Learnability of Quantum Neural Networks

Yuxuan Du,[1] Min-Hsiu Hsieh,[2,*] Tongliang Liu,[1] Shan You,[3] and Dacheng Tao[1,†]

[1] *School of Computer Science, Faculty of Engineering, The University of Sydney, Darlington, New South Wales 2008, Australia*

[2] *Hon Hai Quantum Computing Research Center, Taipei 114, Taiwan*

[3] *SenseTime Research, Beijing, China*

(Received 27 March 2021; revised 8 September 2021; accepted 20 October 2021; published 17 November 2021)

Quantum neural network (QNN), or equivalently, the parameterized quantum circuit (PQC) with a gradient-based classical optimizer, has been broadly applied to many experimental proposals for noisy intermediate-scale quantum (NISQ) devices. However, the learning capability of QNN remains largely unknown due to the nonconvex optimization landscape, the measurement error, and the unavoidable gate noise introduced by NISQ machines. In this study, we theoretically explore the learnability of QNN in the view of the trainability and generalization. Particularly, we derive the convergence performance of QNN under the NISQ setting, and identify classes of computationally hard concepts that can be efficiently learned by QNN. Our results demonstrate that large gate noise, few quantum measurements, and deep circuit depth will lead to poor convergence rates of QNN towards the empirical risk minimization. Moreover, we prove that any concept class, which is efficiently learnable by a quantum statistical query (QSQ) learning model, can also be efficiently learned by PQCs. Since the QSQ learning model can tackle certain problems such as parity learning with a runtime speedup, our result suggests that PQCs established on NISQ devices will retain the quantum advantage measured by generalization ability. Our work provides theoretical guidance for developing advanced QNNs and opens up avenues for exploring quantum advantages beyond hybrid quantum-classical learning protocols in the NISQ era.

DOI: 10.1103/PRXQuantum.2.040337

I. INTRODUCTION

Deep neural network (DNN) has substantially impacted the field of artificial intelligence in the past decade [1] because numerous real-world applications, such as object detection [2], question answering [3], and social recommendation [4], could be accomplished by DNN-based learning algorithms with state-of-the-art performance. The success of DNN is mainly attributed to its versatile architecture, which is best understood by the following multilayer scheme. As shown in Fig. 1(a), the inputs are processed through the feature embedding layers $\mathcal{F}_{\mathbf{x}}(\cdot)$, followed by the fully connected layers $\prod_{\ell} W_{\ell}(\cdot)$, where the choice of each layer and the combination rule can be tailormade for various learning tasks. Training DNN is a process to uncover the intrinsic relation between the input and the output of the given dataset. However, theoretical results to explain how DNN discovers such a relation are largely unknown, hurdled by its flexible architectures and the nonconvex optimization landscape. To this end, a huge amount of effort has been dedicated to understanding the *learnability* of DNN. Concretely, based on the formula "*learnability* = *trainability* + *generalization*" [5], there are two pipelines to explore the learnability of DNN. For the trainability, several studies [6–9] illustrated that DNN with specific structures can converge to the global minima of the training objective function in polynomial time. The generalization concerns whether DNN can effectively output a hypothesis that well approximates the target concept for a certain learning problem. For instance, Ref. [5] proved that overparameterized DNN can learn important concept classes, including the two- and three-layer DNN with fewer parameters, in polynomial samples; while Ref. [10] proved that two-layer DNN can effectively learn polynomial functions.

Quantum machine learning has emerged as a central application of quantum computing [11]. With the aim of solving real-world problems beyond the reach of classical computers, firm and steady progress has been developed during the past decade [12–14]. In addition, a

*min-hsiu.hsieh@foxconn.com

†dacheng.tao@sydney.edu.au

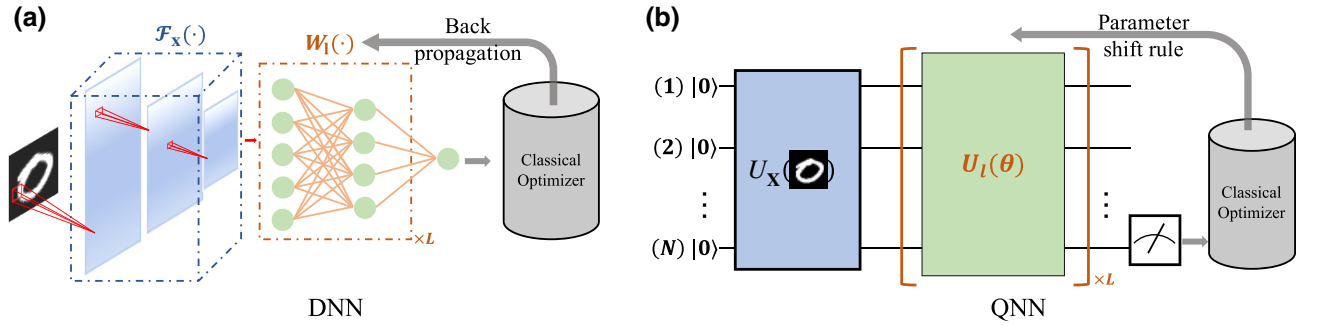

FIG. 1. Illustration of DNN and QNN. The left and right panel shows a DNN and QNN, respectively. For DNN, the feature embedding layers $\mathcal{F}_{\mathbf{x}}(\cdot)$, which contain a sequence of operations with the arbitrary combination such as convolution and attention, maps the input "0" to the feature space. $W_l(\cdot)$ is the lth fully connected layer. For QNN, an encoding quantum circuit $U_{\mathbf{x}}$ maps the classical input "0" to the quantum feature space. $U_l(\boldsymbol{\theta})$ is the lth trainable quantum circuit layer. Classical information for optimization is extracted by quantum measurements.

quantum extension of DNN, i.e., the quantum neural network (QNN), which is separately proposed in Refs. [15–20], received great attention due to the huge success of DNN and the superior computational power of quantum devices [21]. As shown in Fig. 1(b), QNN also adopts the multilayer architecture: the inputs were converted into quantum states by the encoding quantum circuit $U_{\mathbf{x}}$, followed by the trainable quantum circuits $U(\boldsymbol{\theta}) = \prod_{l=1}^{L} U_l(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are adjustable parameters of quantum gates, and a classical optimizer. There is a close correspondence between DNN and QNN: the feature embedding layer "$\mathcal{F}_{\mathbf{x}}$" of DNN corresponds to the encoding quantum circuit $U_{\mathbf{x}}$ of QNN, while the fully connected layer $W_l(\cdot)$ of DNN coincides with the trainable quantum circuit $U_l(\boldsymbol{\theta})$ of QNN. Despite the high similarity, there are two key features separating DNNs with QNNs. First, QNNs generally lack nonlinear activation functions, caused by the linearity of quantum mechanics. Second, $U(\boldsymbol{\theta})$ generally possesses a strong expressive power to prepare classical distributions [22,23]. Noticeably, the latter property enables the advance of QNNs for a wide range of machine-learning problems over their classical counterparts.

In parallel to empirically evaluate performance of QNNs on different learning tasks such as classification [17,19,24] and regression [18,25], there is a growing interest to understand the learning capabilities of QNNs, i.e., their trainability and generalization. Recent studies have partially investigated these two issues from different angles such as optimization [26,27], expressivity [28–30], and generalization [31–38]. Nevertheless, to date, many fundamental theoretical results of QNNs still remain largely unknown. Firstly, a rigorous analysis of the learning performance of QNNs is lacking. The obstruction that impedes the theoretic progress is due to a combination of the following factors: the versatile structures of QNN, the nonconvex optimization landscapes, the unavoidable gate noise, and measurement errors. Classically, the empirical risk-minimization (ERM) principle [39,40] is a learning paradigm that has been broadly employed to benchmark the training performance of the supervised learning algorithms without prior knowledge of the data distributions. To be more specific, ERM measures how fast the objective function used in the learning algorithm converges to the stationary point in terms of the input size and feature dimensions. Following the same routine, it is natural to ask what is the convergence rate of QNN towards the empirical risk minimizer with a specified optimization rule? Answering this question not only enables the theoretical evaluation of the performance of various QNN-based supervised learning algorithms, but more importantly, it also provides guidelines to the design of better quantum supervised learning protocols. Particularly, we believe that the achieved convergence rates can guide us to devise more advanced quantum learning protocols to avoid the barren-plateau (i.e., the vanishing gradients) phenomenon in training QNNs [41]. More discussion comes after we formally introduce Theorem 1.

Secondly, understanding the generalization of QNNs can facilitate the exploration of its applicability with provable advantages. Specifically, generalization concerns whether the learning model can efficiently output (i.e., using a polynomial sample or query complexity) a hypothesis that can well approximate a target concept under a specific learning paradigm. Despite the significance, theoretical analysis of the generalization property of QNN remains largely open. That is, the existing literature mainly focuses on studying the generalization of QNNs under the quantum probably approximate correct (QPAC) learning paradigm [42–45]. However, it remains inconclusive whether QNN possesses any theoretical advantages over classical learning models under other learning paradigms such as noisy QPAC learning [46–49], quantum statistical query learning (QSQ) [50], and quantum differentially private learning [51,52]. A key reason for exploring the

generalization of QNNs under different learning paradigms is that different from QPAC learning, noisy QPAC and QSQ learning can be applied to analyze the generalization of QNNs when the gate noise and measurement error are considered (see Sec. IV for elaborations). In this study, we delve into investigating the generalization of QNNs under the QSQ learning paradigm. In particular, we aim to answer whether there exists any class of concepts that can be efficiently learned by (noisy) QNNs but are computationally hard for the classical learning models in the regime of the statistical query learning. If the answer is affirmative, it enables us to employ QNN implemented on NISQ devices to accomplish certain tasks with theoretical advantages.

The outline of this study is as follows. In Sec. II, we theoretically explore the trainability of QNNs through the lens of ERM. Then, in Sec. III, we conduct numerical simulations to validate the achieved theoretical results. Subsequently, we investigate the generalization ability of QNNs under the statistical query learning protocol in Sec. IV. We conclude this study in Sec. V. All proof details are deferred to the Appendix.

II. TRAINABILITY OF QNN TOWARDS ERM

Before elaborating our theoretical results, we first formulate ERM and the mechanism of QNN. Let $\mathbf{z} = \{\mathbf{z}_j\}_{j=1}^n \in \mathcal{Z}$ be the given dataset with $\mathcal{Z}$ being the sample domain, where the jth sample $\mathbf{z}_j = (\mathbf{x}_j, y_j)$ includes a feature vector $\mathbf{x}_j \in \mathbb{R}^{D_c}$ and a label $y_j \in \mathbb{R}$. ERM aims to find the optimal $\boldsymbol{\theta}^* \in \mathbb{R}^d$ by minimizing the objective function $\mathcal{L}$ within the constraint set $\mathcal{C} \subseteq \mathbb{R}^d$, i.e.,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathcal{C}} \mathcal{L}(\boldsymbol{\theta}, \mathbf{z}) := \frac{1}{2n}\sum_{j=1}^{n} \ell(y_j, \hat{y}_j) + r(\boldsymbol{\theta}), \quad (1)$$

where $\hat{y}_j$ is the predicted label that is determined by $\boldsymbol{\theta}$ and $\mathbf{x}_j$, ℓ is the loss function that measures the disparity between true labels $\{y_j\}_{j=1}^n$ and the predicted labels $\{\hat{y}_j\}_{j=1}^n$, and $r(\cdot)$ is a regularizer. To ease the discussion, throughout the paper, we consider the mean square error loss ℓ with $\ell(y_j, \hat{y}_i) = (\hat{y}_j - y_j)^2$, and use $r(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_2^2/2$ with $\lambda \geq 0$. Note that our analysis can be easily generalized to other loss functions that satisfy S-smooth and G-Lipschitz properties [53].

The common optimization rule to tackle ERM is the batch gradient-descent method [1]. Depending on the available resources, the sample indices are divided into B disjoint batches $\{\mathcal{B}_i\}_{i=1}^B$ with equal size B_s, namely, $\mathbf{z} = \cup_{j\in\{\mathcal{B}_i\}_{i=1}^B} \mathbf{z}_j$. The optimization rule at the tth iteration is $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - (\eta/B)\sum_{i=1}^B \nabla\mathcal{L}(\boldsymbol{\theta}^{(t)}, \mathcal{B}_i)$, where η is the learning rate, the gradient $\nabla\mathcal{L}(\cdot)$ is

$$\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)}, \mathcal{B}_i) = \left(\hat{Y}_i^{(t)} - Y_i\right)\frac{\partial \hat{Y}_i^{(t)}}{\partial \boldsymbol{\theta}^{(t)}} + \lambda\boldsymbol{\theta}^{(t)}, \quad (2)$$

$Y_i = (1/B_s)\sum_{j\in\mathcal{B}_i} y_j$ and $\hat{Y}_i^{(t)} = (1/B_s)\sum_{j\in\mathcal{B}_i} \hat{y}_j^{(t)}$ are the sum average of the true labels and the predicted labels for the ith batch $\mathcal{B}_i$, respectively. When no confusion will occur, we use $\mathcal{L}(\boldsymbol{\theta}^{(t)})$ and $\mathcal{L}_i(\boldsymbol{\theta}^{(t)})$ instead of $\mathcal{L}(\boldsymbol{\theta}^{(t)}, \mathbf{z})$ and $\mathcal{L}(\boldsymbol{\theta}^{(t)}, \mathcal{B}_i)$ in the rest of study.

The general workflow of QNN is summarized in Fig. 1(b). Specifically, QNN first employs a state preparation unitary $U_{\mathbf{x}}$ to encode classical inputs $\{\mathbf{x}_j | j \in \mathcal{B}_i\}$ into quantum states, followed by the quantum circuit $U(\boldsymbol{\theta})$ with tunable parameter $\boldsymbol{\theta}$ to produce the state $\gamma_{\mathcal{B}_i} \in \mathbb{C}^{D\times D}$. Note that some quantum kernel encoding methods may lead to the varied feature dimensions, i.e., $D_c \neq D$. We refer the interested reader to Appendix C for implementation details of $U_{\mathbf{x}}$ and $U(\boldsymbol{\theta})$. Finally, a quantum measurement, e.g., a two-outcome positive operator-valued measure (POVM) $\{\Pi, I - \Pi\}$, is applied to the state $\gamma_{\mathcal{B}_i}$ and produces the outcome V_i that can be viewed as a binary random variable with the Bernoulli distribution $\mathrm{Ber}(\hat{Y}_i)$, where $\hat{Y}_i := \mathrm{Tr}(\Pi\gamma_{\mathcal{B}_i})$. Note that, for a random variable X that follows the Bernoulli distribution with $X \sim \mathrm{Ber}(p)$, we have $\Pr(X = 1) = p$ and $\Pr(X = 0) = 1 - p$. Denote the obtained statistics, i.e., the sample mean, by $\bar{Y}_i = (1/K)\sum_{k=1}^K V_k$ after repeating the above procedure K times. The law of Born rule ensures $\bar{Y}_i \to \hat{Y}_i$ when $K \to \infty$. However, in reality, only a finite number of measurements is allowed, and this results in the sample error (measurement error).

In addition, the quantum gates in NISQ chips, which are used to implement $U_{\mathbf{x}}$ and $U(\boldsymbol{\theta})$, are prone to having errors [54]. The gate noise can be simulated by applying certain quantum channels to each circuit layer, and this can be done by considering the worst-case scenario, i.e., modeling the gate noise at each circuit depth by a quantum depolarization channel [55]. Specifically, given a quantum state $\rho \in \mathbb{C}^{D\times D}$, the depolarization channel $\mathcal{N}_p$ acts on a D-dimensional Hilbert space follows $\mathcal{N}_p(\rho) = (1-p)\rho + p\mathbb{I}/D$, where $\mathbb{I}/D$ is the maximally mixed state [55]. Throughout the whole study, we consider the case that applying $\mathcal{N}_p$ after each circuit depth of QNN, the quantum state before measurement is denoted by $\tilde{\gamma}_{\mathcal{B}_i}$. When the measurement is applied to $\tilde{\gamma}_{\mathcal{B}_i}$, the obtained outcome V_i follows the Bernoulli distribution $\mathrm{Ber}(\tilde{Y}_i)$ with $\tilde{Y}_i := \mathrm{Tr}(\Pi\tilde{\gamma}_{\mathcal{B}_i})$ instead of $\mathrm{Ber}(\hat{Y}_i)$. We remark that while all results presented in the main text assuming the depolarization noise, they can be easily extended to a more general noisy channel (see Appendix I for details).

The optimization of QNN towards ERM is similar to that of DNN. In particular, QNN also generates a sum average of the predicted labels, based on $\boldsymbol{\theta}$ and $\mathcal{B}_i$, after the measurement component in Fig. 1(b). However, the main difference between the gradient-based optimization of QNN and DNN is as follows. In DNN, the gradient in Eq. (2) can be easily obtained via backpropagation [1]. However, due to the nature of quantum mechanics,

the gradient of a quantum unitary operator [e.g., trainable quantum circuit layer $U_l(\boldsymbol{\theta})$] is, in general, not a legitimate quantum operator anymore [56]. To overcome this shortcoming, the *parameter shift rule* [18,56] is proposed to estimate the gradients of a quantum unitary operator using K measurements (see Appendix B for details).

Now we quantify the convergence of QNN towards the empirical risk minimizer under the batch gradient-descent optimization rule in Eq. (2). Particularly, analyzing the convergence of QNN amounts to checking the following two standard utility metrics:

$$\begin{aligned} R_1\left(\boldsymbol{\theta}^{(T)}\right) &:= \mathbb{E}\left[\left\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(T)})\right\|^2\right], \\ R_2\left(\boldsymbol{\theta}^{(T)}\right) &:= \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}^{(T)})] - \mathcal{L}(\boldsymbol{\theta}^*), \end{aligned} \tag{3}$$

where the expectation is taken over the randomness of QNN resulted from the measurement error and gate noise, $\boldsymbol{\theta}^{(T)}$ is the output of QNN after T iterations and $\nabla\mathcal{L}(\cdot)$ denotes the gradient of the objective function $\mathcal{L}(\cdot)$ defined in Eq. (1), and $\boldsymbol{\theta}^*$ is the optimal parameters in Eq. (1) without noise. The metric R_1 evaluates how far QNN is away from the stationary point, $\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(T)}, \mathbf{z})\|^2 = 0$, in expectation [57,58]. The utility metric R_2 evaluates the expected excess empirical risk [59,60].

The utility bounds of noisy QNN are summarized in the following theorem.

Theorem 1: *Let K be the number of measurements per iteration, L_Q be the circuit depth, p be the gate noise, and B be the batch size. QNN outputs $\boldsymbol{\theta}^{(T)} \in \mathbb{R}^d$ after T iterations with the utility bound*

$$R_1 \leq \tilde{O}\left[\text{poly}\left(\frac{d}{T(1-p)^{L_Q}}, \frac{d}{BK(1-p)^{L_Q}}, \frac{d}{(1-p)^{L_Q}}\right)\right].$$

When λ satisfies a technical assumption such that $\lambda \in [0, (1/3\pi)] \cup [(1/\pi), \infty]$, QNN outputs $\boldsymbol{\theta}^{(T)} \in (\pi, 3\pi]^d$ after $T = \tilde{O}[d^3/(1-p)^{L_Q}]$ with the utility bound

$$R_2 \leq \tilde{O}\left[\text{poly}\left(\frac{d}{K^2B(1-p)^{L_Q}} + \frac{d}{(1-p)^{L_Q}}\right)\right].$$

Proof sketch. Here we present the proof sketch of Theorem 1. Refer to Appendix F for the full proof details.

