# On the learnability of quantum neural networks

**Yuxuan Du**
The University of Sydney
**Min-Hsiu Hsieh** ( Min-Hsiu.Hsieh@uts.edu.au )
University of Technology, Sydney https://orcid.org/0000-0002-3396-8427
**Tongliang Liu**
The University of Sydney
**Shan You**
Sensetime
**Dacheng Tao**
The University of Sydney https://orcid.org/0000-0001-7225-5449

**Article**

**Keywords:** Quantum neural network, QNN, learning capability, trainability, NISQ devices

**Posted Date:** September 25th, 2020

**DOI:** https://doi.org/10.21203/rs.3.rs-80242/v1



**Version of Record:** A version of this preprint was published at PRX Quantum on November 17th, 2021. See the published version at https://doi.org/10.1103/PRXQuantum.2.040337.

# On the learnability of quantum neural networks

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

[1] *UBTECH Sydney AI Centre, School of Computer Science, Faculty of Engineering, The University of Sydney, Australia*

[2] *Centre for Quantum Software and Information, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia*

[3] *SenseTime*

**Quantum neural network (QNN), or equivalently, the variational quantum circuits 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 non-convex optimization landscape, the measurement error, and the unavoidable gate noise introduced by NISQ machines. In this study, we theoretically explore the learnability of QNN from the perspective 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 restricted quantum statistical query (QSQ) learning model, can also be efficiently learned by QNN. Since the restricted QSQ learning model can tackle certain problems such as parity learning with a runtime speedup, our result suggests that QNN established on NISQ devices will retain the quantum advantage. Our work provides the theoretical guidance for developing advanced QNNs and opens up avenues for exploring quantum advantages using NISQ devices.**

* Corresponding authors

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 multi-layer scheme. As shown in Figure 1(a), the inputs are processed through the feature embedding layers $\mathcal{F}_{\boldsymbol{x}}(\cdot)$, followed by the fully-connected layers $\prod_{\ell} W_{\ell}(\cdot)$, where the choice of each layer and the combination rule can be tailor-made 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 non-convex 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, the study [5] proved that over-parameterized DNN can learn important concept classes, including the two and three-layer DNN with fewer parameters, in polynomial samples; while the study [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 quantum extension of DNN, i.e., the quantum neural network (QNN), which is separately proposed in [15–20], received great attention due to the huge success of DNN and the superior computational power of quantum devices [21]. As shown in Figure 1(b), QNN also adopts the multi-layer architecture: the inputs were converted into quantum states by the encoding quantum circuit $U_{\boldsymbol{x}}$, 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 '$\mathcal{F}_{\boldsymbol{x}}$' of DNN coincides with the encoding quantum circuit $U_{\boldsymbol{x}}$ of QNN, while the fully-connected layer $W_l(\cdot)$ of DNN coincides with the trainable quantum circuit $U_l(\boldsymbol{\theta})$ of QNN. Celebrated by the strong power of quantum circuits to prepare classical distributions [22, 23], QNN could possess a stronger expressive power than its classical counterparts [24] and advance a wide range of machine learning problems.

Despite the promising prospects, the learning capabilities of QNNs, i.e., their trainability and generalization, remain largely unknown. Firstly, even though empirical studies have shown that QNN can accomplish various supervised learning tasks, e.g., classification [17, 19, 25] and regression [18, 26], a rigorous analysis of the learning performance is lacking. The obstruction that impedes the theoretic progress origins from the combination of the following factors, including the versatile structures of QNN, the non-convex optimization landscapes, the unavoidable gate noise and measurement errors. Classically, the empirical risk minimization (ERM) principle [27, 28] is employed as a universal framework 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 ERM? 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 QNN [29]. More discussion will come after we formally introduce Theorem 1.

Secondly, understanding the generalization of QNN can facilitate the exploration of its applicability with provable advantages; however, theoretical analysis of the generalization property of QNN remains largely open. The difficulty mainly comes from the universality of the generalization, which concerns an entire concept class instead of a specific training dataset. Note that the investigation of the generalization for certain concept classes also lies in the center of the probably approximately correct (PAC) learning, as a building block of learning theory [30]. Analogous to the QNN's generalization, learning theory also concerns whether the learning model can efficiently output a hypothesis that can well approximate a target concept. Due to such a similarity, theoretical results from PAC learning have been broadly exploited to study the generalization of DNN [5, 10]. Unfortunately, quantum PAC (QPAC) learning [31–34], that is built on the noiseless assumption, is not suitable for studying the generalization of QNN because QNN is always associated with the unavoidable gate and measurement noise [21, 35]. A potential alternative is the recently proposed quantum statistical query (QSQ) learning model [36], and QSQ learning models can use exponentially fewer samples than their classical counterparts to learn certain concepts. If we could connect QNN with QSQ learning models, we can answer affirmatively whether there exists any class of concepts that can be efficiently learned by (noisy) QNN but are computationally hard for the classical learning models. Moreover, it enables us to employ QNN implemented on NISQ devices to accomplish certain tasks with

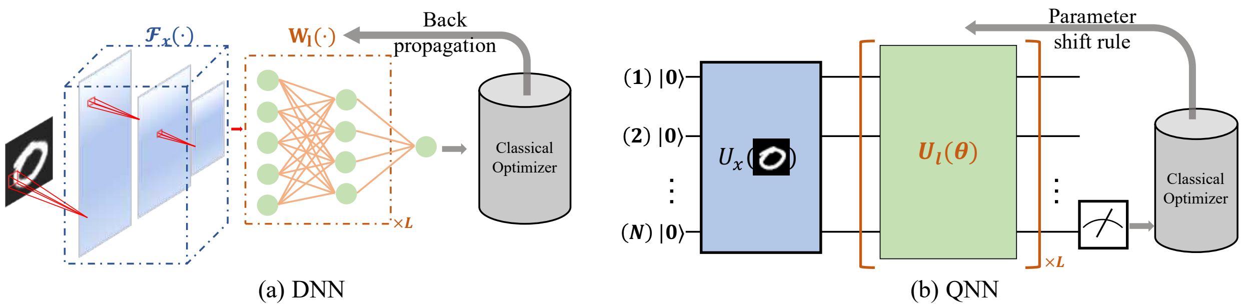



FIG. 1: Illustration of DNN and QNN. The left and right panel shows DNN and QNN, respectively. For DNN, the feature embedding layers $\mathcal{F}_{\boldsymbol{x}}(\cdot)$, which contains a sequence of operations with the arbitrary combination such as convolution and attention, maps the input '0' to the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

theoretical advantages.

## Results

**Trainability of QNN towards ERM.** Before elaborating our theoretical results, we first formulate ERM and the mechanism of QNN. Let $\boldsymbol{z} = \{\boldsymbol{z}_j\}_{j=1}^n \in \mathcal{Z}$ be the given dataset with $\mathcal{Z}$ being the sample domain, where the $j$-th sample $\boldsymbol{z}_j = (\boldsymbol{x}_j, y_j)$ includes a feature vector $\boldsymbol{x}_j \in \mathbb{R}^{D_c}$ 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.,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathcal{C}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{z}) := \frac{1}{n}\sum_{j=1}^{n} \ell(y_i, \hat{y}_i) + r(\boldsymbol{\theta}) \ , \qquad (1)$$

where $\hat{y}_i$ is the predicted label that is determined by $\boldsymbol{\theta}$ and $\boldsymbol{x}_i$, $\ell$ is the loss function that measures the disparity between true labels $\{y_j\}_{j=1}^n$ and the predicted labels $\{\hat{y}_i\}_{i=1}^n$, and $r(\cdot)$ is a regularizer. To ease the discussion, throughout the paper, we consider the mean square error loss $\ell$ with $\ell(y_i, \hat{y}_i) = (\hat{y}_i - y_i)^2$, 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 [37].

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, $\boldsymbol{z} = \cup_{j\in\{\mathcal{B}_i\}_{i=1}^B} \boldsymbol{z}_j$. The optimization rule at the $t$-th iteration is $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \frac{\eta}{B}\sum_{i=1}^B \nabla\mathcal{L}(\boldsymbol{\theta}^{(t)}, \mathcal{B}_i)$, where $\eta$ is the learning rate, the gradient $\nabla\mathcal{L}(\cdot)$ is

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

$Y_i = \frac{1}{B_s}\sum_{j\in\mathcal{B}_i} y_j$ and $\hat{Y}_i^{(t)} = \frac{1}{B_s}\sum_{j\in\mathcal{B}_i} \hat{y}_j^{(t)}$ are the sum average of the true labels and the predicted labels for the $i$-th 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 $\mathcal{L}(\boldsymbol{\theta}^{(t)}, \boldsymbol{z})$ and $\mathcal{L}(\boldsymbol{\theta}^{(t)}, \mathcal{B}_i)$ in the rest of study.

The general workflow of QNN is summarized in Figure 1(b). Specifically, QNN first employs a state preparation unitary $U_{\boldsymbol{x}}$ to encode classical inputs $\{\boldsymbol{x}_j | j \in \mathcal{B}_i\}$ 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 B for implementation details of $U_{\boldsymbol{x}}$ 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 $\bar{Y}_i = \frac{1}{K}\sum_{k=1}^K V_k$ after repeating the above procedure $K$ times. The law of quantum mechanics 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 machines, which are used to implement $U_{\boldsymbol{x}}$ and $U(\boldsymbol{\theta})$, are prone to having errors [35]. The gate noise can be simulated by applying certain quantum channels to each quantum 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 [38]. 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 is defined as $\mathcal{N}_p(\rho) = (1-p)\rho + p\mathbb{I}/D$, where $\mathbb{I}/D$ refers to the maximally mixed state [38]. After applying $\mathcal{N}_p$ to QNN, the quantum state before measurement is $\tilde{\gamma}_{\mathcal{B}_i} = \mathcal{N}_p(\gamma_{\mathcal{B}_i})$. When the measurement is applied to the state $\tilde{\gamma}_{\mathcal{B}_i}$, 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 all results presented in the main text assuming the depolarization noise; however, they can be easily extended to a more general noisy channel. Confer Appendix

G 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 Figure 1(b). However, the main difference between the gradient-based optimization of QNN and DNN is as follows. In DNN, the gradient in Eqn. (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 [39]. To overcome this shortcoming, the *parameter shift rule* [18, 39] is proposed to estimate the gradients of a quantum unitary operator using $K$ measurements. We will elaborate this step in the Methods section.

Now we quantify the convergence of QNN towards ERM. Paricularly, analyzing the convergence of QNN amounts to checking the following two standard utility metrics:

$$R_1\left(\boldsymbol{\theta}^{(T)}\right) := \mathbb{E}\left[\left\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(T)})\right\|^2\right],$$

$$R_2\left(\boldsymbol{\theta}^{(T)}\right) := \mathbb{E}[\mathcal{L}(\boldsymbol{\theta}^{(T)})] - \mathcal{L}(\boldsymbol{\theta}^*), \tag{3}$$

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 Eqn. (1). The metric $R_1$ evaluates how far QNN is away from the stationary point, $\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(T)}, \boldsymbol{z})\|^2 = 0$, in expectation [40, 41]. The utility metric $R_2$ evaluates the expected excess empirical risk [42, 43].

The utility bounds of noisy QNN are summarized in the following theorem, where the full proof is provided in Appendix E.

