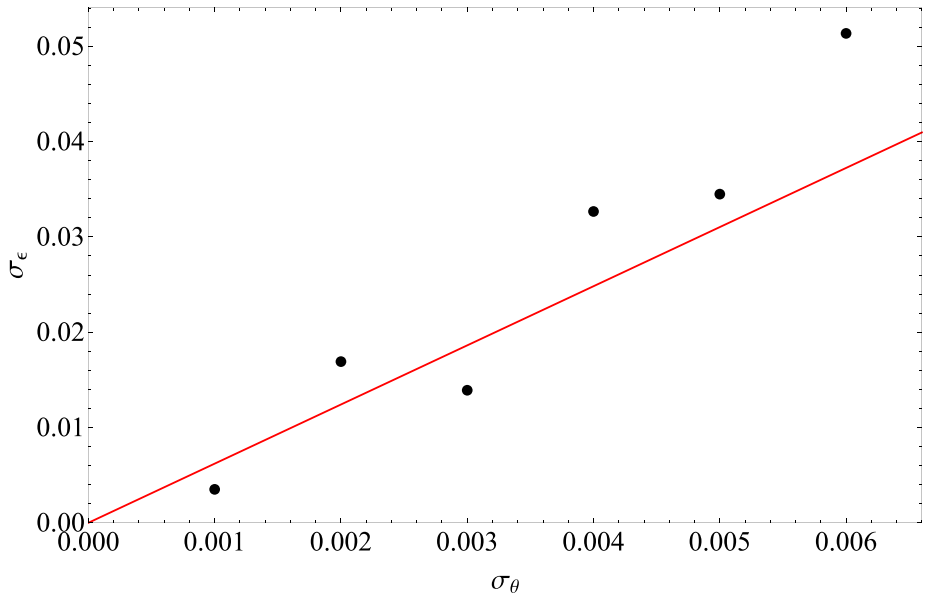

Figure 3. Noise standard deviation σ_θ as a function of standard deviation of final residual error σ_ε after training long enough time, with both numerical result (black dots) and theoretical prediction (red line). In this figure, $\eta = 0.005$, $K \approx 25$, $\varepsilon(0) \approx 1$.

More precisely, in figure 3, we get the relationship between residual error fluctuation and noise. For each σ_θ value, we calculated the standard deviation with final residual error data from 10 experiments, shown as black dots. The final residual error that we get from the numerical experiments is taken absolute value for the benefit of the log scale. We find the numerical results follow the theoretical prediction in a reasonable confidence interval. Moreover, we verify the extent of our final residual error that can achieve as a function of noise σ_θ with numerical evidence.

