Appendix

Junyu Liu
Pritzker School of Molecular Engineering, The University of Chicago, Chicago, IL 60637, USA
Chicago Quantum Exchange, Chicago, IL 60637, USA
Kadanoff Center for Theoretical Physics, The University of Chicago, Chicago, IL 60637, USA
qBraid Co., Harper Court 5235, Chicago, IL 60615, USA
junyuliu@uchicago.edu

Zexi Lin
Pritzker School of Molecular Engineering, The University of Chicago, Chicago, IL 60637, USA
zexil@uchicago.edu

Liang Jiang
Pritzker School of Molecular Engineering, The University of Chicago, Chicago, IL 60637, USA
Chicago Quantum Exchange, Chicago, IL 60637, USA
liangjiang@uchicago.edu

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

Now we consider a classical neural network, the MLP model (see [1]). The definition is

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

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

$$\begin{aligned}
&\mathbb{E}\left[b_{i_1}^{(\ell)} b_{i_2}^{(\ell)}\right] = \delta_{i_1 i_2} C_b^{(\ell)}, \\
&\mathbb{E}\left[W_{i_1 j_1}^{(\ell)} W_{i_2 j_2}^{(\ell)}\right] = \delta_{i_1 i_2} \delta_{j_1 j_2} \frac{C_W^{(\ell)}}{n_{\ell-1}}.
\end{aligned} \tag{2}$$

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,

$$\theta_\mu(t+1) = \theta_\mu(t) - \eta \frac{d\mathcal{L}_\mathcal{A}}{d\theta_\mu}\bigg|_{\theta(t)}. \tag{3}$$

The loss function is

$$\mathcal{L}_{\mathcal{A}} \equiv \frac{1}{2} \sum_{i,\tilde{\alpha} \in \mathcal{A}} \left(z_i \left(x_{\tilde{\alpha}}; \theta \right) - y_{i,\tilde{\alpha}} \right)^2 = \frac{1}{2} \sum_{i,\tilde{\alpha} \in \mathcal{A}} \varepsilon_{i,\tilde{\alpha}}^2 \, , \tag{4}$$

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,

$$\varepsilon_{i,\tilde{\alpha}} = z_i \left(x_{\tilde{\alpha}} \right) - y_{i,\tilde{\alpha}} \, . \tag{5}$$

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 [2, 3]. 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 [4] (see also [1]).

The *barren plateau problem* is a term invented from the quantum community since [5]. 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 [5] is the following, where we define the argument as *laziness*. If we consider the gradient descent process of the variational angles,

$$\theta_\mu(t+1) = \theta_\mu(t) - \eta \frac{d\mathcal{L}_{\mathcal{A}}}{d\theta_\mu} \bigg|_{\theta(t)} \, . \tag{6}$$

and if we make a sufficiently random variational ansatz, the factor $\text{poly}(\dim \mathcal{H})$ where $\dim \mathcal{H}$ is the dimension of the Hilbert space, will appear in the formula of $d\mathcal{L}_A/d\theta_\mu$. 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 $\text{poly}(\dim \mathcal{H})$ will be the integration formula over a 2-design,

$$\int dU U_{ij} U^\dagger_{kl} = \frac{\delta_{il}\delta_{jk}}{\dim \mathcal{H}} \, , \tag{7}$$

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 7 and the LeCun parametrization 2, 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 [6]), 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

$$\begin{aligned} \frac{d\mathcal{L}_{\mathcal{A}}}{d\theta_\mu} &= \sum_{i,\tilde{\alpha}} \varepsilon_{i,\tilde{\alpha}} \frac{d\varepsilon_{i,\tilde{\alpha}}}{d\theta_\mu} = \sum_{i,\tilde{\alpha}} \varepsilon_{i,\tilde{\alpha}} \frac{dz_{i,\tilde{\alpha}}}{d\theta_\mu} \\ &= \sum_{i,\tilde{\alpha}} y_{i,\tilde{\alpha}} \frac{dz_{i,\tilde{\alpha}}}{d\theta_\mu} + \sum_{i,\tilde{\alpha}} z_{i,\tilde{\alpha}} \frac{dz_{i,\tilde{\alpha}}}{d\theta_\mu} \, . \end{aligned} \tag{8}$$