**Theorem 1.** *Let $K$ be the number of measurements, $L_Q$ 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*

$$R_1 \le \tilde{O}\left(poly\left(\frac{d}{T(1-p)^{L_Q}}, \frac{d}{BK(1-p)^{L_Q}}, \frac{d}{(1-p)^{L_Q}}\right)\right)$$

*When $\lambda$ satisfies a technical assumption, QNN outputs $\boldsymbol{\theta}^{(T)} \in \mathbb{R}^d$ after $T = \tilde{O}(\frac{d^3}{(1-p)^{L_Q}})$ with the utility bound*

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

Our result shows that a larger amount of measurements $K$, a lager 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 $R_1$. This 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 [37]. Moreover, the dependence of gate and measurement noise in $R_1$ and $R_2$ accords with the empirical observations [44] 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 (QAOA) [20] is applied to accomplish maximum cut problem on 3-regular graphs, the success probability drops to zero once the gate error level is larger than 0.1.

We note that the achieved utility bounds $R_1$ and $R_2$ are very general, and cover various types of encoding quantum circuits $U_{\boldsymbol{x}}$ and trainable quantum circuits $U(\boldsymbol{\theta})$. Specifically, our results cover all typical encoding circuits, e.g., amplitude encoding [45–47], kernel mapping [17–19], the dimension reduction methods [48], 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 qubits gates [49]. Theorem 1 provides theoretical guidances to design QNN-based learning algorithms on NISQ devices, considering that the gate and measurement noise are ubiquitous in these devices. Lastly, the convergence towards the global optima as shown in $R_2$ provides an insight about how to employ regularization techniques to avoid the barren plateau encountered in training QNN [29]. Particularly, the barren plateau phenomenon stated that, despite the gate noise, the optimization may be terminated at a point that is far away from the global minimum, since the gradient will vanish exponentially with respect to the increased number of qubits and the circuit depth. By contrast, the achieved utility bound $R_2$ shows that with the increasing number of measurements, QNN will converge to the global optima (at least in the noiseless setting). This observation suggests that the regularization techniques allow the optimization of QNN to be relieved from the barren plateau dilemma. Moreover, our result enlightens the path to apply QNN to accomplish large-scale quantum machine learning tasks that require the deep circuit depth and the huge number of qubits.

We also make the following technical contributions along the way to prove Theorem 1. In order to make use of a well-known result in optimization theory [50], namely the stationary point of a *smooth* function can be efficiently located by a simple analytic gradient-based algorithm, to prove the utility bound $R_1$, we have to analytically derive the gradient of QNN. However, it is impossible to obtain an exact gradient of QNN because of the inevitable gate noise and measurement error. To overcome this difficulty, we proved a bounded discrepancy between the estimated and analytic gradient of QNN (confer Theorem 3 in Method for details). This result, accompanied with the smooth property of $\mathcal{L}$, enables us to establish the utility bound $R_1$. Secondly, due to the hardness of finding the global optima $\mathcal{L}(\boldsymbol{\theta}^*)$ in the non-convex landscape, $R_2$ can only be applied to some special non-convex objective functions, i.e., the objective functions satisfy the Polyak-Lojasiewicz (PL) condition [51, 52]. In particular, the study [51] indicates that, if a non-convex function satisfies the PL condition, then every stationary point is the global minimum. In other words, PL enables us to leverage the convergence rate to a stationary point to evaluate $R_2$. Therefore, through proving that the objective function $\mathcal{L}$ also meets the PL condition under a technical assumption, we achieve the utility bounds of $R_2$. Note that the employed technical assumption allowed to bypass the barren plateau phenomenon surprisingly.

**Generalization of QNN.** Next we examine the generalization property of QNN by leveraging the results from quantum learning theory [31]. To achieve this goal, we establish an explicit connection between QNN and QSQ learning models [36], which differs from QPAC learning model via its noise-tolerant feature. Let us first recall the definition of QSQ learning model. Let $\mathcal{C} \subseteq \{c : \{0,1\}^N \rightarrow \{0,1\}\}$ be a concept class and $\mathcal{D} : \{0,1\}^N \rightarrow [0,1]$ be an unknown distribution. Define a QSQ oracle which takes a tolerance parameter $\tau$ and an observable $\mathbb{M} \in \mathbb{C}^{2^{N+1}\times 2^{N+1}}$ and returns a number $\alpha$ satisfying

$$|\alpha - \langle \psi_{c^*} | \mathbb{M} | \psi_{c^*} \rangle | \leq \tau \ , \tag{4}$$

where $|\psi_{c^*}\rangle = \sum_{\boldsymbol{x}\in\{0,1\}^N} \sqrt{\mathcal{D}(\boldsymbol{x})}\, |\boldsymbol{x}, c^*(\boldsymbol{x})\rangle$ refers to a quantum example. The QSQ learning algorithm adaptively feeds a sequence of $\{\mathbb{M}_i, \tau_i\}_i$ into a QSQ oracle, and exploits the responses of $\{\alpha_i\}_i$ to output a hypothesis $h : \{0,1\}^N \rightarrow \{0,1\}$. The goal of the learner is to achieve $\Pr_{\boldsymbol{x}\sim\mathcal{D}}(h(\boldsymbol{x}) \neq c^*(\boldsymbol{x})) \leq \varepsilon$ for all possible $\mathcal{D}$ and $c^*$.

Intuitively, QSQ model can only obtain the estimates of measurement statistics of quantum examples instead of directly accessing them. Notably, the QSQ oracle formulated in Eqn. (4) yields a similar behavior of the variational quantum circuit used in QNN. In particular, we show that QNN can efficiently simulate any QSQ learning algorithms with a restricted set of inputs; namely, when the distribution $\mathcal{D}$ is fixed to be uniform and the observables $\mathbb{M}$ can be implemented by at most *poly*$(n)$ single and two-qubit gates. By leveraging such an observation, we reach the following theorem whose proof is given in Appendix F.

**Theorem 2.** *A* QSQ *learning algorithm, where the distribution over the quantum example $|\psi_{c^*}\rangle$ is fixed to be uniform and the observable $\mathbb{M}$ can be implemented by at most poly$(n)$ single and two-qubit gates, can be efficiently simulated by noisy QNN using polynomial samples.*

The result of Theorem 2 indicates that a noisy QNN can effectively simulate a *restricted* QSQ learning model. Notably, the restricted QSQ learning model can efficiently tackle parity learning, juntas learning, and DNF (disjunctive normal form) learning problems, which are computational hard for the classical SQ model [36]. As a result, we attain a positive answer about the generalization of QNN. Namely, any learning concept class that can be solved by the restricted QSQ learning model with quantum advantages, e.g., parity learning, can also be tackled by a noisy QNN with preserved advantages. Furthermore, the efficacy to simulate the restricted QSQ model by noisy QNN paves a novel way to seeking diverse learning tasks that possess quantum merits, motivated by the fact that SQ learning algorithms have been broadly applied to support vector machines, linear and convex optimization, simulated annealing, matrix decomposition, and so on [53, 54]. In particular, we can first examine whether the restricted 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 accomplishes these tasks with quantum advantages.

**Numerical simulations.** We employ the UCI ML handwritten digits datasets [55] 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 digits images with 10 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 only collect 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 Figure 2. Second, we utilize a feature reduction technique, i.e., principal component analysis (PCA) [56], to reduce the feature dimension of each data example from 64 to 3. The lower left panel of Figure 2, highlighted by the gray region, exhibits the reconstructed hand-written 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 present-day. After applying PCA, we denote the employed dataset as $\boldsymbol{z} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{360}$, where $\boldsymbol{x}_i \in \mathbb{R}^3$ is the $i$-th data feature and $y_i \in \{0,1\}$ is the $i$-th label. The last step is randomly splitting $\boldsymbol{z}$ into two groups, i.e., the training dataset $\boldsymbol{z}_t$ and the test dataset $\boldsymbol{z}_p$. The size of the training dataset $\boldsymbol{z}_t$ and the test dataset $\boldsymbol{z}_p$ is 280 and 80, respectively.

We now employ the preprocessed hand-written digits dataset $\boldsymbol{z}$ and quantum circuits as used in [17] (Confer Methods 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 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 aims to estimate the optimal parameter $\boldsymbol{\theta}^*$ and the minimized objective function $\mathcal{L}^*$. The number of iterations for all numerical simulations

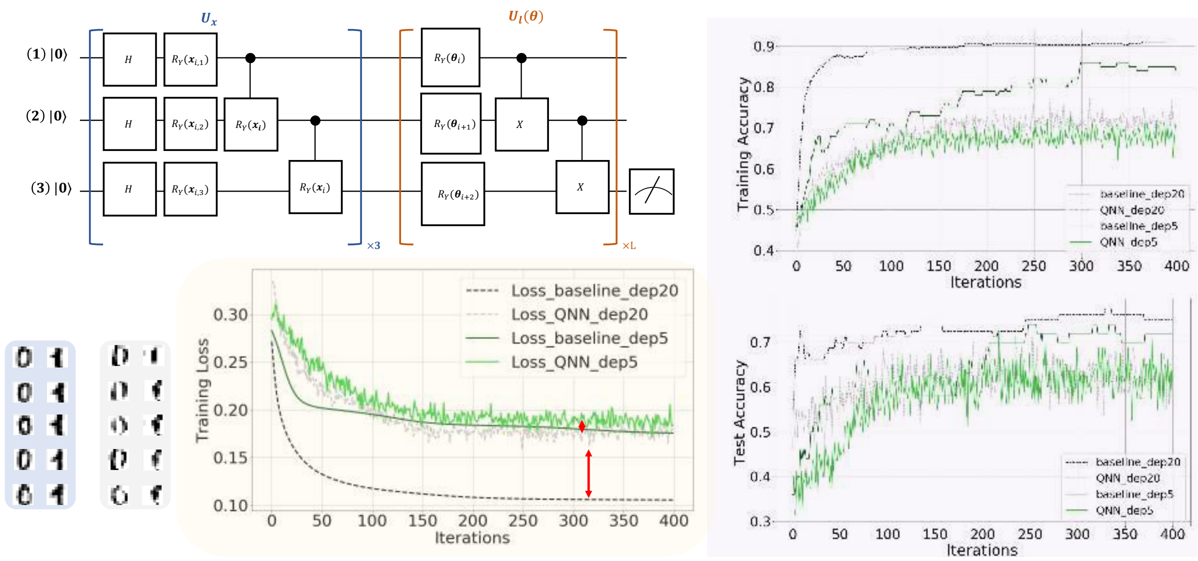



FIG. 2: The implementation of quantum circuits and the simulation results on hand-written digit dataset. The lower left panel illustrates the original and reconstructed training examples, as highlighted by the blue and gray regions, respectively. The upper left panel demonstrates the implementation of data encoding circuit and trainable circuit used in QNN. The label 'x3' and 'x$L$' 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 hyper-parameters settings. In particular, the label 'Loss_baseline_dep20' ('Loss_baseline_dep5') refers to the obtained loss under the setting $L = 20$ ($L = 5$), $p = 0$, and $K \to \infty$, where $L$, $p$, and $K$ refer to the circuit depth, depolarization rate, the number of measurements to estimate expectation value used in QNN, respectively. Similarly, the label 'Loss_QNN_dep20' ('Loss_QNN_dep5') refers to the obtained loss of QNN under the setting $L = 20$ ($L = 5$), $p = 0.0025$, $K = 20$. The upper right and lower right panels separately demonstrate the training accuracy and test accuracy of the quantum classifiers with different hyper-parameters settings.

described above is set as $T = 400$.

The simulation results, as shown in Figure 2, accord with our theoretical results. Specifically, as shown in the lower center of Figure 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 a poorer utility bound. In addition, the achieved training and test accuracies as shown in the right panel of Figure 2 implies that the noisy QNN can also learn a useful decision rule while its performance has slightly degenerated. These observations support the applicability of QNN on NISQ devices.