The proof of deriving the upper bound of R_1 is established on a well-known result in optimization theory [61], i.e., when a function satisfies the smooth property, its stationary point can be efficiently located by a simple gradient-based algorithm. To this end, we first prove that the loss function $\mathcal{L}(\boldsymbol{\theta}, \mathbf{z})$ in Eq. (1) is S smooth with $\nabla^2\mathcal{L}(\boldsymbol{\theta}, \mathbf{z}) \preceq S\mathbb{I}$ with $S > 0$ and $\forall\boldsymbol{\theta} \in \mathcal{C}$. Then, supported by an alternative representation of S smooth, we establish a relationship between the loss difference and the gradients at the tth iteration, i.e., for $\forall t \in [T]$

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)}) \\ &\quad\leq \langle\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)}), \boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\rangle + \frac{S}{2}\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\|^2. \end{aligned} \tag{4}$$

Note that in the NISQ scenario, the term $\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}$ is equal to the estimated gradients $-(\eta/B)\sum_{i=1}^{B}\nabla\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)}, \mathcal{B}_i)$ instead of the analytic gradient $-(\eta/B)\sum_{i=1}^{B}\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)}, \mathcal{B}_i)$ as defined in Eq. (2), where the error is caused by the inevitable gate noise and sample error. Our *key technical contribution* is deriving the relation between the analytic and estimated gradients, i.e., the gradient for the jth parameter with $j \in [d]$ follows

$$\nabla_j\bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (1-\tilde{p})^2\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + \boldsymbol{\varsigma}_i^{(t,j)}, \tag{5}$$

where $\tilde{p} = 1 - (1-p)^{L_Q}$, L_Q is the circuit depth, the constant $C_{j,1}^{(i,t)}$ depends only on Y_i, $\boldsymbol{\theta}^{(t)}$, and $\tilde{p}$, and $\boldsymbol{\varsigma}_i^{(t,j)}$ follows the distribution $\mathcal{P}_Q$ that is formed by Y_i, $\boldsymbol{\theta}^{(t)}$, the number of measurements K, and $\tilde{p}$ with zero mean.

In conjunction with Eqs. (4) and (5), we obtain the relation between the loss discrepancy and the analytic gradeint when the classical optimizer can only access the estimated gradients, i.e.,

$$\begin{aligned} &\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)}) \\ &\quad\leq -\frac{1}{S}\sum_{j=1}^{d}\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})[(1-\tilde{p})^2\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + \boldsymbol{\varsigma}_i^{(t,j)}] \\ &\qquad+ \frac{1}{2S}\sum_{j=1}^{d}[(1-\tilde{p})^2\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + \boldsymbol{\varsigma}_i^{(t,j)}]^2. \end{aligned} \tag{6}$$

After taking expectation over the randomness and simplification, we can obtain the norm of gradients at the tth iteration is upper bounded by the loss difference and other factors, i.e., $\|\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 \leq 2S[\mathcal{L}(\boldsymbol{\theta}^{(t)}) - \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})]/(1-\tilde{p})^2] + [(2\tilde{p} - \tilde{p}^2)(2G + d)(1+10\lambda)^2/(1-\tilde{p})^2] + [(6dK + 8d)/(1-\tilde{p})^2BK^2]$. By induction, with summing over $t = 0, \ldots, T-1$, we achieve the upper bound of R_1.

The proof of deriving the upper bound of R_2 utilizes the Polyak-Lojasiewicz (PL) condition to connect stationary points with the global minimum. Mathematically, a function f satisfies the PL condition if there exists $\mu > 0$ and for every possible $\boldsymbol{\theta} \in \mathcal{C}$, $\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 \geq 2\mu(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*)$, where $\mathcal{L}^* = \min_{\boldsymbol{\theta}\in\mathcal{C}}\mathcal{L}(\boldsymbol{\theta})$. Intuitively, if a nonconvex function satisfies the PL condition [62], every stationary point of such a function is the global minimum [62,63]. To this end, we prove that when the hyperparameter λ satisfies a mild technical assumption with $\lambda \in [0, (1/3\pi)] \cup$

$[(1/\pi),\infty]$, the loss function $\mathcal{L}(\boldsymbol{\theta},\mathbf{z})$ in Eq. (1) obeys the PL condition. Combining the PL condition and the results achieved in analyzing R_1, we have

$$\begin{aligned}&\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})-\mathcal{L}(\boldsymbol{\theta}^{(t)})]\\&\quad\le-\frac{1}{2S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2+g\\&\quad\le-\frac{\mu(1-\tilde{p})^2}{S}[\mathcal{L}(\boldsymbol{\theta}^{(t)})-\mathcal{L}^*]+g,\end{aligned}\tag{7}$$

where $g=[(2G+d)/2S](2-\tilde{p})\tilde{p}(1+10\lambda)^2+[(6dK+8d)/2SBK^2]$. By induction, we can effectively achieve the upper bound of R_2. ■

We emphasize two key points in our proof. First, the consequence of the derived bounded discrepancy between the estimated and analytic gradient of QNN is one of our *key technical contributions*, which may be of independent interest to understand the trainability of other variational quantum algorithms. Namely, the collected gradient information of QNN is generally biased. This property *separates the convergence analysis of QNNs with DNNs*, where the randomness assigned to classical DNNs is unbiased to ensure good convergence. This biased gradient information requests us to adopt different relaxation techniques to analyze the upper bound of R_1 and R_2. Second, the introduced technical assumption of λ is reasonable. Due to the hardness of finding the global optima $\mathcal{L}(\boldsymbol{\theta}^*)$ in the nonconvex landscape, R_2 can only be applied to some special nonconvex objective functions. Compared with other conditions such as the strong convexity and the restricted strong convexity to achieve the linear convergence towards the global minimum, the PL condition is much easier to satisfy by QNN with $\lambda\in[0,(1/3\pi)]\cup[(1/\pi),\infty]$.

The result achieved in Theorem 1 shows that a larger amount of measurements K, a larger batch size B, a smaller depolarizing error p, a smaller parameter space d, and a shallower quantum circuit depth L_Q, can yield better utility bounds R_1 and R_2. In addition, the achieved utility bound R_1 explains how the unavoidable gate noise affects the convergence behavior of QNN. Specifically, no matter how large T or K would become, QNN could still diverge for large d, p, and L_Q because of the term $d/(1-p)^{L_Q}$ in both R_1 and R_2. Notably, unlike other factors, the circuit depth L_Q associated with the system noise p exponentially scales the utility bounds R_1 and R_2. Such a dependence suggests that to enhance the trainability of QNNs and seek a near-optimal solution on NISQ devices, it is significant to control the circuit depth and suppress the system noise p. The above observation coincides with the classical ERM results, where a sufficiently large perturbation noise imposed on the gradient may result in the optimization of ERM to diverge [53]. Moreover, the dependence of gate and measurement noise in R_1 and R_2 accords with the empirical observations [64] that certain quantum learning models, which achieve the promising performances under the ideal setting, may not be applicable to experiments. For example, when the quantum approximate optimization algorithm [20] is applied to accomplish maximum cut problem on three-regular graphs, the success probability drops to zero once the gate error level is larger than 0.1.

The convergence towards the global optima as shown in R_2, which evaluates the convergence rate of QNN to the global minima, implies that regularization techniques may contribute to avoiding the barren plateau encountered in training QNN [41]. The barren-plateau phenomenon claimed that the optimization may be terminated at a point that is far away from the global minimum, since the gradient will be exponentially vanished with respect to the increased number of qubits and the circuit depth. By contrast, R_2 shows that when λ in Eq. (1) is sufficient large, with improving the number of measurements K, the optimized result of QNN will converge to the global optima once the gate noise is not too large. Hence, the regularization techniques allow the optimization of QNN to be released from the barren-plateau dilemma. Remarkably, the prior literature delves into developing effective methods to alleviate barren plateaus for all variational quantum algorithms, including adopting local measurements [26,65], operating trainable parameters by involving correlation or specific optimization strategy [66–68], and controlling entanglement such as the number of two-qubit quantum gates [69–72]. Unlike these general methods, regularization techniques, which modify only the loss function instead of variational quantum circuits, are another efficient approach to alleviate barren plateaus when focusing on conventional machine-learning tasks, where near-optimal results are sufficient to attain good performance [73]. To this end, the introduced regularization term aims to slightly reshape the loss landscape to facilitate optimization. In other words, the setup of this work excludes all other known explanations of the absence of barren plateaus. Our results provide a positive response towards the conjecture raised by Refs. [41,67], where both of them speculate that regularization techniques may eliminate barren plateaus but lack theoretical evidence.

Remark: We note that the achieved utility bounds R_1 and R_2 are very general, and cover various types of encoding quantum circuits $U_{\mathbf{x}}$ and trainable quantum circuits $U(\boldsymbol{\theta})$. Specifically, our results cover all typical encoding circuits, e.g., amplitude encoding [74–76], kernel mapping [17–19], the dimension reduction methods [77], basis encoding methods [16], and diverse architectures of the trainable quantum circuits, as long as it is composed of the parameterized single-qubit gates and two-qubit gates [78]. Theorem 1 provides theoretical guidances to design QNN-based learning algorithms on NISQ devices, considering

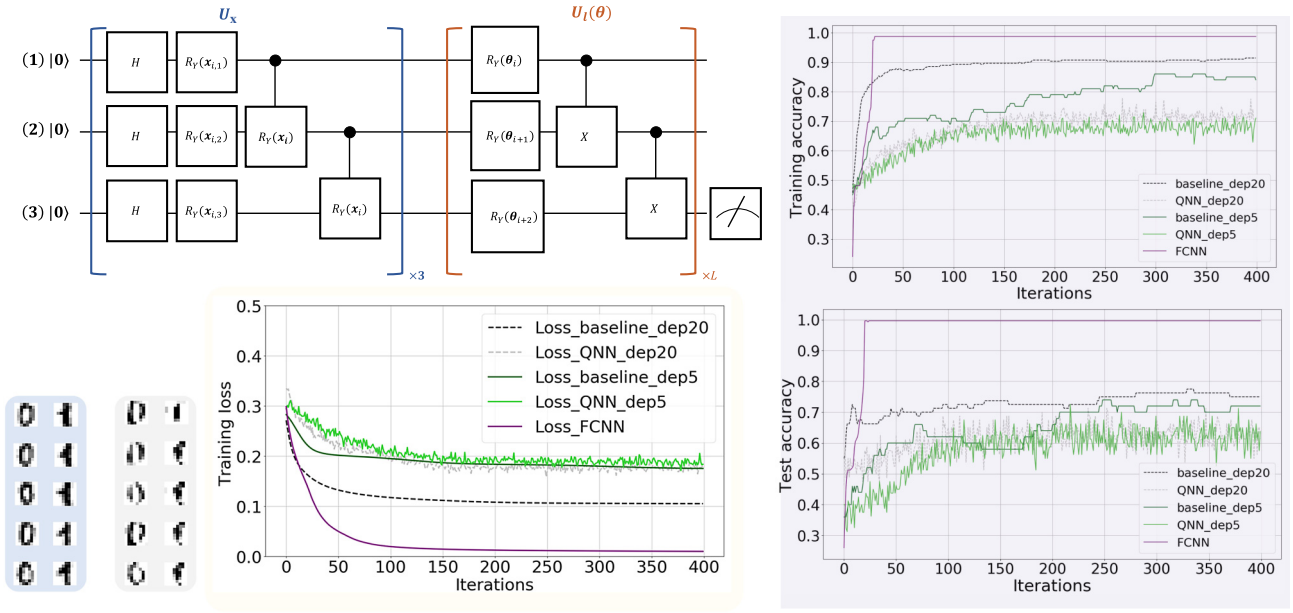

FIG. 2. The simulation results of using QNN to learn a handwritten digit dataset. The lower left panel illustrates the original and reconstructed training examples, as highlighted by the blue and gray regions, respectively. The upper left panel shows the implementation of the data encoding circuit and trainable circuit used in QNN. The label "×3" and "×L" means repeating the quantum gates in blue and brown boxes with 3 and L times, respectively. The lower center panel, highlighted by the yellow region, shows the training loss under different hyperparameter settings. In particular, the label "Loss_baseline_depL" refers to the obtained loss with circuit depth L (i.e., $L = 20$ or $L = 5$), $p = 0$, and $K \to \infty$, where p, and K refer to the depolarization rate and the number of measurements to estimate expectation value used in QNN. Similarly, the label "Loss_QNN_depL" refers to the obtained loss of QNN with setting the circuit depth as L (i.e., $L = 5, 20$), $p = 0.0025$, and $K = 20$. The label "Loss_FCNN" represents the obtained loss of FCNN. The upper right and lower right panels separately exhibit the training and test accuracy of FCNN and QNN with different hyperparameter settings.

that the gate and measurement noise are ubiquitous in these devices.

The tantalizing results indicated by the utility bound R_2 requests a technical assumption such that the loss function applied to QNNs should satisfy the PL condition (see Appendix B), which may not be applicable for most loss functions without regularization operations. In other words, the key message delivered by the utility bound R_2 is that the barren plateaus encountered by QNNs could be alleviated by reshaping the loss landscape assisted by the regularization operations. How to effectively modify the loss landscape without shifting the optimal solution is left as future work.

III. NUMERICAL SIMULATIONS

We employ the UCI ML handwritten digit datasets [79] to exhibit the correctness of utility bounds R_1 and R_2 of QNN, as achieved in Theorem 1. The employed dataset includes in total 1797 handwritten digit images with ten class labels, where each label refers to a digit and each image has 64 attributes. The data preprocessing has three steps. First, we clean the dataset and collect only images with labels 0 and 1. After cleaning, the total number of images is 360, where the number of examples with label 0 (label 1) is 178 (172). In other words, our simulation focuses on the binary classification task. Some collected examples are shown in the lower left panel of Fig. 2. Second, we utilize a feature reduction technique, i.e., principal component analysis (PCA) [80], to reduce the feature dimension of each data example from 64 to 3. The lower left panel of Fig. 2, highlighted by the gray region, exhibits the reconstructed handwritten digit images using the reduced data features. Such a step aims to balance the relatively high-dimension features of the data example and the limited quantum resources available in the present day. After applying PCA, we denote the employed dataset as $\mathbf{z} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{360}$, where $\mathbf{x}_i \in \mathbb{R}^3$ is the ith data feature and $y_i \in \{0, 1\}$ is the ith label. The last step is randomly splitting $\mathbf{z}$ into two groups, i.e., the training dataset $\mathbf{z}_t$ and the test dataset $\mathbf{z}_p$. The size of the training dataset $\mathbf{z}_t$ and the test dataset $\mathbf{z}_p$ is 280 and 80, respectively.

We now employ the preprocessed handwritten digit dataset $\mathbf{z}$ and quantum learning model as used in Ref. [17] (see Appendix G for the implementation details) to study the learnability of QNN under the depolarization noise. Specifically, we apply depolarization channel $\mathcal{N}_p$ to every quantum circuit depth, where the depolarization rate is set as $p = 0.0025$. The depth of trainable circuits $U(\boldsymbol{\theta})$ is set as $L = 5$ and $L = 20$, respectively. The corresponding

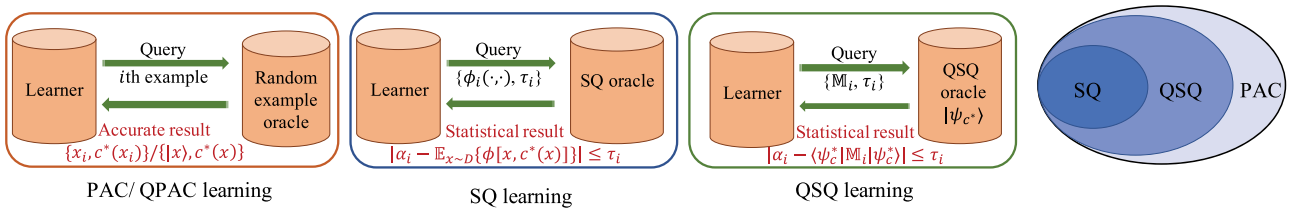

FIG. 3. The mechanism of different quantum learning models. The key difference of PAC (or QPAC), SQ, and QSQ learning is the way to acquire accessible data information, which is used to infer hypothesis $h(\cdot)$. In particular, the left panel exhibits the mechanism of PAC or QPAC learning. To inference $h(\cdot)$, the PAC learner requires access to the example oracle to acquire the accurate information of each example. The two middle panels represent the mechanism of SQ and QSQ learning, respectively. The SQ (QSQ) learner feeds a sequence of queries $\{[\phi_i(\cdot,\cdot), \tau_i]\}$ into a SQ (QSQ) oracle and receives the statistical results $\{\alpha_i\}$ about the target concept $c^*(\cdot)$. The collected results $\{\alpha_i\}$ are then employed to infer the hypothesis $h(\cdot)$ to approximate $c^*(\cdot)$. The right panel compares the power of SQ, QSQ, and PAC learning modes. Namely, restricted by the accessible information of the dataset, some problems can not be efficiently tackled by SQ and QSQ learning models, while they can be effectively solved by PAC learning models.

number of trainable parameters is 15 and 60, respectively. The number of measurements to estimate the expectation value is set as $K = 20$. We also train QNN without noisy channels $\mathcal{N}_p$ under the setting $L = 5, 20$ with the infinite measurements, which is utilized to evaluate how the system noise and the measurement shots affect the learning performance of QNNs. Since seeking the optimal solution $\boldsymbol{\theta}$ and the minimized objective function $\mathcal{L}^*$ is NP hard [81], in the following, we employ the results achieved by noiseless QNN to approximate the optimal results. As a reference, we utilize classical deep neural network learning models, i.e., fully connected neural network (FCNN) [1], to tackle the same binary classification task. The number of trainable parameters of FCNN is set as 15, which is at the same level with QNNs. The optimization strategy is identical to QNNs as described in the main text, where the mean square error is exploited as the loss function and the batch gradient-descent method is adopted to update trainable parameters (see Appendix G for the implementation details). The number of iterations for all numerical simulations described above is set as $T = 400$. The source code related to numerical simulations is available at the Github repository [82].

The simulation results, as shown in Fig. 2, accord with our theoretical results. Specifically, as shown in the lower center of Fig. 2, even though the gate noise and the finite number of measurements are considered, the training loss can still converge after a sufficient number of iterations. Moreover, the gap between the optimal result $\mathcal{L}^*$ (noiseless) and the results $\mathcal{L}(\boldsymbol{\theta}^{(T)})$ under the varied noise setting, as indicated by two red arrows, becomes large with increasing the circuit depth L. Such a phenomenon echoes with the result such that a larger L and p lead to a poorer utility bound. In addition, the achieved training and test accuracies as shown in the right panel of Fig. 2 implies that the noisy QNN can also learn a useful decision rule while its performance has slightly degenerated. In Appendix G, we quantitatively investigate whether the exponential dependence on L and the inverse dependence on K claimed in Theorem 1 can be observed in the above binary classification task. Compared with QNNs, FCNN attains a better performance with respect to the investigated learning task. In particular, after 25 iterations, its train accuracy (test accuracy) achieves 98.7% (99.4%), respectively. Moreover, its training loss converges to 0.01 after 150 iterations. We note that the superiority of QNNs versus DNNs highly depends on the explored dataset. This topic is systematically studied in our recent work [34].

IV. GENERALIZATION OF QNN

We next examine the generalization of QNN in the regime of the statistical query learning [83]. For concreteness, let us first elucidate the key difference between QPAC [42] and QSQ learning [50] paradigms before moving on to present the achieved main results. Recall that for both the classical and quantum machine learning, the generalization concerns whether the exploited learning model can effectively output a hypothesis $h \in \mathcal{H}$ with $\mathcal{H}$ being the hypothesis set that well approximates the target concept for a certain learning problem; namely, using *a polynomial sample or query complexity*. Define $\mathcal{C} \subseteq \{c : \{0,1\}^N \to \{0,1\}\}$ as a concept class and $\mathcal{D} : \{0,1\}^N \to [0,1]$ as an unknown distribution. In QPAC learning [40,42], the learner continuously collects the *labeled examples* $\{|\mathbf{x}_i\rangle, c^*(\mathbf{x}_i)\}$ and then uses these examples to infer a hypothesis $h(\cdot)$ to approximate $c^*(\cdot) \in \mathcal{C}$, as shown in the right panel of Fig. 3. In other words, the QPAC learners can directly access accurate information about a sequence of quantum examples to proceed inference. It is noteworthy that in the NISQ scenario, the generalization of QNNs can not be analyzed by QPAC learning theory, since the system and measurement noise forbids us to access the accurate information $c^*(\mathbf{x}_i)$. A potential solution to address this issue is QSQ learning paradigm given below.

