Trainability of Dissipative Perceptron-Based Quantum Neural Networks

Kunal Sharma, M. Cerezo, Lukasz Cincio, and Patrick J. Coles

Phys. Rev. Lett. **128**, 180505 — Published 6 May 2022

DOI: 10.1103/PhysRevLett.128.180505

Trainability of Dissipative Perceptron-Based Quantum Neural Networks

Kunal Sharma,[1, 2, *] M. Cerezo,[1, 3, *] Lukasz Cincio,[1] and Patrick J. Coles[1]

[1] *Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA*
[2] *Hearne Institute for Theoretical Physics and Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA USA*
[3] *Center for Nonlinear Studies, Los Alamos National Laboratory, Los Alamos, NM, USA*

Several architectures have been proposed for quantum neural networks (QNNs), with the goal of efficiently performing machine learning tasks on quantum data. Rigorous scaling results are urgently needed for specific QNN constructions to understand which, if any, will be trainable at a large scale. Here, we analyze the gradient scaling (and hence the trainability) for a recently proposed architecture that we call dissipative QNNs (DQNNs), where the input qubits of each layer are discarded at the layer's output. We find that DQNNs can exhibit barren plateaus, i.e., gradients that vanish exponentially in the number of qubits. Moreover, we provide quantitative bounds on the scaling of the gradient for DQNNs under different conditions, such as different cost functions and circuit depths, and show that trainability is not always guaranteed. Our work represents the first rigorous analysis of the scalability of a perceptron-based QNN.

Introduction.—Neural networks (NN) have impacted many fields such as neuroscience, engineering, computer science, chemistry, and physics [1]. However, their historical development has seen periods of great progress interleaved with periods of stagnation, due to serious technical challenges [2]. The perceptron was introduced early on as an artificial neuron [3], but it was only realized later that a multi-layer perceptron (now known as a feedforward NN) had much greater power than a single-layer one [1, 2]. Still there was the major issue of how to train multiple layers, and this was eventually addressed by the backpropagation method [4].

Motivated by the success of NNs and the advent of Noisy Intermediate-Scale Quantum devices [5], there has been tremendous effort to develop Quantum Neural Networks (QNNs) [6]. The hope is that QNNs will harness the power of quantum computers to outperform their classical counterparts on machine learning tasks [7, 8], especially for quantum data or tasks that are inherently quantum in nature [9].

Despite several QNN proposals that have been successfully implemented [10–17], more research is needed on the advantages and limitations of specific architectures. Delving into potential scalability issues of QNNs could help to prevent a "winter" for these models, like what was seen historically for classical NNs. This has motivated recent works studying the scaling of gradients in QNNs [18, 19]. There, it was shown that variational quantum algorithms [20–30], which aim to train QNNs to accomplish specific tasks, may exhibit gradients that vanish exponentially with the system size. This so-called barren-plateau phenomenon, where the parameters cannot be efficiently trained for large implementations, was demonstrated for hardware-efficient QNNs, where quantum gates are arranged in a brick-like structure that matches the connectivity of the quantum device [18, 19].

Analyzing the existence of barren plateaus in QNNs is paramount to determining if they can lead to a quantum speedup. This is due to the fact that exponentially vanishing gradients imply that the precision needed to estimate such gradients grows exponentially. Since the standard goal of quantum algorithms is polynomial scaling as opposed to the typical exponential scaling of classical algorithms, a QNN with exponentially vanishing gradients has no hope of achieving this goal. On the other hand, a QNN with gradients that vanish polynomially means that the algorithm requires a polynomial precision, and hence that the hope of quantum speedup is preserved.

Here, we analyze the trainability and the existence of barren plateaus in a class of QNNs that we refer to as *dissipative QNNs* (DQNNs). In a DQNN each node within the network corresponds to a qubit [31], and the connections in the network are modelled by quantum perceptrons [32–37]. The term dissipative refers to the fact that ancillary qubits form the output layer, while the qubits from the input layer are discarded. This architecture has seen significant recent attention and has been proposed as a scalable approach to QNNs [37–39]. In particular, in [37], based on small scale numerical experiments, it was speculated that dissipative quantum neural networks do not suffer from the barren plateau (vanishing gradient) problem. However, contrary to [37], we here analytically prove that DQNNs are not immune to barren plateaus. For example, DQNNs with deep global perceptrons are untrainable despite the dissipative nature of the architecture.

Here we study the large-scale trainability of DQNNs. In particular, we focus on tasks where DQNNs are employed to learn a unitary matrix connecting input and output quantum states and for general supervised quantum machine learning tasks where training data consists of quantum states and corresponding classical labels. For these tasks, we show that the barren plateau phenomenon can also arise in DQNNs. We also discuss certain conditions (e.g., the structure and depth of the DQNN) under which one could avoid a barren plateau and achieve train-

[*] The first two authors contributed equally to this work.