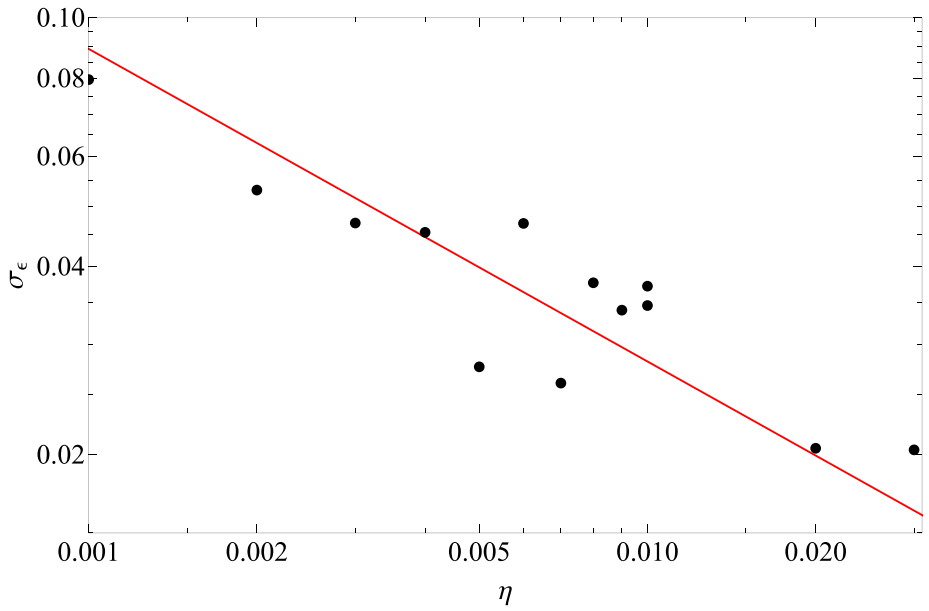

Figure 4. Standard deviation of final residual error σ_ε as a function of learning rate η after training long enough time, with both numerical result (black dots) and theoretical prediction (red line). In this figure, $\sigma_\theta = 0.005$, $K \approx 35$, $\varepsilon(0) \approx 1$, $t = 100$.

In figure 4, we verify the prediction of standard deviation of $\varepsilon(\infty)$, σ_ε, in the small η regime. In these numerical experiments, the inaccuracy comes mainly from a limited number of experiments and a limited time scale ($t = 100$). Especially for experiments with a small learning rate η with random initial states, T_{noise} may be large for 100 steps to cover.

ORCID iD

Junyu Liu https://orcid.org/0000-0003-1669-8039

References

[1] Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love P J, Aspuru-Guzik A and O'brien J L 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 1–7
[2] Yung M-H, Casanova J, Mezzacapo A, Mcclean J, Lamata L, Aspuru-Guzik A and Solano E 2014 From transistor to trapped-ion computers for quantum chemistry *Sci. Rep.* **4** 1–7
[3] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 The theory of variational hybrid quantum-classical algorithms *New J. Phys.* **18** 023023
[4] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets *Nature* **549** 242–6
[5] Cerezo M, Arrasmith A, Babbush R, Benjamin S C, Endo S, Fujii K, McClean J R, Mitarai K, Yuan X and Cincio L *et al* 2021 Variational quantum algorithms *Nat. Rev. Phys.* **3** 1–20
[6] Farhi E, Goldstone J and Gutmann S 2014 A quantum approximate optimization algorithm (arXiv:1411.4028)
[7] Wittek P 2014 *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic)
[8] Wiebe N, Kapoor A and Svore K M 2014 Quantum deep learning (arXiv:1412.3489)
[9] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202
[10] Schuld M and Killoran N 2019 Quantum machine learning in feature hilbert spaces *Phys. Rev. Lett.* **122** 040504
[11] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12
[12] Liu Y, Arunachalam S and Temme K 2021 A rigorous and robust quantum speed-up in supervised machine learning *Nat. Phys.* **17** 1–5
[13] Liu J 2021 Does Richard Feynman dream of electric sheep? Topics on quantum field theory, quantum computing, and computer science *PhD Thesis* Caltech
[14] Farhi E and Neven H 2018 Classification with quantum neural networks on near term processors. (arXiv:1802.06002)
[15] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 Barren plateaus in quantum neural network training landscapes *Nat. Commun.* **9** 1–6
[16] Roberts D A and Yoshida B 2017 Chaos and complexity by design *J. High Energy Phys.* JHEP04(2017)121
[17] Cotler J, Hunter-Jones N, Liu J and Yoshida B 2017 Chaos, complexity and random matrices *J. High Energy Phys.* JHEP11(2017)048
[18] Liu J 2018 Spectral form factors and late time quantum chaos *Phys. Rev.* D **98** 086026
[19] Liu J 2020 Scrambling and decoding the charged quantum information *Phys. Rev. Res.* **2** 043164
[20] Fukuda M, König R and Nechita I 2019 RTNI: a symbolic integrator for haar-random tensor networks *J. Phys. A: Math. Theor.* **52** 425303