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

$$\frac{dz_{i;\alpha}^{(L)}}{db_j^{(\ell)}} = \frac{dz_{i;\alpha}^{(L)}}{dz_{j;\alpha}^{(\ell)}} ,$$
$$\frac{dz_{i;\alpha}^{(L)}}{dW_{jk}^{(\ell)}} = \sum_m \frac{dz_{i;\alpha}^{(L)}}{dz_{m;\alpha}^{(\ell)}} \frac{dz_{m;\alpha}^{(\ell)}}{dW_{jk}^{(\ell)}} = \frac{dz_{i;\alpha}^{(L)}}{dz_{j;\alpha}^{(\ell)}} \sigma_{k;\alpha}^{(\ell-1)} . \tag{9}$$

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,

$$\frac{dz_{i;\alpha}^{(L)}}{dz_{j;\alpha}^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{dz_{i;\alpha}^{(L)}}{dz_{k;\alpha}^{(\ell+1)}} \frac{dz_{k;\alpha}^{(\ell+1)}}{dz_{j;\alpha}^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{dz_{i;\alpha}^{(L)}}{dz_{k;\alpha}^{(\ell+1)}} W_{kj}^{(\ell+1)} \sigma_{j;\alpha}^{(\ell)\prime}$$
$$\text{for} \quad \ell < L ,$$
$$\frac{dz_{i;\alpha}^{(L)}}{dz_{j;\alpha}^{(L)}} = \delta_{ij} . \tag{10}$$

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

$$\begin{aligned}
\frac{dz_{i;\alpha}^{(L)}}{dz_{j;\alpha}^{(\ell)}} &= \sum_{k=1}^{n_{\ell+1}} \frac{dz_{i;\alpha}^{(L)}}{dz_{k;\alpha}^{(\ell+1)}} \frac{dz_{k;\alpha}^{(\ell+1)}}{dz_{j;\alpha}^{(\ell)}} = \sum_{k=1}^{n_{\ell+1}} \frac{dz_{i;\alpha}^{(L)}}{dz_{k;\alpha}^{(\ell+1)}} W_{kj}^{(\ell+1)} \sigma_{j;\alpha}^{(\ell)\prime} \\
&= \sum_{k_{\ell+1},k_{\ell+2}}^{n_{\ell+1},n_{\ell+2}} \frac{dz_{i;\alpha}^{(L)}}{dz_{k_{\ell+2};\alpha}^{(\ell+2)}} W_{k_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} \sigma_{j;\alpha}^{(\ell+1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \\
&= \sum_{k_{\ell+1},k_{\ell+2},\ldots,k_L}^{n_{\ell+1},n_{\ell+2},\ldots,n_L} \frac{dz_{i;\alpha}^{(L)}}{dz_{k_L;\alpha}^{(L)}} W_{k_Lj}^{(L)} W_{k_{L-1}j}^{(L-1)} \ldots W_{k_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} \\
&\times \sigma_{j;\alpha}^{(L-1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \ldots \sigma_{j;\alpha}^{(\ell+1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \\
&= \sum_{k_{\ell+1},k_{\ell+2},\ldots,k_{L-1}}^{n_{\ell+1},n_{\ell+2},\ldots,n_{L-1}} W_{i,j}^{(L)} W_{k_{L-1}j}^{(L-1)} \ldots W_{k_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} \\
&\sigma_{j;\alpha}^{(L-1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \ldots \sigma_{j;\alpha}^{(\ell+1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} .
\end{aligned} \tag{11}$$