Definition 1 (**Quantum statistical query learning oracle, [50]**): *Let* $c^*:\{0,1\}^N\to\{0,1\}$ *be an unknown concept sampled from a known concept class* $\mathcal{C}\subseteq\{c:\{0,1\}^N\to\{0,1\}\}$. *Define the quantum example as*

$$|\psi_{c^*}\rangle=\sum_{\mathbf{x}\in\{0,1\}^N}\sqrt{\mathcal{D}(\mathbf{x})}|\mathbf{x}\rangle|c^*(\mathbf{x})\rangle, \tag{8}$$

where $\mathcal{D}:\{0,1\}^N\to[0,1]$ *is some unknown distribution. A* QSQ *oracle receives a tolerance* $\tau\geq 0$ *and an observable* $\mathbb{M}\in(\mathbb{C}^2)^{\otimes N+1}\times(\mathbb{C}^2)^{\otimes N+1}$ *with* $\mathrm{Tr}(\mathbb{M})\leq 1$, *and outputs a number* α *satisfying*

$$|\alpha-\langle\psi_{c^*}|\mathbb{M}|\psi_{c^*}\rangle|\leq\tau. \tag{9}$$

The above definition indicates that in the case of QSQ learning, the learner can receive only the collected behavior of the whole examples $|\psi_{c^*}\rangle$ instead of information of the individual example $|\mathbf{x}\rangle|c^*(\mathbf{x})\rangle$. The characteristic of employing statistics about $|\psi_{c^*}\rangle$ to infer $h\in\mathcal{H}$ ensures that QSQ learners are more practically feasible than general QPAC learners, especially in the NISQ era [50]. As shown in the middle panel of Fig. 3, the generalization of QSQ learning models means that if there is an algorithm $\mathcal{A}$ such that for every $c^*\in\mathcal{C}$, the learner makes *at most* Q *queries* to the QSQ oracle, i.e., $\{\mathbb{M}_i,\tau_i\}_{i=1}^Q$, and then uses the returned $\{\alpha_i\}_{i=1}^Q$ to output a hypothesis h satisfying

$$\Pr_{\mathbf{x}\sim\mathcal{D}}[h(\mathbf{x})\neq c^*(\mathbf{x})]\leq\varepsilon. \tag{10}$$

When the required number of queries Q for the QSQ learning model to achieve Eq. (10) is less than the classical SQ learning model (as shown in the left panel of Fig. 3), we say that the QSQ learning model possesses *a better generalization property*.

The QSQ oracle can be well realized by noisy QNNs. Recall that three ingredients in the QSQ oracle are the preparation of the quantum example $|\psi_{c^*}\rangle$, the implementation of observable $\mathbb{M}$, and a tolerance τ. Following the description in Fig. 1(b), $|\psi_{c^*}\rangle$ can be prepared by a predefined input oracle $U_{\mathbf{x}}$ with basis encoding, as discussed in Appendix C. Moreover, the observable $\mathbb{M}$ can be implemented by the trainable unitary $U(\boldsymbol{\theta})$ associated with a fixed measurement Π, supported by Stinespring's dilation theorem [55]. With this regard, the expectation value of the QSQ oracle $\langle\psi_{c^*}|\mathbb{M}|\psi_{c^*}\rangle$ is equivalent to $\langle 0^{\otimes(N+1)}U_{\mathbf{x}}^\dagger U(\boldsymbol{\theta})^\dagger\Pi U(\boldsymbol{\theta})U_{\mathbf{x}}0^{\otimes(N+1)}\rangle$. Last, the estimation error τ between the output value α and the expectation value is determined by the system noise and the finite number of measurements. The following theorem analytically quantifies the relationship between QSQ oracle and noisy QNN, whose proof is provided in Appendix H.

Theorem 2: *Denote the total number of measurements applied to QNN as shown in Fig. 1(b) as* K *and the measured result at the* kth *time as* V_k. *Following notations in Definition 1, suppose that* $U_{\mathbf{x}}$ *prepares* $|\psi_{c^*}\rangle$, $U(\boldsymbol{\theta})^\dagger\Pi U(\boldsymbol{\theta})$ *is equal to* $\mathbb{M}$, *and the system noise is modeled by the depolarization channel* $\mathcal{N}_{\tilde{p}}$. *When* $K=\ln(2/b)/2(\tau-5\tilde{p}/4)^2$, *with probability* $1-b$, *we have*

$$\left|\frac{1}{K}\sum_{k=1}^K V_k-\langle\psi_{c^*}|\mathbb{M}|\psi_{c^*}\rangle\right|\leq\tau. \tag{11}$$

The achieved results imply that noisy QNNs with access to a unitary $U_{\mathbf{x}}$, which prepares a quantum example state $|\psi_{c^*}\rangle$, provide a concrete framework for realizing QSQ oracles. In other words, the sample mean $(1/K)\sum_{k=1}^K V_k$ amounts to the output of QSQ oracle α in Eq. (9). According to the definition of generalization in Eq. (10), given a sequence of queries $\{\mathbb{M}_i,\tau_i\}_{i=1}^Q$, the sample mean of noisy QNNs corresponds to $\{\alpha_i\}_{i=1}^Q$ and hence can be used to *infer* a hypothesis h that well approximates c^*. According to Refs. [50,83], these quantum statistical query oracles, or equivalently noisy QNNs, allow for more efficient learning of parities, juntas, and disjunctive normal form over classical SQ models. As a result, we attain a positive answer that noisy QNNs have the potential to possess superior generalization abilities over classical learning models in the regime of statistical query learning.

We note that the process of *inferring* h can be categorized into two classes, depending on how the set $\{\alpha_i\}_{i=1}^Q$ is collected. In the first class, the observables $\{\mathbb{M}_i,\tau_i\}_{i=1}^Q$ used to obtain $\{\alpha_i\}_{i=1}^Q$ are agnostic or correlated. That is, the observable $\mathbb{M}_i$, or equivalently $U(\boldsymbol{\theta})^\dagger\Pi U(\boldsymbol{\theta})$, should be *adaptively constructed* via updating $\boldsymbol{\theta}$ [as shown in Fig. 1(b)] or may be dependent on previous observables, e.g., $\{\mathbb{M}_{i-1},\mathbb{M}_{i-2},\ldots,\mathbb{M}_1\}$. For example, the optimization of $\boldsymbol{\theta}$ can be completed using some available training data. Interestingly, recent studies have envisioned the power of data in pursuing quantum advantages [36,84]. Nevertheless, how to conduct an efficient optimization on $U(\boldsymbol{\theta})$ to accurately estimate $\{\mathbb{M}_i\}_{i=1}^Q$ with provable quantum advantages is beyond the scope of this study and will be left as future work.

In the second class, the observables $\{\mathbb{M}_i,\tau_i\}_{i=1}^Q$ are *predefined and fixed*. Under this setting, for each $i\in[Q]$, the optimal parameters in $U(\boldsymbol{\theta})$, i.e., $\boldsymbol{\theta}^*$, can be explicitly calculated at the initialization step *without optimization*, i.e., $U(\boldsymbol{\theta}^*)^\dagger\Pi U(\boldsymbol{\theta}^*)=\mathbb{M}$. Note that under this scenario, noisy QNNs can be exploited to efficiently construct QSQ oracles and hence attain quantum advantages. For illustration, we exemplify how to use noisy QNN to construct the QSQ oracle adopted in parity learning without optimization. According to the analysis in Ref. [50, Lemma 4.2], for $\forall i\in[Q]$ and $Q=N$ the observable $\mathbb{M}_i$ is equal to $|1\rangle\langle 1|_i\otimes H^{\otimes N-1}\left(|0\rangle\langle 0|^{\otimes[N]\setminus\{i\}}\right)H^{\otimes N-1}$, where the ith qubit is fixed to $|1\rangle\langle 1|$ and the remaining $N-1$ qubits can be obtained by applying the Hadamard gates. When Π is set as

$|0\rangle\langle 0|^{\otimes N}$, the trainable quantum circuit aims to simulate $U^* = X_i \otimes H^{\otimes N-1}$, where the ith qubit is applied to and X gate and the remaining $N-1$ qubits are operated with the Hadamard gates. Note that U^* can be effectively constructed by the universal parameterized quantum circuits such as hardware-efficient ansatz and the ansatz proposed by Ref. [85, Sec. II.B and Theorem 1] with $O[\text{poly}(N)]$ gate complexity. With this regard, we conclude that noisy QNNs allow for efficient learning of parities.

We remark that the results achieved in Theorem 2 paves a new way to explore generalization advantages of QNNs. More specifically, besides the QPAC learning paradigm, QNNs can also achieve certain superiority over their classical counterparts under the statistical query learning paradigm. More precisely, it is intriguing to understand the potential of noisy QNNs on diverse statistical learning problems such as support vector machines, linear and convex optimization, simulated annealing, matrix decomposition, and so on [86,87]. In particular, we can first examine whether QSQ learning models can tackle these tasks that outperform their classical counterparts. If the answer is positive, we can leverage the result in Theorem 2 to design a noisy QNN that achieves these tasks with quantum advantages.

V. DISCUSSION AND CONCLUSION

To summarize, we explore the learnability of QNNs from the aspect of the trainability and generalization. The achieved utility bounds towards ERM indicate that, more measurements, lower noise, and shallower circuit depth contribute to a better performance of QNNs. These results can guide us to devise more advanced QNN-based learning models that are robust to inevitable gate noise and insensitive to the barren-plateau phenomenon.

The analysis technique established in this study can be applied to explain the heuristic result achieved in other quantum learning protocols. More precisely, we can compare the performance of different optimizers (e.g., quantum natural gradient-descent methods [88] and other high-order gradient-descent methods [89]) and different loss functions (e.g., the cross-entropy loss) in the measure of the utility bounds R_1 and R_2. In other words, our study contributes to a better understanding of QNNs and other variational quantum algorithms.

We stress that optimization theory and statistical learning theory adopted in this study are powerful tools to facilitate us to deeply understand the capabilities and limitations of QNNs. Indeed, there is an incremental interest of utilizing these tools to explore the power of QNNs and variational quantum algorithms from different angles, where representative examples include analyzing the generalization error bounds of quantum machine learning models and deriving the expressivity of various ansatzes [29–31,35–37]. Notably, a recent study [27] has analyzed how the utility bound R_2 is bounded by a quantity related to the quantum fisher information of the variational state for a fixed number of iterations T. The corresponding consequences allow us to identify the fundamental difference between QNNs and DNNs, which is crucial to seek quantum advantages in the NISQ era.

Besides studying the trainability of QNNs, we also demonstrate that in the regime of statistical query learning, noisy QNN can be applied to accomplish parity, juntas, and disjunctive normal form (DNF) with better generalization property over classical SQ learning models. Although the achieved results are established on the setting in which the query set $\{\mathbb{M}_i, \tau_i\}$ is predefined at the initialization step, it is of great importance to explore where noisy QNNs possess better generalization ability when the query set $\{\mathbb{M}_i, \tau_i\}$ is adaptive. A positive answer could broaden the applications of NISQ machines.

ACKNOWLEDGMENTS

This work received support from the Faculty of Engineering and Information Technologies at the University of Sydney (the Engineering and Information Technologies Research Scholarship) and Australian Research Council (Australian Research Council Project with ID FL-170100117).

APPENDIX

The organization of the Appendix is as follows. In Appendix A, we unify the notations used in the whole Appendix. In Appendix B, we introduce the parameter shift rule and analyze the analytic and estimateed gradients of QNN. In Appendix C, we elaborate the implementation details of the quantum encoding circuit $U_{\mathbf{x}}$ and the trainable quantum circuit $U(\boldsymbol{\theta})$ used in QNN. In Appendix D, we quantify the properties of the objective function with respect to the optimization theory, which will be employed to prove the utility bounds of QNN. Then, in Appendix E, we exhibit the proof of Theorem 3, as the precondition to achieve utility bounds of QNN. In Appendix F, we exhibit the proof details of Theorem 1 that achieves the utility bounds of QNN towards ERM. In Appendix G, we present more simulation details. Next, in Appendix H, we prove Theorem 2, which shows that any QSQ oracle can be efficiently simulated by noisy QNN. Finally, in Appendix I, we generalize all achieved results to a more general quantum channel.

APPENDIX A: THE SUMMARY OF NOTATIONS

We unify the notations throughout the whole paper. We denote d as the number of training parameters ($\boldsymbol{\theta} \in \mathbb{R}^d$). Define N as the number of qubits and n as the number of training examples. Denote the set $\{1, 2, \ldots, m\}$ as $[m]$. With a slight abuse of notations, we denote ℓ_b as the b

norm, while ℓ (without subscript) is the loss function. We denote the ℓ_p norm of $\mathbf{v}$ as $\|\mathbf{v}\|_p$. In particular, $\|\mathbf{v}\|$ refers to the ℓ_2 norm. We use $O(\cdot)$ [or $\tilde{O}(\cdot)$] to denote the complexity bound (hide polylogarithmic factors). A random variable X that follows Delta distribution is denoted as $X \sim \mathrm{Del}(x_0)$, i.e., $\Pr(X = x_0) = 1$ and $\Pr(X \neq x_0) = 0$. A random variable X that follows uniform distribution is denoted as $X \sim U(a, b)$, where $P(X = x_0) = 1/(b - a)$ with $a \leq x_0 \leq b$.

APPENDIX B: THE PROPERTY OF GRADIENTS IN QNNS

In this section, we first review the parameter shift rule, which is used to calculate the gradients of QNN. We next leverage the parameter shift rule to analyze the relation between the analytic and estimated gradients of QNN.

1. Parameter shift rule

Denote the updating rule of QNN at the tth iteration as

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \frac{\eta}{B}\sum_{i=1}^{B}\nabla\mathcal{L}_i(\boldsymbol{\theta}^{(t)}).$$

To acquire the analytic gradient $\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) = (\hat{Y}_i^{(t)} - Y_i)\partial\hat{Y}_i^{(t)}/\partial\boldsymbol{\theta}_j^{(t)} + \lambda\boldsymbol{\theta}_j^{(t)}$ with $j \in [d]$, the parameter shift rule proceeds by separately feeding tunable parameters $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(t,\pm_j)} := \boldsymbol{\theta}^{(t)} \pm (\pi/2)\mathbf{e}_j$ to the trainable circuit $U(\boldsymbol{\theta})$, where $\mathbf{e}_j$ is the basis vector with the jth entry being 1 and zero otherwise. Following the above notations, we denote $\hat{Y}_i^{(t)}$ and $\hat{Y}_i^{(t,\pm_j)}$ as expectation values of quantum measurements when feeding parameters $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(t,\pm_j)}$ into the trainable quantum circuit $U(\boldsymbol{\theta})$ in the noiseless scenario. The corresponding analytic gradient of QNN is

$$\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) = (\hat{Y}_i^{(t)} - Y_i)\frac{\hat{Y}_i^{(t,+_j)} - \hat{Y}_i^{(t,-_j)}}{2} + \lambda\boldsymbol{\theta}_j^{(t)}.$$

However, in practice, QNN could generate only statistics $\bar{Y}_i^{(t)} = (1/K)\sum_{k=1}^{K} V_k^{(t)}$ and $\bar{Y}_i^{(t,\pm_j)} = (1/K)\sum_{k=1}^{K} V_k^{(t,\pm_j)}$, where $V_k^{(t)} \sim \mathrm{Ber}(\tilde{Y}_i^{(t)})$ and $V_k^{(t,\pm_j)} \sim \mathrm{Ber}(\tilde{Y}_i^{(t,\pm_j)})$, and $\tilde{Y}_i^{(t)}$ and $\tilde{Y}_i^{(t,\pm_j)}$ refer to expectation values of quantum measurements when feeding parameters $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(t,\pm_j)}$ into the noisy trainable quantum circuit $U(\boldsymbol{\theta})$. This leads to the estimated gradient as

$$\nabla_j\bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (\bar{Y}_i^{(t)} - Y_i)\frac{\bar{Y}_i^{(t,+_j)} - \bar{Y}_i^{(t,-_j)}}{2} + \lambda\boldsymbol{\theta}_j^{(t)}.$$

Note that the difficulties of optimizing QNN arise when only the approximated $\hat{Y}_i^{(t)}$ and $\partial\hat{Y}_i^{(t)}/\partial\boldsymbol{\theta}^{(t)}$ are available caused by the finite number of measurements, and the precision deteriorates when more iterations occur.

The analytic and estimated gradients of QNN. As explained in the main text, the key component to prove Theorem 1 is quantifying the relation between the analytic and the estimated gradient of QNN. Here we show that the estimated gradient, which is caused by the gates' noise and the sample errors, can be explicitly formulated to relate with its analytic gradient. An informal result is summarized below (see Appendix E for details).

Theorem 3: *It follows that*

$$\nabla_j\bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (1-\tilde{p})^2\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + \boldsymbol{\varsigma}_i^{(t,j)},$$

where $\tilde{p} = 1 - (1-p)^{L_Q}$, L_Q is the circuit depth, the constant $C_{j,1}^{(i,t)}$ depends only on Y_i, $\boldsymbol{\theta}^{(t)}$, and $\tilde{p}$, and $\boldsymbol{\varsigma}_i^{(t,j)}$ follows the distribution $\mathcal{P}_Q$ that is formed by Y_i, $\boldsymbol{\theta}^{(t)}$, the number of measurements K, and $\tilde{p}$ with zero mean.

Theorem 3 indicates that the estimated gradient $\nabla_j\bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ is centralized around the $(1-\tilde{p})^2\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)}$ and perturbed by a random variable $\boldsymbol{\varsigma}_i^{(t,j)}$. This enables us to quantitively measure how far the estimated gradient is away from the analytic gradient, which is the precondition to leverage the optimization theory to analyze the performance of QNN. Moreover, the result of Theorem 3 implies that, compared with the finite measurements, the gate error is more harmful for the QNN's optimization, which may lead to diverging. In particular, the term $C_{j,1}^{(i,t)}$, which is independent with K, will always exist and induce a biased optimization direction when $\tilde{p} \neq 0$. For the worst case, with $\tilde{p} = 1$, the analytic gradient information is exactly lost. In contrast, K determines only the variance of the distribution $\mathcal{P}_Q$ with zero mean, where classical and quantum literatures [90,91] have provided the convergence guarantee even if $K = 1$.

APPENDIX C: IMPLEMENTATION DETAILS OF ENCODING CIRCUIT AND TRAINABLE CIRCUIT OF QNN

The selection of encoding circuits $U_{\mathbf{x}}$ and trainable circuit $U(\boldsymbol{\theta})$ is flexible in QNN. We now separately explain the implementation details of these two circuits supported by QNN.