## Discussion

In this study, we explore the learnability of QNN 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 QNN. 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. Moreover, we demonstrate that QNN can efficiently learn parity, juntas, and DNF with quantum advantages even with gate noise. Our work also generates plausible new directions for NISQ study that we plan to explore in the future. First, we will use other advanced results in optimization theory to analyze various variational hybrid models on NISQ machines with provable guarantees. In particular, beyond solving classification and regression tasks, QNNs, or equivalently, the variational hybrid quantum-classical learning models, have also been empirically applied to explore fundamental properties of physical systems, e.g., ground energies approximation and thermal averages computation [57, 58]. These problems are generally more sensitive to the global minimum than that of machine learning problems. We expect that the analysis technique established on this study can be applied to explain heuristic result achieved in these learning problems. Second, we aim to exploit more advanced quantum models developed in quantum learning theory to explore the potential advantages that can be achieved by QNN.

## Methods.

*Parameter shift rule.* Denote the updating rule of QNN at the $t$-th 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 $\boldsymbol{\theta}^{(t,\pm_j)} := \boldsymbol{\theta}^{(t)} \pm \frac{\pi}{2}\boldsymbol{e}_j$ to the trainable circuit $U(\boldsymbol{\theta})$, where $\boldsymbol{e}_j$ is the basis vector with the $j$-th 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 only generate statistics $\bar{Y}_i^{(t)} = \frac{1}{K}\sum_{k=1}^{K} V_k^{(t)}$ and $\bar{Y}_i^{(t,\pm_j)} = \frac{1}{K}\sum_{k=1}^{K} V_k^{(t,\pm_j)}$, 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 D for details).

**Theorem 3.** *It follows that*

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

*where* $\tilde{p} = 1 - (1-p)^{L_Q}$*,* $L_Q$ *is the circuit depth, the constant* $C_{j,1}^{(i,t)}$ *only depends on* $Y_i$*,* $\boldsymbol{\theta}^{(t)}$*, and* $\tilde{p}$*, and* $\varsigma_i^{(t,j)}$ *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 $\varsigma_i^{(t,j)}$. 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, $K$ only determines the variance of the distribution $\mathcal{P}_Q$ with zero mean, where classical and quantum literatures [59, 60] have provided the convergence guarantee even if $K = 1$.

*The construction details of numerical simulations.* The implementation of the data encoding circuit $U_{\boldsymbol{x}}$ and the trainable unitary $U(\boldsymbol{\theta})$ follows the proposal [17]. In particular, the data encoding circuit $U_{\boldsymbol{x}}$ uses the kernel encoding method, and the architecture of the trainable unitary $U(\boldsymbol{\theta})$ follows the multi-layer structure. The right panel of Figure 2 illustrates the implementation of data encoding circuit and the trainable circuit used in QNN. Three qubits are employed to build such two circuits. The data encoding circuits $U_{\boldsymbol{x}}$ 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 $R_Y(\boldsymbol{x})$ is $(\pi - \boldsymbol{x}_{i,1})(\pi - \boldsymbol{x}_{i,2})(\pi - \boldsymbol{x}_{i,3})$. 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 $X = \left(\begin{smallmatrix}0 & 1\\ 1 & 0\end{smallmatrix}\right)$.

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[3] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[5] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pages 6158–6169, 2019.

[6] Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *Advances in neural information processing systems*, pages 855–863, 2014.

[7] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in neural information processing systems*, pages 597–607, 2017.

[8] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252, 2019.

[9] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685, 2019.

[10] Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning polynomials with neural networks. In *International conference on machine learning*, pages 1908–1916, 2014.

[11] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195, 2017.

[12] Carlo Ciliberto, Mark Herbster, Alessandro Davide Ialongo, Massimiliano Pontil, Andrea Rocchetto, Simone Severini, and Leonard Wossnig. Quantum machine learning: a classical perspective. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2209):20170551, 2018.

[13] Vedran Dunjko and Hans J Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018.

[14] Aram W Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203, 2017.

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

[16] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.

[17] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209, 2019.

[18] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

[19] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.

[20] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[21] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[22] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2011.

[23] Michael J Bremner, Richard Jozsa, and Dan J Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 467(2126):459–472, 2011.

[24] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. Expressive power of parametrized quantum circuits. *Phys. Rev. Research*, 2:033125, Jul 2020.

[25] Carsten Blank, Daniel K Park, June-Koo Kevin Rhee, and Francesco Petruccione. Quantum classifier with tailored quantum kernel. *npj Quantum Information*, 6(1):1–7, 2020.

[26] Nathan Killoran, Thomas R Bromley, Juan Miguel Arrazola, Maria Schuld, Nicolás Quesada, and Seth Lloyd. Continuous-variable quantum neural networks. *Physical Review Research*, 1(3):033063, 2019.

[27] Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

[28] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

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

[30] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[31] Srinivasan Arunachalam and Ronald de Wolf. Guest column: A survey of quantum learning theory. *ACM SIGACT News*, 48(2):41–67, 2017.

[32] Alp Atici and Rocco A Servedio. Improved bounds on quantum learning algorithms. *Quantum Information Processing*, 4(5):355–386, 2005.

[33] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on computing*, 26(5):1411–1473, 1997.

[34] Rocco A Servedio and Steven J Gortler. Equivalences and separations between quantum and classical learnability.

*SIAM Journal on Computing*, 33(5):1067–1092, 2004.

[35] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[36] Srinivasan Arunachalam, Alex B Grilo, and Henry Yuen. Quantum statistical query learning. *arXiv preprint arXiv:2002.08240*, 2020.

[37] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[38] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.

[39] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

[40] Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole d'Eté de Probabilités de Saint-Flour XXXVIII-2008*, volume 2033. Springer Science & Business Media, 2011.

[41] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3922–3928. AAAI Press, 2017.

[42] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[43] Peter L Bartlett and Shahar Mendelson. Empirical minimization. *Probability theory and related fields*, 135(3):311–334, 2006.

[44] Kevin J. Sung, Matthew P. Harrigan, Nicholas C. Rubin, Zhang Jiang, Ryan Babbush, and Jarrod R. McClean. An exploration of practical optimizers for variational quantum algorithms on superconducting qubit processors, 2020.

[45] Martin Plesch and Časlav Brukner. Quantum-state preparation with universal gate decompositions. *Physical Review A*, 83(3):032302, 2011.

[46] Maria Schuld, Mark Fingerhuth, and Francesco Petruccione. Implementing a distance-based classifier with a quantum interference circuit. *EPL (Europhysics Letters)*, 119(6):60002, 2017.

[47] Maria Schuld, Alex Bocharov, Krysta M Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, 2020.

[48] CM Wilson, JS Otterbach, Nikolas Tezak, RS Smith, GE Crooks, and MP da Silva. Quantum kitchen sinks: An algorithm for machine learning on near-term quantum computers. *arXiv preprint arXiv:1806.08321*, 2018.

[49] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.

[50] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1724–1732. JMLR. org, 2017.

[51] Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[52] Di Wang, Changyou Chen, and Jinhui Xu. Differentially private empirical risk minimization with non-convex loss functions. In *International Conference on Machine Learning*, pages 6526–6535, 2019.

[53] Vitaly Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer and System Sciences*, 78(5):1444–1459, 2012.

[54] Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1265–1277. SIAM, 2017.

[55] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[56] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[57] Mario Motta, Chong Sun, Adrian TK Tan, Matthew J O'Rourke, Erika Ye, Austin J Minnich, Fernando GSL Brandão, and Garnet Kin-Lic Chan. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nature Physics*, 16(2):205–210, 2020.

[58] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5:4213, 2014.

[59] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K Fährmann, Barthélémy Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. *arXiv preprint arXiv:1910.01155*, 2019.

[60] Mo Zhou, Tianyi Liu, Yan Li, Dachao Lin, Enlu Zhou, and Tuo Zhao. Towards understanding the importance of noise in training neural networks. *arXiv preprint arXiv:1909.03172*, 2019.

[61] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. Implementable quantum classifier for nonlinear data. *arXiv preprint arXiv:1809.06056*, 2018.

[62] M Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost-function-dependent barren plateaus in shallow quantum neural networks. *arXiv preprint arXiv:2001.00550*, 2020.

[63] William Huggins, Piyush Patil, Bradley Mitchell, K Birgitta Whaley, and E Miles Stoudenmire. Towards quantum machine learning with tensor networks. *Quantum Science and technology*, 4(2):024001, 2019.

[64] Edward Grant, Marcello Benedetti, Shuxiang Cao, Andrew Hallam, Joshua Lockhart, Vid Stojevic, Andrew G Green, and Simone Severini. Hierarchical quantum classifiers. *arXiv preprint arXiv:1804.03680*, 2018.

[65] Davide Ferrari and Michele Amoretti. Demonstration of envariance and parity learning on the ibm 16 qubit processor. *arXiv preprint arXiv:1801.02363*, 2018.

[66] Diego Riste, Marcus P da Silva, Colm A Ryan, Andrew W Cross, Antonio D Córcoles, John A Smolin, Jay M Gambetta, Jerry M Chow, and Blake R Johnson. Demonstration of quantum advantage in machine learning. *npj Quantum Information*, 3(1):1–5, 2017.

[67] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, 5(1):1–9, 2019.

[68] Kunal Sharma, Sumeet Khatri, Marco Cerezo, and Patrick J Coles. Noise resilience of variational quantum compiling. *New Journal of Physics*, 22(4):043006, 2020.

The organization of the appendix is as follows. In Appendix A, we unify the notations used in the whole appendix. In Appendix B, we elaborate the implementation details of the quantum encoding circuit $U_{\boldsymbol{x}}$ and the trainable quantum circuit $U(\boldsymbol{\theta})$ used in QNN. In Appendix C, we quantifies 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 D, we exhibit the proof of Theorem 3, as the precondition to achieve utility bounds of QNN. In Appendix E, we exhibit the proofs details of Theorem 1 that achieves the utility bounds of QNN towards ERM. Next, in Appendix F, we prove Theorem 2, which shows that any QSQ oracle can be efficiently simulated by noisy QNN. Eventually, in Appendix G, we generalize all achieved results to a more general quantum channel.

## 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 $\{1, 2, ..., m\}$ as $[m]$. With a slight abuse of notations, we denote $\ell_b$ as the $b$-norm, while $\ell$ (without subscript) is the loss function. We denote the $\ell_p$ norm of $\mathbf{v}$ as $\|\mathbf{v}\|_p$. In particular, $\|\mathbf{v}\|$ refers to the $\ell_2$ norm. We use $O(\cdot)$ (or $\tilde{O}(\cdot)$) to denote the complexity bound (hide poly-logarithmic 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 $a \leq x_0 \leq b$.

