

Power of Quantum Generative Learning

Yuxuan Du,¹ Zhuozhuo Tu,² Bujiao Wu,^{3, 4} Xiao Yuan,^{3, 4} and Dacheng Tao¹

¹*JD Explore Academy, Beijing 101111, China*

²*School of Computer Science, The University of Sydney, Darlington, NSW 2008, Australia*

³*Center on Frontiers of Computing Studies, Peking University, Beijing 100871, China*

⁴*School of Computer Science, Peking University, Beijing 100871, China*

The intrinsic probabilistic nature of quantum mechanics invokes endeavors of designing quantum generative learning models (QGLMs). Despite the empirical achievements, the foundations and the potential advantages of QGLMs remain largely obscure. To narrow this knowledge gap, here we explore the generalization property of QGLMs, the capability to extend the model from learned to unknown data. We consider two prototypical QGLMs, quantum circuit Born machines and quantum generative adversarial networks, and explicitly give their generalization bounds. The result identifies superiorities of QGLMs over classical methods when quantum devices can directly access the target distribution and quantum kernels are employed. We further employ these generalization bounds to exhibit potential advantages in quantum state preparation and Hamiltonian learning. Numerical results of QGLMs in loading Gaussian distribution and estimating ground states of parameterized Hamiltonians accord with the theoretical analysis. Our work opens the avenue for quantitatively understanding the power of quantum generative learning models.

I. INTRODUCTION

Learning is a generative activity that constructs its own interpretations of information and draws inferences on them [1]. This comprehensive philosophy sculpts a substantial subject in artificial intelligence, which is designing powerful generative learning models (GLMs) [2, 3] to capture the distribution \mathbb{Q} describing the real-world data (see Fig. 1(a)). Concisely, a fundamental concept behind GLMs is estimating \mathbb{Q} by a tunable probability distribution \mathbb{P}_{θ} . In the past decades, a plethora of GLMs, e.g., the Helmholtz machine [4], variational auto-encoders [5], and generative adversarial networks (GANs) [6, 7], have been proposed. Attributed to the efficacy and flexibility of handling \mathbb{P}_{θ} , these GLMs have been broadly applied to myriad scientific domains and gained remarkable success, including image synthesis and editing [8–10], medical imaging [11], molecule optimization [12, 13], and quantum computing [14–19]. Despite the wide success, their limitations have recently been recognized from different perspectives, such as expensive runtime and inferior performance towards complex datasets [20–25].

Envisioned by the intrinsic probabilistic nature of quantum mechanics and the superior power of quantum computers [26–28], quantum generative learning models (QGLMs) are widely believed to enhance the ability of GLMs. Concrete evidence has been provided by Refs. [29, 30], showing that fault-tolerant based QGLMs could surpass GLMs with stronger model expressivity and runtime speedups. Since fault-tolerant quantum computing is still in absence, attention has recently shifted to design QGLMs that can be efficiently carried out on noisy intermediate-scale quantum (NISQ) machines [31–33] with advantages on specific tasks [34–37]. For this purpose, a leading strategy is constructing QGLMs through variational quantum algorithms (VQAs) [38, 39]. Depending on whether the probability distribution \mathbb{P}_{θ} is explicitly formulated or not [40], these QGLMs can be mainly

divided into explicit QGLMs and implicit QGLMs. Primary protocols of these two classes are *quantum circuit Born machines* (QCBMs) [41–44] and *quantum generative adversarial networks* (QGANs) [45–54]. Experimental studies have demonstrated the feasibility of QGLMs for different learning tasks, e.g., image generation [47, 55], state approximation [34, 56], and drug design [57, 58].

A crucial vein in quantum machine learning [27] is comprehending potential advantages of quantum learners from the perspective of *generalization*, which quantifies the capability to extend the model from learned to unknown data [59–62]. In sharp contrast to quantum discriminative learning [63–69], prior literature related to the generalization of QGLMs is scarce and negative [70, 71]. Concretely, Ref. [71] pointed out the hardness for both QCBM and GLMs to efficiently learn the output distributions of local quantum circuits under the statistical query access. In this regard, two key questions are naturally invoked: how to quantify the generalization of different QGLMs in a generic way? And is there any potential advantage of QGLMs in solving nontrivial learning problems?

To shrink the above knowledge gap, here we establish the generalization theory of QGLMs with the maximum mean discrepancy (MMD) loss [72] and utilize these results to affirm potential merits of QGLMs in practical applications. The attention on MMD loss is because it is an efficient measure to evaluate the difference between \mathbb{P}_{θ} and \mathbb{Q} without the curse of dimensionality issue. Our first result is unveiling the power of QCBMs and QGANs through the lens of the statistical learning theory [73]. Specifically, when \mathbb{Q} is discrete, we prove that the generalization error of QCBMs scales with $\tilde{O}(\sqrt{1/n + 1/m})$, where n and m refer to the number of examples sampled from \mathbb{P}_{θ} and \mathbb{Q} . In addition, when \mathbb{Q} can be efficiently prepared by quantum circuits (see Fig. 1(b)), quantum kernels enable QCBMs to attain a strictly lower generalization error than classical kernels. When \mathbb{Q} is continuous, we prove that the generalization error of QGAN is upper

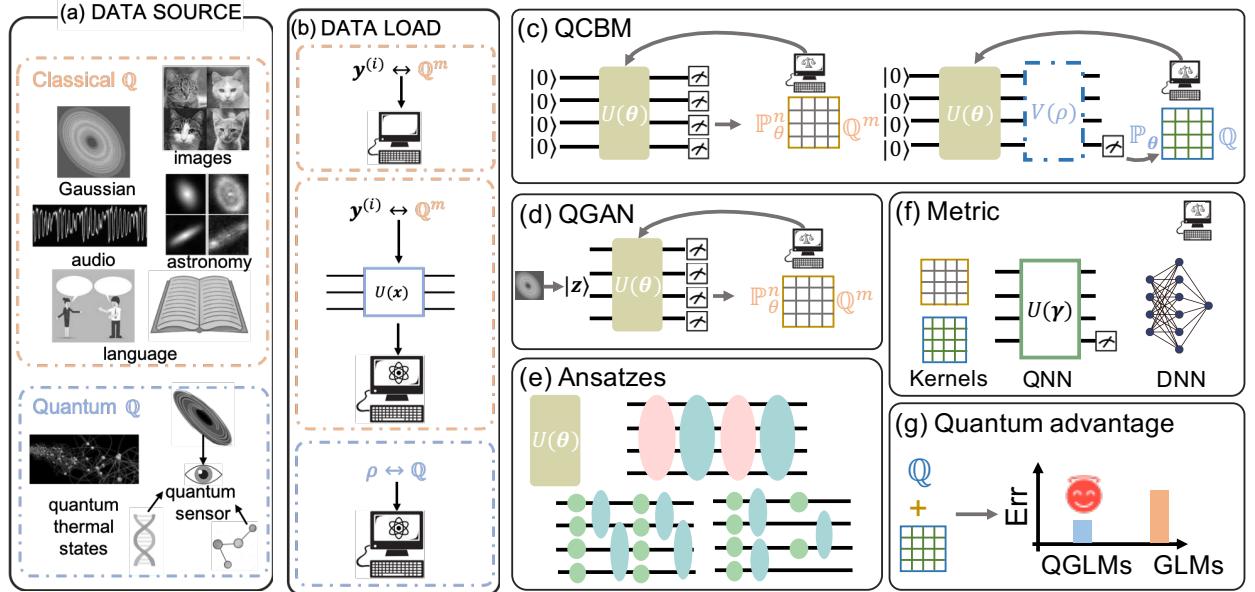


FIG. 1: The paradigm of quantum generative learning models. (a) The data explored in generative learning includes both classical and quantum scenarios. (b) The approaches of QGLMs to access target data distribution \mathbb{Q} . When \mathbb{Q} is classical, QGLMs operate its samples on the classical side or encode its samples into quantum circuits. When \mathbb{Q} is quantum, QGLMs may directly access it without sampling. (c) The paradigm of QCBMs with MMD loss. The left and right panels depict the setup of classical and quantum kernels, respectively. (d) The scheme of QGANs with MMD loss for the continuous distribution \mathbb{Q} . (e) QAOA, hardware-efficient, and tensor-network based Ansätze are covered by $\hat{U}(\boldsymbol{\theta})$ in Eq. (2). (f) The metrics exploited in QGLMs to measure the discrepancy between the generated and target distributions. (g) When \mathbb{Q} and $k(\cdot, \cdot)$ are both quantum, QGLMs may attain generalization advantages over classical GLMs.

bounded by $\tilde{O}(\sqrt{1/n + 1/m} + Nd^k \sqrt{N_{gt} N_{ge}}/n)$, where d , k , N_{ge} , and N_{gt} refer to the dimension of a qudit, the type of quantum gates, the number of encoding gates, and the number of trainable parameters, respectively. This bound explicitly reveals how the encoding strategy and the adopted Ansatz affect the generalization. Taken together, sufficient expressivity and large examples allows $\mathbb{P}_{\theta} \approx \mathbb{Q}$ for QGLMs. In light of this observation, we exhibit the potential advantage of QCBMs and QGANs in quantum state preparation and parameterized Hamiltonian learning.

The remainder of this study is organized as follows. In Secs. II & III, we theoretically explore the power of QCBM and QGAN through the lens of statistical learning theory, respectively. Then, in Sec. IV, we elaborate how QGLMs can advance quantum state preparation and Hamiltonian learning. Subsequently, we conduct numerical simulations to validate our claims about QCBMs and QGANs in Sec. V. We conclude this study in Sec. VI.

II. GENERALIZATION OF QCBM

We first study the generalization property of QCBM. As shown in Fig. 1(a), QCBM applies an N -qudit Ansatz $\hat{U}(\boldsymbol{\theta})$ to a fixed input state $\rho_0 = (|0\rangle\langle 0|)^{\otimes N}$ to form the parameterized distribution $\mathbb{P}_{\theta} \in \mathbb{P}_{\Theta}$, where $\boldsymbol{\theta}$ denotes trainable parameters living in the parameter space Θ .