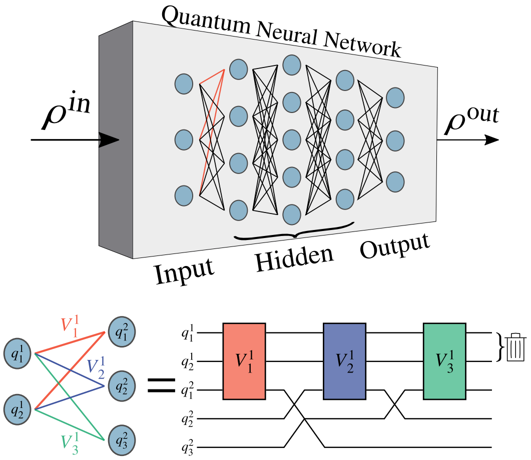

FIG. 1. Schematic diagram of a dissipative perceptron-based quantum neural network (DQNN). Top: The DQNN is composed of input, hidden, and output layers. Each node in the network corresponds to a qubit, which can be connected to qubits in adjacent layers via perceptrons (depicted as lines). The input and output of the DQNN are quantum states denoted as $\rho^{\rm in}$ and $\rho^{\rm out}$, respectively. Bottom: Quantum circuit description of the DQNN. The j-th qubit of the l-th layer is denoted q_j^l. Each perceptron corresponds to a unitary operation on the qubits it connects, with V_j^l denoting the j-th perceptron in the l-th layer.

ability. In particular, our work implies that scalability is not guaranteed, and without careful thought of the structure of DQNNs, their gradients may vanish exponentially in the system sizen. As a by-product of our analysis of specific perceptron architectures, we also show that hardware-efficient QNNs are special cases of DQNNs. Therefore, many important results for hardware-efficient QNNs, such as the ones studied in Refs. [18, 19] also hold for DQNNs. Finally, we remark that we employ novel analytical techniques in our proofs (different from those used in Refs. [18, 19]), which were necessary to develop due to the dissipative nature of DQNNs. Our techniques may be broadly useful in the study of the scaling of other QNN architectures.

Preliminaries.— Let us first introduce the DQNN architecture. As schematically shown in Fig. 1, the DQNN is composed of a series of layers (input, hidden, and ouput) where the qubits at each node are connected via perceptrons. A quantum perceptron is defined as an arbitrary unitary operator with m input and k output qubits. For simplicity, we consider the case when $k=1$, so that each perceptron acts on $m+1$ qubits. The case of arbitrary k is presented in the Supplemental Material.

The qubits in the input layer are initialized to a state $\rho^{\rm in}$, while all qubits in the hidden and output layers are initialized to a fiduciary state such as $|\mathbf{0}\rangle_{\rm hid,out} = |0\dots0\rangle_{\rm hid,out}$. Henceforth we employ the notation "in", "hid", and "out" to indicate operators on qubits in the input, hidden, and output layers, respectively. The output state of the DQNN is a quantum state $\rho^{\rm out}$ (generally mixed) which can be expressed as

$$\rho^{\rm out} \equiv {\rm Tr}_{\rm in,hid}\left[V(\rho^{\rm in}\otimes|\mathbf{0}\rangle_{\rm hid,out}\langle\mathbf{0}|)V^\dagger\right]\,, \qquad (1)$$

with $V = V^{\rm out}_{n_{\rm out}}\dots V^1_{n_1}\dots V^1_1$, and where V_j^l is the perceptron unitary on the l-th layer acting on the j-th output qubit. Here n_l indicates the number of qubits in the l-th layer.

Let us now make two important remarks. First, note that the order in which the perceptrons act is relevant, as in general the unitaries V_j^l will not commute. Second, we remark that for this architecture the perceptrons are applied layer-by-layer, meaning that once all V_j^l (for fixed l) have been applied and the information has propagated forward between layers $l-1$ and l, one can discard the qubits in layer $l-1$. This implies that the width of the DQNN depends on the number of qubits in two adjacent layers and not in the total number of qubits in the network.

To train the DQNN, we assume repeatable access to training data in the form of pairs $\{|\phi_x^{\rm in}\rangle, |\phi_x^{\rm out}\rangle\}$, with $x=1,\dots,N$. We note that, as discussed in the Supplemental Material, our results also hold more generally for supervised quantum machine learning tasks where the training data is of the form $\{|\phi_x^{\rm in}\rangle, y_x\}$, with y_x a label assigned to the input state $|\phi_x^{\rm in}\rangle$ [40].

We then define a cost function (or loss function) which quantifies how well the DQNN reproduces the training data. We assume that the cost is of the form

$$C = \frac{1}{N}\sum_{x=1}^{N} C_x\,, \quad \text{with} \quad C_x = {\rm Tr}[O_x\rho_x^{\rm out}]\,. \qquad (2)$$

As discussed below, in general there are multiple choices for the operator O_x which lead to faithful cost functions, i.e., costs that are extremized if and only if one perfectly learns the mapping on the training data. If the circuit description of output states is provided, one can employ the inverse of the corresponding unitary on the output of a DQNN [41]. Then a measurement in the computational basis estimates the cost function. Otherwise, one can employ a recently developed procedure based on classical shadows to estimate the state overlap [42].