Encoding circuit $U_{\mathbf{x}}$. The typical encoding circuits of QNN can be divided into four categories. A common feature of these encoding methods is that their implementation only costs a low circuit depth, driven by the restricted quantum resources. Let the feature dimension of the classical example $\mathbf{x}_i$ be D_c with $i \in [n]$. The first category is the direct amplitude encoding [74–76,92]. Specifically, the encoder circuit satisfies $U_{\mathbf{x}} : \mathcal{B}_i \to (1/\sqrt{B_s})\sum_{b=1}^{B_s}\sum_{j=1}^{D_c}\hat{\mathbf{x}}_{b,j}^{(i)}|b\rangle|j\rangle$ with $\hat{\mathbf{x}}_{b,j}^{(i)} = \mathbf{x}_{b,j}^{(i)}/\|\mathbf{x}_{b,j}^{(i)}\|$. This method requires a low feature dimension

D_c, since the quantum gates' complexity to build $U_{\mathbf{x}}$ is $O(D_c)$. The second category is the kernel mapping [17–19], where $\mathcal{B}_i$ is encoded into a set of single-qubit gates with a specified arrangement, e.g., $U_{\mathbf{x}}(\mathcal{B}_i) = \sum_{b=1}^{B_s} (|b\rangle\langle b|) \otimes_{j=1}^{D_c} R_Y(\mathbf{x}_{b,j}^{(i)})$. The third category is the dimension-reduction method proposed by Ref. [77]. Specifically, instead of encoding $\mathcal{B}_i$, the amplitude or kernel encoder circuits $U_{\mathbf{x}}$ is exploited to encode a projected features $g(\mathcal{B}_i) \in \mathbb{R}^{B_s \times D_c'}$, where $g(\cdot)$ is a predefined function and $D_c' \ll D_c$. The fourth category is the basis encoding [16,42,50], which is broadly used in quantum learning theory. Specifically, the encoding circuit $U_{\mathbf{x}}$ is employed to prepare a quantum example $|\psi\rangle = \sum_{\mathbf{x}\in\{0,1\}^N} \sqrt{\mathcal{D}(\mathbf{x})}|\mathbf{x}, c(\mathbf{x})\rangle$ with $N = \lceil \log_2 D_c \rceil$, where $\mathcal{D}(\mathbf{x})$ is the data distribution over $\mathbf{x}$, $c(\mathbf{x})$ corresponds to the label of the bit string $\mathbf{x}$ [42,43]. In most cases, the distribution $\mathcal{D}(\mathbf{x})$ is uniform. Hence, the state $|\psi\rangle$ can be efficiently prepared by setting $B = 1$, and applying Hadamard gates and control NOT gates [55] to the initial state $|0\rangle^{\otimes N+1}$.

Trainable quantum circuits $U(\boldsymbol{\theta})$. The trainable quantum circuits, also known as parameterized quantum circuits [78,93], used in QNN can be written as a product of layers of unitaries in the form $U(\boldsymbol{\theta}) = \prod_{l=1}^{L} U_l(\boldsymbol{\theta}_l)$, where $U_l(\boldsymbol{\theta}_l)$ is composed of parameterized single-qubit gates and fixed two-qubit gates. Each trainable layer can be decomposed into $U_l(\boldsymbol{\theta}_l) = [\bigotimes_{k=1}^{N} U_{l,k}(\boldsymbol{\theta}_l)]U_{\text{eng}}$, where $U_{l,k}(\boldsymbol{\theta}_l)$ represents the composition of trainable single-qubit gates and U_{eng} refers to the entanglement layer that contains two-qubit gates. Depending on the detailed architecture, the implementation of $U_l(\boldsymbol{\theta}_l)$ can be categorized into three classes. The first class is the hardware-efficient circuit architecture, where the selection of $U_k(\boldsymbol{\theta}_l)$ and U_{eng} is according to the given NISQ machine that has the specific sparse qubit-to-qubit connectivity and a specified set of quantum gates [26,41]. The second class is the tensor-network-inspired architecture. In particular, the layout of quantum gates is following different tensor networks, e.g., the matrix product state, the tree tensor network, and the multiscale entanglement renormalization ansatz (MERA) [94]. The third class is the Hamiltonian-based architecture, where the entanglement layer U_{eng} refers to a specific Hamiltonian, e.g., Ref. [18] employs $U_{\text{eng}} = e^{-iHT}$ with $H = \sum_{j=1}^{N} a_j X_j + \sum_{j=1}^{N} \sum_{k=1}^{j-1} J_{jk} Z_i Z_k$. Notably, almost all quantum approximate optimization algorithms follow the Hamiltonian-based architecture [20].

APPENDIX D: THE *S*-SMOOTH, *G*-LIPSCHITZ, AND PL CONDITION PROPERTIES FOR THE OBJECTIVE FUNCTION

Before quantifying properties of the objective function used in QNN from the perspective of the optimization theory, we first present the formal definition of *S*-smooth, *G*-Lipschitz, and PL condition properties.

Definition 2: *A function f is S smooth over a set $\mathcal{C}$ if $\nabla^2 f(\boldsymbol{u}) \preceq S\mathbb{I}$ with $S > 0$ and $\forall \boldsymbol{u} \in \mathcal{C}$. A function f is G Lipschitz over a set $\mathcal{C}$ if for all $\boldsymbol{u}, \boldsymbol{w} \in \mathcal{C}$, we have $|f(\boldsymbol{u}) - f(\boldsymbol{w})| \leq G\|\boldsymbol{u} - \boldsymbol{w}\|_2$. A function f satisfies PL condition if there exists $\mu > 0$ and for every possible $\boldsymbol{\theta} \in \mathcal{C}$, $\|\nabla f(\boldsymbol{\theta})\|^2 \geq 2\mu[f(\boldsymbol{\theta}) - f^*]$, where $f^* = \min_{\boldsymbol{\theta}\in\mathcal{C}} f(\boldsymbol{\theta})$.*

To ease the discussion, let us formulate the explicit form of $\mathcal{L}(\boldsymbol{\theta})$. Without loss of generality, we set $B = n$, where each batch $\mathcal{B}_i$ contains only the ith input $\mathbf{x}_i$ with $B_s = 1$. Denote the prepared quantum states as $\{\rho_{\mathcal{B}_i}\}_{i=1}^n$, i.e., $\rho_{\mathcal{B}_i} = |\phi_{\mathcal{B}_i}\rangle\langle\phi_{\mathcal{B}_i}|$ and $|\phi_{\mathcal{B}_i}\rangle \xleftarrow{U_{\mathbf{x}}} \{\mathbf{x}_i\}$ refers to the quantum example corresponding to the classical input batch $\mathcal{B}_i$ (or equivalently, $\mathbf{x}_i$). The explicit form of the objective function is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \left(\hat{y}_i - y_i\right)^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2, \tag{D1}$$

where $\hat{y}_i = \text{Tr}[\Pi U(\boldsymbol{\theta})\rho_{\mathcal{B}_i} U(\boldsymbol{\theta})^\dagger]$ refers to the prediction of QNN given the ith input $\mathbf{x}_i$, $U(\boldsymbol{\theta})$ is the trainable circuit, Π is the employed two-outcome POVM, and y_i is the true label of the ith input. Moreover, since the tunable parameters $\boldsymbol{\theta}$ in QNN refer to the rotation angles, we set its range as $\boldsymbol{\theta} \in [\pi, 3\pi]^d$.

Given Definition 2 and Eq. (D1), the properties of the objective function $\mathcal{L}$ are summarized in the following lemma.

Lemma 1: *Following the notations in Eq. (D1), $\mathcal{L}(\boldsymbol{\theta})$ is S smooth with $S = [(3/2) + \lambda]d^2$ and G Lipschitz with $G = d(1 + 3\pi\lambda)$. Assuming $\lambda \in [0, (1/3\pi)] \cup [(1/\pi), \infty]$, $\mathcal{L}$ satisfies PL condition with $\mu = (-1+\lambda\pi)^2/[1+\lambda d(3\pi)^2]$.*

Proof of Lemma 1. We employ the three lemmas presented below to prove Lemma 1, whose proofs are given in the following subsections. ■

Lemma 2: *The objective function $\mathcal{L}$ is S smooth with $S = (3/2 + \lambda)d^2$.*

Lemma 3: *The objective function $\mathcal{L}$ is G Lipschitz with $G = d(1 + 3\pi\lambda)$.*

Lemma 4: *Assume $\lambda \in [0, (1/3\pi)] \cup [(1/\pi), \infty]$. The objective function $\mathcal{L}$ satisfies the PL condition with $\mu = (-1 + \lambda\pi)^2/[1 + \lambda d(3\pi)^2]$.*

In conjunction with the results of Lemmas 2–4, the proof of Lemma 1 is completed. ■

1. Proof of Lemma 2: *S* smooth

Proof of Lemma 2. Recall the function $\mathcal{L}(\boldsymbol{\theta})$ is S smooth if

$$\nabla^2\mathcal{L}(\boldsymbol{\theta}) \preceq S\mathbb{I}, \tag{D2}$$

with $S > 0$. In other words, to promise $S\mathbb{I} - \nabla^2\mathcal{L}(\boldsymbol{\theta})$ is a positive semidefinite matrix as required in Eq. (D2), we need to obtain the upper bound of the second derivative of $\mathcal{L}(\boldsymbol{\theta})$, i.e., $S \geq \|\nabla^2\mathcal{L}(\boldsymbol{\theta})\|_2$.

Following the notation used in Eq. (D1), the gradient for the parameter $\boldsymbol{\theta}_j$ is

$$\begin{aligned}\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}_j} &= \frac{2}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)\frac{\partial\hat{y}_i}{\partial\boldsymbol{\theta}_j} + \frac{\lambda}{2}\frac{\partial\|\boldsymbol{\theta}\|_2^2}{\partial\boldsymbol{\theta}_j}\\ &= \frac{2}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)\frac{\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}}{2} + \lambda\boldsymbol{\theta}_j\\ &\leq 1 + 3\lambda\pi,\end{aligned} \tag{D3}$$

where $\hat{y}_i^{(\pm_j)} = \mathrm{Tr}\{\Pi U[\boldsymbol{\theta} \pm (\pi/2)\mathbf{e}_j]\rho_{\mathcal{B}_i}U[\boldsymbol{\theta} \pm (\pi/2)\mathbf{e}_j]^\dagger\}$, the second equality employs the conclusion of the parameter shift rule with $\partial\hat{y}_i/\partial\boldsymbol{\theta}_j = (\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)})/2$ [18,56], and the last inequality uses the facts $\pi \leq \boldsymbol{\theta}_j \leq 3\pi$, $(\hat{y}_i - y_i) \leq 1$, and $\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)} \leq 1$, since $\hat{y}_i, y_i, \hat{y}_i^{(\pm_j)} \in [0,1]$.

The upper bound of the derivative $\partial^2\mathcal{L}(\boldsymbol{\theta})/\partial\boldsymbol{\theta}_j\partial\boldsymbol{\theta}_k$ can be derived using the results of Eq. (D3). In particular,

$$\begin{aligned}\frac{\partial^2\mathcal{L}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}_j\partial\boldsymbol{\theta}_k} &= \frac{\partial\left(\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}_j}\right)}{\partial\boldsymbol{\theta}_k} = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial\left[\left(\hat{y}_i - y_i\right)\left(\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}\right) + \lambda\boldsymbol{\theta}_j\right]}{\partial\boldsymbol{\theta}_k}\\ &= \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\partial\hat{y}_i}{\partial\boldsymbol{\theta}_k}\left(\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}\right) + \left(\hat{y}_i - y_i\right)\frac{\partial\left(\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}\right)}{\partial\boldsymbol{\theta}_k} + \lambda\right]\\ &\leq \frac{3}{2} + \lambda,\end{aligned} \tag{D4}$$

where the first equality comes from the last equality of Eq. (D3), and the last inequality employs $(\hat{y}_i - y_i) \leq 1$, $\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)} \leq 1$, and

$$\frac{\partial\hat{y}_i}{\partial\boldsymbol{\theta}_k}, \frac{\partial\hat{y}_i^{(+_j)}}{\partial\boldsymbol{\theta}_k}, \frac{\partial\hat{y}_i^{(-_j)}}{\partial\boldsymbol{\theta}_k} \in [-1/2, 1/2],$$

supported by the parameter shift rule and $\hat{y}_i, \hat{y}_i^{(\pm_j)} \in [0,1]$.

The result of Eq. (D4) implies that $\|\nabla^2\mathcal{L}\|_2 \leq d\|\nabla^2\mathcal{L}\|_\infty \leq d^2(\frac{3}{2} + \lambda)$. In conjunction with Eq. (D2), the objective function is S smooth with $S = d^2(\frac{3}{2} + \lambda)$. ■

2. Proof of Lemma 3: *G* Lipschitz

Proof of Lemma 3. Recall a function $f(\mathbf{x})$ is G Lipschitz if it satisfies

$$|f(\mathbf{b}) - f(\mathbf{a})| \leq G\|\mathbf{b} - \mathbf{a}\|. \tag{D5}$$

Moreover, the mean value theorem gives that, if $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable and $[\mathbf{a}, \mathbf{b}] \subseteq \mathbb{R}^d$, then $\exists\mathbf{c} \in (\mathbf{a}, \mathbf{b})$ such that

$$f(\mathbf{b}) - f(\mathbf{a}) = \langle\nabla f(\mathbf{c}), \mathbf{b} - \mathbf{a}\rangle. \tag{D6}$$

Combining Eqs. (D5) and (D6), the G-Lipschitz condition in Eq. (D5) is equivalent to

$$|\langle\nabla f(\mathbf{c}), \mathbf{b} - \mathbf{a}\rangle| \leq G\|\mathbf{b} - \mathbf{a}\|. \tag{D7}$$

We now replace f, $\mathbf{b}$, and $\mathbf{a}$ used in Eq. (D7) with $\mathcal{L}$, $\boldsymbol{\theta}^{(1)}$, and $\boldsymbol{\theta}^{(2)}$ to prove that the objective function $\mathcal{L}$ is G Lipschitz. Specifically, we need to find a real value G that satisfies

$$\left|\left\langle\nabla\mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)}\right\rangle\right| \leq G\|\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)}\|, \tag{D8}$$

where $\boldsymbol{\theta} \in (\boldsymbol{\theta}^{(2)}, \boldsymbol{\theta}^{(1)})$.

The upper bound of the term $\left\langle\nabla\mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)}\right\rangle$ is

$$\begin{aligned}\left\langle\nabla\mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)}\right\rangle &\leq \|\nabla\mathcal{L}(\boldsymbol{\theta})\|\,\|\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)}\|\\ &\leq d\,\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_\infty\,\|\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)}\|.\end{aligned} \tag{D9}$$

In conjunction with Eqs. (D8) and (D9), G Lipschitz of $\mathcal{L}$ requests

$$d\,\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_\infty \le G. \tag{D10}$$

By leveraging the result of Eq. (D3) with $\nabla_j\mathcal{L}(\boldsymbol{\theta}) \le 1+3\lambda\pi$, we obtain the upper bound of the left side in Eq. (D10) is

$$d\,\|\nabla\mathcal{L}(\boldsymbol{\theta})\|_\infty \le d(1+3\pi\lambda). \tag{D11}$$

This leads to the objective function $\mathcal{L}$ of QNN satisfying G Lipschitz with $G = d(1+3\pi\lambda)$. ■

3. Proof of Lemma 4: the PL condition

Proof of Lemma 4. Recall the definition of Polyak-Lojasiewicz as formulated in Definition 2, it requires that the objective function $\mathcal{L}$ satisfies

$$\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 \ge 2\mu[\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*], \tag{D12}$$

where $\mathcal{L}^* = \min_{\boldsymbol{\theta}\in\mathcal{C}}\mathcal{L}(\boldsymbol{\theta})$.

We first derive a lower bound of $\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2$. In particular, we have

$$\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 = \sum_{j=1}^{d}[\nabla_j\mathcal{L}(\boldsymbol{\theta}_j)]^2 \ge \max_j[\nabla_j\mathcal{L}(\boldsymbol{\theta})]^2. \tag{D13}$$

The lower bound of $\max_j[\nabla_j\mathcal{L}(\boldsymbol{\theta})]^2$ as shown in Eq. (D13) follows

$$\max_j[\nabla_j\mathcal{L}(\boldsymbol{\theta})]^2 \ge \min_{\boldsymbol{\theta}_j\in[\pi,3\pi]}(-1+\lambda\boldsymbol{\theta}_j)^2, \tag{D14}$$

where the last inequality is achieved by exploiting the last second line of Eq. (D3), and the fact $\hat{y}_i, y_i, \hat{y}_i^{(\pm_j)} \in [0,1]$ and $\lambda > 0$, i.e.,

$$\nabla_j\mathcal{L}(\boldsymbol{\theta}) = \frac{2}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)\frac{\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}}{2} + \lambda\boldsymbol{\theta}_j \ge -1 + \lambda\boldsymbol{\theta}_j.$$

Combining the assumption $\lambda \in [0,(1/3\pi)] \cup [(1/\pi),\infty]$ and the above results, the lower bound of Eq. (D13) satisfies

$$\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 \ge (-1+\lambda\boldsymbol{\theta}_j)^2 > 0.$$

We then derive the upper bound of the term $[\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*]$ in Eq. (D12). In particular, we have

$$\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^* \le \mathcal{L}(\boldsymbol{\theta}) + 0 \le 1 + \lambda d(3\pi)^2, \tag{D15}$$

where the first inequality comes from the definitions of $\mathcal{L}^*$, i.e.,

$$-\mathcal{L}^* = -\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i^* - y_i)^2 - \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 \le 0,$$

with $\hat{y}_i^* = \mathrm{Tr}[\Pi U(\boldsymbol{\theta}^*)\rho_i U(\boldsymbol{\theta}^*)^\dagger]$, and the second inequality employs the definition of $\mathcal{L}(\boldsymbol{\theta})$ with

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 \le 1 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2,$$

and $(\lambda/2)\|\boldsymbol{\theta}\|^2 \le (\lambda/2)d\|\boldsymbol{\theta}\|_\infty^2 = (3\pi)^2\lambda d/2$.

By combining Eqs. (D14) and (D15) with Eq. (D12), we obtain the following relation:

$$\begin{aligned}\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 &\ge (-1+\lambda\pi)^2 \ge 2\mu[1+\lambda d(3\pi)^2] \\ &\ge 2\mu[\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*].\end{aligned} \tag{D16}$$

The above relation indicates that the objection function $\mathcal{L}(\boldsymbol{\theta})$ satisfies the PL condition with

$$\mu = \frac{(-1+\lambda\pi)^2}{1+\lambda d(3\pi)^2}.$$ ■

APPENDIX E: PROOF OF THEOREM 3

Theorem 3 establishes the relation between the analytic gradient $\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)})$ and the estimated gradient $\nabla_j\bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ of QNN. Its formal description is as follows.