The probability of sampling $i \in [d^N]$ over the discrete distribution \mathbb{P}_{θ} yields

$$\mathbb{P}_{\theta}(i) = \text{Tr}(\Pi_i \hat{U}(\boldsymbol{\theta}) \rho_0 \hat{U}(\boldsymbol{\theta})^\dagger), \quad (1)$$

where $\Pi_i = |i\rangle\langle i|$ refers to the projector of the computational basis i . Given n reference examples $\{\mathbf{x}^{(j)}\}_{j=1}^n$ sampled from \mathbb{P}_{θ} , its empirical distribution is defined as $\mathbb{P}_{\theta}^n(i) = \sum_{j=1}^n \delta_{\mathbf{x}^{(j)}}(i)/n$ with $\delta_{(.)}(\cdot)$ being the indicator. Throughout the whole study, the Ansatz employed in QCBMs takes the generic form

$$\hat{U}(\boldsymbol{\theta}) = \prod_{l=1}^{N_g} \hat{U}_l(\boldsymbol{\theta}), \quad (2)$$

where $\hat{U}_l(\boldsymbol{\theta}) \in \mathcal{U}(d^k)$ refers to the l -th quantum gate operated with at most k -qudits with $k \leq N$, and $\mathcal{U}(d^k)$ denotes the unitary group in dimension d^k ($d = 2$ for qubits). Note that for simplicity, the definition of N -qudit Ansatz $\hat{U}(\boldsymbol{\theta})$ in Eq. (2) omits some identity operators. The complete description for the l -th layer is $\mathbb{I}_{d^{N-k}} \otimes \hat{U}_l(\boldsymbol{\theta})$. The form of $\hat{U}(\boldsymbol{\theta})$ covers almost all Ansätze in VQAs and some constructions are given in Fig. 1(e).

The training of QCBMs is guided by the maximum mean discrepancy (MMD) loss [72]. Suppose \mathbb{Q} and \mathbb{P}_{θ} live on the same space. The MMD loss measures the difference between \mathbb{P}_{θ} and \mathbb{Q} with

$$\text{MMD}^2(\mathbb{P}_{\theta} || \mathbb{Q}) = \mathbb{E}(k(\mathbf{x}, \mathbf{x}')) + \mathbb{E}(k(\mathbf{y}, \mathbf{y}')) - 2 \mathbb{E}(k(\mathbf{x}, \mathbf{y})),$$

where the expectations are taken over the randomness of examples and $k(\cdot, \cdot)$ denotes a predefined kernel. QCBMs aim to find an estimator minimizing MMD loss,

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} \parallel \mathbb{Q}). \quad (3)$$

If the distribution $\mathbb{P}_{\boldsymbol{\theta}}$ and \mathbb{Q} cannot be accessed directly, the estimator is located by minimizing the empirical MMD loss with finite samples [72], i.e.,

$$\hat{\boldsymbol{\theta}}^{(n,m)} = \arg \min_{\boldsymbol{\theta} \in \Theta} \text{MMD}_U^2(\mathbb{P}_{\boldsymbol{\theta}}^n \parallel \mathbb{Q}^m), \quad (4)$$

where $\mathbb{P}_{\boldsymbol{\theta}}^n$ and \mathbb{Q}^m separately refers to the empirical distribution of $\mathbb{P}_{\boldsymbol{\theta}}$ and \mathbb{Q} , and

$$\begin{aligned} \text{MMD}_U^2(\mathbb{P}_{\boldsymbol{\theta}}^n \parallel \mathbb{Q}^m) &:= \frac{1}{n(n-1)} \sum_{i \neq i'}^n k(\mathbf{x}^{(i)}, \mathbf{x}^{(i')}) \\ &+ \frac{1}{m(m-1)} \sum_{j \neq j'}^m k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')}) - \frac{2}{nm} \sum_{i,j} k(\mathbf{x}^{(i)}, \mathbf{y}^{(j)}). \end{aligned}$$

More explanations of MMD loss are provided in SM B.

The choice of the kernel $k(\cdot, \cdot)$ in MMD loss is flexible. When it is specified to be the *quantum* kernel (i.e., the specified kernel can be efficiently computed by quantum algorithms, where representative examples are linear kernel and polynomial kernels [74]) and \mathbb{Q} can be directly accessed by QGLMs (e.g., \mathbb{Q} can be efficiently prepared by a quantum circuit), the MMD loss can be efficiently calculated.

Lemma 1. Suppose that the distribution \mathbb{Q} can be directly accessed by QCBMs. When the quantum kernel is adopted, the MMD loss in Eq. (3) can be estimated within an error ϵ in $O(1/\epsilon^2)$ runtime complexity.

Proof of Lemma 1. Here we separately elaborate on the calculation of MMD loss when the target distribution \mathbb{Q} can be efficiently prepared by a diagonalized mixed state $\sigma = \sum_{\mathbf{y}} \mathbb{Q}(\mathbf{y}) |\mathbf{y}\rangle \langle \mathbf{y}|$ or a pure quantum state $|\Psi\rangle = \sum_{\mathbf{y}} \sqrt{\mathbb{Q}(\mathbf{y})} |\mathbf{y}\rangle$.

Diagonalized mixed states. In this setting, the quantum kernel corresponds to the linear kernel, i.e.,

$$k(\mathbf{a}, \mathbf{a}') = \langle \mathbf{a}, \mathbf{a}' \rangle.$$

Recall the definition of the MMD loss is $\text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} \parallel \mathbb{Q}) = \mathbb{E}(k(\mathbf{x}, \mathbf{x}')) - 2\mathbb{E}(k(\mathbf{x}, \mathbf{y})) + \mathbb{E}(k(\mathbf{y}, \mathbf{y}'))$. The first term equals to

$$\mathbb{E}(k(\mathbf{x}, \mathbf{x}')) = \mathbb{E}_{\mathbb{P}_{\boldsymbol{\theta}}, \mathbb{P}_{\boldsymbol{\theta}}} (\delta_{\mathbf{x}, \mathbf{x}'}) = \sum_{\mathbf{x}} \mathbb{P}_{\boldsymbol{\theta}}^2(\mathbf{x}).$$

Similarly, the second term equals to

$$\mathbb{E}(k(\mathbf{x}, \mathbf{y})) = \sum_{\mathbf{x}} \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) \mathbb{Q}(\mathbf{x}).$$

And the third term equals to

$$\mathbb{E}(k(\mathbf{y}, \mathbf{y}')) = \sum_{\mathbf{y}} \mathbb{Q}^2(\mathbf{y}).$$

The above three terms can be effectively and analytically evaluated by quantum Swap test when the input state of QCBM in Eq. (1) is a full rank mixed state, e.g., $\rho_0 = \mathbb{I}_{2^n}/2^n$. Denote the output state of QCBM as $\rho = U(\boldsymbol{\theta})\rho_0 U(\boldsymbol{\theta})^\dagger$. This state is also diagonalized and its diagonalized entry records $\mathbb{P}_{\boldsymbol{\theta}}$, i.e.,

$$\rho = \sum_{\mathbf{x}} \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) |\mathbf{x}\rangle \langle \mathbf{x}|.$$

According to [75, 76], given two mixed states ϱ_1 and ϱ_2 , the output of Swap test is $1/2 + \text{Tr}(\varrho_1 \varrho_2)/2$ with an additive error ϵ in $O(1/\epsilon^2)$ runtime. As such, when $\varrho_1 = \varrho_2 = \rho$, the first term $\mathbb{E}(k(\mathbf{x}, \mathbf{x}'))$ can be calculated by Swap test, because $\text{Tr}(\rho \rho) = \sum_{\mathbf{x}} \mathbb{P}_{\boldsymbol{\theta}}^2(\mathbf{x})$. Likewise, through setting $\varrho_1 = \rho$, and $\varrho_2 = \sigma$ ($\varrho_1 = \varrho_2 = \sigma$), the second (third) term can be efficiently evaluated by Swap test with an additive error ϵ . In other words, by leveraging Swap test, we can estimate MMD loss with an additive error ϵ in $O(1/\epsilon^2)$ runtime cost.

Pure states. The quantum kernel in this scenario corresponds to a nonlinear kernel, i.e.,

$$k(\mathbf{a}, \mathbf{a}') = \langle \frac{\mathbf{a}}{\sqrt{\mathbb{P}(\mathbf{a})}}, \frac{\mathbf{a}'}{\sqrt{\mathbb{P}(\mathbf{a}')}} \rangle,$$

where $\mathbb{P}(\mathbf{a})$ stands for the probability of sampling \mathbf{a} and $\sum_{\mathbf{a}} \mathbb{P}(\mathbf{a}) = 1$. With this regard, the explicit form of MMD loss yields

$$\begin{aligned} &\text{MMD}(\mathbb{P}_{\boldsymbol{\theta}} \parallel \mathbb{Q}) \\ &= \mathbb{E}(k(\mathbf{x}, \mathbf{x}')) - 2\mathbb{E}(k(\mathbf{x}, \mathbf{y})) + \mathbb{E}(k(\mathbf{y}, \mathbf{y}')) \\ &= \sum_{\mathbf{x}} \sum_{\mathbf{x}'} \mathbb{P}_{\boldsymbol{\theta}} \mathbb{P}_{\boldsymbol{\theta}} \left\langle \frac{\mathbf{x}}{\sqrt{\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x})}}, \frac{\mathbf{x}'}{\sqrt{\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}')}} \right\rangle \\ &\quad - 2 \sum_{\mathbf{x}} \sum_{\mathbf{y}} \mathbb{P}_{\boldsymbol{\theta}} \mathbb{Q} \left\langle \frac{\mathbf{x}}{\sqrt{\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x})}}, \frac{\mathbf{y}}{\sqrt{\mathbb{Q}(\mathbf{y})}} \right\rangle \\ &\quad + \sum_{\mathbf{y}} \sum_{\mathbf{y}'} \mathbb{Q} \mathbb{Q} \left\langle \frac{\mathbf{y}}{\sqrt{\mathbb{Q}(\mathbf{y})}}, \frac{\mathbf{y}'}{\sqrt{\mathbb{Q}(\mathbf{y}')}} \right\rangle \\ &= \sum_{\mathbf{x}} \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) + \sum_{\mathbf{y}} \mathbb{Q}(\mathbf{y}) - 2 \sum_{\mathbf{x}} \sqrt{\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) \mathbb{Q}(\mathbf{x})} \\ &= 2 - 2 \sum_{\mathbf{x}} \sqrt{\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) \mathbb{Q}(\mathbf{x})}. \end{aligned}$$