When O_x acts non-trivially on all qubits of the output layer, we use the term *global cost function*, denoted as C^G. Here one usually compares objects (states or operators) living in exponentially large Hilbert spaces. For instance, choosing

$$O_x^G = \mathbb{1} - |\phi_x^{\rm out}\rangle\langle\phi_x^{\rm out}|\,, \qquad (3)$$

leads to a global cost function that quantifies the average fidelity between each $\rho_x^{\rm out}$ and $|\phi_x^{\rm out}\rangle$.

As shown in Ref. [19], local cost functions do not exhibit a barren plateau for shallow hardware-efficient QNNs. Therefore, it is important to study if local observables can also lead to trainability guarantees in DQNNs. Henceforth, we use the term *local cost function*, denoted as C^L, for the cases when the operator O_x acts non-trivially on a small number of qubits in the output layer.

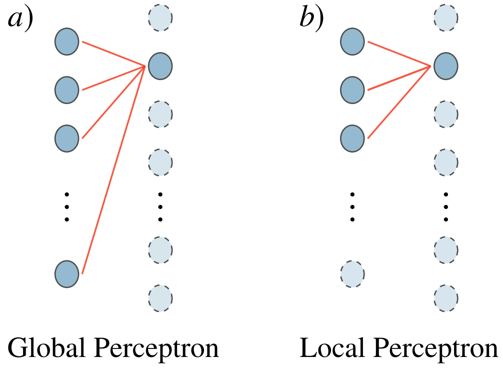

FIG. 2. Global and local perceptrons. a) The global perceptron acts non-trivially on all input qubits, i.e., $m = n$. b) The local perceptron acts non-trivially only on a small number of input qubits. For the case shown, $m = 3$.

Since the global cost in (3) is a state fidelity function, in general it will not be possible to design a corresponding faithful local cost. Therefore, we restrict ourselves to the case when $|\phi_x^{\text{out}}\rangle$ is a tensor-product state across n_{out} qubits $|\phi_x^{\text{out}}\rangle = |\psi_{x,1}^{\text{out}}\rangle \otimes \ldots \otimes |\psi_{x,n_{\text{out}}}^{\text{out}}\rangle$. Then, we can define the following local observable:

$$O_x^L = \mathbb{1} - \frac{1}{n_{\text{out}}} \sum_{j=1}^{n_{\text{out}}} |\psi_{x,j}^{\text{out}}\rangle\langle\psi_{x,j}^{\text{out}}| \otimes \mathbb{1}_{\bar{j}} \,, \tag{4}$$

where $\mathbb{1}_{\bar{j}}$ denotes the identity over all qubits in the output layer except for qubit j. Equation (4) leads to a faithful local cost that vanishes under the same condition as the global cost defined from (3) [41, 43].

Finally let us introduce the term *global perceptron* to refer to the case when the perceptron V_j^l acts non-trivially on *all* qubits in the l-th layer, i.e., when $m = n_{l-1}$. On the other hand, a *local perceptron* is defined as a unitary V_j^l acting on a number of qubits $m \in \mathcal{O}(1)$ which is independent of n_{l-1}. Figure 2 schematically shows a global and a local perceptron.

To analyze the existence of barren plateaus and the trainability of the DQNN one needs to define an ansatz and a training method for the perceptrons. In what follows we consider two general training approaches.

Random parameterized quantum circuits.—We first consider the case where the perceptrons are parametrized quantum circuits (i.e., variational circuits) that can be expressed as a sequence of parameterized and unparameterized gates from a given gate alphabet [18, 44]. That is, the perceptrons are of the form

$$V_j^l(\boldsymbol{\theta}_j^l) = \prod_{k=1}^{\eta_j^l} R_k(\theta^k) W_k \,, \tag{5}$$

with $R_k(\theta^k) = e^{-(i/2)\theta^k \Gamma_k}$, W_k an unparameterized unitary, and where Γ_k is a Hermitian operator with $\text{Tr}[\Gamma_k^2] \leqslant 2^{n+1}$. Such parameterization is widely used as it can allow for a straightforward evaluation of the cost function gradients, and since in general its quantum circuit description can be easily obtained [45–47].

A common strategy for training random parameterized quantum circuits is to randomly initialize the parameters in (5), and employ a training loop to minimize the cost function. To analyze the trainability of the DQNN we compute the variance of the partial derivative $\partial C/\partial\theta^\nu \equiv \partial_\nu C$, where θ^ν belongs to a given V_j^l

$$\text{Var}[\partial_\nu C] = \left\langle (\partial_\nu C)^2 \right\rangle - \left\langle \partial_\nu C \right\rangle^2 \,. \tag{6}$$