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,

$$\begin{aligned}
&\mathbb{E}\left(\left(\frac{dz_{i;\alpha}^{(L)}}{dz_{j;\alpha}^{(\ell)}}\right)^2\right) \\
&= \sum_{k_{\ell+1},k_{\ell+2},\ldots,k_{L-1},\bar{k}_{\ell+1},\bar{k}_{\ell+2},\ldots,\bar{k}_{L-1}}^{n_{\ell+1},n_{\ell+2},\ldots,n_{L-1},n_{\ell+1},n_{\ell+2},\ldots,n_{L-1}} \mathbb{E}\left(\begin{array}{l} W_{i,j}^{(L)} W_{i,j}^{(L)} W_{k_{L-1}j}^{(L-1)} W_{\bar{k}_{L-1}j}^{(L-1)} \cdots \\ W_{k_{\ell+2}j}^{(\ell+2)} W_{\bar{k}_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} W_{\bar{k}_{\ell+1}j}^{(\ell+1)} \end{array} \right) \mathbb{E}\left(\left(\Sigma_{j;\alpha}^{(\ell);(L-1)}\right)^2\right) \\
&= \sum_{k_{\ell+1},k_{\ell+2},\ldots,k_{L-1},\bar{k}_{\ell+1},\bar{k}_{\ell+2},\ldots,\bar{k}_{L-1}}^{n_{\ell+1},n_{\ell+2},\ldots,n_{L-1},n_{\ell+1},n_{\ell+2},\ldots,n_{L-1}} \mathbb{E}\left(\begin{array}{l} W_{i,j}^{(L)} W_{i,j}^{(L)} W_{k_{L-1}j}^{(L-1)} W_{\bar{k}_{L-1}j}^{(L-1)} \cdots \\ W_{k_{\ell+2}j}^{(\ell+2)} W_{\bar{k}_{\ell+2}j}^{(\ell+2)} W_{k_{\ell+1}j}^{(\ell+1)} W_{\bar{k}_{\ell+1}j}^{(\ell+1)} \end{array} \right) \mathbb{E}\left(\left(\Sigma_{j;\alpha}^{(\ell);(L-1)}\right)^2\right) \\
&= \frac{1}{n_L} C_W^{(L)} C_W^{(L-1)} \ldots C_W^{(\ell+1)} \mathbb{E}\left(\left(\Sigma_{j;\alpha}^{(\ell);(L-1)}\right)^2\right) ,
\end{aligned} \tag{12}$$

where

$$\Sigma_{j;\alpha}^{(\ell);(L-1)} = \sigma_{j;\alpha}^{(L-1)\prime} \sigma_{j;\alpha}^{(L-2)\prime} \ldots \sigma_{j;\alpha}^{(\ell+2)\prime} \sigma_{j;\alpha}^{(\ell+1)\prime} . \tag{13}$$

We have used the Wick contraction rule and the LeCun parametrization 2 according to [1]. Plug Equation 12 back to Equation 9, 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 neural tangent kernel (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 [1]. 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

$$H_{i_1 i_2;\alpha_1\alpha_2} \equiv \sum_{\mu} \frac{dz_{i_1;\alpha_1}}{d\theta_\mu} \frac{dz_{i_2;\alpha_2}}{d\theta_\mu} \ . \tag{14}$$

The gradient descent rule will imply,

$$\delta\varepsilon_{i;\delta} = -\eta \sum_{i_1,\tilde{\alpha}\in\mathcal{A}} H_{ii_1;\delta\tilde{\alpha}}\varepsilon_{i_1,\tilde{\alpha}} \ . \tag{15}$$

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

$$H_{i_1 i_2;\alpha_1\alpha_2} = \bar{H}_{i_1 i_2;\alpha_1\alpha_2} + \Delta H_{i_1 i_2;\alpha_1\alpha_2} \ , \tag{16}$$

and we have

$$\begin{aligned} &\mathbb{E}\left(\Delta H_{i_1 i_2;\alpha_1\alpha_2}\Delta H_{i_3 i_4;\alpha_3\alpha_4}\right) = \\ &\frac{1}{n_{L-1}}\left[\delta_{i_1 i_2}\delta_{i_3 i_4}A_{(\alpha_1\alpha_2)(\alpha_3\alpha_4)} + \delta_{i_1 i_3}\delta_{i_2 i_4}B_{\alpha_1\alpha_3\alpha_2\alpha_4} + \delta_{i_1 i_4}\delta_{i_2 i_3}B_{\alpha_1\alpha_4\alpha_2\alpha_3}\right] \ . \end{aligned} \tag{17}$$

The full expressions of A, B are given in Chapter 8 of [1]. Similarly, in the statistics language, one could check [7]. 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 [1], 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.