The above results indicate that the evaluation of MMD loss amounts to calculating $\sum_{\mathbf{x}} \sqrt{\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) \mathbb{Q}(\mathbf{x})}$. Denote the generated state of QCBM as $|\Phi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta}) |0\rangle^{\otimes n} = e^{i\phi} \sum_{\mathbf{x}} \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) |\mathbf{x}\rangle$, where $\rho_0 = (|0\rangle \langle 0|)^{\otimes n}$. When the target distribution \mathbb{Q} refers to a pure quantum state

$|\Psi\rangle = \sum_{\mathbf{y}} \sqrt{\mathbb{Q}(\mathbf{y})} |\mathbf{y}\rangle$, the term $\sum_{\mathbf{x}} \sqrt{\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) \mathbb{Q}(\mathbf{x})}$ can be evaluated by Swap test [75], i.e.,

$$|\langle \Phi(\boldsymbol{\theta}) | \Psi \rangle|^2 = \left(\sum_{i=1}^{2^N} \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) \mathbb{Q}(\mathbf{x}) \right)^2. \quad (5)$$

According to [75], taking account of the sample error, this term can be estimated within an additive error ϵ in $O(1/\epsilon^2)$ runtime complexity. ■

It hints that when both $k(\cdot, \cdot)$ and \mathbb{Q} are quantum, $\text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q})$ can be efficiently calculated in which the runtime cost is *independent* of the dimension of data space. In contrast, for GLMs and QCBMs with classical kernels, the runtime cost in computing the MMD loss polynomially scales with n and m . Such runtime discrepancy warrants the advantage of QCBMs explained in the subsequent context.

We now analyze the generalization ability of QCBMs. To begin with, let us extend the definition of *generalization error* from the classical learning theory [77] to the regime of quantum generative learning.

Definition 1 (Generalization error of QGLMs). *When either the kernel $k(\cdot, \cdot)$ or the target \mathbb{Q} is classical, the generalization error of QGLMs is*

$$\mathfrak{R}^C = \text{MMD}^2(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q}) - \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}). \quad (6)$$

When the kernel $k(\cdot, \cdot)$ is quantum and \mathbb{Q} can be efficiently accessed by quantum machines, the generalization error of QGLMs is

$$\mathfrak{R}^Q = \text{MMD}^2(\mathbb{P}_{\hat{\boldsymbol{\theta}}} || \mathbb{Q}) - \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}). \quad (7)$$

Intuitively, both \mathfrak{R}^C and \mathfrak{R}^Q evaluate the divergence of the estimated and the optimal MMD loss, where a lower \mathfrak{R}^C or \mathfrak{R}^Q suggests a better generalization ability. The following theorem establishes the generalization theory of QCBMs.

Theorem 1. *Assume $\max_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \leq C_1$ with C_1 being a constant. Define $C_2 = \sup_{\mathbf{x}} k(\mathbf{x}, \mathbf{x})$. Following settings in Lemma 1, with probability at least $1 - \delta$, the generalization error of QCBMs yields*

$$\mathfrak{R}^Q \leq \mathfrak{R}^C \leq C_1 \sqrt{\frac{8}{n} + \frac{8}{m}} \sqrt{C_2} (2 + \sqrt{\log \frac{1}{\delta}}). \quad (8)$$

Proof of Theorem 1. To prove Theorem 1, we first derive the upper bound of the generalization error \mathfrak{R}^C under the generic setting. Then, we analyze \mathfrak{R}^Q under the specific setting where the quantum kernel is employed and the target distribution \mathbb{Q} can be directly accessed by quantum machines. The analysis of \mathfrak{R}^C adopts the following lemma.

Lemma 2 (Adapted from Theorem 1, [78]). *Suppose that the kernel $k(\cdot, \cdot)$ is bounded. Following the notations in Theorem 1, when the number of examples sampled from $\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}}$ and \mathbb{Q} is n and m , with probability $1 - \delta$, $\text{MMD}(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q}) \leq \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) + 2\sqrt{\frac{2}{n} + \frac{2}{m}} \sqrt{\sup_{\mathbf{x} \in \mathcal{X}} k(\mathbf{x}, \mathbf{x})} (2 + \sqrt{\frac{2}{\delta}})$.*

The calculation of \mathfrak{R}^C . Recall the definition of $\hat{\boldsymbol{\theta}}^{(n,m)}$ in Eq. (4). Let us first rewrite the generalization error as

$$\begin{aligned} \mathfrak{R}^C &= \text{MMD}^2(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q}) - \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \\ &= \left(\text{MMD}(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q}) - \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \right) \\ &\quad \times \left(\text{MMD}(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q}) + \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \right) \\ &\leq 2C_1 \left| \text{MMD}(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q}) - \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \right|, \end{aligned}$$

where the second equality uses $\inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) = (\inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}))^2$ and the inequality employs the $\text{MMD}(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q}) + \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \leq 2 \text{MMD}(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q}) \leq 2C_1$.

In conjunction with the above equation with the results of Lemma 2, we obtain that with probability at least $1 - \delta$, the upper bound of the generalization error of QGCM yields

$$\mathfrak{R}^C \leq 4C_1 \left(\frac{2}{n} + \sqrt{\frac{2}{m}} \right) \sqrt{\sup_{\mathbf{x} \in \mathcal{X}} k(\mathbf{x}, \mathbf{x})} \left(2 + \sqrt{\log \frac{2}{\delta}} \right). \quad (9)$$

The calculation of \mathfrak{R}^Q . Recall the definition of $\hat{\boldsymbol{\theta}}$ in Eq. (3). When QCBM adopts the quantum kernel and the target distribution \mathbb{Q} can be directly accessed by quantum machines, the minimum argument of the loss function yields $\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q})$. Following the definition of the generalization error, we obtain

$$\begin{aligned} \mathfrak{R}^Q &= \text{MMD}^2(\mathbb{P}_{\hat{\boldsymbol{\theta}}} || \mathbb{Q}) - \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \\ &\leq \text{MMD}^2(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q}) - \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \\ &= \mathfrak{R}^C, \end{aligned} \quad (10)$$

where the inequality is supported by the definition of $\hat{\boldsymbol{\theta}}$, i.e., $\text{MMD}^2(\mathbb{P}_{\hat{\boldsymbol{\theta}}} || \mathbb{Q}) = \min_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \leq \text{MMD}^2(\mathbb{P}_{\hat{\boldsymbol{\theta}}^{(n,m)}} || \mathbb{Q})$.

Combining the results of \mathfrak{R}^C and \mathfrak{R}^Q in Eqs. (9) and (10), we obtain that with probability at least $1 - \delta$,

$$\mathfrak{R}^Q \leq \mathfrak{R}^C \leq 4C_1 \left(\frac{2}{n} + \sqrt{\frac{2}{m}} \right) \sqrt{\sup_{\mathbf{x} \in \mathcal{X}} k(\mathbf{x}, \mathbf{x})} \left(2 + \sqrt{\log \frac{2}{\delta}} \right). \quad \blacksquare$$

It indicates that when \mathbb{Q} is quantum, quantum kernels promise a *strictly lower* generalization error than classical

kernels. Moreover, the decisive factor to improve generalization of QCBMs is to simultaneously increase n and m . Note that this phenomenon results into a tradeoff between the generalization and the runtime efficiency for QCBM with classical kernels. That is, increasing n and m enables the reduced generalization error but increases the runtime cost to compute $\text{MMD}_U^2(\mathbb{P}_{\theta}^n \parallel \mathbb{Q}^m)$, which echoes with the claim in [71]. Conversely, QCBM with quantum kernels can simultaneously obtain both the good generalization and runtime efficacy, supported by Lemma 1. We remark that this statement does not contradict with the result in [71], since our setting does not require the statistical query access. Instead, our results show that a careful design of learning strategies and model constructions can enhance the power of QCBMs to advance GLMs, especially when \mathbb{Q} refers to the tasks in quantum many body physics and quantum information processing [34].

Remark. In SM C, we partially address another long standing problem of quantum probabilistic models, i.e., whether QCBMs are superior to classical GLMs. Briefly, for certain \mathbb{Q} , QCBMs can attain a lower $\inf_{\theta \in \Theta} \text{MMD}^2(\mathbb{P}_{\theta} \parallel \mathbb{Q})$ over restricted Boltzmann machine [79], which may lead to a better learnability.

III. GENERALIZATION OF QGAN

We next analyze the generalization of QGANs for continuous \mathbb{Q} . As shown in Fig. 1(b), QGAN adopts an N -qudit Ansatz $\hat{U}(\theta)$ in Eq. (2) to build a generator $G_{\theta}(\cdot)$, which maps \mathbf{z} sampled from a prior distribution $\mathbb{P}_{\mathbf{z}}$ to a generated example $\mathbf{x} \sim \mathbb{P}_{\theta}$, i.e., $\mathbf{x} := G_{\theta}(\mathbf{z}) \in \mathbb{R}^{d^N}$. The j -th entry of \mathbf{x} is

$$\mathbf{x}_j = \text{Tr}(\Pi_j \hat{U}(\theta) \rho_{\mathbf{z}} \hat{U}(\theta)^{\dagger}), \quad \forall j \in [d^N], \quad (11)$$

where $\rho_{\mathbf{z}}$ refers to the encoded quantum state of \mathbf{z} . Given n reference examples $\{\mathbf{x}^{(i)}\}_{i=1}^n$ produced by $G_{\theta}(\cdot)$, its empirical distribution is $\mathbb{P}_{\theta}^n(d\mathbf{x}) = \sum_{i=1}^n \delta_{\mathbf{x}^{(i)}}(d\mathbf{x})/n$. Similarly, we define \mathbb{Q}^m as the empirical distribution of \mathbb{Q} with m true examples. In the training process, QGAN updates θ to minimize the empirical MMD loss in Eq. (4), i.e., the optimal solution satisfies $\theta^* = \arg \min \text{MMD}_U^2(\mathbb{P}_{\theta}^n \parallel \mathbb{Q}^m)$. Note that QGANs can be applied to estimate both discrete and continuous distributions. When \mathbb{Q} is discrete, the mechanism of QGANs is *equivalent to* QCBMs (refer to SM A & B for more elaborations about QGANs). Due to this reason, here we only focus on applying QGANs to estimate the continuous distribution \mathbb{Q} .