Here the notation $\langle \cdots \rangle$ indicates the average over all randomly initialized perceptrons. From (5), we find

$$\partial_\nu C = \frac{i}{2N} \sum_{x=1}^{N} \text{Tr}\Big[A_j^l \tilde{\rho}_x^{\text{in}} (A_j^l)^\dagger [\mathbb{1}_{\bar{j}}^{\bar{l}} \otimes \Gamma_k, (B_j^l)^\dagger \tilde{O}_x B_j^l] \Big] \,, \tag{7}$$

where we have defined

$$B_j^l = \mathbb{1}_{\bar{j}}^{\bar{l}} \otimes \prod_{k=1}^{\nu-1} R_k(\theta^k) W_k \,, \; A_j^l = \mathbb{1}_{\bar{j}}^{\bar{l}} \otimes \prod_{k=\nu}^{\eta_j^l} R_k(\theta^k) W_k \,, \tag{8}$$

such that $\mathbb{1}_{\bar{j}}^{\bar{l}} \otimes V_j^l = A_j^l B_j^l$, and where $\mathbb{1}_{\bar{j}}^{\bar{l}}$ indicates the identity on all qubits on which V_j^l does not act. Note that the trace in (7) is over *all* qubits in the DQNN. In addition, we define

$$\begin{aligned} \tilde{\rho}_x^{\text{in}} &= V_{j-1}^l \ldots V_1^1 (\rho_x^{\text{in}} \otimes |\mathbf{0}\rangle\langle\mathbf{0}|_{\text{hid,out}}) (V_1^1)^\dagger \ldots (V_{j-1}^l)^\dagger \,, \\ \tilde{O}_x &= (V_{j+1}^l)^\dagger \ldots (V_{n_{\text{out}}}^{\text{out}})^\dagger (\mathbb{1}_{\text{in,hid}} \otimes O_x) V_{n_{\text{out}}}^{\text{out}} \ldots V_{j+1}^l \,. \end{aligned}$$

If the perceptron V_j^l is sufficiently random so that A_j^l, B_j^l, or both, form independent unitary 1-designs, then we find that $\langle \partial_\nu C \rangle = 0$ (see Supplemental Material). In this case, $\text{Var}[\partial_s C]$ quantifies (on average) how much the gradient concentrates around zero. Hence, exponentially small $\text{Var}[\partial_s C]$ values would imply that the slope of the cost function landscape is insufficient to provide cost-minimizing directions.

Here we recall that a t-design is a set of unitaries $\{V_y \in U(d)\}_{y \in Y}$ (of size $|Y|$) on a d-dimensional Hilbert space such that for every polynomial $P_t(V_y)$ of degree at most t in the matrix elements of V_y, and of $V_y^\dagger$ one has [48] $\langle P_t(V) \rangle_V = \frac{1}{|Y|} \sum_{y \in Y} P_t(V_y) = \int d\mu(V) P_t(V)$, where the integral is over the unitary group $U(d)$.

Let us assume for simplicity the case when the DQNN input and output layers have the same number of qubits ($n_{\text{in}} = n_{\text{out}} = n$). As shown in the Supplemental Material, the following theorem holds.

Theorem 1. *Consider a DQNN with deep global perceptrons parametrized as in* (5)*, such that* A_j^l*,* B_j^l *in* (8) *and* V_j^l $(\forall j, l)$ *form independent* 2*-designs over* $n+1$ *qubits. Then, the variance of the partial derivative of the cost function with respect to* θ^ν *in* V_j^l *is upper bounded as*

$$\text{Var}[\partial_\nu C^G] \leqslant g(n), \quad \text{with} \quad g(n) \in \mathcal{O}\left(1/2^{2n}\right) \,, \tag{9}$$

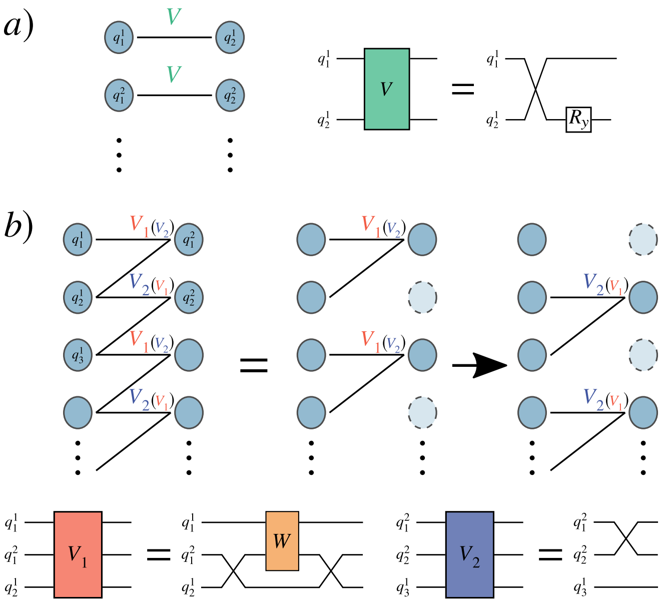

FIG. 3. Shallow local perceptrons ansatzes. a) Here $m = 1$ so that each perceptron acts on a single input and output qubit. Moreover, for all j and l we have $V_j^l = V$. The unitaries V are simply given by a SWAP operator followed by a single qubit rotation around the y axis. b) Local perceptrons V_j^l with $m = 2$. The local perceptrons are given by the unitaries V_1, or V_2. Specifically, for l odd on j odd (even) $V_j^l = V_1(V_2)$, while for l even and j odd (even) we have $V_j^l = V_2(V_1)$. Here we also show the order in which the perceptrons are applied so that we first implement the unitaries with j odd, followed by the unitaries with j even. The W gate in V_1 forms a local 2-design on two qubits.

if O_x is the global operator of (3), *and upper bounded as*

$$\mathrm{Var}[\partial_\nu C^L] \leqslant h(n), \quad \textit{with} \quad h(n) \in \mathcal{O}\left(1/2^n\right) , \qquad (10)$$

when O_x is the local operator in (4).