## B. Implementation details of encoding circuit and trainable circuit of QNN

The selection of encoding circuits $U_{\boldsymbol{x}}$ 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 $U_{\boldsymbol{x}}$.** 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 $\boldsymbol{x}_i$ be $D_c$ with $i \in [n]$. The first category is the direct amplitude encoding [45–47, 61]. Specifically, the encoder circuit satisfies $U_{\boldsymbol{x}} : \mathcal{B}_i \to \frac{1}{\sqrt{B_s}} \sum_{b=1}^{B_s} \sum_{j=1}^{D_c} \hat{\boldsymbol{x}}_{b,j}^{(i)} |b\rangle |j\rangle$ with $\hat{\boldsymbol{x}}_{b,j}^{(i)} = \boldsymbol{x}_{b,j}^{(i)} / \|\boldsymbol{x}_{b,j}^{(i)}\|$. This method requires a low feature dimension $D_c$, since the quantum gates complexity to build $U_{\boldsymbol{x}}$ 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 arrangements, e.g., $U_{\boldsymbol{x}}(\mathcal{B}_i) = \sum_{b=1}^{B_s} (|b\rangle \langle b|) \otimes_{j=1}^{D_c} \mathrm{R_Y}(\boldsymbol{x}_{b,j}^{(i)})$. The third category is the dimension reduction method proposed by [48]. Specifically, instead of encoding $\mathcal{B}_i$, the amplitude or kernel encoder circuits $U_{\boldsymbol{x}}$ 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, 31, 36], which is broadly used in quantum learning theory. Specifically, the encoding circuit $U_{\boldsymbol{x}}$ is employed to prepare a quantum example $|\psi\rangle = \sum_{\boldsymbol{x} \in \{0,1\}^N} \sqrt{\mathcal{D}(\boldsymbol{x})} |\boldsymbol{x}, c(\boldsymbol{x})\rangle$ with $N = \lceil \log_2 D_c \rceil$, where $\mathcal{D}(\boldsymbol{x})$ is the data distribution over $\boldsymbol{x}$, $c(\boldsymbol{x})$ corresponds to the label of the bit-string $\boldsymbol{x}$ [31, 32]. In most cases, the distribution $\mathcal{D}(\boldsymbol{x})$ is uniform. Hence, the state $|\psi\rangle$ can be efficiently prepared by setting $B = 1$, and applying Hadamard gates and control-not gates [38] to the initial state $|0\rangle^{\otimes N+1}$.

**Trainable quantum circuits $U(\boldsymbol{\theta})$.** The trainable quantum circuits, a.k.a, parameterized quantum circuits [24, 49], 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-qubits gates. Each trainable layer can be decomposed into $U_l(\boldsymbol{\theta}_l) = (\bigotimes_{k=1}^{N} U_{l,k}(\boldsymbol{\theta}_l)) U_{eng}$, where $U_{l,k}(\boldsymbol{\theta}_l)$ represents the composition of trainable single-qubit gates and $U_{eng}$ refers to entanglement layer that contains two-qubits 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 [29, 62]. 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 multi-scale entanglement renormalization ansatz (MERA) [63]. The third class is the Hamiltonian based architecture, where the entanglement layer $U_{eng}$ refers to a specific Hamiltonian, e.g., the study [18] employs $U_{eng} = e^{-iHT}$ with $H = \sum_{j=1}^{N} a_j \mathrm{X}_j + \sum_{j=1}^{N} \sum_{k=1}^{j-1} J_{jk} \mathrm{Z}_i \mathrm{Z}_k$. Notably, almost all quantum approximate optimization algorithms follow the Hamiltonian based architecture [20].

### C. 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 1.** *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 $\mathcal{B}_i$ only contains the $i$-th input $\boldsymbol{x}_i$ 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 $|\phi_{\mathcal{B}_i}\rangle \xleftarrow{U_{\boldsymbol{x}}} \{\boldsymbol{x}_i\}$ refers to the quantum example corresponding to the classical input batch $\mathcal{B}_i$ (or equivalently, $\boldsymbol{x}_i$). The explicit form of the objective function is

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 \ , \tag{C1}$$

where $\hat{y}_i = \mathrm{Tr}(\Pi U(\boldsymbol{\theta})\rho_{\mathcal{B}_i}U(\boldsymbol{\theta})^\dagger)$ refers to the prediction of QNN given the $i$-th input $\boldsymbol{x}_i$, $U(\boldsymbol{\theta})$ is the trainable circuit, $\Pi$ is the employed two-outcome POVM, and $y_i$ is the true label of the $i$-th 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 1 and Eqn. (C1), the properties of the objective function $\mathcal{L}$ are summarized in the following lemma.

**Lemma 1.** *Following the notations in Eqn. (C1), $\mathcal{L}(\boldsymbol{\theta})$ is $S$-smooth with $S = (\frac{3}{2} + \lambda)d^2$ and $G$-Lipschitz with $G = d(1 + 3\pi\lambda)$. Assuming $\lambda \in (0, \frac{1}{3\pi}) \cup (\frac{1}{\pi}, \infty)$, $\mathcal{L}$ 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 $\lambda \in (0, \frac{1}{3\pi}) \cup (\frac{1}{\pi}, \infty)$. The objective function $\mathcal{L}$ satisfies PL condition with $\mu = \frac{(-1+\lambda\pi)^2}{1+\lambda d(3\pi)^2}$.*

In conjunction with the results of Lemma 2, 3, and 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

$$\nabla^2\mathcal{L}(\boldsymbol{\theta}) \preceq S\mathbb{I} \ , \tag{C2}$$

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 Eqn. (C2), 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 Eqn. (C1), the gradient for the parameter $\boldsymbol{\theta}_j$ is

$$\begin{aligned}
&\frac{\partial\mathcal{L}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}_j}\\
=&\frac{2}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)\frac{\partial\hat{y}_i}{\partial\boldsymbol{\theta}_j} + \frac{\lambda}{2}\frac{\partial\|\boldsymbol{\theta}\|_2^2}{\partial\boldsymbol{\theta}_j}\\
=&\frac{2}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)\frac{\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}}{2} + \lambda\boldsymbol{\theta}_j\\
\leq&1 + 3\lambda\pi \ ,
\end{aligned} \tag{C3}$$

where $\hat{y}_i^{(\pm_j)} = \mathrm{Tr}(\Pi U(\boldsymbol{\theta} \pm \frac{\pi}{2}\boldsymbol{e}_j)\rho_{\mathcal{B}_i}U(\boldsymbol{\theta} \pm \frac{\pi}{2}\boldsymbol{e}_j)^\dagger)$, the second equality employs the conclusion of the parameter shift rule with $\frac{\partial\hat{y}_i}{\partial\boldsymbol{\theta}_j} = \frac{\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}}{2}$ [18, 39], 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 $\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_j \partial \boldsymbol{\theta}_k}$ can be derived using the results of Eqn. (C3). In particular,

$$\begin{aligned}
\frac{\partial^2 \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_j \partial \boldsymbol{\theta}_k} = \frac{\partial(\frac{\partial \mathcal{L}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_j})}{\partial \boldsymbol{\theta}_k} &= \frac{1}{n}\sum_{i=1}^{n} \frac{\partial\left((\hat{y}_i - y_i)\left(\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}\right) + \lambda \boldsymbol{\theta}_j\right)}{\partial \boldsymbol{\theta}_k} \\
=& \frac{1}{n}\sum_{i=1}^{n}\left[\frac{\partial \hat{y}_i}{\partial \boldsymbol{\theta}_k}\left(\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}\right) + (\hat{y}_i - y_i)\frac{\partial\left(\hat{y}_i^{(+_j)} - \hat{y}_i^{(-_j)}\right)}{\partial \boldsymbol{\theta}_k} + \lambda\right] \\
\leq& \frac{3}{2} + \lambda \ ,
\end{aligned} \tag{C4}$$

where the first equality comes from the last equality of Eqn. (C3), 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 shit rule and $\hat{y}_i, \hat{y}_i^{(\pm_j)} \in [0, 1]$.

The result of Enq. (C4) implies that $\|\nabla^2 \mathcal{L}\|_2 \leq d\|\nabla^2 \mathcal{L}\|_\infty \leq d^2(\frac{3}{2} + \lambda)$. In conjunction with Eqn. (C2), 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 $f(\boldsymbol{x})$ is $G$-Lipschitz if it satisfies

$$|f(\boldsymbol{b}) - f(\boldsymbol{a})| \leq G\|\boldsymbol{b} - \boldsymbol{a}\| \ . \tag{C5}$$

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

$$f(\boldsymbol{b}) - f(\boldsymbol{a}) = \langle \nabla f(\boldsymbol{c}), \boldsymbol{b} - \boldsymbol{a} \rangle \ . \tag{C6}$$

Combining Enq. (C5) and (C6), the $G$-Lipschitz condition in Eqn. (C5) is equivalent to

$$|\langle \nabla f(\boldsymbol{c}), \boldsymbol{b} - \boldsymbol{a} \rangle| \leq G\|\boldsymbol{b} - \boldsymbol{a}\| \ . \tag{C7}$$

We now replace $f$, $\boldsymbol{b}$, and $\boldsymbol{a}$ used in Eqn. (C7) 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

$$\left|\left\langle \nabla \mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)} \right\rangle\right| \leq G\|\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)}\| \ , \tag{C8}$$

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

$$\left\langle \nabla \mathcal{L}(\boldsymbol{\theta}), \boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)} \right\rangle \leq \|\nabla \mathcal{L}(\boldsymbol{\theta})\| \, \|\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)}\| \leq d \, \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_\infty \, \|\boldsymbol{\theta}^{(1)} - \boldsymbol{\theta}^{(2)}\| \ . \tag{C9}$$

In conjunction with Eqn. (C8) and (C9), $G$-Lipschitz of $\mathcal{L}$ requests

$$d \, \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_\infty \leq G \ . \tag{C10}$$

By leveraging the result of Eqn. (C3) with $\nabla_j \mathcal{L}(\boldsymbol{\theta}) \leq 1 + 3\lambda\pi$, we obtain the upper bound of the left side in Eqn. (C10) is

$$d \, \|\nabla \mathcal{L}(\boldsymbol{\theta})\|_\infty \leq d(1 + 3\pi\lambda) \ . \tag{C11}$$

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: PL condition

*Proof of Lemma 4.* Recall the definition of Polyak-Lojasiewicz as formulated in Definition 1, it requires that the objective function $\mathcal{L}$ satisfies

$$\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 \geq 2\mu(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*) \ , \tag{C12}$$

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

$$\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 = \sum_{j=1}^{d}(\nabla_j\mathcal{L}(\boldsymbol{\theta}_j))^2 \geq \max_j(\nabla_j\mathcal{L}(\boldsymbol{\theta}))^2 \ . \tag{C13}$$

The lower bound of $\max_j(\nabla_j\mathcal{L}(\boldsymbol{\theta}))^2$ as shown in Eqn. (C13) follows

$$\max_j(\nabla_j\mathcal{L}(\boldsymbol{\theta}))^2 \geq \min_{\boldsymbol{\theta}_j\in[\pi,3\pi]}(-1+\lambda\boldsymbol{\theta}_j)^2 \ , \tag{C14}$$

where the last inequality is achieved by exploiting the last second line of Eqn. (C3), and the fact $\hat{y}_i, y_i, \hat{y}_i^{(\pm_j)} \in [0,1]$ and $\lambda > 0$, i.e.,

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

Combining the assumption $\lambda \in (0, \frac{1}{3\pi}) \cup (\frac{1}{\pi}, \infty)$ and the above results, the lower bound of Eqn. (C13) satisfies

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

We then derive the upper bound of the term $(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*)$ in Eqn. (C12). In particular, we have

$$\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^* \leq \mathcal{L}(\boldsymbol{\theta}) + 0 \leq 1 + \lambda d(3\pi)^2 \ , \tag{C15}$$

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

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

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

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

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

By combining Eqn. (C14) and (C15) with Eqn. (C12), we obtain the following relation

$$\|\nabla\mathcal{L}(\boldsymbol{\theta})\|^2 \geq (-1+\lambda\pi)^2 \geq 2\mu(1+\lambda d(3\pi)^2) \geq 2\mu(\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*) \ . \tag{C16}$$

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

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

□

## D. 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 $t$-th iteration, we define five constants with*

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

*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 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 Subsection D 1.