When QGANs in Eq. (11) are designed to estimate the continuous distribution \mathbb{Q} , their generalization can only be measured by \mathfrak{R}^C in Definition 1. In this scenario, it is of paramount importance of unveiling how the generalization of QGANs depends on uploading methods and the structure information Ansatz. The following theorem makes a concrete step toward this goal.

Theorem 2. Assume the kernel $k(\cdot, \cdot)$ is C_3 -Lipschitz. Suppose that the employed quantum circuit $\hat{U}(\mathbf{z})$ to pre-

pare $\rho_{\mathbf{z}}$ containing in total N_{ge} parameterized gates and each gate acting on at most k qudits. Following notations in Eqs. (4) and (6), with probability at least $1 - \delta$, the generalization error of QGANs, \mathfrak{R}^C in Eq. (11) is upper bounded by

$$8\sqrt{\frac{8C_2^2(n+m)}{nm} \ln \frac{1}{\delta}} + \frac{48}{n-1} + \frac{144d^k \sqrt{N_{gt} + N_{ge}}}{n-1} C_4, \quad (12)$$

where $C_2 = \sup_{\mathbf{x}} k(\mathbf{x}, \mathbf{x})$, $C_4 = N \ln(441dC_3^2 n N_{ge} N_{gt}) + 1$.

Proof sketch. Here we only deliver the central idea and postpone the whole proof to SM D.

For clearness, we denote $\hat{\theta}^{(n,m)}$ as $\hat{\theta}$. Define $\mathcal{E}(\theta) = \text{MMD}_U^2(\mathbb{P}_{\theta}^n \parallel \mathbb{Q}^m)$ and $\mathcal{T}(\theta) = \text{MMD}^2(\mathbb{P}_{\theta} \parallel \mathbb{Q})$. Then the generalization error is upper bounded by

$$\begin{aligned} \mathfrak{R}^C &= \mathcal{E}(\hat{\theta}) - \mathcal{E}(\hat{\theta}) + \mathcal{T}(\hat{\theta}) - \mathcal{T}(\theta^*) \\ &\leq |\mathcal{T}(\hat{\theta}) - \mathcal{E}(\hat{\theta})| + |\mathcal{T}(\theta^*) - \mathcal{E}(\theta^*)|, \end{aligned} \quad (13)$$

where the first inequality employs the definition of $\hat{\theta}$ with $\mathcal{E}(\theta^*) \geq \mathcal{E}(\hat{\theta})$ and the second inequality uses the property of absolute value function. In this regard, we derive the probability for $\sup_{\theta \in \Theta} |\mathcal{T}(\theta) - \mathcal{E}(\theta)| < \epsilon$, which in turn can achieve the upper bound of \mathfrak{R}^C .

According to the explicit form of the MMD loss and Jensen inequality, $\sup_{\theta \in \Theta} |\mathcal{E}(\theta) - \mathcal{T}(\theta)|$ satisfies

$$\begin{aligned} &\sup_{\theta \in \Theta} |\text{MMD}_U^2(\mathbb{P}_{\theta}^n \parallel \mathbb{Q}^m) - \text{MMD}^2(\mathbb{P}_{\theta} \parallel \mathbb{Q})| \\ &\leq \sup_{\theta \in \Theta} \underbrace{\left| \mathbb{E}_{\mathbf{y}, \mathbf{y}'}(k(\mathbf{y}, \mathbf{y}')) - \frac{\sum_{j \neq j'} k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')})}{m(m-1)} \right|}_{T1} \\ &\quad + \sup_{\theta \in \Theta} \underbrace{\left| \mathbb{E}_{\mathbf{z}, \mathbf{z}'}(k(G_{\theta}(\mathbf{z}), G_{\theta}(\mathbf{z}'))) - \frac{\sum_{i \neq i'} k(G_{\theta}(\mathbf{z}^{(i)}), G_{\theta}(\mathbf{z}^{(i')}))}{n(n-1)} \right|}_{T2} \\ &\quad + 2 \sup_{\theta \in \Theta} \underbrace{\left| \mathbb{E}_{\mathbf{z}, \mathbf{y}}(k(G_{\theta}(\mathbf{z}), \mathbf{y})) - \frac{\sum_{i \in [n], j \in [m]} k(G_{\theta}(\mathbf{z}^{(i)}), \mathbf{y}^{(j)})}{mn} \right|}_{T3}. \end{aligned}$$

In other words, to upper bound \mathfrak{R}^C , we should separately derive the upper bounds of the terms $T1$, $T2$, and $T3$. Specifically, supported by concentration inequality [80], we obtain with probability at least $1 - 3\delta_{T3}$,

$$\mathfrak{R}^C \leq 2(\epsilon_1 + 2\epsilon_2 + 4\epsilon), \quad (14)$$

where $\epsilon = \sqrt{\frac{8C^2(n+m)}{nm \log 1/\delta_{T3}}}$, $\epsilon_1 = \mathbb{E}(T2)$, and $\epsilon_2 = \mathbb{E}(T3)$.

The above inequality indicates that the generalization error of QGAN depends on ϵ_1 and ϵ_2 . Namely, by leveraging the covering number—a tool developed in statistical learning theory, we prove $\epsilon_1 \leq \frac{8}{n-1} + \frac{24\sqrt{d^{2k}(N_{ge} + N_{gt})}}{n-1} (1 + N \ln(441dC_3^2(n-1)N_{ge}N_{gt}))$ and $\epsilon_2 \leq \frac{8}{n} + \frac{24\sqrt{d^{2k}(N_{ge} + N_{gt})}}{n} (1 + N \ln(441dC_3^2 n N_{ge} N_{gt}))$.

Taken together, with probability $1 - 3\delta_{T3}$, the generalization error of QGANs is upper bounded by $8\sqrt{\frac{8C_2^2(n+m)}{nm}} \ln \frac{1}{\delta} + \frac{48}{n-1} + \frac{144d^k\sqrt{N_{gt}+N_{ge}}}{n-1} C_4$. \blacksquare

The results of Theorem 2 convey four-fold implications. First, according to the first term in the right hand-side of Eq. (12), to achieve a tight upper bound of \mathfrak{R}^C , the ratio between the number of examples sampled from \mathbb{Q} and \mathbb{P} should satisfy $m/n = 1$. In this case, \mathfrak{R}^C linearly decreases and finally converges to zero with the increased n or m . Second, \mathfrak{R}^C linearly depends on the kernel term C_2 , exponentially depends on k in Eq. (2), and sublinearly depends on the number of trainable quantum gates N_{gt} . These observations underpin the importance of controlling the expressivity of the adopted Ansatz and selecting proper kernels to ensure both the good learning performance and generalization of QGANs. Third, the way of preparing ρ_z is a determinant factor in generalization of QGANs, which implies the prior distribution \mathbb{P}_z and the number of encoding gates N_{ge} should be carefully designed. Last, the explicit dependence on the architecture of Ansatz connects the generalization of QGANs with their trainability, i.e., a large N_{gt} or k may induce barren plateaus in training QGLMs [81, 82] and results in an inferior learning performance. Meanwhile, it also leads to a degraded generalization error bound.

Remark. Most kernels such as RBF, linear, and Matérn kernels satisfy the Lipschitz condition [83]. Meanwhile, Theorem 2 can be efficiently extended to noisy settings by using the contraction properties of noisy channels explained in Ref. [66].

The derived bound in Theorem 2 is succinct and can be directly employed to quantify the generalization of QGANs with the specified Ansatz. The following corollary quantifies \mathfrak{R}^C of QGANs with two typical Ansätze in VQAs, i.e., hardware-efficient Ansatz and quantum approximate optimization Ansatz.

Corollay 1. *Following notations in Theorem 2, when QGAN is realized by the hardware-efficient Ansatz and quantum approximate optimization Ansatz with L layers, \mathfrak{R}^C is upper bounded by*

$$\tilde{\mathcal{O}}(C_2\sqrt{\frac{n+m}{(nm)}} + \frac{1}{n} + \frac{\sqrt{N(L_E+3L)}(N+1)}{n-1}),$$

and

$$\tilde{\mathcal{O}}(C_2\sqrt{\frac{(n+m)}{(nm)}} + \frac{1}{n-1} + \frac{2^N\sqrt{(N+1)(L+L_E)}(N+1)}{n-1}).$$

Proof of Corollary 1. Here we separately analyze the generalization ability of QGANs with two typical classes of Ansätze, i.e., the hardware-efficient Ansätze and the quantum approximation optimization Ansätze.

Hardware-efficient Ansätze. An N -qubits hardware-efficient Ansatz is composed of L layers, i.e., $U(\boldsymbol{\theta}) = \prod_{l=1}^L U(\boldsymbol{\theta}^l)$ with $L \sim \text{poly}(N)$, where $U(\boldsymbol{\theta}^l)$ is composed

of parameterized single-qubit gates and fixed two-qubit gates. In general, the topology of $U(\boldsymbol{\theta}^l)$ for any $l \in [L]$ is the same and each qubit interacts with at least one parameterized single-qubit gate and two qubits gates. Mathematically, we have $U(\boldsymbol{\theta}^l) = (\otimes_{i=1}^N U_s) U_{eng}$ with $U_s = R_Z(\beta)R_Y(\gamma)R_Z(\nu)$ being realized by three rotational qubit gates and $\gamma, \beta, \nu \in [0, 2\pi]$. The number of two-qubit gates in each layer is set as N and the connectivity of two-qubit gates aims to adapt to the topology restriction of quantum hardware. The entangled layer U_{eng} contains two-qubit gates, i.e., CNOT gates, whose connectivity adapts the topology of the quantum hardware.

An example of 4-qubit QGAN with hardware-efficient Ansatz is illustrated in the upper panel of Fig. 2. Under this setting, when both the encoding unitary and the trainable unitary adopt the hardware-efficient Ansatz, we have $k = 2$, $d = 2$, $N_{ge} = L_E N$, $N_{gt} = L(3N)$, and $N_g = L(3N + N) = 4L$. Based on the above settings, we achieve the generalization error of ab N -qubit QGAN with the hardware-efficient Ansätze, supported by Theorem 2, i.e., with probability at least $1 - \delta$,

$$\begin{aligned} \mathfrak{R}^C \leq & 8\sqrt{\frac{8C_2^2(n+m)}{nm}} \ln \frac{1}{\delta} + \frac{48}{n-1} \\ & + \frac{576\sqrt{N(L_E+3L)}}{n-1} (N \ln(1323dC_3^2nN^2L_E L) + 1), \end{aligned}$$

where $C_2 = \sup_{\mathbf{x}} k(\mathbf{x}, \mathbf{x})$.