Theorem 1 shows that DQNNs with deep global perceptron unitaries that form 2-designs [49, 50] exhibit barren plateaus for global and local cost functions. An immediate question that follows is whether barren plateaus still arise for shallow perceptrons, which cannot form 2-designs on $n+1$ qubits. In what follows we analyze specific cases of shallow local perceptrons for which results can be obtained.

Let us first consider the simple perceptrons of Fig. 3(a), where $m = 1$, and where R_y denotes a single qubit rotation around the y axis: $R_y(\theta^\nu) = e^{-i\theta^\nu Y/2}$ (with all angles randomly initialized). In this case one recovers the toy model example of [19], and we know that if O_x is the global operator of (3), then $\mathrm{Var}[\partial_\nu C^G] = \frac{1}{8}\left(\frac{3}{8}\right)^{n-1}$. On the other hand, if O_x is the local operator in (4), then $\mathrm{Var}[\partial_\nu C^L] = \frac{1}{8n^2}$.

These results suggest that DQNNs with simple shallow local perceptrons and global cost functions are untrainable when randomly initialized. On the other hand, they also indicate that barren plateaus for DQNNs might be avoided by employing: (1) shallow (local) perceptrons, and (2) local cost functions.

Let us now consider the shallow local perceptron of Fig. 3(b), where each unitary W forms a local 2-design on two qubits. For this architecture the ensuing DQNN can be *exactly* mapped into a layered hardware-efficient ansatz as in [19], where two layers of the DQNN correspond to a single layer of the hardware-efficient ansatz [51]. Note that this mapping is not general, but rather valid for the specific architecture in Fig. 3(b). As shown in Ref. [19], when employing a global cost function, with O_x given by (3), one finds that if the number is layers is $\mathcal{O}(\mathrm{poly}(\log(n)))$, then the DQNN cost function exhibits barren plateaus as

$$\mathrm{Var}[\partial_\nu C^G] \leqslant \widehat{f}(n)\,, \quad \text{with} \quad \widehat{f}(n) \in \mathcal{O}\left((\sqrt{3}/4)^n\right) . \quad (11)$$

On the other hand, for a local cost function with O_x given by (4), if the number of layers is in $\mathcal{O}(\log(n))$, then there is no barren plateau [19] as

$$\widehat{g}(n) \leqslant \mathrm{Var}[\partial_\nu C^L]\,, \quad \text{with} \quad \widehat{g}(n) \in \Omega\left(1/\,\mathrm{poly}(n)\right) . \quad (12)$$

Here we remark that (12) was obtained following the same assumptions as those used in Corollary 2 of [19]. Note that obtaining a lower bound for the variance implies that the DQNN trainability is guaranteed.

Parameter matrix multiplication.—While in random parametrized quantum circuits one optimizes and trains a single gate angle at a time, other optimization approaches can also be considered. In what follows we analyze the trainability for a method introduced in Ref. [37] where at each time-step all perceptrons are simultaneously optimized.

In this training approach, which we call parameter matrix multiplication, the perceptrons are not explicitly decomposed into quantum circuits, but rather are treated as unitary matrices. The perceptrons $V_j^l(0)$ are randomly initialized at time-step zero, and at each step s they are updated via

$$V_j^l(s+\varepsilon) = e^{i\varepsilon H_j^l(s)} V_j^l(s)\,. \qquad (13)$$

The matrices H_j^l are such that $\mathrm{Tr}[(H_j^l)^2] \leqslant 2^{n+1}$ and are parametrized as $H_j^l(s) = \sum_{\boldsymbol{uv}} h_{j,\boldsymbol{u},\boldsymbol{v}}^l X^{\boldsymbol{u}} Z^{\boldsymbol{v}}$, with $X^{\boldsymbol{u}} Z^{\boldsymbol{v}} = X_1^{u_1} Z_1^{v_1} \otimes X_2^{u_2} Z_2^{v_2} \ldots$, and where X_j and Z_j are Pauli operators on qubit j. The matrices $K_j^l(s)$ are called *parameter matrices*, and at each time-step the coefficients $h_{j,\boldsymbol{u},\boldsymbol{v}}^l$ need to be optimized. As shown in the Supplemental Material, if at least one perceptron $V_j^l(0)$ is sufficiently random so that it forms a global unitary 1-design, then we find $\langle \partial C/\partial s \rangle \equiv \langle \partial_s C \rangle = 0$.