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

$$\begin{aligned} \nu^{(t)} &= (1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} \ , \\ (\sigma_i^{(t)})^2 &= \frac{-(1-\tilde{p})^2(\hat{Y}_i^{(t)})^2 + (1-\tilde{p})\left(1 - 2\tilde{p}\frac{\mathrm{Tr}(\Pi)}{D}\right)\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - \tilde{p}^2\frac{(\mathrm{Tr}(\Pi))^2}{D^2}}{K} \ . \end{aligned} \tag{D1}$$

*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*

$$\begin{aligned} \nu^{(t,\pm_j)} &= (1-\tilde{p})\hat{Y}_i^{(t,\pm_j)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} \ , \\ (\sigma_i^{(t,\pm_j)})^2 &= \frac{-(1-\tilde{p})^2(\hat{Y}_i^{(t,\pm_j)})^2 + (1-\tilde{p})\left(1 - 2\tilde{p}\frac{\mathrm{Tr}(\Pi)}{D}\right)\hat{Y}_i^{(t,\pm_j)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - \tilde{p}^2\frac{(\mathrm{Tr}(\Pi))^2}{D^2}}{K} \ . \end{aligned} \tag{D2}$$

*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 $t$-th iteration, given the input $\mathcal{B}_i$ and $K$ measurements, the estimated gradient for $j$-th parameter $\boldsymbol{\theta}_j$ of noisy QNN is

$$\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (\bar{Y}_i^{(t)} - Y_i)\left(\bar{Y}_i^{(t,+_j)} - \bar{Y}_i^{(t,-_j)}\right) + \lambda\boldsymbol{\theta}_j^{(t)} \ . \tag{D3}$$

Combining Lemma 5 and Eqn. (D3), the term $\Delta_i^{(t,j)} := \bar{Y}_i^{(t,+_j)} - \bar{Y}_i^{(t,-_j)}$ in Eqn. (D3) can be treated as the difference of two random variables. The term $(\bar{Y}_i^{(t)} - Y_i)$ in Eqn. (D3) 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

$$\Delta_i^{(t,j)} = (1-\tilde{p})(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)}) + \xi^{(t,j)} \ , \tag{D4}$$

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

$$(\bar{Y}_i^{(t)} - \bar{Y}_i^{(t)}) = (1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} + \xi^{(t)} - \bar{Y}_i^{(t)} \ , \tag{D5}$$

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

The reformulated terms as shown in Eqn. (D4) and Eqn. (D5) indicate that the estimated result $\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)})$ can be rewritten as

$$\begin{aligned}
&\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) \\
=&(\bar{Y}_i^{(t)} - Y_i)(\bar{Y}_i^{(t,+_j)} - \bar{Y}_i^{(t,-_j)}) + \lambda \boldsymbol{\theta}_j^{(t)} \\
=&\left((1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right)(1-\tilde{p})(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)}) + \left((1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right)\xi^{(t,j)} \\
&+ (1-\tilde{p})(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)})\xi^{(t)} + \xi^{(t)}\xi^{(t,j)} + \lambda \boldsymbol{\theta}_j^{(t)} \\
=&(1-\tilde{p})^2 \nabla_j \mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + (1-\tilde{p})\tilde{p}\left(\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right)(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)}) + (2\tilde{p} - \tilde{p}^2)\lambda \boldsymbol{\theta}_j^{(t)} \\
&+ (1-\tilde{p})(\hat{Y}^{(t,+_j)} - \hat{Y}^{(t,-_j)})\xi^{(t)} + \left((1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - Y_i\right)\xi^{(t,j)} + \xi^{(t)}\xi^{(t,j)} \ .
\end{aligned} \tag{D6}$$

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

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

where $\varsigma_i^{(t,j)} = C_{j,2}^{(i,t)}\xi_i^{(t)} + C_{j,3}^{(i,t)}\xi_i^{(t,j)} + \xi^{(t)}\xi_i^{(t,j)}$, 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

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

□

### 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 $\mathcal{N}_p(U_{L_Q}(\boldsymbol{\theta})...U_2(\boldsymbol{\theta})\mathcal{N}_p(U_1(\boldsymbol{\theta})\rho U_1(\boldsymbol{\theta})^\dagger)U_2(\boldsymbol{\theta})^\dagger...U_{L_Q}(\boldsymbol{\theta})^\dagger)=\mathcal{N}_{\tilde{p}}(U(\boldsymbol{\theta})\rho U(\boldsymbol{\theta})^\dagger)$, where $\rho$ 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

$$\mathcal{N}_p(\rho^{(1)})=(1-p)\rho^{(1)}+p\frac{\mathbb{I}_D}{D}\ , \tag{D7}$$

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

$$U_2(\boldsymbol{\theta})\mathcal{N}_p(\rho^{(1)})U_2(\boldsymbol{\theta})^\dagger=(1-p)\rho^{(2)}+p\frac{\mathbb{I}_D}{D}\ . \tag{D8}$$

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

$$\mathcal{N}_p(U_2(\boldsymbol{\theta})\mathcal{N}_p(\rho^{(1)})U_2(\boldsymbol{\theta})^\dagger)=(1-p)^2\rho^{(2)}+(1-p)p\frac{\mathbb{I}_D}{D}+p\frac{\mathbb{I}_D}{D}=(1-p)^2\rho^{(2)}+(1-(1-p)^2)\frac{\mathbb{I}_D}{D}\ . \tag{D9}$$

By induction, suppose at $k$-th step, the generated state is

$$\rho^{(k)}=(1-p)^l\rho^{(k)}+(1-(1-p)^k)\frac{\mathbb{I}_D}{D}\ . \tag{D10}$$

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

$$\rho^{(k+1)}=\mathcal{N}_p\left(U_{k+1}(\boldsymbol{\theta})\rho^{(k)}U_{k+1}(\boldsymbol{\theta})^\dagger\right)=(1-p)^{k+1}\rho^{(k+1)}+(1-(1-p)^{k+1})\frac{\mathbb{I}_D}{D}\ . \tag{D11}$$

According to the formula of 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.,

$$\mathcal{N}_{\tilde{p}}(\rho)=(1-p)^{L_Q}\rho^{(L_Q)}+(1-(1-p)^{L_Q})\frac{\mathbb{I}}{D}\ , \tag{D12}$$

where

$$\tilde{p}=1-(1-p)^{L_Q}\ . \tag{D13}$$

□

*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 $t$-th 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 $\Pi$ to measure such an ensemble $K$ times, the generated statistics (sample mean) is $\bar{Y}_i^{(t)}=\frac{1}{K}\sum_{k=1}^K V_k^{(t)}$, 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*

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* $V_k^{(t)}|z=2\sim\mathrm{Ber}(\frac{\mathrm{Tr}(\Pi)}{D})$ *with* $\frac{\mathrm{Tr}(\Pi)}{D}=\mathrm{Tr}(\Pi\gamma_{i,2}^{(t)})$ .

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

$$(1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} \text{ and } -(1-\tilde{p})^2(\hat{Y}_i^{(t)})^2 + (1-\tilde{p})\left(1-2\tilde{p}\frac{\mathrm{Tr}(\Pi)}{D}\right)\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - \tilde{p}^2\frac{(\mathrm{Tr}(\Pi))^2}{D^2} ,$$

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

$$\begin{aligned} \nu^{(t)} &= (1-\tilde{p})\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} , \\ (\sigma_i^{(t)})^2 &= \frac{-(1-\tilde{p})^2(\hat{Y}_i^{(t)})^2 + (1-\tilde{p})\left(1-2\tilde{p}\frac{\mathrm{Tr}(\Pi)}{D}\right)\hat{Y}_i^{(t)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - \tilde{p}^2\frac{(\mathrm{Tr}(\Pi))^2}{D^2}}{K} . \end{aligned} \tag{D14}$$

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

$$\begin{aligned} \nu^{(t,\pm_j)} &= (1-\tilde{p})\hat{Y}_i^{(t,\pm_j)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} , \\ (\sigma_i^{(t,\pm_j)})^2 &= \frac{-(1-\tilde{p})^2(\hat{Y}_i^{(t,\pm_j)})^2 + (1-\tilde{p})\left(1-2\tilde{p}\frac{\mathrm{Tr}(\Pi)}{D}\right)\hat{Y}_i^{(t,\pm_j)} + \tilde{p}\frac{\mathrm{Tr}(\Pi)}{D} - \tilde{p}^2\frac{(\mathrm{Tr}(\Pi))^2}{D^2}}{K} . \end{aligned} \tag{D15}$$

□

## E. 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 $R_1 \leq \tilde{O}\left(poly(\frac{d}{T(1-p)^{L_Q}}, \frac{d}{BK(1-p)^{L_Q}}, \frac{d}{(1-p)^{L_Q}})\right)$ and $R_2 \leq \tilde{O}\left(poly(d, \frac{1}{K^2B}, \frac{1}{(1-p)^{L_Q}})\right)$, 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 [50] 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 study [51], which claims that if a non-convex function satisfies PL condition, then every stationary point is the global minimum. Since the objective function used in QNN satisfies PL condition as shown in Lemma 1, we can effectively combine the PL condition with the result of $R_1$ to obtain the utility bound $R_2$.

*Proof of Theorem 5.* We employ the following two theorems to achieve Theorem 5, whose proofs are given in Subsections E 1 and E 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

$$R_1 \leq \tilde{O}\left(poly(\frac{d}{T(1-p)^{L_Q}}, \frac{d}{BK(1-p)^{L_Q}}, \frac{d}{(1-p)^{L_Q}})\right) \ . \tag{E1}$$

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

$$R_2 \leq \tilde{O}\left(poly(d, \frac{1}{K^2B}, \frac{1}{(1-p)^{L_Q}})\right) \ . \tag{E2}$$

□

## 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 Subsection E 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 $\frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j \bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right]$ with $S$ being the smooth parameter is upper bounded by*

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

*Proof of Theorem 6.* Recall that the optimization rule of noisy QNN at the $t$-th iteration follows

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta\nabla\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)}) \ . \tag{E3}$$

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

$$\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)}) \leq \langle\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)}), \boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\rangle + \frac{S}{2}\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\|^2 \ . \tag{E4}$$

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

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)}) \\
\leq&\langle\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)}), \boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\rangle + \frac{S}{2}\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^{(t)}\|^2 \\
=&-\frac{1}{S}\langle\nabla\mathcal{L}(\boldsymbol{\theta}^{(t+1)}), \nabla\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\rangle + \frac{1}{2S}\|\nabla\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\|^2 \\
=&-\frac{1}{S}\sum_{j=1}^{d}\left(\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t+1)})\nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right) + \frac{1}{2S}\sum_{j=1}^{d}\left(\nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2 \ .
\end{aligned} \tag{E5}$$

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

$$\nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)}) = \frac{1}{B}\sum_{i=1}^{B}(1-\tilde{p})^2\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + C_{j,2}^{(i,t)}\xi_i^{(t)} + C_{j,3}^{(i,t)}\xi_i^{(t,j)} + \xi_i^{(t)}\xi^{(t,j)} \ . \tag{E6}$$

Combining Eqn. (E5) with Eqn. (E6) and taking expectation over $\xi_i^{(t)}$ and $\xi_i^{(t,j)}$, we obtain

$$\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})]$$

$$
\begin{aligned}
&\leq -\frac{1}{S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 - \frac{1}{S}\sum_{j=1}^{d}\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\left(\frac{1}{B}\sum_{i=1}^{B}C_{j,1}^{(i,t)}\right) \\
&\quad -\frac{1}{S}\sum_{j=1}^{d}\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\frac{1}{B}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)}}\left[C_{j,2}^{(i,t)}\xi_i^{(t)}\right] - \frac{1}{S}\sum_{j=1}^{d}\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\frac{1}{B}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t,j)}}\left[C_{j,3}^{(i,t)}\xi_i^{(t,j)}\right] \\
&\quad -\frac{1}{S}\sum_{j=1}^{d}\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\frac{1}{B}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\xi_i^{(t)}\xi_i^{(t,j)}\right] + \frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right] \\
&\leq -\frac{1}{S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{G}{2S}\max_{i,j}C_{j,1}^{(i,t)} + \frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right]\ . \qquad \text{(E7)}
\end{aligned}
$$

The first inequality uses the result of Eqn. (E6). 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 $G$-Lipschitz property.