B Some further details about concentration conditions

For concentration conditions including the quantum meta-kernel, one could see [8] 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 **??** 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 **??**. 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 [9],

$$\Delta K = \sqrt{\mathbb{E}\left(\left(K-\bar{K}\right)^2\right)} \approx \frac{\sqrt{L}}{N^2}\sqrt{\left(8\mathrm{Tr}^2\left(O^2\right) + 12\mathrm{Tr}\left(O^4\right)\right)} \ , \tag{18}$$

in the large-N limit, and we present a detailed calculation in [8] with formulas up to 4-design. Moreover, we could look at higher order corrections to the Taylor expansion by the quantum meta-

kernel (dQNTK) [10],

$$\delta\varepsilon = -\eta \sum_{\ell} \frac{d\varepsilon}{d\theta_\ell}\frac{d\varepsilon}{d\theta_\ell}\varepsilon + \frac{1}{2}\eta^2\varepsilon^2 \sum_{\ell_1,\ell_2} \frac{d^2\varepsilon}{d\theta_{\ell_1}d\theta_{\ell_2}}\frac{d\varepsilon}{d\theta_{\ell_1}}\frac{d\varepsilon}{d\theta_{\ell_2}}$$
$$\equiv -\eta K\varepsilon + \frac{1}{2}\eta^2\varepsilon^2\mu \, . \tag{19}$$

Here $\mu = \sum_{\ell_1,\ell_2} \frac{d^2\varepsilon}{d\theta_{\ell_1}d\theta_{\ell_2}}\frac{d\varepsilon}{d\theta_{\ell_1}}\frac{d\varepsilon}{d\theta_{\ell_2}}$ 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 [1]), and we have

$$\Delta\mu = \sqrt{\mathbb{E}\left(\mu^2\right)} \approx \frac{\sqrt{32}L}{N^3}\mathrm{Tr}^{3/2}\left(O^2\right) \, , \tag{20}$$

in the large-N limit. The condition where the QNTK estimation in Equation **??** is valid when

$$\Delta K \ll K \Leftrightarrow L \gg 1 \, , \tag{21}$$
$$\frac{1}{2}\eta^2\varepsilon^2\Delta\mu \ll \eta\bar{K}\varepsilon \Leftrightarrow \eta\varepsilon(0)\frac{L}{N^3}\mathrm{Tr}^{3/2}\left(O^2\right) \ll \frac{L\mathrm{Tr}\left(O^2\right)}{N^2}$$
$$\Leftrightarrow \frac{\eta\Omega_O}{N}\varepsilon(0) \ll 1 \, . \tag{22}$$

We call the conditions 21 and 22 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 22 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 $\Omega_O \geq \mathcal{O}(N)$, so Equation 22 is automatically satisfied. The condition that usually matters is Equation 21, 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

$$\frac{\Delta K}{\bar{K}} = \mathcal{O}\left(\frac{1}{\sqrt{L}}\right) \, , \frac{\Delta\mu}{\bar{K}} = \mathcal{O}\left(\frac{1}{N}\right) \, . \tag{23}$$

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 [1]. The origin of this equivalence comes from the similarity between Equation **??** and Equation 24, while a higher level (but heuristic) understanding comes from a connection between quantum field theory and the large-width expansion [1, 11, 12] and a similarity between Feynman rules in quantum field theory and matrix models [13], 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.

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 [1, 11, 12, 14] 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,

$$\mathbb{E}\left(W_{ij}W_{kl}\right) = \frac{\sigma_W^2}{\text{width}}\delta_{ik}\delta_{jl} \, , \tag{24}$$

or more generally,

$$\mathbb{E}\left(W_{i_1j_1}W_{i_2j_2}\ldots W_{i_{2k-1}j_{2k-1}}W_{i_{2k}j_{2k}}\right) = \mathcal{O}(\frac{1}{\text{poly(width)}}) \, , \tag{25}$$

for all positive integer k. Here, we are considering the multilayer perceptron (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 [15, 16]. Matrix models have been studied for a long time, around and after the second string theory revolution [13], and they have deep connections to the holographic principle [17] and the AdS/CFT correspondence [18, 19]. Haar ensembles are toy versions of matrix models, which have been widely studied as toy models of chaotic quantum black holes [20, 21]. The similarity between the LeCun parametrization 24 and the 1-design Haar integral formula

$$\mathbb{E}(U_{ij}U^{\dagger}_{kl}) = \frac{1}{\dim \mathcal{H}}\delta_{il}\delta_{jk} \, , \tag{26}$$

or more generally,

$$\mathbb{E}\left(U_{i_1 j_1}U^{\dagger}_{i_2 j_2}\dots U_{i_{2k-1}j_{2k-1}}U^{\dagger}_{i_{2k}j_{2k}}\right) = \mathcal{O}(\frac{1}{\text{poly}(\dim \mathcal{H})}) \, , \tag{27}$$

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.