As proved in the Supplemental Material, the following theorem holds.

Theorem 2. *Consider a DQNN with deep global perceptrons, which are updated via the parameter matrix multiplication of* (13). *Suppose that for all j, l, the $V_j^l(0)$ perceptrons form independent 2-designs over $n+1$ qubits. Then the variance of the partial derivative of the cost function with respect to the time-step parameter s is upper bounded as*

$$\mathrm{Var}[\partial_s C] \leqslant f(n), \quad \textit{with} \quad f(n) \in \mathcal{O}\left(1/2^n\right), \tag{14}$$

when O_x is the global operator of (3), *or the local operator in* (4).

Although the updating method in (13) simultaneously updates all perceptrons at each time-step, Theorem 2 implies that barren plateaus also arise when using the parameter matrix multiplication method.

We note that our proof techniques invoke the pure state properties of input and output states. Since the output state of a randomly initialized DQNN will be close to a maximally mixed across any bipartite cut [52], we speculate that our results can be extended to expectation values of arbitrary Hamiltonian. We leave this question for future work.

Conclusions.—In this work we analyzed the trainability of a special class of Quantum Neural Networks (QNNs) called Dissipative QNNs (DQNNs). We first proved that the trainability of DQNNs is not always guaranteed as they can exhibit barren plateaus in their cost function landscape. The existence of such barren plateaus was linked to the localities (i.e., the number of qubits they act non-trivially on) of the perceptrons and of the cost function. Specifically, we showed that: (1) DQNNs with deep global perceptrons are untrainable despite the dissipative nature of the architecture, and (2) for shallow and local perceptrons, employing global cost functions leads to barren plateaus, while using local costs avoids them. We note that our results are completely general for DQNN architectures, e.g., covering arbitrary numbers of hidden layers and general perceptrons acting on any number of qubits.

In addition, we provided a specific architecture for DQNNs with local shallow perceptrons that can be exactly mapped to a layered hardware-efficient ansatz. This result not only indicates that hardware-efficient QNNs can be represented as DQNNs, but it also allows us to derive trainability guarantees for these DQNNs. In this case, since the perceptrons are local, each neuron only receives information from a small number of qubits in the previous layer. Such architecture is reminiscent of classical convolutional neural networks, which are known to avoid some of the trainability problems of fully connected networks [53].

These results show that much work needs to be done to understand the trainability of QNNs and guarantee that they can provide a quantum speedup over classical neural networks. For instance, interesting future research directions are QNN-specific-optimizers [54–57], analyzing the resilience of QNNs to noise [22, 41], and strategies to prevent barren plateaus [58–61]. Another interesting direction is to extend our results to the case when the input and output states are mixed states, particularly when the goal is to match marginals of the target output state and the output of a DQNN [62]. Furthermore, exploring architectures beyond DQNNs and hardware-efficient QNNs would be of interest, particularly if such architectures have large-scale trainability.

Acknowledgments

We thank Jarrod McClean, Tobias Osborne, and Andrew Sornborger for helpful conversations. All authors acknowledge support from LANL's Laboratory Directed Research and Development (LDRD) program. MC was also supported by the Center for Nonlinear Studies at LANL. PJC also acknowledges support from the LANL ASC Beyond Moore's Law project. This work was also supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under the Accelerated Research in Quantum Computing (ARQC) program.

Supplemental Material.—The Supplemental Material contains details of our proofs and References [63, 64].

Note Added.—Our work is the first to analyze barren plateaus in the context of data science applications, and also the first to consider perceptron-based quantum neural networks (QNNs). Our work has inspired more recent studies of trainability for other QNN architectures, such as quantum convolutional neural networks [65], tree-based architectures [66], and others [67–69]. We also note that our results can also be interpreted as a type of entanglement-induced barren plateau. Here, a large amount of entanglement in a parameterized quantum circuit can lead to trainability issues when qubits are discarded, and the output qubits are concentrated around the maximally mixed state. This phenomenon was further studied in [52, 70].

[1] Simon Haykin, *Neural networks: a comprehensive foundation* (Prentice Hall PTR, 1994).

[2] Marvin Minsky and Seymour A Papert, *Perceptrons: An introduction to computational geometry* (MIT press, 2017).

[3] Frank Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para* (Cornell Aeronautical Laboratory, 1957).

[4] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams, "Learning representations by back-propagating

errors," nature **323**, 533–536 (1986).

[5] J. Preskill, "Quantum computing in the NISQ era and beyond," Quantum **2**, 79 (2018).

[6] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione, "The quest for a quantum neural network," Quantum Information Processing **13**, 2567–2586 (2014).

[7] Michael A Nielsen, *Neural networks and deep learning*, Vol. 2018 (Determination press San Francisco, CA, USA:, 2015).

[8] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd, "Quantum machine learning," Nature **549**, 195–202 (2017).