Quantum approximate optimization Ansätze. The lower panel of Fig. 2 plots the quantum approximate optimization Ansätze. The mathematical expression of this Ansatz takes the form $U(\boldsymbol{\theta}) = \prod_{l=1}^L U(\boldsymbol{\theta}^l)$, where the l -th layer $U(\boldsymbol{\theta}^l) = U_B(\boldsymbol{\theta}^l)U_C(\boldsymbol{\theta}^l)$ is implemented by the driven Hamiltonian $U_B(\boldsymbol{\theta}^l) = \otimes_{i=1}^N R_X(\theta_i^l)$ and the target Hamiltonian $U_C(\boldsymbol{\theta}^l) = \exp(-i\theta_{l+1}^l H_C)$ with H_C being a specified Hamiltonian. Under this setting, when the encoding unitary is constructed by the hardware-efficient Ansatz and the trainable unitary is realized by the quantum approximate optimization Ansätze, we have $k = N$, $d = 2$, $N_{ge} = L_E N$, and $N_{gt} = L(N+1)$. Based on the above settings, we achieve the generalization error of an N -qubit QGAN with the quantum approximate optimization Ansätze, supported by Theorem 2, i.e., with probability at least $1 - \delta$,

$$\begin{aligned} \mathfrak{R}^C \leq & 8\sqrt{\frac{8C_2^2(n+m)}{nm}} \ln \frac{1}{\delta} + \frac{48}{n-1} \\ & + \frac{144 \times 2^N \sqrt{(N+1)(L+L_E)}}{n-1} \\ & \times (N \ln(441dC_3^2nLL_E N(N+1)) + 1), \end{aligned}$$

where $C_2 = \sup_{\mathbf{x}} k(\mathbf{x}, \mathbf{x})$. \blacksquare

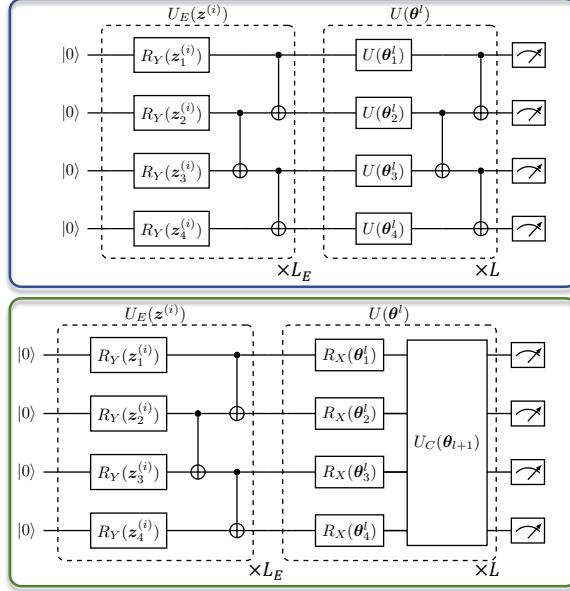


FIG. 2: Illustration of QGANs with different Ansätze. The upper panel presents that both the encoding method and the trainable unitary of QGANs employ the hardware-efficient Ansätze. The lower panel presents a class of QGANs such that the encoding method uses the hardware-efficient Ansätze and the trainable unitary is implemented by the quantum approximate optimization Ansätze.

IV. APPLICATIONS OF QGLMS WITH MERITS

A crucial message conveyed by Theorems 1 and 2 is that when $\mathbb{Q} \in \mathbb{P}_\Theta$, the distance between the learned \mathbb{P}_θ and \mathbb{Q} is continuously decreased by increasing n and m . This evidence contributes to theoretically study the merits of QGLMs in many practical problems such as quantum state preparation and parameterized Hamiltonian learning.

A. Quantum state preparation

Quantum state preparation is a crucial subroutine in quantum computing and quantum sensing, since its efficiency determines the achievable runtime speedups of quantum machine learning and Hamiltonian simulation algorithms, and improves the precision of the device. Nevertheless, theoretical results indicate that exactly loading the generic discrete distribution with d^N dimensions to an N -qudit state requests $O(d^N)$ gates [84, 85], which prevents any merits. Moreover, there is no deterministic method of encoding continuous distribution into the multi-qudit system. QGLMs open the door to efficiently encode both discrete and continuous distributions into quantum circuits with certain estimation error [45]. Theorems 1 and 2 establish the theoretical foundation about such an error. Specifically, for $\mathbb{Q} \in \mathbb{P}_\Theta$, large examples promise

a low estimation error and the learned QGLM can be easily realized and reused. For $\mathbb{Q} \notin \mathbb{P}_\Theta$, the estimation error could be considerable, even though n and m are infinite. In this view, the power of QGLMs in quantum state preparation highly depends on their expressivity.

B. Parameterized Hamiltonian Learning

QGLMs may advance GLMs in parameterized Hamiltonian learning (PHL). Define an N -qubit parameterized Hamiltonian as $H(\mathbf{a})$ and the corresponding ground state as $|\phi(\mathbf{a})\rangle$, where \mathbf{a} is the interaction parameter sampled from a prior distribution \mathbb{D} . PHL aims to use m training samples $\{\mathbf{a}^{(i)}, |\phi(\mathbf{a}^{(i)})\rangle\}_{i=1}^m$ to approximate the distribution of the ground states for $H(\mathbf{a})$ with $\mathbf{a} \sim \mathbb{D}$, i.e., $|\phi(\mathbf{a})\rangle \sim \mathbb{Q}$. If the estimation error is low, then the trained model can prepare the ground state of $H(\mathbf{a}')$ for an unseen parameter $\mathbf{a}' \sim \mathbb{D}$. This property can be used to explore many crucial behaviors in condensed-matter systems. Concisely, there exists an N -qubit Hamiltonian $H(\mathbf{a})$ and a distribution \mathbb{D} such that the formed ground state distribution \mathbb{Q} can be efficiently estimated by QGLMs with $O(\text{poly}(N))$ trainable parameters but is computational hardness for GLMs with $O(\text{poly}(N))$ trainable parameters.

The separated power between QGLMs and GLMs relies on the following lemma.

Lemma 3. Suppose that \mathbb{Q} refers to the distribution of the ground states for parameterized Hamiltonians $H(\mathbf{a})$ with $\mathbf{a} \sim \mathbb{D}$. Under the quantum threshold assumption, there exists a distribution \mathbb{Q} that can be efficiently represented by QGLMs but is computationally hard for GLMs.

Proof sketch. The proof is provided in SM E. Conceptually, the proof is established on the results of quantum random circuits, which is widely believed to be classically computationally hard and in turn can be used to demonstrate quantum advantages on NISQ devices [86, 87]. Namely, Ref. [88] proposed a Heavy Output Generation (HOG) problem to separate the power between classical and quantum computers. That is, under the quantum threshold assumption, there do not exist classical samples that can spoof the k samples output by a random quantum circuit with success probability at least 0.99 [88], while quantum computers can solve the HOG problem with high success probability.

Based on this observation, we connect $|\phi(\mathbf{a})\rangle$ with the output state of random quantum circuits. To this end, we prove that there exists a ground state $|\phi(\mathbf{a})\rangle$ of a Hamiltonian $H(\mathbf{a})$, which can be efficiently prepared by quantum computers but is computationally hard for classical algorithms. ■

It indicates that in PHL, the expressivity of QGLMs outperforms GLMs with $\mathbb{Q} \in \mathbb{P}_\Theta^Q$ and $\mathbb{Q} \notin \mathbb{P}_\Theta^C$. When the number of trainable parameters of QGLMs scales with $O(\text{poly}(N))$, there may exist a kernel leading to

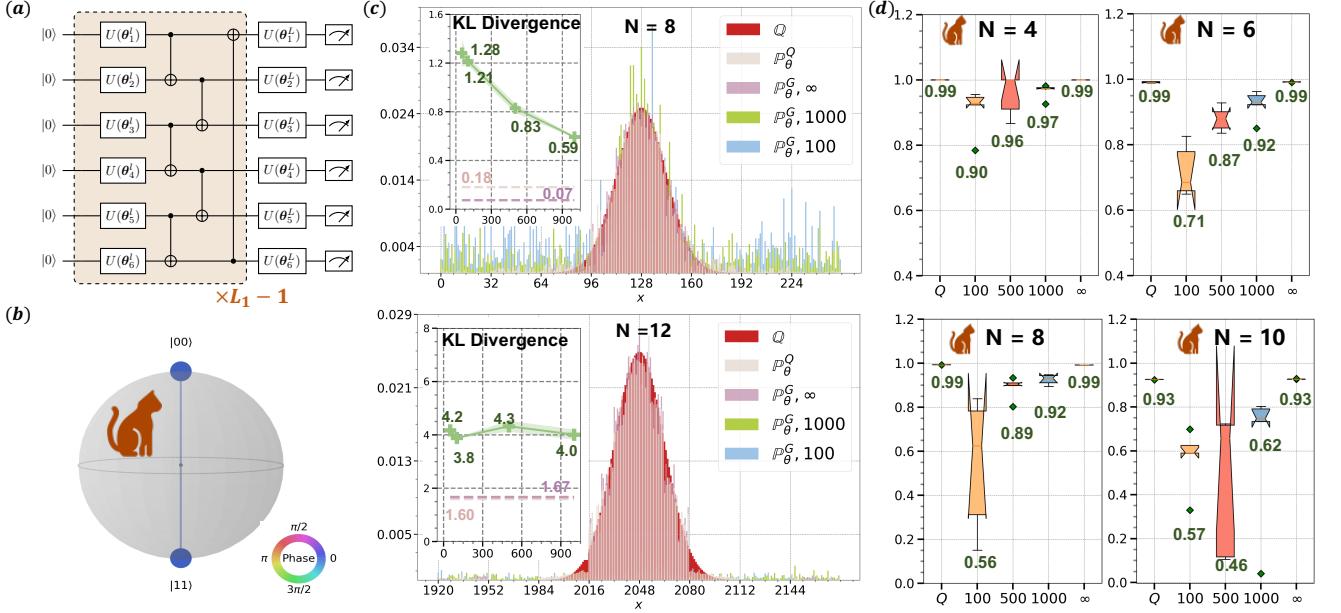


FIG. 3: Simulation results of QCBM. (a) The implementation of QCBMs when $N = 6$. The label ‘ $L_1 - 1$ ’ refers to repeating the architecture highlighted by the brown color $L_1 - 1$ times. The gate $U(\theta_i^l)$ refers to the R_Y gate applied on the i -th qubit in the l -th layer of Ansatz $\hat{U}(\theta)$. (b) The visualization of a two-qubit GHZ state. (c) The upper and lower panels separately show the simulation results of QCBMs in the task of estimating the discrete Gaussian distribution when $N = 8$ and $N = 12$. The labels ‘ Q ’, P_θ^Q , ‘ $P_\theta^{G,n}$ ’, stand for the target distribution, the output of QCBM with the quantum kernel, and the output of QCBM with the RBF kernel and n samples, respectively. The inner plots evaluate the statistical performance of QCBM through KL divergence, where the x-axis labels the number of examples n . (d) The four box-plots separately show the simulation results of QCBM in the task of approximating N -qubit GHZ state with $N = 4, 6, 8, 10$. The y-axis refers to the fidelity. The x-axis refers to the applied kernels in QCBM, where the label ‘ Q ’ represents the quantum kernel and the rest four labels refers to the RBF kernel with n samples.