[21] Mohri M, Rostamizadeh A and Talwalkar A 2018 *Foundations of Machine Learning* (MIT Press)
[22] Roberts D A, Yaida S and Hanin B 2021 The principles of deep learning theory (arXiv:2106.10165)
[23] Cerezo M, Sone A, Volkoff T, Cincio L and Coles P J 2021 Cost function dependent barren plateaus in shallow parametrized quantum circuits *Nat. Commun.* **12** 1–12
[24] Arthur Pesah M C, Wang S, Volkoff T, Sornborger A T and Coles P J 2021 Absence of barren plateaus in quantum convolutional neural networks *Phys. Rev.* X **11** 041011
[25] Cerezo M and Coles P J 2021 Higher order derivatives of quantum neural networks with barren plateaus *Quantum Sci. Technol.* **6** 035006
[26] Andrew Arrasmith M C, Czarnik P, Cincio L and Coles P J 2021 Effect of barren plateaus on gradient-free optimization *Quantum* **5** 558
[27] Brown T B *et al* 2020 Language models are few-shot learners (arXiv:2005.14165)
[28] Liu J, Tacchino F, Glick J R, Jiang L and Mezzacapo A 2022 Representation Learning via Quantum Neural Tangent Kernels *PRX Quantum* **3** 030323
[29] Liu J, Najafi K, Sharma K, Tacchino F, Jiang L and Mezzacapo A 2023 Analytic theory for the dynamics of wide quantum neural networks *Phys. Rev. Lett.* **130** 150601
[30] Lee J, Bahri Y, Novak R, Schoenholz S S, Pennington J and Sohl-Dickstein J 2017 Deep neural networks as gaussian processes (arXiv:1711.00165)
[31] Jacot A, Gabriel F and Hongler C 2018 Neural tangent kernel: convergence and generalization in neural networks (arXiv:1806.07572)
[32] Lee J, Xiao L, Schoenholz S, Bahri Y, Novak R, Sohl-Dickstein J and Pennington J 2019 Wide neural networks of any depth evolve as linear models under gradient descent *Advances in Neural Information Processing Systems* vol 32 pp 8572–83
[33] Sohl-Dickstein J, Novak R, Schoenholz S S and Lee J 2020 On the infinite width limit of neural networks with a standard parameterization (arXiv:2001.07301)
[34] Yang G and Edward J H 2020 Feature learning in infinite-width neural networks (arXiv:2011.14522)
[35] Yaida S 2020 Non-gaussian processes and neural networks at finite widths *Mathematical and Scientific Machine Learning* (PMLR) pp 165–92
[36] Arora S, Simon S D, Wei H, Zhiyuan L, Salakhutdinov R and Wang R 2019 On exact computation with an infinitely wide neural net (arXiv:1904.11955)
[37] Dyer E and Gur-Ari G 2019 Asymptotics of wide networks from feynman diagrams (arXiv:1909.11304)
[38] Halverson J, Maiti A and Stoner K 2021 Neural networks and quantum field theory *Mach. Learn.: Sci. Technol.* **2** 035002
[39] Roberts D A 2021 Why is AI hard and physics simple? (arXiv:2104.00008)
[40] Roberts D A and Yaida S 2021 Effective theory of deep learning: beyond the infinite-width limit *Deep Learning Theory Summer School* (Princeton)
[41] Rudolph M S, Sim S, Raza A, Stechly M, McClean J R, Anschuetz E R, Serrano L and Perdomo-Ortiz A 2021 Orqviz: visualizing high-dimensional landscapes in variational quantum algorithms (arXiv:2111.04695)
[42] Lei W *et al* 2017 Towards understanding generalization of deep learning: perspective of loss landscapes (arXiv:1706.10239)
[43] Kawaguchi K, Huang J and Pack Kaelbling L 2019 Every local minimum value is the global minimum value of induced model in nonconvex machine learning *Neural Comput.* **31** 2293–323
[44] Nielsen M A and Chuang I 2002 *Quantum Computation and Quantum Information* (Cambridge University Press) p 700 (available at: https://books.google.com/books/about/Quantum_Computation_and_Quantum_Informat.html?id=65FqEKQOfP8C)