[9] Kunal Sharma, M Cerezo, Zoë Holmes, Lukasz Cincio, Andrew Sornborger, and Patrick J Coles, "Reformulation of the no-free-lunch theorem for entangled data sets," arXiv preprint arXiv:2007.04900 (2020).

[10] J. Romero, J. P. Olson, and A. Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," Quantum Science and Technology **2**, 045001 (2017).

[11] Vedran Dunjko and Hans J Briegel, "Machine learning & artificial intelligence in the quantum domain: a review of recent progress," Reports on Progress in Physics **81**, 074001 (2018).

[12] Guillaume Verdon, Jason Pye, and Michael Broughton, "A universal training algorithm for quantum deep learning," arXiv preprint arXiv:1806.09729 (2018).

[13] Edward Farhi and Hartmut Neven, "Classification with quantum neural networks on near term processors," arXiv preprint arXiv:1802.06002 (2018).

[14] Carlo Ciliberto, Mark Herbster, Alessandro Davide Ialongo, Massimiliano Pontil, Andrea Rocchetto, Simone Severini, and Leonard Wossnig, "Quantum machine learning: a classical perspective," Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **474**, 20170551 (2018).

[15] Nathan Killoran, Thomas R Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd, "Continuous-variable quantum neural networks," Physical Review Research **1**, 033063 (2019).

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

[17] Zhih-Ahn Jia, Biao Yi, Rui Zhai, Yu-Chun Wu, Guang-Can Guo, and Guo-Ping Guo, "Quantum neural network states: A brief review of methods and applications," Advanced Quantum Technologies **2**, 1800077 (2019).

[18] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven, "Barren plateaus in quantum neural network training landscapes," Nature communications **9**, 4812 (2018).

[19] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," Nature communications **12**, 1–12 (2021).

[20] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," Nature Communications **5**, 4213 (2014).

[21] Bela Bauer, Dave Wecker, Andrew J Millis, Matthew B Hastings, and Matthias Troyer, "Hybrid quantum-classical approach to correlated materials," Physical Review X **6**, 031045 (2016).

[22] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," New Journal of Physics **18**, 023023 (2016).

[23] A. Arrasmith, L. Cincio, A. T. Sornborger, W. H. Zurek, and P. J. Coles, "Variational consistent histories as a hybrid algorithm for quantum foundations," Nature communications **10**, 3438 (2019).

[24] Tyson Jones, Suguru Endo, Sam McArdle, Xiao Yuan, and Simon C Benjamin, "Variational quantum algorithms for discovering hamiltonian spectra," Physical Review A **99**, 062304 (2019).

[25] X. Xu, J. Sun, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, "Variational algorithms for linear algebra," arXiv:1909.03898 [quant-ph].

[26] Carlos Bravo-Prieto, Ryan LaRose, M. Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J. Coles, "Variational quantum linear solver: A hybrid algorithm for linear systems," arXiv:1909.05820 (2019).

[27] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C Benjamin, "Theory of variational quantum simulation," Quantum **3**, 191 (2019).

[28] Cristina Cirstoiu, Zoe Holmes, Joseph Iosue, Lukasz Cincio, Patrick J Coles, and Andrew Sornborger, "Variational fast forwarding for quantum simulation beyond the coherence time," arXiv preprint arXiv:1910.04292 (2019).

[29] Marco Cerezo, Alexander Poremba, Lukasz Cincio, and Patrick J Coles, "Variational quantum fidelity estimation," Quantum **4**, 248 (2020).

[30] M Cerezo, Kunal Sharma, Andrew Arrasmith, and Patrick J Coles, "Variational quantum state eigensolver," arXiv preprint arXiv:2004.01372 (2020).

[31] Noriaki Kouda, Nobuyuki Matsui, Haruhiko Nishimura, and Ferdinand Peper, "Qubit neural network and its learning efficiency," Neural Computing & Applications **14**, 114–121 (2005).

[32] MV Altaisky, "Quantum neural network," arXiv preprint quant-ph/0107012 (2001).

[33] Alaa Sagheer and Mohammed Zidan, "Autonomous quantum perceptron neural network," arXiv preprint arXiv:1312.4149 (2013).

[34] Michael Siomau, "A quantum model for autonomous learning automata," Quantum information processing **13**, 1211–1221 (2014).

[35] Erik Torrontegui and Juan José García-Ripoll, "Unitary quantum perceptron as efficient universal approximator," EPL (Europhysics Letters) **125**, 30004 (2019).

[36] Francesco Tacchino, Chiara Macchiavello, Dario Gerace, and Daniele Bajoni, "An artificial neuron implemented on an actual quantum processor," npj Quantum Information **5**, 1–8 (2019).

[37] Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J Osborne, Robert Salzmann, Daniel Scheiermann, and Ramona Wolf, "Training deep quantum neural networks," Nature Communications **11**, 1–6 (2020).

[38] Dmytro Bondarenko and Polina Feldmann, "Quantum autoencoders to denoise quantum data," Physical Review Letters **124**, 130502 (2020).

[39] Kyle Poland, Kerstin Beer, and Tobias J Osborne, "No free lunch for quantum machine learning," arXiv preprint arXiv:2003.14103 (2020).