$\inf_{\theta \in \Theta} \text{MMD}^2(\mathbb{P}_\theta^Q \parallel Q) \rightarrow 0$. Considering that \mathbb{P}_θ^Q and Q are intractable, the training of QGANs amounts to minimizing $\text{MMD}_U^2(\mathbb{P}_\theta^Q \parallel Q)$. Theorems 1 and 2 imply that when the empirical loss tends to zero and n and m are sufficient large, the generalization error vanishes and thus the estimated distribution recovers the target distribution with $\mathbb{P}_{\theta^{(n,m)}}^Q \approx \mathbb{P}_\theta^Q \approx Q$. Conversely, the result $Q \notin \mathbb{P}_\theta^C$ for GLMs means that even though the empirical loss and the generalization error are zero, the estimated distribution $\mathbb{P}_{\theta^{(n,m)}}^C$ fails to recover Q .

V. NUMERICAL SIMULATIONS

In this section, we apply QCBMs and QGANs to accomplish the distribution preparation and quantum state approximation tasks. More implementation details and simulation results are deferred to SM F.

A. Gaussian distribution preparation by QCBM

The first task is applying QCBMs to prepare the discrete Gaussian distribution $N(N, \mu, \sigma)$, where N specifies the range of events with $x \in [2^N]$, and μ and σ refer

to the mean and variance, respectively. An intuition of $N(N, \mu, \sigma)$ is shown in Fig. 3(c), labeled by Q . The hyper-parameter settings are as follows. For all simulations, we fix $\mu = 1$ and $\sigma = 8$. The qubit count is set as $N = 12$. The hardware-efficient Ansätze is employed to construct QCBM with $L_1 = 8$ for $N = 8$ ($L_1 = 12$ for $N = 12$). An intuition is in illustrated in Fig. 3(a). The quantum kernel and RBF kernel are adopted to compute the MMD loss. For RBF kernel, the number of samples is set as $n = 100, 1000$, and ∞ . The maximum number of iterations is $T = 50$. For each setting, we repeat the training 5 times to collect the statistical results.

The simulation results of QCBMs are illustrated in Fig. 3(c). The two outer plots exhibit the approximated distributions under different settings. In particular, for both $N = 8, 12$, the approximated distribution generated by QCBM with the quantum kernel well approximates Q . In the measure of KL divergence, the similarity of these two distributions is 0.18 and 1.6 for $N = 8, 12$, respectively. In contrast, when the adopted kernels are classical and the number of measurements is finite, QCBMs encounter the inferior performance. Namely, by increasing n and m from 50 to 1000, the KL divergence between the approximated distribution and the target distribution only decreases from 1.28 to 0.59 in the case of $N = 8$. Moreover, under the same setting, the KL divergence does not manifestly

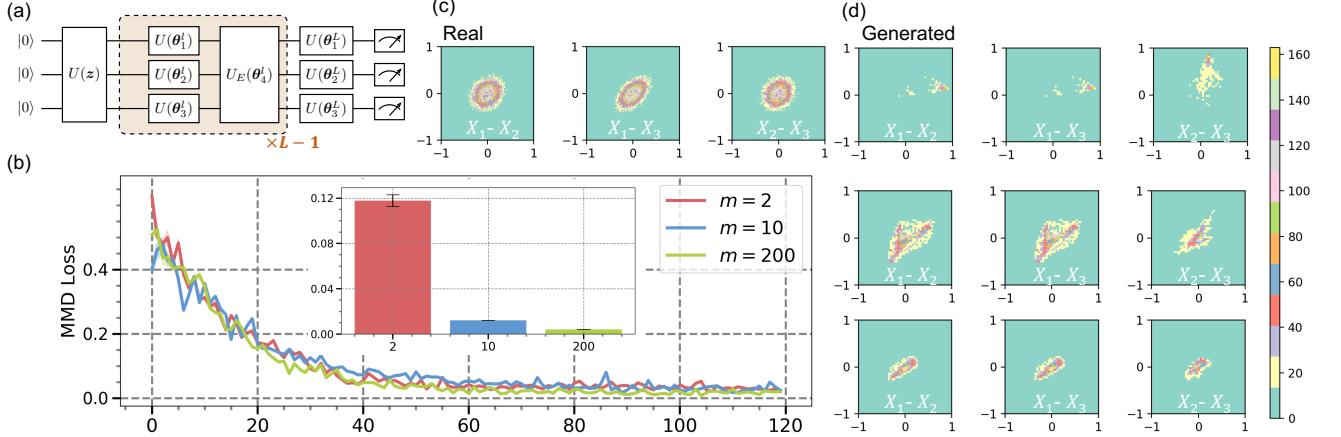


FIG. 4: Simulation results of QGANs. (a) The implementation of QGANs when the number of qubits is $N = 3$. $U(z)$ refers to the encoding circuit to load the example \mathbf{z} . The meaning of ‘ $L_1 - 1$ ’ is identical to the one explained in Fig. 1. The gate $U(\theta_i^l)$ refers to the $R_Y R_Z$ gates applied on the i -th qubit in the l -th layer of $\hat{U}(\boldsymbol{\theta})$. (b) The outer plot shows the training loss of QGANs with varied settings of m . The x-axis refers to the number of iterations. The inner plot shows the generalization property of trained QGANs by evaluating MMD loss. (c) The visualization of the exploited 3D Gaussian distribution. The label ‘ X_a-X_b ’ means projects the 3D Gaussian into the X_a-X_b plane with a, b belonging to x, y, z-axis. (d) The generated data sampled from the trained QGAN with varied settings of m . From upper to lower panel, m equals to 2, 10, and 200, respectively.

decrease when $N = 12$, which requires a larger n and m to attain a good approximation as suggested by Theorem 1. This argument is warranted by the numerical results with the setting $n = m \rightarrow \infty$, where the achieved KL divergence is comparable with QCBM with the quantum kernel. Nevertheless, the runtime complexity of QCBMs with the classical kernel polynomially scales with n and m . According to Lemma 1, under this scenario, QCBMs with the quantum kernel embraces the runtime advantages.

B. GHZ state approximation by QCBM

We apply QCBMs to accomplish the task of preparing GHZ states, a.k.a., “cat states” [89]. An intuition is depicted in Fig. 3(b). The choice of GHZ states is motivated by their importance in quantum information. The formal expression of an N -qubit GHZ state is $|\text{GHZ}\rangle = (|0\rangle^{\otimes N} + |1\rangle^{\otimes N})/\sqrt{2}$. The hyper-parameter settings are as follows. The number of qubits is set as $N = 6, 8, 10$ and the corresponding depth is $L_1 = 4, 6, 8, 10$. The quantum kernel and RBF kernel are adopted to compute the MMD loss. For RBF kernel, the number of samples is set as $n = 100, 1000$, and ∞ . The maximum number of iterations is $T = 50$. For each setting, we repeat the training 5 times to get a better understanding of the robustness of the results. The other settings are identical to those used in the task of preparing Gaussian distributions.

The simulation results, as illustrated in Fig. 3(d), indicate that QCBMs with quantum kernels outperforms RBF kernels when n and m are finite. This observation becomes apparent with an increased N . For all settings of N , the averaged fidelity between the generated states of QCBMs with the quantum kernel and the target $|\text{GHZ}\rangle$

is above 0.99, whereas the obtained averaged fidelity for QCBMs with the RBF kernel is 0.46 for $N = 10$ and $n = m = 100$. Meanwhile, as with the prior task, RBF kernel attain a competitive performance with the quantum kernel unless $n = m \rightarrow \infty$, while the price to pay is an unaffordable computational overhead.

C. 3D-correlated Gaussian preparation by QGAN

The last tasks is using QGANs to prepare 3D correlated Gaussian distributions with varied settings. The target distribution \mathbb{Q} is a 3D correlated Gaussian distribution centered at $\mu = (0, 0, 0)$ with covariance matrix $\sigma = \begin{pmatrix} 0.5 & 1 & 0.25 \\ 0.1 & 0.5 & 0.25 \\ 0.25 & 0.1 & 0.5 \end{pmatrix}$. The sampled examples from \mathbb{Q} are visualized in Fig. 4(a). Here we use a variant of the style-QGAN proposed by [90] to accomplish the task. Two key modifications in our protocol are constructing the quantum generator and replacing the trainable discriminator by MMD loss. Different from the original proposal applying the re-uploading method, the modified quantum generator first uploads the prior example \mathbf{z} using $U(z)$ followed by the Ansatz $U(\boldsymbol{\theta})$. Such a modification facilitates the analysis of the generalization behavior of QGANs as claimed in Theorem 2. The hyper-parameter settings are as follows. The number of reference samples n ranges from 2 to 200 and we keep $n = m$. The layer depth of $G_{\boldsymbol{\theta}}(\cdot)$ is set as $L \in \{2, 4, 6, 8\}$. Each setting is repeated with 5 times to collect the statistical results.

The simulation results are exhibited in Figs. 4(b)-(c). For illustration, we depicts the generated distribution of the trained QGANs in Fig. 4(b). With increasing m , the learned distribution is close to the real distribution.