Theorem 4 (The formal description of Theorem 3): *Denote $\tilde{p} = 1-(1-p)^{L_Q}$ with L_Q being the quantum circuit depth. At the tth iteration, we define five constants with*

$$C_{j,a}^{(i,t)} = \begin{cases} (1-\tilde{p})\tilde{p}(1/2 - Y_i)(\hat{Y}_i^{(t,+_j)} - \hat{Y}_i^{(t,-_j)}) - (2\tilde{p}-\tilde{p}^2)\lambda\boldsymbol{\theta}_j^{(t)}, & a=1 \\ (1-\tilde{p})(\hat{Y}_i^{(t,+_j)} - \hat{Y}_i^{(t,-_j)}), & a=2 \\ [(1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}/2 - Y_i], & a=3 \\ \frac{-(1-\tilde{p})(\hat{Y}_i^{(t)})^2 + (1-\tilde{p})^2\hat{Y}_i^{(t)} + (\tilde{p}/2) - (\tilde{p}^2/4)}{K}, & a=4 \\ \frac{-(1-\tilde{p})[(\hat{Y}_i^{(t,+_j)})^2 + (\hat{Y}_i^{(t,-_j)})^2] + (1-\tilde{p})^2(\hat{Y}_i^{(t,+_j)} + \hat{Y}_i^{(t,-_j)}) + \tilde{p} - (\tilde{p}^2/2)}{K}, & a=5, \end{cases}$$

where $\hat{Y}_i^{(t,\pm_j)} = \mathrm{Tr}[\Pi U(\boldsymbol{\theta} \pm \boldsymbol{e}_j)\rho_{\mathcal{B}_i} U(\boldsymbol{\theta} \pm \boldsymbol{e}_j)^\dagger]$*, K refers to the number of quantum measurements, and* $\hat{Y}_i^{(t)}$ *and* Y_i *are the sum average of the predicted and true labels for the* i*th batch* $\mathcal{B}_i$*.*

The relation between the estimated and analytic gradients of QNN follows

$$\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (1-\tilde{p})^2 \nabla_j \mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + \varsigma_i^{(t,j)}$$

with $\varsigma_i^{(t,j)} = C_{j,2}^{(i,t)} \xi_i^{(t)} + C_{j,3}^{(i,t)} \xi_i^{(t,j)} + \xi^{(t)} \xi_i^{(t,j)}$*, where* $\xi_i^{(t)}$ *and* $\xi_i^{(t,j)}$ *are two random variables with zero mean and variances* $C_{j,4}^{(i,t)}$ *and* $C_{j,5}^{(i,t)}$*, respectively.*

The intuition to achieve Theorem 4 is as follows. As explained in the main text, the discrepancy between the estimated gradient $\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ and the analytic gradient $\nabla_j \mathcal{L}_i(\boldsymbol{\theta}^{(t)})$ is caused by the difference between the estimated results $\bar{Y}_i^{(t)}$ (or $\bar{Y}_i^{(t,\pm_j)}$) and the expected results $\hat{Y}_i^{(t)}$ (or $\hat{Y}_i^{(t,\pm_j)}$), due to the involved depolarization noise $\mathcal{N}_p$ and the finite number of measurements K. Specifically, the noisy channel $\mathcal{N}_p$ shifts the expectation values, and the finite number of measurements K turns the output of the quantum circuit from the determination to be random. Under the above observation, the estimated gradients $\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ can be treated as the random variable that is formed by three random variables $\bar{Y}_i^{(t)}$ and $\bar{Y}_i^{(t,\pm_j)}$, where the probability distributions of $\bar{Y}_i^{(t)}$ and $\bar{Y}_i^{(t,\pm_j)}$ are determined by $K, \mathcal{N}_p, \hat{Y}_i^{(t)}$, and $\hat{Y}_i^{(t,\pm j)}$. Therefore, to explicitly build the relation between $\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ and $\nabla_j \mathcal{L}_i(\boldsymbol{\theta}^{(t)})$, we should first formulate the distribution of the estimated gradients using $\bar{Y}_i^{(t)}$ and $\bar{Y}_i^{(t,\pm_j)}$, and then connect the obtained distribution with the analytic gradients. The following lemma summarizes the distribution of the estimated gradients using $\bar{Y}_i^{(t)}$ and $\bar{Y}_i^{(t,\pm_j)}$, whose proof is given in Sec. 1.

Lemma 5: *The mean* $\nu_i^{(t)}$ *and variance* $(\sigma_i^{(t)})^2$ *of the estimated result* $\bar{Y}_i^{(t)}$ *are*

$$\begin{aligned} \nu^{(t)} &= (1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D}, \\ (\sigma_i^{(t)})^2 &= \frac{-(1-\tilde{p})^2(\hat{Y}_i^{(t)})^2 + (1-\tilde{p})\{1-2\tilde{p}[\mathrm{Tr}(\Pi)/D]\}\hat{Y}_i^{(t)} + \tilde{p}[\mathrm{Tr}(\Pi)/D] - \tilde{p}^2\{[\mathrm{Tr}(\Pi)]^2/D^2\}}{K}. \end{aligned} \tag{E1}$$

The mean $\nu_i^{(t,\pm_j)}$ *and variance* $(\sigma_i^{(t,\pm_j)})^2$ *of the estimated results* $\bar{Y}_i^{(t,\pm_j)}$ *are*

$$\begin{aligned} \nu^{(t,\pm_j)} &= (1-\tilde{p})\hat{Y}_i^{(t,\pm_j)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D}, \\ (\sigma_i^{(t,\pm_j)})^2 &= \frac{-(1-\tilde{p})^2(\hat{Y}_i^{(t,\pm_j)})^2 + (1-\tilde{p})\{1-2\tilde{p}[\mathrm{Tr}(\Pi)/D]\}\hat{Y}_i^{(t,\pm_j)} + \tilde{p}[\mathrm{Tr}(\Pi)/D] - \tilde{p}^2\{[\mathrm{Tr}(\Pi)]^2/D^2\}}{K}. \end{aligned} \tag{E2}$$

Proof of Theorem 4. We now utilize the established relations as shown in Lemma 5 to obtain the relation between the estimated and the analytic gradients. Recall that, at the tth iteration, given the input $\mathcal{B}_i$ and K measurements, the estimated gradient for the jth parameter $\boldsymbol{\theta}_j$ of noisy QNN is

$$\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (\bar{Y}_i^{(t)} - Y_i)\left(\bar{Y}_i^{(t,+_j)} - \bar{Y}_i^{(t,-_j)}\right) + \lambda \boldsymbol{\theta}_j^{(t)}. \tag{E3}$$

Combining Lemma 5 and Eq. (E3), the term $\Delta_i^{(t,j)} := \bar{Y}_i^{(t,+_j)} - \bar{Y}_i^{(t,-_j)}$ in Eq. (E3) can be treated as the difference of two random variables. The term $(\bar{Y}_i^{(t)} - Y_i)$ in Eq. (E3) can also be treated as a random variables. We now separately investigate their moment properties.

The term $\Delta_i^{(t,j)}$. Following the notations used in Lemma 5, the mean and variance of the term $\Delta_i^{(t,j)}$ are $\nu_i^{(t,+_j)} - \nu_i^{(t,-_j)}$ and $(\sigma_i^{(t,j)})^2 = (\sigma_i^{(t,+_j)})^2 + (\sigma_i^{(t,-_j)})^2$, supported by the definition of moments and the independent relation between $\bar{Y}_i^{(t,+_j)}$ and $\bar{Y}_i^{(t,-_j)}$.

By leveraging the explicit form of $\nu_i^{(t,\pm_j)}$, the random variable $\Delta_i^{(t,j)}$ can be rewritten as

$$\Delta_i^{(t,j)} = (1-\tilde{p})(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)}) + \xi^{(t,j)}, \tag{E4}$$

where $\xi^{(t,j)}$ is a random variable with zero mean and variance $(\sigma_i^{(t,j)})^2$.

The term $(\bar{Y}_i^{(t)} - Y_i)$. Following the notations used in Lemma 5, an equivalent representation of $(\bar{Y}_i^{(t)} - \bar{Y}_i^{(t)})$ is

$$(\bar{Y}_i^{(t)} - \bar{Y}_i^{(t)}) = (1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} + \xi^{(t)} - \bar{Y}_i^{(t)}, \tag{E5}$$

where $\xi^{(t)}$ is a random variable with zero mean and variance $(\sigma_i^{(t)})^2$.

The reformulated terms as shown in Eq. (E4) and Eq. (E5) indicate that the estimated result $\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ can be rewritten as

$$\begin{aligned}
\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) &= (\bar{Y}_i^{(t)} - Y_i)(\bar{Y}_i^{(t,+_j)} - \bar{Y}_i^{(t,-_j)}) + \lambda\boldsymbol{\theta}_j^{(t)} \\
&= \left((1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right)(1-\tilde{p})(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)}) \\
&\quad + \left((1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right)\xi^{(t,j)} \\
&\quad + (1-\tilde{p})(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)})\xi^{(t)} + \xi^{(t)}\xi^{(t,j)} + \lambda\boldsymbol{\theta}_j^{(t)} \\
&= (1-\tilde{p})^2\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + (1-\tilde{p})\tilde{p}\left(\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right)(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)}) + (2\tilde{p} - \tilde{p}^2)\lambda\boldsymbol{\theta}_j^{(t)} \\
&\quad + (1-\tilde{p})(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)})\xi^{(t)} + \left((1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right)\xi^{(t,j)} + \xi^{(t)}\xi^{(t,j)}.
\end{aligned} \tag{E6}$$

Combining the above equation and the explicit expression of $\xi^{(t)}$ and $\xi^{(t,j)}$, we obtain the relation between the estimated and the analytic gradients. Specifically, the estimated gradient can be formulated as

$$\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (1-\tilde{p})^2\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + \boldsymbol{\varsigma}_i^{(t,j)},$$

where $\boldsymbol{\varsigma}_i^{(t,j)} = C_{j,2}^{(i,t)}\xi_i^{(t)} + C_{j,3}^{(i,t)}\xi_i^{(t,j)} + \xi^{(t)}\xi_i^{(t,j)}$, the first three constants $\{C_{j,1}^{(i,t)}\}_{i=1}^3$ are defined as

$$C_{j,a}^{(i,t)} = \begin{cases} (1-\tilde{p})\tilde{p}\left(\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right)(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)}) + (2\tilde{p} - \tilde{p}^2)\lambda\boldsymbol{\theta}_j^{(t)}, & a = 1 \\ (1-\tilde{p})(\hat{Y}_i^{(t,+_j)} - \hat{Y}_i^{(t,-_j)}), & a = 2 \\ \left((1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right), & a = 3, \end{cases}$$

and the last two constants, which separately correspond to the variance $(\sigma_i^{(t)})^2$ and $(\sigma_i^{(t,j)})^2$ of the random variables $\xi_i^{(t)}$ and $\xi_i^{(t,j)}$, are

$$C_{j,a}^{(i,t)} = \begin{cases} \frac{-(1-\tilde{p})^2(\hat{Y}_i^{(t)})^2 + (1-\tilde{p})\{1-2\tilde{p}[\mathrm{Tr}(\Pi)D]\}\hat{Y}_i^{(t)} + \tilde{p}[\mathrm{Tr}(\Pi)/D] - \tilde{p}^2\{[\mathrm{Tr}(\Pi)]^2/D^2\}}{K}, & a = 4 \\ \frac{-(1-\tilde{p})^2[(\hat{Y}_i^{(t,+_j)})^2 + (\hat{Y}_i^{(t,-_j)})^2] + (1-\tilde{p})\{1-2\tilde{p}[\mathrm{Tr}(\Pi)/D]\}(\hat{Y}_i^{(t,+_j)} + \hat{Y}_i^{(t,-_j)}) + 2\tilde{p}[\mathrm{Tr}(\Pi)/D] - 2\tilde{p}^2\{[\mathrm{Tr}(\Pi)]^2/D^2\}}{K}, & a = 5. \end{cases}$$

■

1. Proof of Lemma 5

To achieve Lemma 5, we first simplify the learning model of QNN with the depolarization noise. In particular, all noisy channels $\mathcal{N}_p$, which are separately applied to each quantum circuit depth, can be merged together to a specific circuit depth and presented by a new depolarization channel $\mathcal{N}_{\tilde{p}}$.

Lemma 6: *Let $\mathcal{N}_p$ be the depolarization channel. There always exists a depolarization channel $\mathcal{N}_{\tilde{p}}$ with $\tilde{p} = 1 - (1-p)^{L_Q}$ that satisfies $\mathcal{N}_p\{U_{L_Q}(\boldsymbol{\theta})\cdots U_2(\boldsymbol{\theta})\mathcal{N}_p[U_1(\boldsymbol{\theta})\rho U_1(\boldsymbol{\theta})^\dagger]U_2(\boldsymbol{\theta})^\dagger\cdots U_{L_Q}(\boldsymbol{\theta})^\dagger\} = \mathcal{N}_{\tilde{p}}[U(\boldsymbol{\theta})\rho U(\boldsymbol{\theta})^\dagger]$, where ρ is the input quantum state.*

Proof of Lemma 6. Denote $\rho^{(k)}$ as $\rho^{(k)} = \prod_{l=1}^{k} U_l(\boldsymbol{\theta})\rho U_l(\boldsymbol{\theta})^\dagger$. Applying $\mathcal{N}_p$ to $\rho^{(1)}$ gives

$$\mathcal{N}_p(\rho^{(1)}) = (1-p)\rho^{(1)} + p\frac{\mathbb{I}_D}{D}, \tag{E7}$$

where D refers to the dimensions of Hilbert space interacted with $\mathcal{N}_p$.

Supporting by the above equation, applying $U_2(\boldsymbol{\theta})$ to the state $\mathcal{N}_p(\rho^{(1)})$ gives

$$U_2(\boldsymbol{\theta})\mathcal{N}_p(\rho^{(1)})U_2(\boldsymbol{\theta})^\dagger = (1-p)\rho^{(2)} + p\frac{\mathbb{I}_D}{D}. \tag{E8}$$

Then interacting $\mathcal{N}_p$ with the state $U_2(\boldsymbol{\theta})\mathcal{N}_p(\rho^{(1)})U_2(\boldsymbol{\theta})^\dagger$ gives

$$\begin{aligned}
&\mathcal{N}_p[U_2(\boldsymbol{\theta})\mathcal{N}_p(\rho^{(1)})U_2(\boldsymbol{\theta})^\dagger] \\
&\quad = (1-p)^2\rho^{(2)} + (1-p)p\frac{\mathbb{I}_D}{D} + p\frac{\mathbb{I}_D}{D} \\
&\quad = (1-p)^2\rho^{(2)} + [1-(1-p)^2]\frac{\mathbb{I}_D}{D}.
\end{aligned} \tag{E9}$$

By induction, suppose at kth step, the generated state is

$$\rho^{(k)} = (1-p)^l\rho^{(k)} + [1-(1-p)^k]\frac{\mathbb{I}_D}{D}. \tag{E10}$$

Then applying $U_{k+1}(\boldsymbol{\theta})$ followed by $\mathcal{N}_p$ gives

$$\begin{aligned}
\rho^{(k+1)} &= \mathcal{N}_p\left(U_{k+1}(\boldsymbol{\theta})\rho^{(k)}U_{k+1}(\boldsymbol{\theta})^\dagger\right) \\
&= (1-p)^{k+1}\rho^{(k+1)} + [1-(1-p)^{k+1}]\frac{\mathbb{I}_D}{D}.
\end{aligned} \tag{E11}$$

According to the formula of the depolarization channel, an immediate observation is that the noisy QNN is equivalent to applying a single depolarization channel $\mathcal{N}_{\tilde{p}}$ at the last circuit depth L_Q, i.e.,

$$\mathcal{N}_{\tilde{p}}(\rho) = (1-p)^{L_Q}\rho^{(L_Q)} + [1-(1-p)^{L_Q}]\frac{\mathbb{I}}{D}, \tag{E12}$$

where

$$\tilde{p} = 1-(1-p)^{L_Q}. \tag{E13}$$

■

Proof of Lemma 5. We now use the simplified QNN given by Lemma 6 to explore the relation between the generated statistic $\bar{Y}_i^{(t)}$ and the expectation value $\hat{Y}^{(t)}$ (the same rule applies to connect $\bar{Y}_i^{(t,\pm_j)}$ with $\hat{Y}^{(t,\pm_j)}$).

At the tth iteration, given the tunable parameters $\boldsymbol{\theta}^{(t)}$ and inputs $\mathcal{B}_i$, the ensemble corresponding to the generated state of QNN before taking quantum measurements is $\{p_l, \gamma_{i,l}^{(t)}\}_{l=1}^2$, i.e., $p_1 = 1-\tilde{p}$ with $\gamma_{i,1}^{(t)} = U(\boldsymbol{\theta}^{(t)})\rho_{\mathcal{B}_i}U(\boldsymbol{\theta}^{(t)})^\dagger$ and $p_2 = \tilde{p}$ with $\gamma_{i,2}^{(t)} = \mathbb{I}_D/D$. After applying a two-outcome POVM Π to measure such an ensemble K times, the generated statistics (sample mean) is $\bar{Y}_i^{(t)} = (1/K)\sum_{k=1}^{K} V_k^{(t)}$, where each measured outcome $V_k^{(t)}$ with $k \in [K]$ is a random variable that satisfies Fact 1. ■

Fact 1: *$V_k^{(t)}$ is a random variable that follows the distribution $\mathcal{P}_{Q'}(V_k^{(t)}) = \sum_{c=1}^{2}\Pr(z=c)\Pr(V_k^{(t)}|z=c)$. The explicit formula of $\mathcal{P}_{Q'}$ is as follows.*

1. $\Pr(z=1) = 1-\tilde{p}$ *with* $V_k^{(t)}|z=1 \sim \mathrm{Ber}(\hat{Y}_i^{(t)})$ *and* $\hat{Y}_i^{(t)} = \mathrm{Tr}(\Pi\gamma_{i,1}^{(t)})$*;*
2. $\Pr(z=2) = \tilde{p}$ *with* $V_k^{(t)}|z=2 \sim \mathrm{Ber}[\mathrm{Tr}(\Pi)/D]$ *with* $\mathrm{Tr}(\Pi)/D = \mathrm{Tr}(\Pi\gamma_{i,2}^{(t)})$.

Fact 1 implies that the mean and variance of $V_k^{(t)}$ are

$$\begin{aligned}
&(1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} \quad \text{and} \quad -(1-\tilde{p})^2(\hat{Y}_i^{(t)})^2 \\
&\quad + (1-\tilde{p})\left(1-2\tilde{p}\frac{\mathrm{Tr}(\Pi)}{D}\right)\hat{Y}_i^{(t)} \\
&\quad + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - \tilde{p}^2\frac{[\mathrm{Tr}(\Pi)]^2}{D^2},
\end{aligned}$$

respectively. Moreover, since each outcome $V_k^{(t)}$ follows the distribution $\mathcal{P}_{Q'}$, the mean $\nu_i^{(t)}$ and the variance $(\sigma_i^{(t)})^2$ of the sample mean $\bar{Y}_i^{(t)}$ are

$$\begin{aligned}
\nu^{(t)} &= (1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D}, \\
(\sigma_i^{(t)})^2 &= \frac{-(1-\tilde{p})^2(\hat{Y}_i^{(t)})^2 + (1-\tilde{p})\{1-2\tilde{p}[\mathrm{Tr}(\Pi)/D]\}\hat{Y}_i^{(t)} + \tilde{p}[\mathrm{Tr}(\Pi)/D] - \tilde{p}^2\{[\mathrm{Tr}(\Pi)]^2/D^2\}}{K}.
\end{aligned} \tag{E14}$$

Following the same routine, the mean $\nu_i^{(t,\pm_j)}$ and the variance $(\sigma_i^{(t,\pm_j)})^2$ of the sample mean $\bar{Y}_i^{(t,\pm_j)}$ satisfy

$$\begin{aligned}\nu^{(t,\pm_j)} &= (1-\tilde{p})\hat{Y}_i^{(t,\pm_j)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D},\\ (\sigma_i^{(t,\pm_j)})^2 &= \frac{-(1-\tilde{p})^2(\hat{Y}_i^{(t,\pm_j)})^2 + (1-\tilde{p})\{1-2\tilde{p}[\mathrm{Tr}(\Pi)/D]\}\hat{Y}_i^{(t,\pm_j)} + \tilde{p}[\mathrm{Tr}(\Pi)/D] - \tilde{p}^2\{[\mathrm{Tr}(\Pi)]^2/D^2\}}{K}.\end{aligned} \tag{E15}$$

APPENDIX F: PROOF OF THEOREM 1

Theorem 1 quantifies the utility bounds R_1 and R_2 of QNN under the depolarization noise towards ERM framework. For ease of illustration, we restate Theorem 1 below.