D Noises

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

$$\theta_\ell(t+1) - \theta_\ell(t) \equiv \delta\theta_\mu = -\eta\frac{\partial \mathcal{L}}{\partial \theta_\ell} = i\eta\left\langle \Psi_0 \left| V^{\dagger}_{+,\ell}\left[X_\ell, V^{\dagger}_{-,\ell}OV_{-,\ell}\right]V_{+,\ell}\right|\Psi_0\right\rangle \, , \tag{28}$$

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 $\mathcal{N}(0, \sigma^2_\theta)$.

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

$$\delta\varepsilon = -\eta\varepsilon K + \sum_\ell \frac{\partial \varepsilon}{\partial \theta_\ell}\Delta\theta_\ell \, . \tag{29}$$

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

$$\sum_\ell \frac{\partial \varepsilon}{\partial \theta_\ell}\Delta\theta_\ell \sim \mathcal{N}(0, K\sigma^2_\theta) \, . \tag{30}$$

Thus, we could write the recursion relation as

$$\delta\varepsilon = -\eta\varepsilon K + \sqrt{K}\Delta\theta \, . \tag{31}$$

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

$$\varepsilon(t) = (1-\eta K)^t\varepsilon(0) + \sqrt{K}\sum_{i=0}^{t-1}(1-\eta K)^i\Delta\theta(t-1-i) \, . \tag{32}$$

Now, we have

$$\sqrt{K}\sum_{i=0}^{t-1}(1-\eta K)^i\Delta\theta(t-1-i)\sim\mathcal{N}(0,K\sigma_\theta^2\sum_{i=0}^{t-1}(1-\eta K)^{2i})$$
$$=\mathcal{N}(0,\sigma_\theta^2\frac{1-(1-\eta K)^{2t}}{\eta(2-\eta K)})\,. \tag{33}$$

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

$$\overline{\varepsilon^2}(t)=(1-\eta K)^{2t}\left(\varepsilon^2(0)-\frac{\sigma_\theta^2}{\eta(2-\eta K)}\right)+\frac{\sigma_\theta^2}{\eta(2-\eta K)}\,. \tag{34}$$

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

$$\overline{\varepsilon^2}(\infty)=\frac{\sigma_\theta^2}{\eta(2-\eta K)}\approx\mathcal{O}(\frac{\sigma_\theta^2}{\eta})\,, \tag{35}$$

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,

$$(1-\eta K)^{T_{\text{noise}}}\varepsilon(0)\approx\sigma_\theta\sqrt{\frac{1-(1-\eta K)^{2T_{\text{noise}}}}{\eta(2-\eta K)}}\,. \tag{36}$$

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

$$T_{\text{noise}}\approx\frac{\log\left(\frac{\sigma_\theta}{\sqrt{2\varepsilon^2(0)\eta-\varepsilon^2(0)\eta^2K+\sigma_\theta^2}}\right)}{\log(1-\eta K)}\,,$$
$$\varepsilon(T_{\text{noise}})=2(1-\eta K)^{T_{\text{noise}}}\varepsilon(0)=\frac{2\sigma_\theta^2}{\sqrt{\varepsilon(0)^2(2\eta-\eta^2K)+\sigma_\theta^2}}\varepsilon(0)\,. \tag{37}$$

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.