[45] Shor P W 1994 Algorithms for quantum computation: discrete logarithms and factoring *Proc. 35th Annual Symposium on Foundations of Computer Science* (IEEE) pp 124–34
[46] Brandao F G S L, Harrow A W and Horodecki M 2016 Local random quantum circuits are approximate polynomial-designs *Commun. Math. Phys.* **346** 397–434
[47] Anschuetz E R and Kiani B T 2022 Beyond barren plateaus: quantum variational algorithms are swamped with traps (available at: www.nature.com/articles/s41467-022-35364-5)
[48] Abedi E, Beigi S and Taghavi L 2022 Quantum lazy training (arXiv:2202.08232)
[49] Chizat L, Oyallon E and Bach F 2019 On lazy training in differentiable programming *Advances in Neural Information Processing Systems* p 32
[50] Shirai N, Kubo K, Mitarai K and Fujii K 2021 Quantum tangent kernel (arXiv:2111.02951)
[51] Wang S, Fontana E, Cerezo M, Sharma K, Sone A, Cincio L and Coles P J 2021 Noise-induced barren plateaus in variational quantum algorithms *Nat. Commun.* **12** 1–11
[52] Knill E, Ortiz G and Somma R D 2007 Optimal quantum measurements of expectation values of observables *Phys. Rev.* A **75** 012328
[53] Liu J, Wilde F, Anna Mele A, Jiang L and Eisert J 2022 Noise can be helpful for variational quantum algorithms (arXiv:2210.06723)
[54] Vandersypen L M K and Chuang I L 2005 Nmr techniques for quantum control and computation *Rev. Mod. Phys.* **76** 1037
[55] You X, Chakrabarti S and Xiaodi W 2022 A convergence theory for over-parameterized variational quantum eigensolvers (arXiv:2205.12481)
[56] Cong I, Choi S and Lukin M D 2019 Quantum convolutional neural networks *Nat. Phys.* **15** 1273–8
[57] Larocca M, Nathan J, García-Martín D, Coles P J and Cerezo M 2021 Theory of overparametrization in quantum neural networks (arXiv:2109.11676)
[58] Canatar A, Bordelon B and Pehlevan C 2021 Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks *Nat. Commun.* **12** 1–12
[59] Hochreiter S 1998 The vanishing gradient problem during learning recurrent neural nets and problem solutions *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **6** 107–16
[60] Hochreiter S *et al* 2001 Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (available at: https://ieeexplore.ieee.org/document/5264952)
[61] Kaiming H, Zhang X, Ren S and Sun J 2015 Delving deep into rectifiers: surpassing human-level performance on imagenet classification *Proc. IEEE Int. Conf. on Computer Vision* pp 1026–34
[62] Golubeva A, Neyshabur B and Gur-Ari G 2020 Are wider nets better given the same number of parameters? (arXiv:2010.14495)
[63] Witten E 1995 String theory dynamics in various dimensions *Nucl. Phys.* B **443** 85–126
[64] Tom Banks W F, Shenker S H and Susskind L 1997 M theory as a matrix model: a Conjecture *Phys. Rev.* D **55** 5112–28

[65] Eliecer Berenstein D, Martin Maldacena J and Stefan Nastase H 2002 Strings in flat space and pp waves from $N = 4$ superYang-Mills *J. High Energy Phys.* JHEP04(2002)013
[66] Susskind L 1995 The World as a hologram *J. Math. Phys.* **36** 6377–96
[67] Martin Maldacena J 1998 The Large N limit of superconformal field theories and supergravity *Adv. Theor. Math. Phys.* **2** 231–52
[68] Witten E 1998 Anti-de Sitter space and holography *Adv. Theor. Math. Phys.* **2** 253–91
[69] Hayden P and Preskill J 2007 Black holes as mirrors: quantum information in random subsystems *J. High Energy Phys.* JHEP09(2007)120