Theorem 5 (Restate of Theorem 1): *QNN outputs $\boldsymbol{\theta}^{(T)} \in \mathbb{R}^d$ after T iterations with utility bounds $R_1 \leq \tilde{O}\{poly[d/T(1-p)^{L_Q}, d/BK(1-p)^{L_Q}, d/(1-p)^{L_Q}]\}$ and $R_2 \leq \tilde{O}\{poly[d, 1/K^2B, 1/(1-p)^{L_Q}]\}$, where K is the number of quantum measurements, L_Q is the quantum circuit depth, p is the gate noise, and B is the number of batches.*

The high level idea to achieve the utility bounds R_1 and R_2 is as follows. Recall that R_1 measures how far the trainable parameter of QNN is away from the stationary point. A well-known result in optimization theory [61] is that when a function satisfies the smooth property, its stationary point can be efficiently located by a simple gradient-based algorithm. By leveraging this observation and the relation between the estimated and analytic gradients as achieved in Theorem 4, we can quantify how the estimated gradients of QNN converge to the stationary point, which corresponds to the utility bound R_1.

Recall that the utility bound R_2 evaluates the disparity between the expected empirical risk and the optimal risk that is determined by the global minimum. To achieve R_2, we utilize the result of the PL condition [63]. Concretely, the PL condition is a sufficient condition for the gradient-descent methods to achieve a linear convergence rate towards the optimal solution, i.e., if a nonconvex function satisfies the PL condition [62], every stationary point of such a function is the global minimum [62,63]. In other words, the PL condition connects stationary points with the global minimum. Meanwhile, compared with other conditions such as the strong convexity and the restricted strong convexity to achieve the linear convergence towards the global minimum, the PL condition is much easier to satisfy by QNN with a mild technical assumption, as proved in Lemma 1. Therefore, by leveraging the result of R_1, which measures how far the optimized loss of QNN is away from a stationary point and the result of Lemma 1 that QNN satisfies the PL condition, we can effectively obtain the utility bound R_2 for QNN.

Proof of Theorem 5. We employ the following two theorems to achieve Theorem 5, whose proofs are given in Secs. 1 and 2, respectively. ■

Theorem 6: *Given the dataset $\boldsymbol{z}$, QNN outputs $\boldsymbol{\theta}^{(T)}$ after T iterations with utility bound*

$$R_1 \leq \frac{2S(1+90\lambda d)}{T(1-\tilde{p})^2} + \frac{(2\tilde{p}-\tilde{p}^2)(2G+d)(1+10\lambda)^2}{(1-\tilde{p})^2} + \frac{6dK+8d}{(1-\tilde{p})^2BK^2}.$$

Theorem 7: *Given the dataset $\boldsymbol{z}$, QNN outputs $\boldsymbol{\theta}^{(T)}$ after T iterations with utility bound*

$$R_2 \leq (1+90\lambda d)\exp\left(-\frac{\mu(1-\tilde{p})^2T}{S}\right) + T\frac{(2\tilde{p}-\tilde{p}^2)(G+2d)(1+10\lambda)^2BK^2+6dK+8d}{2SBK^2}.$$

As for R_1, with setting $T \leftarrow \infty$ and after the simplification, the utility bound as shown in Theorem 6 follows

$$R_1 \leq \tilde{O}\left[\mathrm{poly}\left(\frac{d}{T(1-p)^{L_Q}}, \frac{d}{BK(1-p)^{L_Q}}, \frac{d}{(1-p)^{L_Q}}\right)\right]. \tag{F1}$$

As for R_2, with setting $T = \mathcal{O}\{[S/\mu(1-\tilde{p})^2]\ln[(1+90\lambda d)2SBK^2/(2\tilde{p}-\tilde{p}^2)(G+2d)(1+10\lambda)^2BK^2+6dK+8d]\}$ and after simplification, the utility bound as shown in Theorem 7 follows

$$R_2 \leq \tilde{O}\left[\mathrm{poly}\left(d, \frac{1}{K^2B}, \frac{1}{(1-p)^{L_Q}}\right)\right]. \tag{F2}$$

1. Proof of Theorem 6: The utility bound R_1

The proof of Theorem 6 employs the following lemma, where its proof is given in Sec. 3.

Lemma 7: *Taking expectation over the randomness of* $\xi_i^{(t)}$ *and* $\xi_i^{(t,j)}$ *in the estimated gradient* $\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})$ *as formulated in Theorem 4, the term* $(1/2S)\sum_{j=1}^{d} \mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}} \left\{ \left[\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)}) \right]^2 \right\}$ *with* S *being the smooth parameter is upper bounded by*

$$\frac{(1-\tilde{p})^4}{2S}\|\nabla \mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{(1-\tilde{p})^2 G}{2S}\max_{i,j} C_{j,1}^{(i,t)} + \frac{d}{2S}\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2 + \frac{6dK+8d}{2SBK^2}.$$

Proof of Theorem 6. Recall that the optimization rule of noisy QNN at the tth iteration follows

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)}). \tag{F3}$$

Since the objective function $\mathcal{L}(\boldsymbol{\theta})$ is S smooth, as indicated in Lemma 1, we have

$$\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)}) \leq \langle \nabla \mathcal{L}(\boldsymbol{\theta}^{(t)}), \boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)} \rangle + \frac{S}{2}\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\|^2. \tag{F4}$$

Combine the above two equations and setting $\eta = 1/S$, we have

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)}) &\leq \langle \nabla \mathcal{L}(\boldsymbol{\theta}^{(t)}), \boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)} \rangle + \frac{S}{2}\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\|^2 \\
&= -\frac{1}{S}\langle \nabla \mathcal{L}(\boldsymbol{\theta}^{(t)}), \nabla \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)}) \rangle + \frac{1}{2S}\|\nabla \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\|^2 \\
&= -\frac{1}{S}\sum_{j=1}^{d}\left(\nabla_j \mathcal{L}(\boldsymbol{\theta}^{(t)}) \nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right) + \frac{1}{2S}\sum_{j=1}^{d}\left(\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2.
\end{aligned} \tag{F5}$$

Recall the definition of the estimated gradient is $\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)}) = (1/B)\sum_{i=1}^{B} \nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ and the explicit expression of $\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ is

$$\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (1-\tilde{p})^2 \nabla_j \mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + C_{j,2}^{(i,t)}\xi^{(t)} + C_{j,3}^{(i,t)}\xi_i^{(t,j)} + \xi_i^{(t)}\xi_i^{(t,j)}.$$

Alternatively, the gradient for the j th parameter $\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})$ follows

$$\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)}) = \frac{1}{B}\sum_{i=1}^{B}(1-\tilde{p})^2 \nabla_j \mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + C_{j,2}^{(i,t)}\xi_i^{(t)} + C_{j,3}^{(i,t)}\xi_i^{(t,j)} + \xi_i^{(t)}\xi^{(t,j)}. \tag{F6}$$

Combining Eq. (F5) with Eq. (F6) and taking expectation over $\xi_i^{(t)}$ and $\xi_i^{(t,j)}$, we obtain

$$\begin{aligned}
\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})] \leq & -\frac{1}{S}(1-\tilde{p})^2\|\nabla \mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 - \frac{1}{S}\sum_{j=1}^{d}\nabla_j \mathcal{L}(\boldsymbol{\theta}^{(t)})\left(\frac{1}{B}\sum_{i=1}^{B} C_{j,1}^{(i,t)}\right) \\
& - \frac{1}{S}\sum_{j=1}^{d}\nabla_j \mathcal{L}(\boldsymbol{\theta}^{(t)})\frac{1}{B}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)}}\left[C_{j,2}^{(i,t)}\xi_i^{(t)}\right] - \frac{1}{S}\sum_{j=1}^{d}\nabla_j \mathcal{L}(\boldsymbol{\theta}^{(t)})\frac{1}{B}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t,j)}}\left[C_{j,3}^{(i,t)}\xi_i^{(t,j)}\right] \\
& - \frac{1}{S}\sum_{j=1}^{d}\nabla_j \mathcal{L}(\boldsymbol{\theta}^{(t)})\frac{1}{B}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\xi_i^{(t)}\xi_i^{(t,j)}\right] + \frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right] \\
& - \frac{1}{S}(1-\tilde{p})^2\|\nabla \mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{G}{2S}\max_{i,j} C_{j,1}^{(i,t)} + \frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right].
\end{aligned} \tag{F7}$$

The first inequality uses the result of Eq. (F6). The second inequality uses $\mathbb{E}[\xi_i^{(t)}] = 0$, $\mathbb{E}[\xi_i^{(t,j)}] = 0$ as shown in Theorem 4, and $-G/d \leq \nabla_j \mathcal{L}(\boldsymbol{\theta}^{(t)}) \leq G/d$ supported by the G-Lipschitz property.

By leveraging Lemma 7, Eq. (F7) can be further simplified as

$$\begin{aligned}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})-\mathcal{L}(\boldsymbol{\theta}^{(t)})] &\le -\frac{1}{S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2+\frac{G}{2S}\max_{i,j}C_{j,1}^{(i,t)}+\frac{(1-\tilde{p})^4}{2SB}\|\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2\\ &\quad+\frac{(1-\tilde{p})^2G}{2S}\max_{i,j}C_{j,1}^{(i,t)}+\frac{d}{2S}\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2+\frac{6dK+8d}{2SBK^2}\\ &\le -\frac{1}{2S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2+\frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2+\frac{6dK+8d}{2SBK^2}.\end{aligned}\tag{F8}$$

The first inequalities comes from Lemma 7, and the second inequality employs $(1-\tilde{p})^4/2SB \le (1-\tilde{p})^2/2S$ and the following result:

$$\begin{aligned}&\frac{G}{2S}\max_{i,j}C_{j,1}^{(i,t)}+\frac{(1-\tilde{p})^2G}{2S}\max_{i,j}C_{j,1}^{(i,t)}+\frac{d}{2S}\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2\\ &\le\frac{[1+(1-\tilde{p})^2]G}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)+\frac{d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2\\ &\le\frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2,\end{aligned}\tag{F9}$$

where the first inequality uses the upper bound of $C_{j,1}^{(i,t)}$ and $(C_{j,1}^{(i,t)})^2$, i.e., $\max_{i,j}C_{j,1}^{(i,t)} \le (1-\tilde{p})\tilde{p}+10(2-\tilde{p})\tilde{p}\lambda \le (2-\tilde{p})\tilde{p}(1+10\lambda)$ and $\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2 \le [(2-\tilde{p})\tilde{p}(1+10\lambda)]^2 \le (2-\tilde{p})\tilde{p}(1+10\lambda)^2$, and the second inequality uses $(1-\tilde{p})^2 \le 1$.

An equivalent representation of Eq. (F8) is

$$\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 \le 2S\frac{\mathcal{L}(\boldsymbol{\theta}^{(t)})-\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})]}{(1-\tilde{p})^2}+\frac{(2\tilde{p}-\tilde{p}^2)(2G+d)(1+10\lambda)^2}{(1-\tilde{p})^2}+\frac{6dK+8d}{(1-\tilde{p})^2BK^2}.\tag{F10}$$

By induction, with summing over $t=0,\ldots,T-1$ and taking expectation of Eq. (F10), we obtain

$$\begin{aligned}\mathbb{E}_t\left[\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2\right] &\le 2S\frac{\mathcal{L}(\boldsymbol{\theta}^{(0)})-\mathbb{E}_{\xi_i^{(T)},\xi_i^{(T,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(T)})]}{T(1-\tilde{p})^2}\\ &\quad+\frac{(2\tilde{p}-\tilde{p}^2)(2G+d)(1+10\lambda)^2}{(1-\tilde{p})^2}+\frac{6dK+8d}{(1-\tilde{p})^2BK^2}\\ &\le\frac{2S+2S\lambda d(3\pi)^2}{T(1-\tilde{p})^2}+\frac{(2\tilde{p}-\tilde{p}^2)(2G+d)(1+10\lambda)^2}{(1-\tilde{p})^2}+\frac{6dK+8d}{(1-\tilde{p})^2BK^2}\\ &\le\frac{2S(1+90\lambda d)}{T(1-\tilde{p})^2}+\frac{(2\tilde{p}-\tilde{p}^2)(2G+d)(1+10\lambda)^2}{(1-\tilde{p})^2}+\frac{6dK+8d}{(1-\tilde{p})^2BK^2},\end{aligned}\tag{F11}$$

where the second inequality uses $\mathcal{L}(\boldsymbol{\theta}^{(0)})-\mathbb{E}_{\xi_i^{(T)},\xi_i^{(T,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(T)})] \le \mathcal{L}(\boldsymbol{\theta}^{(0)})-\mathcal{L}^*$, $\mathcal{L}^*>0$ and $\mathcal{L}(\boldsymbol{\theta}^{(0)}) \le 1+\lambda d(3\pi)^2$. ■

2. Proof of Theorem 7: The utility bound R_2

Proof of Theorem 7. The proof of Theorem 7 is similar with that of Theorem 6. In particular, following the same routine, we obtain the result of Eq. (F8), i.e.,

$$\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})-\mathcal{L}(\boldsymbol{\theta}^{(t)})] \le -\frac{1}{2S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2+\frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2+\frac{6dK+8d}{2SBK^2}.\tag{F12}$$

Then, we call the conclusion of the PL condition as formulated in Lemma 1 and acquire

$$\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})] \leq -\frac{\mu(1-\tilde{p})^2}{S}[\mathcal{L}(\boldsymbol{\theta}^{(t)}) - \mathcal{L}^*] + \frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2 + \frac{6dK+8d}{2SBK^2}. \tag{F13}$$

An equivalent reformulation of Eq. (F13) is

$$\mathbb{E}_{\varsigma^{(t)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})] - \mathcal{L}^* \leq \left(1 - \frac{\mu(1-\tilde{p})^2}{S}\right)[\mathcal{L}(\boldsymbol{\theta}^{(t)}) - \mathcal{L}^*] + \frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2 + \frac{6dK+8d}{2SBK^2}. \tag{F14}$$

By induction, with summing over $t = 0, \ldots, T$ and taking expectation, we obtain

$$\begin{aligned}\mathbb{E}_{\varsigma^{(t)}}[\mathcal{L}(\boldsymbol{\theta}^{(T)})] - \mathcal{L}^* &\leq \left(1 - \frac{\mu(1-\tilde{p})^2}{S}\right)^T [\mathcal{L}(\boldsymbol{\theta}^{(0)}) - \mathcal{L}^*] + T\frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2 + T\frac{6dK+8d}{2SBK^2} \\ &\leq (1+90\lambda d)\exp\left(-\frac{\mu(1-\tilde{p})^2 T}{S}\right) + T\frac{(2\tilde{p}-\tilde{p}^2)(G+2d)(1+10\lambda)^2 BK^2 + 6dK + 8d}{2SBK^2}, \end{aligned} \tag{F15}$$

where the second inequality uses $\mathcal{L}(\boldsymbol{\theta}^{(0)}) - \mathcal{L}^* \leq 1 + 90\lambda d$ and $1 + x \leq e^x$ for all real x. ■

3. Proof of Lemma 7

Proof of Lemma 7. As shown in Theorem 4, the explicit formula of the estimated gradient is

$$\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)}) = \frac{1}{B}\sum_{i=1}^{B}(1-\tilde{p})^2 \nabla_j \mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + C_{j,2}^{(i,t)}\xi_i^{(t)} + C_{j,3}^{(i,t)}\xi_i^{(t,j)} + \xi_i^{(t)}\xi^{(t,j)}. \tag{F16}$$

By using the above result, we obtain

$$\begin{aligned}&\frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right] \\ &\quad\leq \frac{(1-\tilde{p})^4}{2S}\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{(1-\tilde{p})^2}{2SB}\sum_{j=1}^{d}\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\left(\sum_{i=1}^{B}C_{j,1}^{(i,t)}\right) \\ &\qquad+ \frac{(1-\tilde{p})^2}{SB}\sum_{j=1}^{d}\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)}}[\xi_i^{(t)}] \\ &\qquad+ \frac{(1-\tilde{p})^2}{SB}\sum_{j=1}^{d}\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t,j)}}[\xi_i^{(t,j)}] + \frac{(1-\tilde{p})^2}{SB}\sum_{j=1}^{d}\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)}\xi_i^{(t,j)}}[\xi_i^{(t)}\xi_i^{(t,j)}] \\ &\qquad+ \frac{d}{2SB^2}\left(\sum_{i=1}^{B}C_{j,1}^{(i,t)}\right)^2 + \frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)}}[\xi_i^{(t)}] + \frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t,j)}}[\xi_i^{(t,j)}] + \frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\xi_i^{(t)}\xi_i^{(t,j)}] \\ &\qquad+ \frac{1}{2SB^2}\sum_{j=1}^{d}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)}}[(\xi_i^{(t)})^2] + \frac{1}{SB^2}\sum_{j=1}^{d}\sum_{i=1}^{B}\left(\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\xi_i^{(t)}\xi_i^{(t,j)}] + \mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[(\xi_i^{(t)})^2\xi_i^{(t,j)}]\right) \\ &\qquad+ \frac{1}{2SB^2}\sum_{j=1}^{d}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t,j)}}[(\xi_i^{(t,j)})^2] + \frac{1}{SB^2}\sum_{j=1}^{d}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\xi_i^{(t)}(\xi_i^{(t,j)})^2] + \frac{1}{2SB^2}\sum_{j=1}^{d}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)}\xi_i^{(t,j)}}[(\xi_i^{(t)})^2(\xi_i^{(t,j)})^2]\end{aligned}$$

$$\leq \frac{(1-\tilde{p})^4}{2S}\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{(1-\tilde{p})^2 G}{2S}\max_{i,j} C_{j,1}^{(i,t)} + \frac{d}{2S}\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2 + \frac{dC_{j,4,\max}^{(t)}}{2SB} + \frac{dC_{j,5,\max}^{(t,j)}}{2SB} + \frac{dC_{j,4,\max}^{(t)} C_{j,5,\max}^{(t,j)}}{2SB}. \tag{F17}$$

The first and second inequalities uses $C_{j,2}^{(i,t)} \leq 1$, $C_{j,3}^{(i,t)} \leq 1$, $\mathbb{E}[\xi_i^{(t)}] = 0$, $\mathbb{E}[\xi_i^{(t,j)}] = 0$, and $-G/d \leq \nabla_j \mathcal{L}(\boldsymbol{\theta}^{(t)}) \leq G/d$ supported by the G-Lipschitz property. The term $C_{j,4,\max}^{(t)}$ refers to $C_{j,4,\max}^{(t)} = \max_i C_{j,4}^{(i,t)}$. Similarly, the term $C_{j,5,\max}^{(t,j)}$ refers to $C_{j,5,\max}^{(t,j)} = \max_i C_{j,5}^{(i,t)}$.

Since Theorem 4 indicates that

$$C_{j,4,\max}^{(t)} \leq \frac{(1-\tilde{p})\{1-2\tilde{p}[\mathrm{Tr}(\Pi)/D]\}}{K} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{DK} \leq \frac{2}{K},$$

and

$$C_{j,5,\max}^{(t,j)} \leq \frac{(1-\tilde{p})\{1-2\tilde{p}[\mathrm{Tr}(\Pi)/D]\}(\hat{Y}_i^{(t,+_j)} + \hat{Y}_i^{(t,-_j)}) + 2\tilde{p}[\mathrm{Tr}(\Pi)/D]}{K} \leq \frac{4}{K},$$

we obtain

$$\frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right] \leq \frac{(1-\tilde{p})^4}{2S}\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{(1-\tilde{p})^2 G}{2S}\max_{i,j} C_{j,1}^{(i,t)} + \frac{d}{2S}\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2 + \frac{6dK+8d}{2SBK^2}. \tag{F18}$$

■

APPENDIX G: MORE NUMERICAL SIMULATION DETAILS

1. The construction of QNNs

The implementation of the data encoding circuit $U_{\mathbf{x}}$ and the trainable unitary $U(\boldsymbol{\theta})$ follows the proposal [17]. In particular, the data encoding circuit $U_{\mathbf{x}}$ uses the kernel encoding method, and the architecture of the trainable unitary $U(\boldsymbol{\theta})$ follows the multilayer structure. The right panel of Fig. 2 illustrates the implementation of data encoding circuit and the trainable circuit used in QNN. Three qubits are employed to build two such circuits. The data encoding circuit $U_{\mathbf{x}}$ is composed of Hadamard gates $H = \frac{1}{\sqrt{2}}\left(\begin{smallmatrix}1 & 1\\ 1 & -1\end{smallmatrix}\right)$, R_Y gates with $R_Y(2a) = \left(\begin{smallmatrix}\cos(a) & -\sin(a)\\ \sin(a) & \cos(a)\end{smallmatrix}\right)$, and controlled-$R_Y$ gates with $\mathrm{CRY}(2a) = |0\rangle\langle 0| \otimes \mathbb{I}_2 + |1\rangle\langle 1| \otimes R_Y(2a)$. Specifically, the rotation angle in $R_Y(\mathbf{x})$ is $(\pi - \mathbf{x}_{i,1})(\pi - \mathbf{x}_{i,2})(\pi - \mathbf{x}_{i,3})$. The construction of the trainable circuit $U(\boldsymbol{\theta})$ uses R_Y gates and controlled-NOT gates $CX = |0\rangle\langle 0| \otimes \mathbb{I}_2 + |1\rangle\langle 1| \otimes X$ with $X = \left(\begin{smallmatrix}0 & 1\\ 1 & 0\end{smallmatrix}\right)$.