By leveraging Lemma 7, Eqn. (E7) can be further simplified as

$$
\begin{aligned}
&\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})] \\
&\leq -\frac{1}{S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{G}{2S}\max_{i,j}C_{j,1}^{(i,t)} + \frac{(1-\tilde{p})^4}{2SB}\|\nabla_j\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 \\
&\quad + \frac{(1-\tilde{p})^2G}{2S}\max_{i,j}C_{j,1}^{(i,t)} + \frac{d}{2S}\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2 + \frac{6dK+8d}{2SBK^2} \\
&\leq -\frac{1}{2S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2 + \frac{6dK+8d}{2SBK^2}\ . \qquad \text{(E8)}
\end{aligned}
$$

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

$$
\begin{aligned}
&\frac{G}{2S}\max_{i,j}C_{j,1}^{(i,t)} + \frac{(1-\tilde{p})^2G}{2S}\max_{i,j}C_{j,1}^{(i,t)} + \frac{d}{2S}\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2 \\
\leq&\frac{(1+(1-\tilde{p})^2)G}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda) + \frac{d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2 \\
\leq&\frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2\ , \qquad \text{(E9)}
\end{aligned}
$$

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

An equivalent representation of Eqn. (E8) is

$$
\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 \leq 2S\frac{\mathcal{L}(\boldsymbol{\theta}^{(t)}) - \mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})]}{(1-\tilde{p})^2} + \frac{(2\tilde{p}-\tilde{p}^2)(2G+d)(1+10\lambda)^2}{(1-\tilde{p})^2} + \frac{6dK+8d}{(1-\tilde{p})^2BK^2}\ . \qquad \text{(E10)}
$$

By induction, with summing over $t = 0, ..., T-1$ and taking expectation of Eqn. (E10), we obtain

$$
\begin{aligned}
&\mathbb{E}_t\left[\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2\right] \\
\leq&2S\frac{\mathcal{L}(\boldsymbol{\theta}^{(0)}) - \mathbb{E}_{\xi_i^{(T)},\xi_i^{(T,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(T)})]}{T(1-\tilde{p})^2} + \frac{(2\tilde{p}-\tilde{p}^2)(2G+d)(1+10\lambda)^2}{(1-\tilde{p})^2} + \frac{6dK+8d}{(1-\tilde{p})^2BK^2} \\
\leq&\frac{2S+2S\lambda d(3\pi)^2}{T(1-\tilde{p})^2} + \frac{(2\tilde{p}-\tilde{p}^2)(2G+d)(1+10\lambda)^2}{(1-\tilde{p})^2} + \frac{6dK+8d}{(1-\tilde{p})^2BK^2} \\
\leq&\frac{2S(1+90\lambda d)}{T(1-\tilde{p})^2} + \frac{(2\tilde{p}-\tilde{p}^2)(2G+d)(1+10\lambda)^2}{(1-\tilde{p})^2} + \frac{6dK+8d}{(1-\tilde{p})^2BK^2}\ , \qquad \text{(E11)}
\end{aligned}
$$

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

## 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 Eqn.(E8), i.e.,

$$
\begin{aligned}
&\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})-\mathcal{L}(\boldsymbol{\theta}^{(t)})]\\
\leq& -\frac{1}{2S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2+\frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2+\frac{6dK+8d}{2SBK^2}\ .
\end{aligned}
\tag{E12}
$$

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

$$
\begin{aligned}
&\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})-\mathcal{L}(\boldsymbol{\theta}^{(t)})]\\
\leq& -\frac{\mu(1-\tilde{p})^2}{S}(\mathcal{L}(\boldsymbol{\theta}^{(t)})-\mathcal{L}^*)+\frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2+\frac{6dK+8d}{2SBK^2}\ .
\end{aligned}
\tag{E13}
$$

An equivalent reformulation of Eqn. (E13) is

$$
\begin{aligned}
&\mathbb{E}_{\varsigma^{(t)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)})]-\mathcal{L}^*\\
\leq&\left(1-\frac{\mu(1-\tilde{p})^2}{S}\right)(\mathcal{L}(\boldsymbol{\theta}^{(t)})-\mathcal{L}^*)+\frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2+\frac{6dK+8d}{2SBK^2}\ .
\end{aligned}
\tag{E14}
$$

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

$$
\begin{aligned}
&\mathbb{E}_{\varsigma^{(t)}}[\mathcal{L}(\boldsymbol{\theta}^{(T)})]-\mathcal{L}^*\\
\leq&\left(1-\frac{\mu(1-\tilde{p})^2}{S}\right)^T(\mathcal{L}(\boldsymbol{\theta}^{(0)})-\mathcal{L}^*)+T\frac{2G+d}{2S}(2-\tilde{p})\tilde{p}(1+10\lambda)^2+T\frac{6dK+8d}{2SBK^2}\\
\leq&(1+90\lambda d)\exp\left(-\frac{\mu(1-\tilde{p})^2T}{S}\right)+T\frac{(2\tilde{p}-\tilde{p}^2)(G+2d)(1+10\lambda)^2BK^2+6dK+8d}{2SBK^2}\ ,
\end{aligned}
\tag{E15}
$$

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

$$
\nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})=\frac{1}{B}\sum_{i=1}^{B}(1-\tilde{p})^2\nabla_j\mathcal{L}_i(\boldsymbol{\theta}^{(t)})+C_{j,1}^{(i,t)}+C_{j,2}^{(i,t)}\xi_i^{(t)}+C_{j,3}^{(i,t)}\xi_i^{(t,j)}+\xi_i^{(t)}\xi^{(t,j)}\ .
\tag{E16}
$$

By using the above result, we obtain

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

$$
\begin{aligned}
&+\frac{1}{2SB^2}\sum_{j=1}^{d}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t,j)}}[(\xi_i^{(t,j)})^2]+\frac{1}{SB^2}\sum_{j=1}^{d}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\xi_i^{(t)}(\xi_i^{(t,j)})^2]+\\
&+\frac{1}{2SB^2}\sum_{j=1}^{d}\sum_{i=1}^{B}\mathbb{E}_{\xi_i^{(t)}\xi_i^{(t,j)}}[(\xi_i^{(t)})^2(\xi_i^{(t,j)})^2]\\
\leq&\frac{(1-\tilde{p})^4}{2S}\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2+\frac{(1-\tilde{p})^2G}{2S}\max_{i,j}C_{j,1}^{(i,t)}+\frac{d}{2S}\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2\\
&+\frac{dC_{j,4,\max}^{(t)}}{2SB}+\frac{dC_{j,5,\max}^{(t,j)}}{2SB}+\frac{dC_{j,4,\max}^{(t)}C_{j,5,\max}^{(t,j)}}{2SB}\ .
\end{aligned}
\tag{E17}
$$

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 $G$-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

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

and

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

we obtain

$$
\begin{aligned}
&\frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right]\\
\leq&\frac{(1-\tilde{p})^4}{2S}\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2+\frac{(1-\tilde{p})^2G}{2S}\max_{i,j}C_{j,1}^{(i,t)}+\frac{d}{2S}\max_{i,j}\left(C_{j,1}^{(i,t)}\right)^2+\frac{6dK+8d}{2SBK^2}\ .
\end{aligned}
\tag{E18}
$$

□

## F. Proof of Theorem 2

To ease the understanding, we first explain how to use variational quantum circuits of QNN to conduct a similar task of a QSQ oracle in Subsection F 1. We then complete the proof of Theorem 2 in Subsection F 2.

### 1. The similarity between the restricted QSQ oracle and QNN

Let us first recap the formal definition of the general QSQ learning model, i.e., the quantum example and the QSQ oracle.

**Definition 2** (Quantum example)**.** *Let $c^*:\{0,1\}^N\to\{0,1\}$ be an unknown concept sampled from a known concept class $\mathcal{C}\subseteq\{c:\{0,1\}^N\to\{0,1\}\}$. Denote the labeled examples as $(\boldsymbol{x},c^*(\boldsymbol{x}))$, where $\boldsymbol{x}$ is drawn from some unknown distribution $\mathcal{D}:\{0,1\}^N\to[0,1]$. The quantum example is defined as*

$$
|\psi_{c^*}\rangle=\sum_{\boldsymbol{x}\in\{0,1\}^N}\sqrt{\mathcal{D}(\boldsymbol{x})}\,|\boldsymbol{x}\rangle\,|c^*(\boldsymbol{x})\rangle\,.
\tag{F1}
$$

**Definition 3** (QSQ oracle, [36])**.** *A quantum statistical query oracle for some $c^*\in\mathcal{C}$ receives as inputs 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 $\alpha$ satisfying*

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

*where the quantum example $\psi_{c^*}$ is defined in Eqn. (F1).*

The efficiency of QSQ learning model is quantified by the $\varepsilon$-learnirng.

**Definition 4** ($\varepsilon$-learning). *Let $\mathcal{C} \subseteq \{c : \{0,1\}^N \to \{0,1\}\}$ be a concept class and $\mathcal{D} : \{0,1\}^N \to [0,1]$ be a distribution. We say that $\mathcal{C}$ can be $\varepsilon$-learned in the* QSQ *model with $Q$ queries, if there is an algorithm $\mathcal{A}$ such that for every $c^* \in \mathcal{C}$, $\mathcal{A}$ makes at most $Q$ queries to the* QSQ *oracle and outputs a hypothesis $h$ satisfying* $\Pr_{\boldsymbol{x}\sim\mathcal{D}}[h(\boldsymbol{x}) \neq c^*(\boldsymbol{x})] \leq \varepsilon$.

The above definitions indicate that a QSQ oracle takes the tuple $\{|\psi_{c^*}\rangle, \mathbb{M}, \tau\}$, and returns a classical result $\alpha$ that estimates the target result $\langle\psi_{c^*}|\mathbb{M}|\psi_{c^*}\rangle$ within the threshold $\tau$. Moreover, $\varepsilon$-learning implies that the QSQ algorithm adaptively chooses a sequence of $\{|\psi_{c^*}\rangle, \mathbb{M}_i, \tau_i\}_i$ and exploits the received feedback $\{\alpha_i\}_i$ to obtain the hypothesis $h$. As proved in [36], there exists a *poly*$(N)$ queries QSQ algorithm with tolerance $\tau = \tilde{O}(\varepsilon)$ that $\varepsilon$-learns some concept classes under the *uniform distribution*, while these concept classes are computational hard for SQ models.