E Numerical results

In this part, we show some simple numerical evidences based on the analysis done in [8]. We will use the randomized version of the hardware-efficient variational ansatz defined in [8]. In Figure 1, 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 [8]. 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

$$\bar{\varepsilon}=\sqrt{\frac{2}{\pi}}\cdot\frac{\sigma_\theta}{\sqrt{2\eta-\eta^2K}}\,, \tag{38}$$

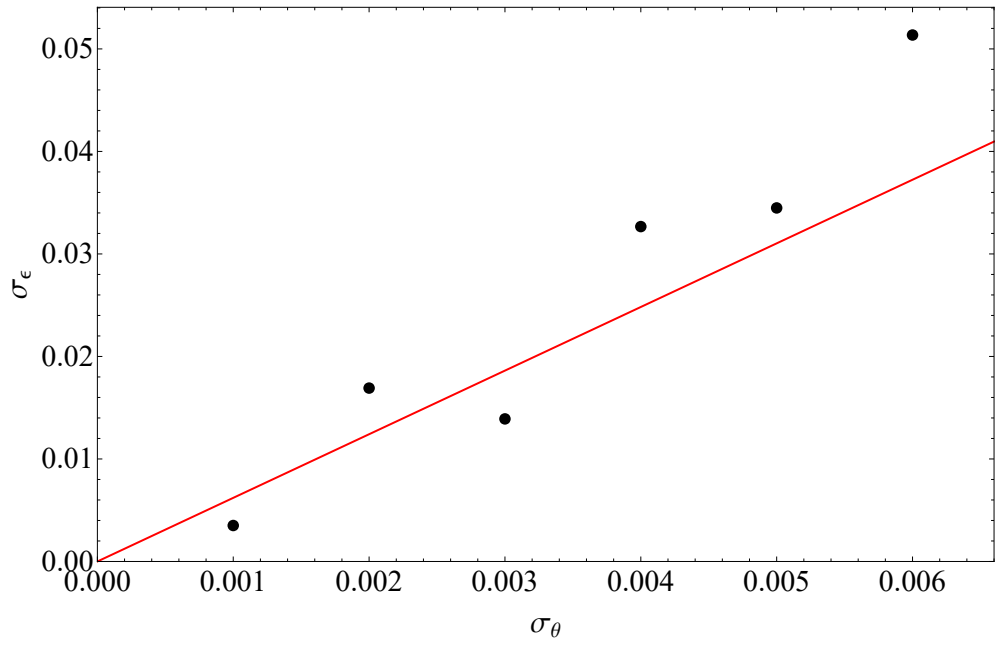

Figure 1: Noise standard deviation σ_θ as a function of standard deviation of final residual error σ_ε after training long enough time, with both numerical result (black dots) and theoretical prediction (red line). In this figure, $\eta = 0.005$, $K \approx 25$, $\varepsilon(0) \approx 1$.

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.

More precisely, in Figure 1, 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.

In Figure 2, 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.

References

[1] Daniel A Roberts, Sho Yaida, and Boris Hanin. The principles of deep learning theory. *arXiv preprint arXiv:2106.10165*, 2021.

[2] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[3] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