The implementation of FCNN is exhibited in Fig. 4. The employed FCNN consists of one input layer, one hidden layer, and one output layer with dimensions 3, 3, 2, respectively. The softmax function is applied to the output layer for normalization.

The hyperparameter settings adopted in the numerical simulations are as follows. For QNNs, the learning rate η for all simulations is set as 2. For FCNN, the learning rate η for all simulations is set as 0.1. The batch size in each updating for both QNNs and FCNN is set as 280.

2. More simulation results for the utility R_1

We conduct additional numerical simulations to quantitively investigate whether the exponential dependence on the circuit depth L and the inverse dependence on the number of measurements K claimed in Theorem 1 can be observed in the binary classification task introduced in the paper. The appended numerical simulations mainly follow the setup introduced in the original submission. Particularly, a three-qubit QNN with the hardware-efficient ansatz $U(\boldsymbol{\theta}) = \prod_{l=1}^{L} U_l(\boldsymbol{\theta})$ as shown in the upper left panel in Fig. 2 is employed to classify the preprocessed handwritten

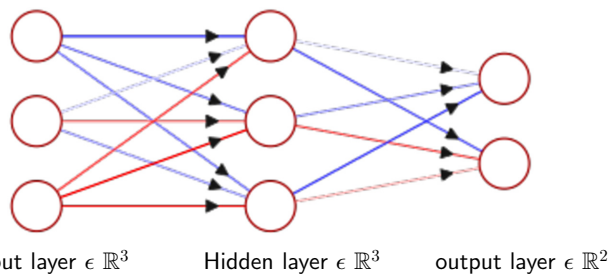

FIG. 4. The implementation of FCNN used in numerical simulations.

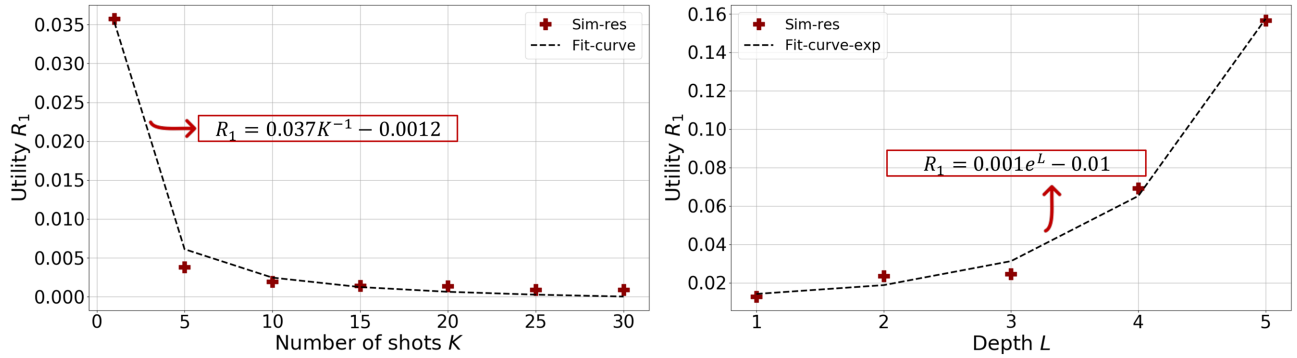

FIG. 5. The left panel depicts how the utility R_1 scales with the number of shots K. The label "Sim-res" refers to the collected simulation results with the varied settings. The label "Fit-curve" refers to the fitting curve, i.e., the mathematical form is $aK^{-1} + b$ with $a, b \in \mathbb{R}$, with respect to the collected results. The right panel depicts how the utility R_1 scales with L. The label "Sim-res" has the same meaning as the left panel. The label "Fit-curve-exp" refers to the fitting curve for the exponential function, i.e., the mathematical form is $ae^K + b$ with $a, b \in \mathbb{R}$, with respect to the collected results.

digit dataset $\mathbf{z} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{360}$, where $\mathbf{x}_i \in \mathbb{R}^3$ is the ith data feature and $y_i \in \{0, 1\}$ is the ith label. The gate encoding method is adopted to load classical data $\mathbf{x}_i$ into quantum states.

To examine whether the utility bound R_1 *inversely scales* with the number of shots K, as claimed in Theorem 1, we proceed with the following numerical simulations. We fix the depth of trainable quantum circuits as $L = 3$ and the depolarization rate as $p = 0.1$. The total number of iterations is set as $T = 50$. Through varying the number of shots from $K = 1$ to $K = 30$, we calculate the utility R_1 at the last iteration. For computational efficiency, the size of the training dataset $\mathbf{z}_t$ and the test dataset $\mathbf{z}_p$ is 80 and 280, respectively. Each setting is repeated 5 times to collect the statistical results. The obtained results are exhibited in the left panel of Fig. 5. Through fitting the simulation results, we observe $R_1 \propto O(1/K)$, which accords with our theoretical results.

We last explore whether the utility bound R_1 *exponentially scales* with the circuit depth L. To do so, we fix the number of shots as $K = 20$ and the depolarization rate as $p = 0.2$. The total number of iterations is set as $T = 20$. Through varying the depth from $L = 1$ to $L = 5$, we calculate the utility R_1 at the last iteration. For computational efficiency, we set $|\mathbf{z}_t| = 5$. Each setting is repeated 5 times to collect the statistical results. The achieved results are exhibited in the left panel of Fig. 5. Through fitting the simulation results with both the exponential and polynomial functions, we observe $R_1 \propto O[\exp(L)]$, which accords with our theoretical results.

APPENDIX H: PROOF OF THEOREM 2

Proof of Theorem 2. Following Definition 1, we observe that the QSQ algorithm can be efficiently simulated by QNN once each query $\{\mathbb{M}_i, \tau_i\}_{i=1}^{Q}$ can be efficiently simulated by noisy QNN, i.e., given the query $\{\mathbb{M}_i, \tau_i\}$, the noisy QNN returns an estimated result α_i that is ε-close to $\nu = \langle\psi_{c^*}|\mathbb{M}|\psi_{c^*}\rangle$ by taking $O[\text{poly}(N)]$ copies of $|\psi_{c^*}\rangle$. In the following, we prove that each query to the QSQ oracle can be efficiently simulated by noisy QNN up to a polynomial overhead.

Without loss of generality, we set the tuple fed into the QSQ oracle as $\{\mathbb{M}, \tau\}$. Let $|\psi_{c^*}\rangle$ be the quantum example given in Definition 1. In this way, following notations in Theorem 2, the expectation value of quantum measurements for noisy QNN under the depolarization noise setting $\mathcal{N}_{\tilde{p}}$ yields $\tilde{\nu} = (1 - \tilde{p})\nu + [\tilde{p}\,\text{Tr}(\mathbb{M})/2^{N+1}]$ with $\nu = \langle\psi_{c^*}|\mathbb{M}|\psi_{c^*}\rangle$. In addition, the measurement outcome V_k is a random variable that satisfies $V_k \sim \text{Ber}(\tilde{\nu})$.

By the Chernoff-Hoeffding bound for real-valued variables, we obtain the relation between the sample mean $\tilde{Y} = (1/K)\sum_{k=1}^{K} V_k$ with K measurements and the target result $\tilde{\nu}$, i.e.,

$$\Pr\left(\left|\frac{1}{K}\sum_{i=1}^{K} V_k - \tilde{\nu}\right| \geq \frac{\delta}{2}\right) \leq 2\exp(-\delta^2 K/2). \tag{H1}$$

Denote $b = 2\exp(-\delta^2 K/2)$. Equation (H1) implies that when $K = 2\ln(2/b)/\delta^2$, with probability at least $1 - b$, we have $|(1/K)\sum_{i=1}^{K} V_k - \tilde{\nu}| \leq \delta/2$. Moreover, the distance between the result ν (i.e., the target value of the QSQ oracle) and the shifted expectation value $\tilde{\nu}$ follows

$$|\nu - \tilde{\nu}| \leq \tilde{p}\nu + \tilde{p}\frac{\text{Tr}(\mathbb{M})}{2^{N+1}}. \tag{H2}$$

In conjunction with the above two equations, we obtain that with probability at least $1-b$,

$$\left|\frac{1}{K}\sum_{k=1}^{K}V_k-\nu\right|=\left|\frac{1}{K}\sum_{k=1}^{K}V_k-\tilde{\nu}+\tilde{\nu}-\nu\right|\le\tilde{p}\,\nu+\tilde{p}\,\frac{\mathrm{Tr}(\mathbb{M})}{2^{N+1}}+\frac{\delta}{2}\le\tilde{p}\left(\nu+\frac{1}{2^{N+1}}\right)+\frac{\delta}{2},\tag{H3}$$

where the last equality uses $\mathrm{Tr}(\mathbb{M})\le 1$ given in Definition 1.

Note that, to guarantee that QNN can simulate the QSQ oracle as formulated in Definition 1, the most right term in Eq. (H3) should be upper bounded by τ, i.e.,

$$\left|\frac{1}{K}\sum_{k=1}^{K}V_k-\nu\right|\le\tilde{p}\left(\nu+\frac{1}{2^{N+1}}\right)+\frac{\delta}{2}\le\frac{5}{4}\tilde{p}+\frac{\delta}{2}\le\tau,$$

where the last second inequality uses the upper bounds $\nu\le 1$ and $(1/2^{N+1})\le\frac{1}{4}$. Note that the above inequality implicitly requests that $\tilde{p}<\frac{4}{5}$, since the threshold τ is in the range $(0,1)$. After simplification, we have

$$\delta\le 2\left(\tau-\tilde{p}\,\frac{5}{4}\right).$$

In other words, when $\delta=2(\tau-\tilde{p}\,\frac{5}{4})$, with probability at least $1-b$, the sample mean of noisy QNN satisfies

$$\left|\frac{1}{K}\sum_{k=1}^{K}V_k-\nu\right|\le\tau,\tag{H4}$$

which accords with the output of the QSQ oracle.

We now quantify the number of measurements K to promise Eq. (H4). Recall $K=2\ln(2/b)/\delta^2$. By employing the explicit form of δ, we obtain

$$K=\frac{\ln(2/b)}{2[\tau-\tilde{p}\,(5/4)]^2}.$$

The achieved result indicates that the successful probability of noisy QNN (i.e., $1-2b$) to estimate the QSQ oracle can be exponentially improved by linearly increasing the number of measurements. Moreover, the term $1/[\tau-\tilde{p}\,(5/4)]$ implies that the lower gate noise and lower circuit depth result in the smaller number of measurements, which guarantees the efficiency of noisy QNN to simulate the QSQ oracle. ■

APPENDIX I: GENERALIZATION OF THE RESULTS TO MORE GENERAL QUANTUM CHANNELS

Here we generalize the achieved results in the main text from the depolarization channel to a more general channel $\mathcal{E}_{p_1}$. Specifically, after applying $\mathcal{E}_{p_1}$ to each circuit depth, the generated state of QNN follows

$$\mathcal{E}_{p_1}\{U_L(\boldsymbol{\theta})\cdots U_2(\boldsymbol{\theta})\mathcal{E}_{p_1}[U_1(\boldsymbol{\theta})\rho U_1(\boldsymbol{\theta})^\dagger]U_2(\boldsymbol{\theta})^\dagger\cdots U_L(\boldsymbol{\theta})^\dagger\}=(1-p_1)^{L_Q}\,[U(\boldsymbol{\theta})U_{\mathbf{x}}]\,\rho\,[U(\boldsymbol{\theta})U_{\mathbf{x}}]^\dagger+p_2'\kappa+p_3^{L_Q}\frac{\mathbb{I}_D}{D},\tag{I1}$$

where $(1-p_1)^{L_Q}+p_2'+p_3^{L_Q}=1$, and κ is a mixed state that can either be correlated or uncorrelated with $[U(\boldsymbol{\theta})U_{\mathbf{x}}]\,\rho\,[U(\boldsymbol{\theta})U_{\mathbf{x}}]^\dagger$. Without confusion, we set $\tilde{p}=1-(1-p_1)^{L_Q}$. It is worth noting that the quantum channel $\mathcal{E}_{p_1}$ formulated above is sufficiently universal, which closely relates to most Pauli channels associated with the depolarization channel [55,95].

The outline of this section is as follows. In Sec. 1, we discuss the utility bounds of QNN under ERM. Then, in Sec. 2, we quantify the generalization property of QNN.

1. Utility bounds of QNN

We now employ the noisy quantum model, i.e., the right-hand side of Eq. (I1), to establish the relation between the estimated gradients $\nabla_j\bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ and the analytic gradients $\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)})$. Recall that

$$\nabla_j\bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})=(\bar{Y}_i^{(t)}-Y_i)\left(\bar{Y}_i^{(t,+_j)}-\bar{Y}_i^{(t,-_j)}\right)+\lambda\boldsymbol{\theta}_j^{(t)},$$

where $\bar{Y}_i^{(t)}=\sum_{k=1}^{K}V_k^{(t)}/K$ and $\bar{Y}_i^{(t,\pm_j)}=\sum_{k=1}^{K}V_k^{(t,\pm_j)}/K$ refer to the sample means when feeding $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^{(t,\pm_j)}$ into the trainable circuit. As with the depolarization channel, the sample mean $\bar{Y}_i^{(t)}$ or $\bar{Y}_i^{(t,\pm_j)}$ is a random variable that follows a certain distribution. In particular, following the notations used in Theorem 4, the mean and variance of $\bar{Y}_i^{(t)}$ follows

$$\begin{cases}\nu^{(t)}=(1-\tilde{p})\hat{Y}_i^{(t)}+p_2'\,\mathrm{Tr}(\Pi\kappa^{(t)})+\frac{p_3^{L_Q}}{2},\\ \sigma^{(t)}=-\frac{\left((1-\tilde{p})\hat{Y}_i^{(t)}+p_2'\,\mathrm{Tr}(\Pi\kappa^{(t)})\right)^2}{K}+\frac{(1-p_3^{L_Q})\left((1-\tilde{p})\hat{Y}_i^{(t)}+p_2'\,\mathrm{Tr}(\Pi\kappa^{(t)})\right)}{K}+\frac{p_3^{L_Q}}{2}-\frac{(p_3^{L_Q})^2}{4}.\end{cases}$$

Similarly, the mean and variance of $\bar{Y}_i^{(t,\pm_j)}$ follows

$$\begin{cases} \nu^{(t,\pm_j)} = (1-\tilde{p})\hat{Y}_i^{(t,\pm_j)} + p_2' \operatorname{Tr}(\Pi\kappa^{(t,\pm_j)}) + \frac{p_3^{L_Q}}{2}, \\ \sigma^{(t,\pm_j)} = -\frac{\left((1-\tilde{p})\hat{Y}_i^{(t,\pm_j)} + p_2' \operatorname{Tr}(\Pi\kappa^{(t,\pm_j)})\right)^2}{K} + \frac{(1-p_3^{L_Q})\left((1-\tilde{p})\hat{Y}_i^{(t,\pm_j)} + p_2' \operatorname{Tr}(\Pi\kappa^{(t,\pm_j)})\right)}{K} + \frac{p_3^{L_Q}}{2} - \frac{(p_3^{L_Q})^2}{4}. \end{cases}$$

By expanding the sample means using their explicit forms as shown above, we obtain the relation between the estimated and analytic gradients, i.e.,

$$\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (1-\tilde{p})^2 \nabla_j \mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + \varsigma_i^{(t,j)}, \tag{I2}$$

where $\varsigma_i^{t,j} = C_{j,2}^{(i,t)}\xi_i^{(t)} + C_{j,2}^{(i,t)}\xi_i^{(t,j)} + \xi_i^{(t)}\xi_i^{(t,j)}$, and two random variables $\xi_i^{(t)}$ and $\xi_i^{(t)}$ have zero means and their variances are $C_{j,4}^{(i,t)}$ and $C_{j,5}^{(i,t)}$, respectively. The explicit formula of the five parameters $\{C_{j,a}^{(i,t)}\}_{a=1}^t$ is

$$\begin{cases} C_{j,1}^{(i,t)} &= \left(p_2' \operatorname{Tr}(\Pi\kappa^{(t)}) + \frac{p_3^{L_Q}}{2} - \tilde{p}\, Y_i\right)(1-\tilde{p})(\hat{Y}_i^{(t,+_j)} - \hat{Y}_i^{(t,-_j)}) \\ &\quad + p_2'(1-\tilde{p})(\hat{Y}_i^{(t)} - Y_i)[\operatorname{Tr}(\Pi\kappa^{(t,+_j)}) - \operatorname{Tr}(\Pi\kappa^{(t,-_j)})] \\ &\quad + \left(p_2' \operatorname{Tr}(\Pi\kappa^{(t)}) + \frac{p_3^{L_Q}}{2} - \tilde{p}\, Y_i\right)[\operatorname{Tr}(\Pi\kappa^{(t,+_j)}) - \operatorname{Tr}(\Pi\kappa^{(t,-_j)})] + [1-(1-\tilde{p})^2]\lambda\boldsymbol{\theta}_j^{(t)}, \\ C_{j,2}^{(i,t)} &= \left((1-\tilde{p})(\hat{Y}_i^{(t,+_j)} - \hat{Y}_i^{(t,-_j)}) + p_2'[\operatorname{Tr}(\Pi\kappa^{(t,+_j)}] - \operatorname{Tr}[\Pi\kappa^{(t,-_j)})]\right), \\ C_{j,3}^{(i,t)} &= \left[(1-\tilde{p})(\hat{Y}_i^{(t)} - Y_i) + \left(p_2' \operatorname{Tr}(\Pi\kappa^{(t)}) + \frac{p_3^{L_Q}}{2} - \tilde{p}\, Y_i\right)\right], \\ C_{j,4}^{(i,t)} &= -\frac{\left((1-\tilde{p})\hat{Y}_i^{(t)} + p_2' \operatorname{Tr}(\Pi\kappa^{(t)})\right)^2}{K} + \frac{(1-p_3^{L_Q})\left((1-\tilde{p})\hat{Y}_i^{(t)} + p_2' \operatorname{Tr}(\Pi\kappa^{(t)})\right)}{K} + \frac{p_3^{L_Q}}{2K} - \frac{(p_3^{L_Q})^2}{4K}, \\ C_{j,5}^{(i,t)} &= -\frac{\left((1-\tilde{p})\hat{Y}_i^{(t,+_j)} + p_2' \operatorname{Tr}(\Pi\kappa^{(t,+_j)})\right)^2}{K} - \frac{\left((1-\tilde{p})\hat{Y}_i^{(t,-_j)} + p_2' \operatorname{Tr}(\Pi\kappa^{(t,-_j)})\right)^2}{K} \\ &\quad + \frac{(1-p_3^{L_Q})\left((1-\tilde{p})(\hat{Y}_i^{(t,+_j)} - \hat{Y}_i^{(t,-_j)}) + p_2'[\operatorname{Tr}(\Pi\kappa^{(t,+_j)}) - \operatorname{Tr}(\Pi\kappa^{(t,-_j)})]\right)}{K} + \frac{p_3^{L_Q}}{K} - \frac{(p_3^{L_Q})^2}{2K}. \end{cases}$$

We next use the relation between the estimated and analytic gradients to separately quantify the utility bounds R_1 and R_2 of QNN under the noisy channel $\mathcal{E}_{p_1}$ setting.