The outer plot in Fig. 4(c) shows that for all settings of m , the empirical MMD loss, i.e., $\text{MMD}_U(\mathbb{P}_{\theta}^{n,m} \parallel \mathbb{Q}^m)$, converges after 60 iterations, where the averaged loss is 0.0114, 0.0077, and 0.0054 for $m = 2, 10, 200$, respectively. The inner plot measures the expected MMD loss, i.e., the trained QGANs are employed to generate new 10000 examples and then evaluate MMD_U to estimate MMD. The averaged expected MMD loss for $m = 2, 10, 200$ is 0.1178, 0.0122, and 0.0041 respectively. These observations echo with Theorem 2, where a large m allows a better generalization ability.

VI. DISCUSSIONS

We provide a succinct and direct way to compare generalization of QGLMs with different Ansätze, which deepens our understanding about the capabilities of QGLMs and benefit the design of advanced QGLMs. For QCBMs, Theorem 1 unveils that quantum kernels can greatly benefit their generalization and reduce computational overhead over classical kernels when the target distribution is quantum. Theorem 2 hints that the generalization error of QGANs has the explicit dependence on the qudits count,

the structure information of the employed Ansätze, the adopted encoding method, and the choice of prior distribution. These results suggest that a possible way to enhance the learning performance of QGLMs is devising novel Ansätze via quantum circuit architecture design techniques [91–96]. Although the attained theoretical results do not exhibit the generic exponential advantages of QGLMs, we clearly show that their potentials in quantum state preparation and parameterized Hamiltonian learning.

The developed techniques in this study are general and provide a novel approach to theoretically investigate the power of QGLMs. For instance, a promising direction is uncovering the generalization of other QGLMs. Furthermore, it is intriguing to explore the generalization of QGLMs with other loss functions [78, 97]. Another important future direction will be identifying how to use QGLMs to gain substantial quantum advantages for practical applications, e.g., quantum many body physics, quantum sensing, and quantum information processing. Last, the entangled relation between expressivity and generalization in QGLMs queries a deeper understanding from each side.

-
- [1] Merlin C Wittrock. Generative processes of comprehension. *Educational psychologist*, 24(4):345–376, 1989.
 - [2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
 - [3] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.
 - [4] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.
 - [5] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
 - [6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
 - [7] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017*, 2017.
 - [8] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. APDrawingGAN: Generating artistic portrait drawings from face photos with hierarchical gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’19)*, pages 10743–10752, 2019.
 - [9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.
 - [10] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.
 - [11] Karim Armanious, Chenming Jiang, Marc Fischer, Thomas Küstner, Tobias Hepp, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. Medgan: Medical image translation using gans. *Computerized Medical Imaging and Graphics*, 79:101684, 2020.
 - [12] Łukasz Maziarka, Agnieszka Pocha, Jan Kaczmarczyk, Krzysztof Rataj, Tomasz Danel, and Michał Warchał. Mol-cyclegan: a generative model for molecular optimization. *Journal of Cheminformatics*, 12(1):1–18, 2020.
 - [13] Oscar Méndez-Lucio, Benoit Baillif, Djork-Arné Clevert, David Rouquier, and Joerg Wichard. De novo generation of hit-like molecules from gene expression signatures using artificial intelligence. *Nature communications*, 11(1):1–10, 2020.
 - [14] Shahnawaz Ahmed, Carlos Sánchez Muñoz, Franco Nori, and Anton Frisk Kockum. Quantum state tomography with conditional generative adversarial networks. *Physical Review Letters*, 127(14):140502, 2021.
 - [15] Juan Carrasquilla, Giacomo Torlai, Roger G Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, 2019.
 - [16] Alistair WR Smith, Johnnie Gray, and MS Kim. Efficient quantum state sample tomography with basis-dependent neural networks. *PRX Quantum*, 2(2):020348, 2021.
 - [17] Andrea Rocchetto, Edward Grant, Sergii Strelchuk, Giuseppe Carleo, and Simone Severini. Learning hard quantum distributions with variational autoencoders.

- npj Quantum Information*, 4(1):1–7, 2018.
- [18] Roger G Melko, Giuseppe Carleo, Juan Carrasquilla, and J Ignacio Cirac. Restricted boltzmann machines in quantum physics. *Nature Physics*, 15(9):887–892, 2019.
 - [19] Giuseppe Carleo, Yusuke Nomura, and Masatoshi Imada. Constructing exact representations of quantum many-body systems with deep neural networks. *Nature communications*, 9(1):1–11, 2018.
 - [20] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
 - [21] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *Advances in neural information processing systems*, 29:658–666, 2016.
 - [22] Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? some theory and empirics. In *International Conference on Learning Representations*, 2018.
 - [23] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016.
 - [24] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.
 - [25] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*, 2021.
 - [26] Richard P Feynman. Quantum mechanical computers. *Between Quantum and Cosmos*, pages 523–548, 2017.
 - [27] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195, 2017.
 - [28] Aram W Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203, 2017.
 - [29] Xun Gao, Z-Y Zhang, and L-M Duan. A quantum machine learning algorithm based on generative models. *Science advances*, 4(12):eaat9004, 2018.
 - [30] Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. *Physical review letters*, 121(4):040502, 2018.
 - [31] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.
 - [32] 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.
 - [33] Yulin Wu, Wan-Su Bao, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, et al. Strong quantum computational advantage using a superconducting quantum processor. *Physical review letters*, 127(18):180501, 2021.
 - [34] Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, et al. Quantum advantage in learning from experiments. *arXiv preprint arXiv:2112.00778*, 2021.
 - [35] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):1–9, 2021.
 - [36] Xinbiao Wang, Yuxuan Du, Yong Luo, and Dacheng Tao. Towards understanding the power of quantum kernels in the NISQ era. *Quantum*, 5:531, August 2021.
 - [37] Yuxuan Du and Dacheng Tao. On exploring practical potentials of quantum auto-encoder with advantages. *arXiv preprint arXiv:2106.15432*, 2021.
 - [38] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.
 - [39] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022.
 - [40] Implicit probabilistic models do not specify the distribution of the data itself, but rather define a stochastic process that, after training, aims to draw samples from the underlying data distribution. Refer to SM A for details.
 - [41] Marcello Benedetti, Delfina Garcia-Pintos, Oscar Perdomo, Vicente Leyton-Ortega, Yunseong Nam, and Alejandro Perdomo-Ortiz. A generative modeling approach for benchmarking and training shallow quantum circuits. *npj Quantum Information*, 5(1):1–9, 2019.
 - [42] Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machines. *Physical Review A*, 98(6):062324, 2018.
 - [43] Brian Coyle, Maxwell Henderson, Justin Chan Jin Le, Niran Kumar, Marco Paini, and Elham Kashefi. Quantum versus classical generative modelling in finance. *Quantum Science and Technology*, 6(2):024013, 2021.
 - [44] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, and Dacheng Tao. Expressive power of parametrized quantum circuits. *Phys. Rev. Research*, 2:033125, Jul 2020.
 - [45] 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.
 - [46] Jinfeng Zeng, Yufeng Wu, Jin-Guo Liu, Lei Wang, and Jiangping Hu. Learning and inference on generative adversarial quantum circuits. *Physical Review A*, 99(5):052306, 2019.
 - [47] He-Liang Huang, Yuxuan Du, Ming Gong, Youwei Zhao, Yulin Wu, Chaoyue Wang, Shaowei Li, Futian Liang, Jin Lin, Yu Xu, et al. Experimental quantum generative adversarial networks for image generation. *Physical Review Applied*, 16(2):024051, 2021.
 - [48] Yiming Huang, Hang Lei, Xiaoyu Li, and Guowu Yang. Quantum maximum mean discrepancy gan. *Neurocomputing*, 454:88–100, 2021.
 - [49] Jonathan Romero and Alán Aspuru-Guzik. Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions. *Advanced Quantum Technologies*, 4(1):2000003, 2021.
 - [50] Kaitlin Gili, Marta Mauri, and Alejandro Perdomo-Ortiz. Evaluating generalization in classical and quantum generative models. *arXiv preprint arXiv:2201.08770*, 2022.
 - [51] Kouhei Nakaji and Naoki Yamamoto. Quantum semi-supervised generative adversarial network for enhanced data classification. *Scientific reports*, 11(1):1–10, 2021.