**Lemma 8** (Modified from Lemma 4.2, 4.3, and 4.5 in [36]). *Let $\mathcal{C}$ be the concept class of parities, $k$-juntas, or poly$(N)$-sized DNFs (Disjunctive Normal Forms), then there exists a poly$(N)$-query* QSQ *algorithm with tolerance $\tau = \tilde{O}(\varepsilon)$ that $\varepsilon$-learns $\mathcal{C}$ under the uniform distribution. All of these concepts are computational hard for* SQ *models.*

Here we propose a restricted QSQ learning model, motivated by the result of Lemma 8 such that the quantum advantages achieved by QSQ learning model are based on the uniform distribution setting. In particular, we impose two restrictions on the tuple $\{|\psi_{c^*}\rangle, \mathbb{M}, \tau\}$ that is feeding into the QSQ oracle. As for the quantum example, we require $|\psi_{c^*}\rangle$ to follow the the uniform distribution, i.e., let $c^* : \{0,1\}^N \to \{0,1\}$ be an unknown concept sampled from a known concept class $\mathcal{C}$, the labeled examples as $(\boldsymbol{x}, c^*(\boldsymbol{x}))$ is drawn from the uniform distribution $\mathcal{D}$ with

$$|\psi_{c^*}\rangle = \sum_{\boldsymbol{x}\in\{0,1\}^N} \sqrt{\mathcal{D}(\boldsymbol{x})}\, |\boldsymbol{x}\rangle\, |c^*(\boldsymbol{x})\rangle = \sum_{\boldsymbol{x}\in\{0,1\}^N} \frac{1}{\sqrt{2^N}}\, |\boldsymbol{x}\rangle\, |c^*(\boldsymbol{x})\rangle \ . \tag{F2}$$

Second, we require that the observable $\mathbb{M}$ can be implemented by using at most *poly*$(N)$ single and two qubits gates. We define a restricted QSQ oracle that can only query these restricted quantum examples and observables.

**Definition 5** (Restricted QSQ oracle). *A restricted quantum statistical query oracle for some $c^* \in \mathcal{C}$ 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 $\mathrm{Tr}(\mathbb{M}) \leq 1$ as inputs, and outputs a number $\alpha$ satisfying*

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

*where $|\psi_{c^*}\rangle$ is the restricted quantum example defined in Eqn. (F2) and the observable $\mathbb{M}$ can be implemented using at most $O(poly(N))$ single and two qubits gates.*

Supported by Definition 5, the criteria to quantify the efficiency of the restricted QSQ learning model is as follows.

**Definition 6** (restricted $\varepsilon$-learning). *Let $\mathcal{C} \subseteq \{c : \{0,1\}^N \to \{0,1\}\}$ be a concept class and $\mathcal{D}$ be a uniform distribution. We say that $\mathcal{C}$ can be $\varepsilon$-learned in the restricted* QSQ *model with $Q$ queries, if there is an algorithm $\mathcal{A}$ such that for every $c^* \in \mathcal{C}$, $\mathcal{A}$ makes at most $Q$ queries to the restricted* QSQ *oracle and outputs a hypothesis $h$ satisfying* $\Pr_{\boldsymbol{x}\sim\mathcal{D}}[h(\boldsymbol{x}) \neq c^*(\boldsymbol{x})] \leq \varepsilon$.

We remark that the proposed restricted QSQ learning model can also be used to achieve quantum advantages in learning parities, $k$-juntas, or *poly*$(N)$-sized DNFs, supported by Lemma 8 and the fact that the gate complexity to implement the related $\mathbb{M}$ is *poly*$(N)$ [36].

In the following, we will demonstrate that the quantum examples and observables of the restricted QSQ oracle can be effectively represented by the variational quantum circuits used in QNN. In particular, the flexibility of QNN allows us to specify a quantum observable as the quantum measurement conducted in the variational quantum circuit [49, 62, 64]. This implies that the observable $\mathbb{M}$ that can be constructed by $O(poly(N))$ quantum gates, as formulated in Definition 5, can be effectively represented by QNN. Moreover, the restricted quantum example given in Eqn. (F2) can also be efficiently prepared by the quantum encoding circuit $U_{\boldsymbol{x}}$, since $|\psi_{c^*}\rangle$ only involves the bit-string encoding and its probability amplitude satisfies $\sqrt{\mathcal{D}(\boldsymbol{x})} = \frac{1}{\sqrt{2^N}}$ for all $\boldsymbol{x}$. As explained in Appendix B, the flexibility of $U_{\boldsymbol{x}}$ allows the efficacy to prepare the restricted quantum example by leveraging Hadamard gates and two qubits gates, e.g., CNOT gates. For example, the gate complexity of $U_{\boldsymbol{x}}$ to prepare $|\psi_{c^*}\rangle$ that is employed to accomplish parity learning is at most $2N$, where $N$ Hadamard gates separately apply to $N$ qubits, followed by at most $N$ CNOT gates to label $c^*(\boldsymbol{x})$ [65, 66].

The efficiency of exploiting the variational quantum circuit to simulate the restricted quantum example $|\psi_{c^*}\rangle$ and $\mathbb{M}$ ensures the similar statistical property between noisy QNN and the restricted QSQ oracle. Specifically, when the number of measurements goes to infinity, the noisy QNN returns a classical result that estimates the target result within the a certain error. Let the encoding circuit $U_{\boldsymbol{x}}$ prepare the state $|\psi_{c^*}\rangle$ and the quantum measurement constructed from $\mathbb{M}$. Under the depolarization noise, the expectation value of quantum measurements of the noisy QNN yields

$$\tilde{\nu} = \mathrm{Tr}(\mathbb{M}\mathcal{N}_{\tilde{p}}(|\psi_{c^*}\rangle\langle\psi_{c^*}|)) = (1-\tilde{p})\nu + \tilde{p}\frac{\mathrm{Tr}(\mathbb{M})}{2^{N+1}} \ , \tag{F3}$$

where $\tilde{p}$ is defined in Eqn. (D13) and $\nu = \langle\psi_{c^*}|\mathbb{M}|\psi_{c^*}\rangle$, supported by Lemma 6. Combining Definition 6 and Eqn. (F3), it is easy to see the similar behavior between a QSQ oracle and a noisy QNN, where both of them can only output the estimates of statistical properties of the labeled examples.

We end this subsection by addressing the potential to apply noisy QNN to simulate the general QSQ oracle. Recall that a major difference between the restricted and general setting is the uniform distribution setting exerting on the quantum example. This restriction ensures that $U_{\boldsymbol{x}}$ can efficiently load the quantum example into QNN. Besides the uniform setting, $U_{\boldsymbol{x}}$ has the capability of loading quantum example under certain non-uniform distribution $\mathcal{D}$ with $O(poly(N))$ gate complexity. A representative example is quantum generative adversarial network, which encodes the generic probability distributions that implicitly given by data samples into quantum states [67]. In other words, it is possible to employ noisy QNN to simulate a more general QSQ oracle that covers a large class of distributions. However, connecting noisy QNN with the restricted QSQ oracle in Definition 6 is sufficient to answer the main focus of this study, i.e., what concept classes can be efficiently learned by noisy QNN that are computational hard for classical models, since the concept classes that separates QSQ learning with SQ learning are all based on the uniform distribution setting.

### 2. proof of Theorem 2

*Proof of Theorem 2.* Following Definition 6, we observe that the restricted QSQ algorithm can be efficiently simulated by QNN once each query $\{|\psi_{c^*}\rangle, \mathbb{M}_i, \tau_i\}_i$ can be efficiently simulated by the variational quantum circuits of QNN, i.e., given $\mathbb{M}_i$, and $\tau_i$, the quantum circuit returns an estimated result that $\varepsilon$-close to $\nu = \langle\psi_{c^*}|\mathbb{M}|\psi_{c^*}\rangle$ by querying $|\psi_{c^*}\rangle$ at most $O(poly(N))$ times. In the following, we exploit the results obtained in Subsection F 1 to prove that each query to the restricted 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 $\{|\psi_{c^*}\rangle, \mathbb{M}, \tau\}$, where $|\psi_{c^*}\rangle$ is the restricted quantum example given in Eqn. (F2). In this way, as shown in Eqn. (F3), the expectation value of quantum measurements for noisy QNN under the depolarization noise setting $\mathcal{N}_{\tilde{p}}$ yields $\tilde{\nu} = (1-\tilde{p})\nu + \frac{\tilde{p}\,\mathrm{Tr}(\mathbb{M})}{2^{N+1}}$ with $\nu = \langle\psi_{c^*}|\mathbb{M}|\psi_{c^*}\rangle$. In addition, the measurement outcome $V_k$ is a random variable that satisfies $V_k \sim \mathrm{Ber}(\tilde{\nu})$.

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

$$\Pr\left(\left|\frac{1}{K}\sum_{i=1}^{K} V_k - \tilde{\nu}\right| \geq \frac{\delta}{2}\right) \leq 2\exp(-\delta^2 K/2) \ . \tag{F4}$$

Denote $b = 2\exp(-\delta^2 K/2)$. Eqn. (F4) implies that, when $K = \frac{2\ln(2/b)}{\delta^2}$, with probability at least $1-b$, we have $|\frac{1}{K}\sum_{i=1}^{K} V_k - \tilde{\nu}| \leq \delta/2$.

Moreover, supported by Eqn. (F3), the distance between the result $\nu$ (i.e., the target value of the restricted QSQ oracle) and the shifted expectation value $\tilde{\nu}$ follows

$$|\nu - \tilde{\nu}| \leq \tilde{p}\nu + \tilde{p}\frac{\mathrm{Tr}(\mathbb{M})}{2^{N+1}} \ . \tag{F5}$$

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

$$\left|\frac{1}{K}\sum_{k=1}^{K} V_k - \nu\right| = \left|\frac{1}{K}\sum_{k=1}^{K} V_k - \tilde{\nu} + \tilde{\nu} - \nu\right| \leq \tilde{p}\nu + \tilde{p}\frac{\mathrm{Tr}(\mathbb{M})}{2^{N+1}} + \frac{\delta}{2} \leq \tilde{p}(\nu + \frac{1}{2^{N+1}}) + \frac{\delta}{2} \ , \tag{F6}$$

where the last equality uses $\mathrm{Tr}(\mathbb{M}) \leq 1$ given in Definition 5.

Note that, to guarantee that QNN can simulate the restricted QSQ oracle as formulated in Definition 5, the rightest term in Eqn. (F6) should be upper bounded by $\tau$, i.e.,

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

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

$$\delta \leq 2(\tau - \tilde{p}\frac{5}{4}) \ .$$

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

$$\left| \frac{1}{K}\sum_{k=1}^{K} V_k - \nu \right| \leq \tau \ , \tag{F7}$$

which accords with the output of the restricted $\mathsf{QSQ}$ oracle.

We now quantify the number of measurements $K$ to promise Eqn. (F7). Recall $K = \frac{2\ln(2/b)}{\delta^2}$. By employing the explicit form of $\delta$, we obtain

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

The achieved result indicates that the successful probability of noisy QNN (i.e., $1-2b$) to estimate the restricted $\mathsf{QSQ}$ oracle can be exponentially improved by linearly increasing the number of measurements. Moreover, the term $\frac{1}{(\tau - \tilde{p}\frac{5}{4})}$ 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 restricted $\mathsf{QSQ}$ oracle. □

## G. Generalization the results to more general quantum channels

Here we generalize the achieved results in 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

$$\begin{aligned}&\mathcal{E}_{p_1}(U_L(\boldsymbol{\theta})...U_2(\boldsymbol{\theta})\mathcal{E}_{p_1}(U_1(\boldsymbol{\theta})\rho U_1(\boldsymbol{\theta})^\dagger)U_2(\boldsymbol{\theta})^\dagger...U_L(\boldsymbol{\theta})^\dagger)\\ =&(1-p_1)^{LQ}\left(U(\boldsymbol{\theta})U_{\boldsymbol{x}}\right)\rho\left(U(\boldsymbol{\theta})U_{\boldsymbol{x}}\right)^\dagger + p_2'\kappa + p_3^{LQ}\frac{\mathbb{I}_D}{D} \ ,\end{aligned} \tag{G1}$$

where $(1-p_1)^{LQ} + p_2' + p_3^{LQ} = 1$, and $\kappa$ is a mixed state that can either be correlated or uncorrelated with $\left(U(\boldsymbol{\theta})U_{\boldsymbol{x}}\right)\rho\left(U(\boldsymbol{\theta})U_{\boldsymbol{x}}\right)^\dagger$. Without confusion, we set $\tilde{p} = 1-(1-p_1)^{LQ}$. 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 [38, 68].