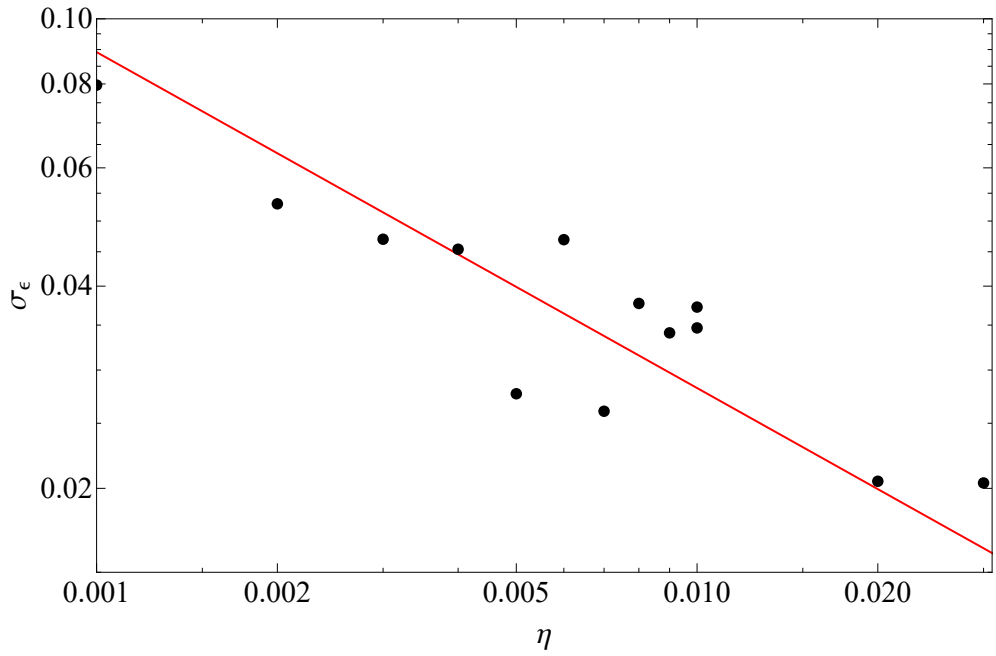

Figure 2: Standard deviation of final residual error σ_ε as a function of learning rate η after training long enough time, with both numerical result (black dots) and theoretical prediction (red line). In this figure, $\sigma_\theta = 0.005$, $K \approx 35$, $\varepsilon(0) \approx 1$, $t = 100$.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[5] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1): 1–6, 2018.

[6] Anna Golubeva, Behnam Neyshabur, and Guy Gur-Ari. Are wider nets better given the same number of parameters? *arXiv preprint arXiv:2010.14495*, 2020.

[7] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

[8] Junyu Liu, Khadijeh Najafi, Kunal Sharma, Francesco Tacchino, Liang Jiang, and Antonio Mezzacapo. Analytic theory for the dynamics of wide quantum neural networks. *Physical Review Letters*, 130(15):150601, 2023.

[9] Marco Cerezo and Patrick J Coles. Higher order derivatives of quantum neural networks with barren plateaus. *Quantum Science and Technology*, 6(3):035006, 2021.

[10] Junyu Liu, Francesco Tacchino, Jennifer R. Glick, Liang Jiang, and Antonio Mezzacapo. Representation Learning via Quantum Neural Tangent Kernels. *PRX Quantum*, 3(3):030323, 2022. doi: 10.1103/PRXQuantum.3.030323.

[11] Ethan Dyer and Guy Gur-Ari. Asymptotics of wide networks from feynman diagrams. *arXiv preprint arXiv:1909.11304*, 2019.

[12] James Halverson, Anindita Maiti, and Keegan Stoner. Neural networks and quantum field theory. *Machine Learning: Science and Technology*, 2(3):035002, 2021.

[13] Edward Witten. String theory dynamics in various dimensions. *Nucl. Phys. B*, 443:85–126, 1995. doi: 10.1016/0550-3213(95)00158-O.

[14] Daniel A Roberts. Why is ai hard and physics simple? *arXiv preprint arXiv:2104.00008*, 2021.

[15] Tom Banks, W. Fischler, S. H. Shenker, and Leonard Susskind. M theory as a matrix model: A Conjecture. *Phys. Rev. D*, 55:5112–5128, 1997. doi: 10.1103/PhysRevD.55.5112.

[16] David Eliecer Berenstein, Juan Martin Maldacena, and Horatiu Stefan Nastase. Strings in flat space and pp waves from N=4 superYang-Mills. *JHEP*, 04:013, 2002. doi: 10.1088/1126-6708/2002/04/013.

[17] Leonard Susskind. The World as a hologram. *J. Math. Phys.*, 36:6377–6396, 1995. doi: 10.1063/1.531249.

[18] Juan Martin Maldacena. The Large N limit of superconformal field theories and supergravity. *Adv. Theor. Math. Phys.*, 2:231–252, 1998. doi: 10.1023/A:1026654312961.

[19] Edward Witten. Anti-de Sitter space and holography. *Adv. Theor. Math. Phys.*, 2:253–291, 1998. doi: 10.4310/ATMP.1998.v2.n2.a2.

[20] Patrick Hayden and John Preskill. Black holes as mirrors: Quantum information in random subsystems. *JHEP*, 09:120, 2007. doi: 10.1088/1126-6708/2007/09/120.

[21] Daniel A. Roberts and Beni Yoshida. Chaos and complexity by design. *JHEP*, 04:121, 2017. doi: 10.1007/JHEP04(2017)121.