Utility bound R_1. As with Eq. (F7), with taking expectation over $\xi_i^{(t)}$ and $\xi_i^{(t,j)}$, we obtain

$$\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})] \le -\frac{1}{S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{G}{2S}\left(\frac{1}{B}\sum_{i=1}^B C_{j,1}^{(i,t)}\right) + \frac{1}{2S}\sum_{j=1}^d \mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right], \tag{I3}$$

where the inequality employs $\mathbb{E}[\xi_i^{(t)}] = 0$, $\mathbb{E}[\xi_i^{(t,j)}] = 0$, and $-G/d \le \nabla_j \mathcal{L}(\boldsymbol{\theta}^{(t)}) \le G/d$.

For the term $(1/2S)\sum_{j=1}^d \mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\{\left[\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right]^2\}$ in the above equation, its upper bound satisfies

$$\begin{aligned} \frac{1}{2S}\sum_{j=1}^d \mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right] \le& \frac{(1-\tilde{p})^4}{2S}\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{(1-\tilde{p})^2 G}{2SB}\sum_{i=1}^B C_1^{(i,t)} \\ &+ \frac{d}{2SB^2}\left(\sum_{i=1}^B C_1^{(i,t)}\right)^2 + d\frac{\sigma_{\max}^{(t)} + \sigma_{\max}^{(t,j)} + \sigma_{\max}^{(t)}\sigma_{\max}^{(t,j)}}{SB}, \end{aligned} \tag{I4}$$

where the first and second inequalities uses $C_2^{(i,t)} \le 2$, $C_3^{(i,t)} \le 2$, $\mathbb{E}[\xi_i^{(t)}] = 0$, and $\mathbb{E}[\xi_i^{(t,j)}] = 0$. The term $\sigma_{\max}^{(t)}$ refers to $\sigma_{\max}^{(t)} = \max_i \sigma_i^{(t)} \le 3/K$. Similarly, the term $\sigma_{\max}^{(t,j)}$ refers to $\sigma_{\max}^{(t,j)} = \max_i \sigma_i^{(t,+_j)} + \sigma_i^{(t,-_j)} \le 3/K$.

In conjunction with the above two equations, we achieve

$$\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})] \leq -\frac{1}{2S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{(2G+d)\{5+3[1-(1-\tilde{p})^2]\lambda\pi\}}{2S} + \frac{6dK+9d}{SBK^2}, \tag{I5}$$

where the inequality uses $C_{j,1}^{(i,t)} \leq 5 + 3[1-(1-\tilde{p})^2]\lambda\pi$.

After rewriting and taking induction, we have

$$\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 \leq 2S\frac{1+9\lambda d}{T(1-\tilde{p})^2} + \frac{(2G+d)\{5+3[1-(1-\tilde{p})^2]\lambda\pi\}}{(1-\tilde{p})^2} + \frac{12dK+18d}{(1-\tilde{p})^2BK^2}. \tag{I6}$$

With setting $T \to \infty$, we achieve the utility bound R_1, i.e.,

$$R_1 \leq \tilde{O}\left(\frac{1}{(1-\tilde{p})^2}, d, \frac{1}{BK}\right). \tag{I7}$$

Utility bound R_2. With combining Eq. (I5) and the PL condition, we obtain

$$\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})] \leq -\frac{\mu(1-\tilde{p})^2}{S}[\mathcal{L}(\boldsymbol{\theta}^{(t)}) - \mathcal{L}^*] + \frac{(2G+d)\{5+3[1-(1-\tilde{p})^2]\lambda\pi\}}{2S} + \frac{6dK+9d}{SBK^2}. \tag{I8}$$

After rewriting and induction, we have

$$\mathbb{E}_{\boldsymbol{\varsigma}^{(t)}}[\mathcal{L}(\boldsymbol{\theta}^{(T)})] - \mathcal{L}^* \leq 15\lambda d\exp\left(-\frac{\mu(1-\tilde{p})^2T}{S}\right) + T\frac{(2G+d)\{5+3[1-(1-\tilde{p})^2]\lambda\pi\}}{2S} + T\frac{6dK+9d}{SBK^2}. \tag{I9}$$

With setting $T = O\{[S/\mu(1-\tilde{p})^2]\ln\left(30\lambda dSBK^2/(2G+d)\{5+3[1-(1-\tilde{p})^2]\lambda\pi\}BK^2 + 12dK + 18d\right)\}$, the utility bound is

$$R_2 \leq O\left(\frac{1}{(1-\tilde{p})^2}, \frac{1}{SBK^2}, d\right). \tag{I10}$$

2. Generalization property of (noisy) QNN

The generalization of Theorem 2. Analogous to the depolarization noise setting, the distance between the target result $\nu = \mathrm{Tr}(\mathbb{M}|\psi_{c^*}\rangle\langle\psi_{c^*}|)$ and the shifted expectation value $\tilde{\nu} = (1-\tilde{p})\nu + p_2'\,\mathrm{Tr}(\mathbb{M}\kappa) + p_3^{L_Q}\,\mathrm{Tr}(\mathbb{M})/D$ of QNN under the noisy channel $\mathcal{E}_{p_1}$ follows $|\nu - \tilde{\nu}| \leq \tilde{p}\,\nu + p_2' + p_3^{L_Q}/D$. Then by employing Chernoff-Hoeffding bound, we achieve, with probability at least $1 - 2\exp(-\delta^2 n/2)$,

$$\left|\frac{1}{k}\sum_{k=1}^{K}V_k - \nu\right| \leq \left|\frac{1}{k}\sum_{k=1}^{K}V_k - \tilde{\nu} + \tilde{\nu} - \nu\right| \leq \tilde{p}\,\nu + p_2' + \frac{p_3^{L_Q}}{D} + \frac{\delta}{2}.$$

With setting $\delta = 2(\tau - \tilde{p}\,\nu - p_2' - p_3^{L_Q}/D)$, the relation between the number of measurements K and the successful probability b obeys

$$\Pr\left[\left|\frac{1}{K}\sum_{k=1}^{K}V_k - \tilde{\nu}\right| \geq \left(\tau - \tilde{p}\,\nu - p_2' - \frac{p_3^{L_Q}}{D}\right)\right] \leq 2\exp\left[-2\left(\tau - \tilde{p}\,\nu - p_2' - \frac{p_3^{L_Q}}{D}\right)^2 K\right] = b. \tag{I11}$$

After simplification, we conclude that, when $\tilde{p} \leq [\tau - p_2' - (p_3^{LQ}/D) - (\delta/2)]/\nu$ (to promise the existence of the feasible solution), with the successful probability at least $1 - b$, the required number of measurements to attain $\left|1/K \sum_{k=1}^{K} V_k - \nu\right| \leq \tau$ is

$$K = \frac{\ln(2/b)}{4\left[\tau - \tilde{p}\nu - p_2' - (p_3^{LQ}/D)\right]^2}. \tag{I12}$$

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, Cambridge, MA, USA, 2016).

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE, Manhattan, New York, USA, 2017), p. 2961.

[3] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., New York, NY, USA, 2019), p. 5754.

[4] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, in *Proceedings of the 26th International Conference on World Wide Web* (Association for Computing Machinery, New York, NY, USA, 2017), p. 173.

[5] Z. Allen-Zhu, Y. Li, and Y. Liang, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., New York, NY, USA, 2019), p. 6158.

[6] R. Livni, S. Shalev-Shwartz, and O. Shamir, in *Advances in Neural Information Processing Systems* (MIT Press, Cambridge, MA, USA, 2014), p. 855.

[7] Y. Li and Y. Yuan, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., New York, NY, USA, 2017), p. 597.

[8] Z. Allen-Zhu, Y. Li, and Z. Song, in *International Conference on Machine Learning* (Microtome Publishing, Brookline, MA, USA, 2019), p. 242.

[9] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, in *International Conference on Machine Learning* (Microtome Publishing, Brookline, MA, USA, 2019), p. 1675.

[10] A. Andoni, R. Panigrahy, G. Valiant, and L. Zhang, in *International Conference on Machine Learning* (Microtome Publishing, Brookline, MA, USA, 2014), p. 1908.

[11] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195 (2017).

[12] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: A classical perspective, Proc. R. Soc. A **474**, 20170551 (2018).

[13] V. Dunjko and H. J. Briegel, Machine learning & artificial intelligence in the quantum domain: A review of recent progress, Rep. Prog. Phys. **81**, 074001 (2018).

[14] A. W. Harrow and A. Montanaro, Quantum computational supremacy, Nature **549**, 203 (2017).

[15] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, Training deep quantum neural networks, Nat. Commun. **11**, 1 (2020).

[16] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, ArXiv:1802.06002 (2018, to be published).

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

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

[19] M. Schuld and N. Killoran, Quantum Machine Learning in Feature Hilbert Spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[20] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, ArXiv:1411.4028 (2014, to be published).

[21] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[22] S. Aaronson and A. Arkhipov, in *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, USA, 2011), p. 333.

[23] M. J. Bremner, R. Jozsa, and D. J. Shepherd, Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy, Proc. R. Soc. A **467**, 459 (2011).

[24] C. Blank, D. K. Park, J.-K. K. Rhee, and F. Petruccione, Quantum classifier with tailored quantum kernel, npj Quantum Inf. **6**, 1 (2020).

[25] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, Continuous-variable quantum neural networks, Phys. Rev. Res. **1**, 033063 (2019).

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

[27] L. Gentini, A. Cuccoli, S. Pirandola, P. Verrucchi, and L. Banchi, Noise-resilient variational hybrid quantum-classical optimization, Phys. Rev. A **102**, 052414 (2020).

[28] C.-C. Chen, M. Watabe, K. Shiba, M. Sogabe, K. Sakamoto, and T. Sogabe, On the expressibility and overfitting of quantum circuit learning, ACM Trans. Quantum Comput. **2**, 1 (2021).

[29] Y. Du, Z. Tu, X. Yuan, and D. Tao, An efficient measure for the expressivity of variational quantum algorithms, ArXiv:2104.09961 (2021, to be published).

[30] K. Bu, D. E. Koh, L. Li, Q. Luo, and Y. Zhang, On the statistical complexity of quantum circuits, ArXiv:2101.06154 (2021, to be published).

[31] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, Nat. Comput. Sci. **1**, 403 (2021).

[32] M. C. Caro and I. Datta, Quantum learning with noise and decoherence: A robust quantum neural network, Quantum Mach. Intell. **2**, 1 (2020).

[33] M. C. Caro, E. Gil-Fuster, J. J. Meyer, J. Eisert, and R. Sweke, Encoding-dependent generalization bounds for parametrized quantum circuits, ArXiv:2106.03880 (2021, to be published).

[34] Y. Qian, X. Wang, Y. Du, X. Wu, and D. Tao, The dilemma of quantum neural networks, ArXiv:2106.04975 (2021, to be published).

[35] L. Banchi, J. Pereira, and S. Pirandola, Generalization in quantum machine learning: a quantum information perspective, ArXiv:2102.08991 (2021, to be published).

[36] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Anomalous collapses of nares strait ice arches leads to enhanced export of arctic sea ice, Nat. Commun. **12**, 1 (2021).

[37] H.-Y. Huang, R. Kueng, and J. Preskill, Information-Theoretic Bounds on Quantum Advantage in Machine Learning, Phys. Rev. Lett. **126**, 190505 (2021).

[38] C. Gyurik, D. van Vreumingen, and V. Dunjko, Generalization in quantum machine learning: a quantum information perspective, ArXiv:2105.05566 (2021, to be published).

[39] V. Vapnik, in *Advances in Neural Information Processing Systems* (Morgan Kaufmann Publishers, San Francisco, CA, USA, 1992), p. 831.

[40] V. Vapnik, *The Nature of Statistical Learning Theory* (Springer-Verlag, Berlin, Germany, 1995).

[41] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 1 (2018).

[42] S. Arunachalam and R. de Wolf, Guest column, ACM SIGACT News **48**, 41 (2017).

[43] A. Atici and R. A. Servedio, Improved bounds on quantum learning algorithms, Quantum Inf. Process. **4**, 355 (2005).

[44] E. Bernstein and U. Vazirani, Quantum complexity theory, SIAM J. Comput. **26**, 1411 (1997).

[45] R. A. Servedio and S. J. Gortler, Equivalences and separations between quantum and classical learnability, SIAM J. Comput. **33**, 1067 (2004).

[46] M. C. Caro, Measurement-device-independent quantum key distribution with uncharacterized coherent sources, Quantum Inf. Process. **19**, 1 (2020).

[47] A. W. Cross, G. Smith, and J. A. Smolin, Quantum learning robust against noise, Phys. Rev. A **92**, 012327 (2015).

[48] A. B. Grilo, I. Kerenidis, and T. Zijlstra, Learning-with-errors problem is easy with quantum samples, Phys. Rev. A **99**, 032314 (2019).

[49] A. Gollakota and D. Liang, On the hardness of PAC-learning stabilizer states with noise, ArXiv:2102.05174 (2021, to be published).

[50] S. Arunachalam, A. B. Grilo, and H. Yuen, Quantum statistical query learning, ArXiv:2002.08240 (2020, to be published).

[51] S. Arunachalam, Y. Quek, and J. Smolin, Private learning implies quantum stability, ArXiv:2102.07171 (2021, to be published).

[52] Y. Du, M.-H. Hsieh, T. Liu, S. You, and D. Tao, Quantum differentially private sparse regression learning, ArXiv:2007.11921 (2020, to be published).

[53] S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, United Kingdom, 2004).

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

[55] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, United Kingdom, 2010).

[56] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).

[57] V. Koltchinskii, *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008* (Springer Science & Business Media, Berlin, Germany, 2011), Vol. 2033.

[58] J. Zhang, K. Zheng, W. Mou, and L. Wang, in *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (AAAI Press, Cambridge, USA, 2017), p. 3922.

[59] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, Convexity, classification, and risk bounds, J. Am. Stat. Assoc. **101**, 138 (2006).

[60] P. L. Bartlett and S. Mendelson, Empirical minimization, Probab. Theory. Relat. Fields **135**, 311 (2006).

[61] C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan, in *Proceedings of the 34th International Conference on Machine Learning* (Microtome Publishing, Brookline, MA, USA, 2017), Vol. 70, p. 1724.

[62] H. Karimi, J. Nutini, and M. Schmidt, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (Springer, Berlin, Germany, 2016), p. 795.

[63] Y. Nesterov and B. T. Polyak, Cubic regularization of Newton method and its global performance, Math. Program. **108**, 177 (2006).

[64] K. J. Sung, M. P. Harrigan, N. C. Rubin, Z. Jiang, R. Babbush, and J. R. McClean, An exploration of practical optimizers for variational quantum algorithms on superconducting qubit processors, ArXiv:2005.11011 (2020).

[65] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, Absence of Barren Plateaus in quantum convolutional neural networks, Phys. Rev. X **11**, 041011 (2021).

[66] T. Volkoff and P. J. Coles, Large gradients via correlation in random parameterized quantum circuits, Quantum Sci. Technol. **6**, 025008 (2021).

[67] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, Quantum **3**, 214 (2019).

[68] A. Skolik, J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib, Layerwise learning for quantum neural networks, Quantum Mach. Intell. **3**, 1 (2021).

[69] C. O. Marrero, M. Kieferová, and N. Wiebe, Entanglement induced barren plateaus, ArXiv:2010.15968 (2020, to be published).

[70] K. Zhang, M.-H. Hsieh, L. Liu, and D. Tao, Toward trainability of quantum neural networks, ArXiv:2011.06258 (2020, to be published).

[71] C. Zhao and X.-S. Gao, Analyzing the barren plateau phenomenon in training quantum neural networks with the ZX-calculus, Quantum **5**, 466 (2021).

[72] T. L. Patti, K. Najafi, X. Gao, and S. F. Yelin, Entanglement devised barren plateau mitigation, Phys. Rev. Res. **3**, 033090 (2021).

[73] K. Kawaguchi, Deep learning without poor local minima, Adv. Neural Inf. Process. Syst. **29**, 586 (2016).

[74] M. Plesch and Č. Brukner, Quantum-state preparation with universal gate decompositions, Phys. Rev. A **83**, 032302 (2011).

[75] M. Schuld, M. Fingerhuth, and F. Petruccione, Implementing a distance-based classifier with a quantum interference circuit, EPL **119**, 60002 (2017).
[76] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).
[77] C. Wilson, J. Otterbach, N. Tezak, R. Smith, G. Crooks, and M. da Silva, Quantum kitchen sinks: An algorithm for machine learning on near-term quantum computers, ArXiv:1806.08321 (2018, to be published).
[78] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. **4**, 043001 (2019).
[79] D. Dua and C. Graff, *UCI Machine Learning Repository* (University of California, Irvine, 2017), http://archive.ics.uci.edu/ml.
[80] S. Wold, K. Esbensen, and P. Geladi, Principal component analysis, Chemometr. Intell. Lab. Syst. **2**, 37 (1987).
[81] L. Bittel and M. Kliesch, Training variational quantum algorithms is NP-hard—even for logarithmically many qubits and free fermionic systems, ArXiv:2101.07267 (2021, to be published).
[82] https://github.com/yuxuan-du/Learnability-of-QNN.
[83] A. Blum, A. Kalai, and H. Wasserman, Noise-tolerant learning, the parity problem, and the statistical query model, J. ACM **50**, 506 (2003).
[84] X. Wang, Y. Du, Y. Luo, and D. Tao, Towards understanding the power of quantum kernels in the NISQ era, Quantum **5**, 531 (2021).
[85] H. Cai, Q. Ye, and D.-L. Deng, Sample complexity of learning quantum circuits, ArXiv:2107.09078 (2021, to be published).
[86] V. Feldman, A complete characterization of statistical query learning with applications to evolvability, J. Comput. Syst. Sci. **78**, 1444 (2012).
[87] V. Feldman, C. Guzman, and S. Vempala, in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms* (SIAM, Philadelphia, PA, USA, 2017), p. 1265.
[88] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, Quantum **4**, 269 (2020).
[89] J. Gacon, C. Zoufal, G. Carleo, and S. Woerner, Simultaneous perturbation stochastic approximation of the quantum fisher information, ArXiv:2103.09232 (2021, to be published).
[90] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, Stochastic gradient descent for hybrid quantum-classical optimization, Quantum **4**, 314 (2020).
[91] M. Zhou, T. Liu, Y. Li, D. Lin, E. Zhou, and T. Zhao, Towards understanding the importance of noise in training neural networks, ArXiv:1909.03172 (2019, to be published).
[92] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, A grover-search based quantum learning scheme for classification, New J. Phys. **23**, 023020 (2021).
[93] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, Expressive power of parametrized quantum circuits, Phys. Rev. Res. **2**, 033125 (2020).
[94] W. Huggins, P. Patil, B. Mitchell, K. B. Whaley, and E. M. Stoudenmire, Towards quantum machine learning with tensor networks, Quantum Sci. Technol. **4**, 024001 (2019).
[95] K. Sharma, S. Khatri, M. Cerezo, and P. J. Coles, Noise resilience of variational quantum compiling, New J. Phys. **22**, 043006 (2020).