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

[41] Kunal Sharma, Sumeet Khatri, Marco Cerezo, and Patrick J Coles, "Noise resilience of variational quantum compiling," New Journal of Physics **22**, 043006 (2020).

[42] Hsin-Yuan Huang, Richard Kueng, and John Preskill, "Predicting many properties of a quantum system from very few measurements," Nature Physics **16**, 1050–1057 (2020).

[43] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, "Quantum-assisted quantum compiling," Quantum **3**, 140 (2019).

[44] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao, "The expressive power of parameterized quantum circuits," arXiv preprint arXiv:1810.11922 (2018).

[45] Gian Giacomo Guerreschi and Mikhail Smelyanskiy, "Practical optimization for hybrid quantum-classical algorithms," arXiv preprint arXiv:1701.01450 (2017).

[46] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," Phys. Rev. A **98**, 032309 (2018).

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

[48] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine, "Exact and approximate unitary 2-designs and their application to fidelity estimation," Physical Review A **80**, 012304 (2009).

[49] Fernando GSL Brandao, Aram W Harrow, and Michał Horodecki, "Local random quantum circuits are approximate polynomial-designs," Communications in Mathematical Physics **346**, 397–434 (2016).

[50] Aram Harrow and Saeed Mehraban, "Approximate unitary t-designs by short random quantum circuits using nearest-neighbor and long-range gates," arXiv preprint arXiv:1809.06957 (2018).

[51] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," Nature **549**, 242 (2017).

[52] Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe, "Entanglement induced barren plateaus," arXiv preprint arXiv:2010.15968 (2020).

[53] Neena Aloysius and M Geetha, "A review on deep convolutional neural networks," in *2017 International Conference on Communication and Signal Processing (ICCSP)* (IEEE, 2017) pp. 0588–0592.

[54] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo, "Quantum natural gradient," arXiv preprint arXiv:1909.02108 (2019).

[55] Jonas M Kübler, Andrew Arrasmith, Lukasz Cincio, and Patrick J Coles, "An adaptive optimizer for measurement-frugal variational algorithms," arXiv preprint arXiv:1909.09083 (2019).

[56] Bálint Koczor and Simon C Benjamin, "Quantum natural gradient generalised to non-unitary circuits," arXiv preprint arXiv:1912.08660 (2019).

[57] Andrew Arrasmith, Lukasz Cincio, Rolando D Somma, and Patrick J Coles, "Operator sampling for shot-frugal optimization in variational algorithms," arXiv preprint arXiv:2004.06252 (2020).

[58] Ryan LaRose, Arkin Tikku, Étude O'Neel-Judy, Lukasz Cincio, and Patrick J Coles, "Variational quantum state diagonalization," npj Quantum Information **5**, 1–10 (2019).

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

[60] Guillaume Verdon, Michael Broughton, Jarrod R McClean, Kevin J Sung, Ryan Babbush, Zhang Jiang, Hartmut Neven, and Masoud Mohseni, "Learning to learn with quantum neural networks via classical neural networks," arXiv preprint arXiv:1907.05415 (2019).

[61] Tyler Volkoff and Patrick J Coles, "Large gradients via correlation in random parameterized quantum circuits," arXiv preprint arXiv:arXiv:2005.12200 (2020).

[62] Adrien Bolens and Markus Heyl, "Reinforcement learning for digital quantum simulation," Phys. Rev. Lett. **127**, 110502 (2021).

[63] Zbigniew Puchała and Jaroslaw Adam Miszczak, "Symbolic integration with respect to the haar measure on the unitary groups," Bulletin of the Polish Academy of Sciences Technical Sciences **65**, 21–27 (2017).

[64] Motohisa Fukuda, Robert König, and Ion Nechita, "RTNI—a symbolic integrator for haar-random tensor networks," Journal of Physics A: Mathematical and Theoretical **52**, 425303 (2019).

[65] Arthur Pesah, M Cerezo, Samson Wang, Tyler Volkoff, Andrew T Sornborger, and Patrick J Coles, "Absence of barren plateaus in quantum convolutional neural networks," Accepted in PRX, In press. (2020).

[66] Kaining Zhang, Min-Hsiu Hsieh, Liu Liu, and Dacheng Tao, "Toward trainability of quantum neural networks," arXiv preprint arXiv:2011.06258 (2020).

[67] Chen Zhao and Xiao-Shan Gao, "Analyzing the barren plateau phenomenon in training quantum neural networks with the ZX-calculus," Quantum **5**, 466 (2021).

[68] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles, "Noise-induced barren plateaus in variational quantum algorithms," Accepted in Nature Communications, In press (2020), arXiv preprint arXiv:2007.14384.

[69] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner, "The power of quantum neural networks," Nature Computational Science **1**, 403–409 (2021).

[70] Taylor L Patti, Khadijeh Najafi, Xun Gao, and Susanne F Yelin, "Entanglement devised barren plateau mitigation," Physical Review Research **3**, 033090 (2021).