The outline of this section is as follows. In Subsection G 1, we discuss the utility bounds of QNN under ERM. Then, in Subsection G 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 Eqn. (G1), 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 depolarization channel, the sample mean $\bar{Y}_i^{(t)}$ or $\bar{Y}_i^{(t,\pm_j)}$ is a random variable follows certain distribution. In particular, following the notations used in Theorem 4, the mean and variance of $\bar{Y}_i^{(t)}$ follows

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

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.,

$$\nabla_j \bar{\mathcal{L}}_i(\boldsymbol{\theta}^{(t)}) = (1-\tilde{p})^2 \nabla_j \mathcal{L}_i(\boldsymbol{\theta}^{(t)}) + C_{j,1}^{(i,t)} + \varsigma_i^{(t,j)} , \tag{G2}$$

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

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

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 Eqn.(E7), with taking expectation over $\xi_i^{(t)}$ and $\xi_i^{(t,j)}$, we obtain

$$\begin{aligned} & \mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})] \\ \leq & -\frac{1}{S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{G}{2S}\left(\frac{1}{B}\sum_{i=1}^{B} C_{j,1}^{(i,t)}\right) + \frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right] , \end{aligned} \tag{G3}$$

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

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

$$\begin{aligned} \frac{1}{2S}\sum_{j=1}^{d}\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}\left[\left(\nabla_j\bar{\mathcal{L}}(\boldsymbol{\theta}^{(t)})\right)^2\right] \leq & \frac{(1-\tilde{p})^4}{2S}\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{(1-\tilde{p})^2 G}{2SB}\sum_{i=1}^{B} C_1^{(i,t)} \\ & + \frac{d}{2SB^2}\left(\sum_{i=1}^{B} C_1^{(i,t)}\right)^2 + d\frac{\sigma_{\max}^{(t)} + \sigma_{\max}^{(t,j)} + \sigma_{\max}^{(t)}\sigma_{\max}^{(t,j)}}{SB} , \end{aligned} \tag{G4}$$

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

In conjunction with the above two equations, we achieve

$$\begin{aligned} & \mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})] \\ \leq & -\frac{1}{2S}(1-\tilde{p})^2\|\nabla\mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 + \frac{(2G+d)(5+3(1-(1-\tilde{p})^2)\lambda\pi)}{2S} + \frac{6dK+9d}{SBK^2} , \end{aligned} \tag{G5}$$

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

$$\|\nabla \mathcal{L}(\boldsymbol{\theta}^{(t)})\|^2 \leq 2S\frac{1+9\lambda d}{T(1-\tilde{p})^2} + \frac{(2G+d)(5+3(1-(1-\tilde{p})^2)\lambda\pi)}{(1-\tilde{p})^2} + \frac{12dK+18d}{(1-\tilde{p})^2BK^2} \ . \tag{G6}$$

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

$$R_1 \leq \tilde{O}\left(\frac{1}{(1-\tilde{p})^2}, d, \frac{1}{BK}\right) \ . \tag{G7}$$

**Utility bound $R_2$.** With combining Eqn. (G5) and PL condition, we obtain

$$\begin{aligned} &\mathbb{E}_{\xi_i^{(t)},\xi_i^{(t,j)}}[\mathcal{L}(\boldsymbol{\theta}^{(t+1)}) - \mathcal{L}(\boldsymbol{\theta}^{(t)})] \\ \leq& -\frac{\mu(1-\tilde{p})^2}{S}(\mathcal{L}(\boldsymbol{\theta}^{(t)}) - \mathcal{L}^*) + \frac{(2G+d)(5+3(1-(1-\tilde{p})^2)\lambda\pi)}{2S} + \frac{6dK+9d}{SBK^2} \ . \end{aligned} \tag{G8}$$

After rewriting and induction, we have

$$\mathbb{E}_{\varsigma^{(t)}}[\mathcal{L}(\boldsymbol{\theta}^{(T)})] - \mathcal{L}^* \leq 15\lambda d \exp\left(-\frac{\mu(1-\tilde{p})^2T}{S}\right) + T\frac{(2G+d)(5+3(1-(1-\tilde{p})^2)\lambda\pi)}{2S} + T\frac{6dK+9d}{SBK^2} \ . \tag{G9}$$

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

$$R_2 \leq O\left(\frac{1}{(1-\tilde{p})^2}, \frac{1}{SBK^2}, d\right) \ . \tag{G10}$$

## 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 $1 - 2\exp(-\delta^2 n/2)$,

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

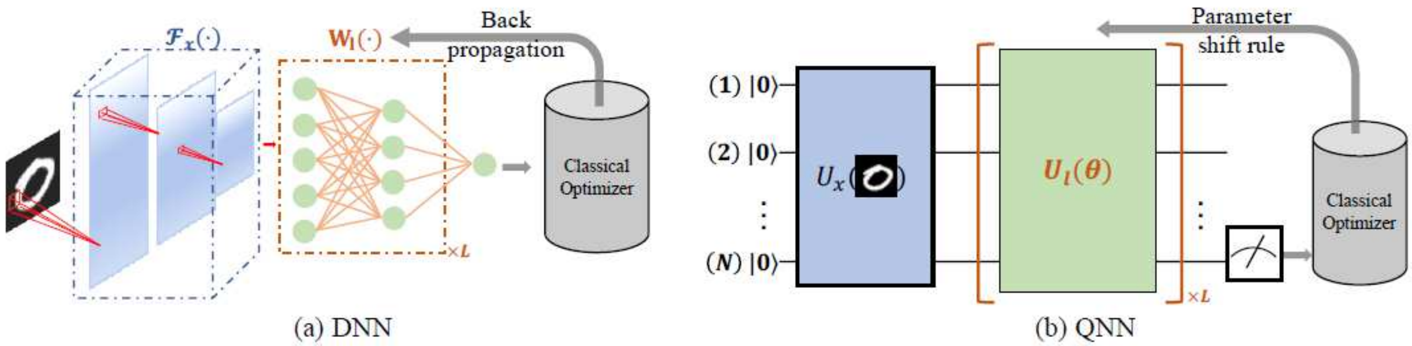
With setting $\delta = 2(\tau - \tilde{p}\nu - p_2' - p_3^{L_Q}/D)$, the relation between the number of measurements $K$ and the successful probability $b$ obeys

$$\Pr\left(\left|\frac{1}{K}\sum_{k=1}^{K} V_k - \tilde{\nu}\right| \geq \left(\tau - \tilde{p}\nu - p_2' - \frac{p_3^{L_Q}}{D}\right)\right) \leq 2\exp\left(-2\left(\tau - \tilde{p}\nu - p_2' - \frac{p_3^{L_Q}}{D}\right)^2 K\right) = b \ . \tag{G11}$$

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

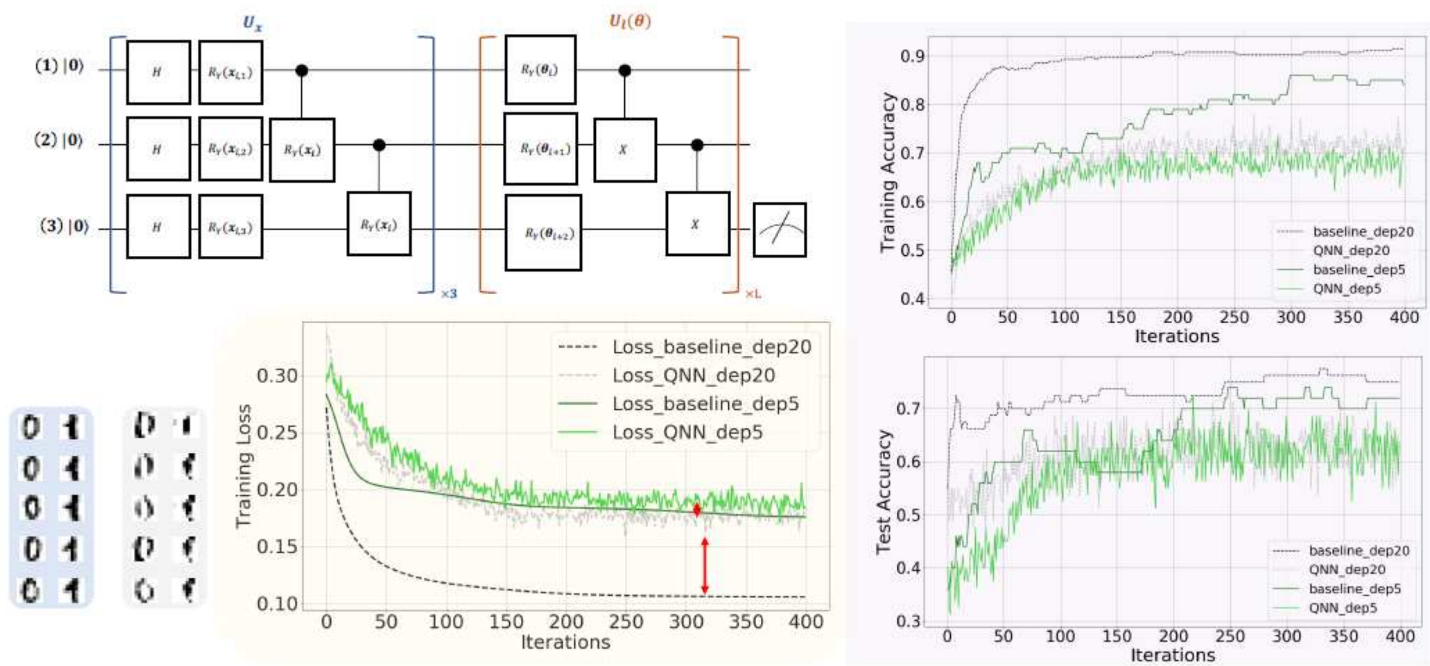
$$K = \frac{\ln\left(\frac{2}{b}\right)}{4\left(\tau - \tilde{p}\nu - p_2' - \frac{p_3^{L_Q}}{D}\right)^2} \ . \tag{G12}$$

# Figures




## Figure 1

Illustration of DNN and QNN. The left and right panel shows DNN and QNN, respectively. For DNN, the feature embedding layers Fx(.), which contains a sequence of operations with the arbitrary combination such as convolution and attention, maps the input '0' to the feature space. Wl(.) is the l-th fully-connected layer. For QNN, an encoding quantum circuit Ux maps the classical input '0' to the quantum feature space. Ul(θ) is the l-th trainable quantum circuit. Classical information for optimization is extracted by quantum measurements.




## Figure 2

The implementation of quantum circuits and the simulation results on hand-written 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 demonstrates the implementation of data encoding circuit and trainable circuit used in QNN. The label ‘x3’ and ‘xL’ 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 hyper-parameters settings. In particular, the label ‘Loss_baseline_dep20’ (‘Loss_baseline_dep5’) refers to the obtained loss under the setting L = 20 (L = 5), p = 0, and K ! 1, where L, p, and K refer to the circuit depth, depolarization rate, the number of measurements to estimate expectation value used in QNN, respectively. Similarly, the label ‘Loss_QNN_dep20’ (‘Loss_QNN_dep5’) refers to the obtained loss of QNN under the setting L = 20 (L = 5), p = 0:0025, K = 20. The upper right and lower right panels separately demonstrate the training accuracy and test accuracy of the quantum classifiers with different hyper-parameters settings.