- [52] Paolo Braccia, Leonardo Banchi, and Filippo Caruso. Quantum noise sensing by generating fake noise. *Physical Review Applied*, 17(2):024002, 2022.
- [53] Abhinav Anand, Jonathan Romero, Matthias Degroote, and Alán Aspuru-Guzik. Noise robustness and experimental demonstration of a quantum generative adversarial network for continuous distributions. *Advanced Quantum Technologies*, 4(5):2000069, 2021.
- [54] Xu-Fei Yin, Yuxuan Du, Yue-Yang Fei, Rui Zhang, Li-Zheng Liu, Yingqiu Mao, Tongliang Liu, Min-Hsiu Hsieh, Li Li, Nai-Le Liu, Dacheng Tao, Yu-Ao Chen, and Jian-Wei Pan. Efficient Bipartite Entanglement Detection Scheme with a Quantum Adversarial Solver. *Phys. Rev. Lett.* 128, 110501 (2022), 2022. arXiv:2203.07749v1.
- [55] Daiwei Zhu, Norbert M Linke, Marcello Benedetti, Kevin A Landsman, Nhungh H Nguyen, C Huerta Alderete, Alejandro Perdomo-Ortiz, Nathan Korda, A Garfoot, Charles Brecque, et al. Training of quantum circuits on a hybrid quantum computer. *Science advances*, 5(10):eaaw9918, 2019.
- [56] Ling Hu, Shu-Hao Wu, Weizhou Cai, Yuwei Ma, Xiang-hao Mu, Yuan Xu, Haiyan Wang, Yipu Song, Dong-Ling Deng, Chang-Ling Zou, et al. Quantum generative adversarial learning in a superconducting quantum circuit. *Science advances*, 5(1):eaav2761, 2019.
- [57] Junde Li, Rasit O Topaloglu, and Swaroop Ghosh. Quantum generative models for small molecule drug discovery. *IEEE Transactions on Quantum Engineering*, 2:1–8, 2021.
- [58] Yu-Xin Jin, Jun-Jie Hu, Qi Li, Zhi-Cheng Luo, Fang-Yan Zhang, Hao Tang, Kun Qian, and Xian-Min Jin. Quantum Deep Learning for Mutant COVID-19 Strain Prediction, 2022. arXiv:2203.03556v1.
- [59] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). *arXiv preprint arXiv:1703.00573*, 2017.
- [60] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 6158–6169, 2019.
- [61] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, Shan You, and Dacheng Tao. Learnability of quantum neural networks. *PRX Quantum*, 2(4):040337, 2021.
- [62] Shengjia Zhao, Hongyu Ren, Arianna Yuan, Jiaming Song, Noah Goodman, and Stefano Ermon. Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31, 2018.
- [63] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021.
- [64] Leonardo Banchi, Jason Pereira, and Stefano Pirandola. Generalization in quantum machine learning: A quantum information standpoint. *PRX Quantum*, 2(4):040321, 2021.
- [65] Matthias C Caro, Hsin-Yuan Huang, M Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J Coles. Generalization in quantum machine learning from few training data. *arXiv preprint arXiv:2111.05292*, 2021.
- [66] Yuxuan Du, Zhuozhuo Tu, Xiao Yuan, and Dacheng Tao. Efficient measure for the expressivity of variational quantum algorithms. *Physical Review Letters*, 128(8):080506, 2022.
- [67] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Physical Review Letters*, 126(19):190505, 2021.
- [68] Junyu Liu, Khadijeh Najafi, Kunal Sharma, Francesco Tacchino, Liang Jiang, and Antonio Mezzacapo. An analytic theory for the dynamics of wide quantum neural networks, 2022. arXiv:2203.16711v1.
- [69] Yang Qian, Xinbiao Wang, Yuxuan Du, Xingyao Wu, and Dacheng Tao. The dilemma of quantum neural networks. *arXiv preprint arXiv:2106.04975*, 2021.
- [70] Ryan Sweke, Jean-Pierre Seifert, Dominik Hangleiter, and Jens Eisert. On the quantum versus classical learnability of discrete distributions. *Quantum*, 5:417, 2021.
- [71] Marcel Hinsche, Marios Ioannou, Alexander Nietner, Jonas Haferkamp, Yihui Quek, Dominik Hangleiter, Jean-Pierre Seifert, Jens Eisert, and Ryan Sweke. Learnability of the output distributions of local quantum circuits. *arXiv preprint arXiv:2110.05517*, 2021.
- [72] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(null):723–773, mar 2012.
- [73] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.
- [74] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122(4):040504, 2019.
- [75] Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001.
- [76] Hirotada Kobayashi, Keiji Matsumoto, and Tomoyuki Yamakami. Quantum merlin-arthur proof systems: Are multiple merlins more helpful to arthur? In *International Symposium on Algorithms and Computation*, pages 189–198. Springer, 2003.
- [77] GK Dziugaite, DM Roy, and Z Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Uncertainty in Artificial Intelligence-Proceedings of the 31st Conference, UAI 2015*, pages 258–267, 2015.
- [78] Francois-Xavier Briol, Alessandro Barp, Andrew B Duncan, and Mark Girolami. Statistical inference for generative models with maximum mean discrepancy. *arXiv preprint arXiv:1906.05944*, 2019.
- [79] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.
- [80] Shahar Mendelson. A few notes on statistical learning theory. In *Advanced lectures on machine learning*, pages 1–40. Springer, 2003.
- [81] 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.
- [82] Maria Kieferova, Ortiz Marrero Carlos, and Nathan Wiebe. Quantum generative training using r\enyi divergences. *arXiv preprint arXiv:2106.09567*, 2021.
- [83] Armin Lederer, Jonas Umlauft, and Sandra Hirche. *Uniform Error Bounds for Gaussian Process Regression with Application to Safe Control*. Curran Associates Inc., Red

- Hook, NY, USA, 2019.
- [84] Martin Plesch and Časlav Brukner. Quantum-state preparation with universal gate decompositions. *Physical Review A*, 83(3):032302, 2011.
- [85] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. Quantum state preparation with optimal circuit depth: Implementations and applications. *arXiv preprint arXiv:2201.11495*, 2022.
- [86] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595, 2018.
- [87] Qingling Zhu, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, Ming Gong, Cheng Guo, Chu Guo, Shaojun Guo, Lianchen Han, Linyin Hong, He-Liang Huang, Yong-Heng Huo, Liping Li, Na Li, Shaowei Li, Yuan Li, Futian Liang, Chun Lin, Jin Lin, Haoran Qian, Dan Qiao, Hao Rong, Hong Su, Lihua Sun, Liangyuan Wang, Shiyu Wang, Dachao Wu, Yulin Wu, Yu Xu, Kai Yan, Weifeng Yang, Yang Yang, Yangsen Ye, Jiahui Yin, Chong Ying, Jiale Yu, Chen Zha, Cha Zhang, Haibin Zhang, Kaili Zhang, Yiming Zhang, Han Zhao, Youwei Zhao, Liang Zhou, Chao-Yang Lu, Cheng-Zhi Peng, Xiaobo Zhu, and Jian-Wei Pan. Quantum computational advantage via 60-qubit 24-cycle random circuit sampling, 2021.
- [88] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. In *32nd Computational Complexity Conference*, page 1, 2017.
- [89] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.
- [90] Carlos Bravo-Prieto, Julien Baglio, Marco Cè, Anthony Francis, Dorota M Grabowska, and Stefano Carrazza. Style-based quantum generative adversarial networks for monte carlo events. *arXiv preprint arXiv:2110.06933*, 2021.
- [91] David Amaro, Carlo Modica, Matthias Rosenkranz, Mattia Fiorentini, Marcello Benedetti, and Michael Lubasch. Filtering variational quantum algorithms for combinatorial optimization. *Quantum Science and Technology*, 7(1):015021, 2022.
- [92] M Bilkis, M Cerezo, Guillaume Verdon, Patrick J Coles, and Lukasz Cincio. A semi-agnostic ansatz with variable structure for quantum machine learning. *arXiv preprint arXiv:2103.06712*, 2021.
- [93] Yuxuan Du, Tao Huang, Shan You, Min-Hsiu Hsieh, and Dacheng Tao. Quantum circuit architecture search: error mitigation and trainability enhancement for variational quantum solvers. *arXiv preprint arXiv:2010.10217*, 2020.
- [94] Kehuan Linghu, Yang Qian, Ruixia Wang, Meng-Jun Hu, Zhiyuan Li, Xuegang Li, Huikai Xu, Jingning Zhang, Teng Ma, Peng Zhao, et al. Quantum circuit architecture search on a superconducting processor. *arXiv preprint arXiv:2201.00934*, 2022.
- [95] En-Jui Kuo, Yao-Lung L Fang, and Samuel Yen-Chi Chen. Quantum architecture search via deep reinforcement learning. *arXiv preprint arXiv:2104.07715*, 2021.
- [96] Shi-Xin Zhang, Chang-Yu Hsieh, Shengyu Zhang, and Hong Yao. Differentiable quantum architecture search. *arXiv preprint arXiv:2010.08561*, 2020.
- [97] Shouvanik Chakrabarti, Huang Yiming, Tongyang Li, Soheil Feizi, and Xiaodi Wu. Quantum wasserstein generative adversarial networks. *Advances in Neural Information Processing Systems*, 32, 2019.
- [98] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223, 2017.
- [99] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.
- [100] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017.
- [101] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.
- [102] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [103] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [104] Marcello Benedetti, Edward Grant, Leonard Wossnig, and Simone Severini. Adversarial quantum circuit learning for pure state approximation. *New Journal of Physics*, 21(4):043023, 2019.
- [105] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.
- [106] Jacob Biamonte. Universal variational quantum computation. *Physical Review A*, 103(3):L030401, 2021.
- [107] Nicolas Le Roux and Yoshua Bengio. Representational power of restricted boltzmann machines and deep belief networks. *Neural computation*, 20(6):1631–1649, 2008.
- [108] Xun Gao and Lu-Ming Duan. Efficient representation of quantum many-body states with deep neural networks. *Nature communications*, 8(1):662, 2017.
- [109] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2 edition, 2018.
- [110] Richard M Dudley. The sizes of compact subsets of hilbert space and continuity of gaussian processes. *Journal of Functional Analysis*, 1(3):290–330, 1967.
- [111] Thomas Barthel and Jianfeng Lu. Fundamental limitations for measurements in quantum many-body systems. *Phys. Rev. Lett.*, 121:080406, Aug 2018.
- [112] Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *Siam journal on computing*, 35(5):1070–1097, 2006.

Supplementary Material: “Power of Quantum Generative Learning”

SM A: Schematic of QGANs in the discrete and continuous settings

For the purpose of elucidating, in this section, we first introduce the basic theory of GANs and QGANs, especially for QGANs with MMD loss, and then demonstrate the equivalence of QCBMs and QGANs in the discrete setting when the loss function is specified to be MMD.

Basic theory of (classical) GANs and QGANs when \mathbb{Q} is continuous. The fundamental mechanism of GAN [6] and its variations [98–101] is as follows. GAN sets up a two-players game: the generator G creates data that pretends to come from the real data distribution \mathbb{Q} to fool the discriminator D , while D tries to distinguish the fake generated data from the real training data. Mathematically, G and D corresponds to two differentiable functions. In particular, the input of G is a latent variable \mathbf{z} and its output is \mathbf{x} , i.e., $G : G(\mathbf{z}, \boldsymbol{\theta}) \rightarrow \mathbf{x}$ with $\boldsymbol{\theta}$ being trainable parameters for G . The role of the latent variable \mathbf{z} is ensuring GAN to be a structured probabilistic model [102]. The input of D can either be the generated data \mathbf{x} or the real data $\mathbf{y} \sim \mathbb{Q}$ and its output corresponds to the binary classification result (real or fake), respectively. The mathematical expression of D yields $D : D(\mathbf{x}, \mathbf{y}, \boldsymbol{\gamma}) \rightarrow (0, 1)$ with $\boldsymbol{\gamma}$ being trainable parameters for D . If the distribution learned by G equals to the real data distribution, i.e., $\mathbb{P}_{\boldsymbol{\theta}} = \mathbb{Q}$, then D can never discriminate between the generated data and the real data and this unique solution is called Nash equilibrium [6].

To reach the Nash equilibrium, the training process of GANs corresponds to the minimax optimization. Namely, the discriminator D updates $\boldsymbol{\gamma}$ to maximize the classification accuracy, while the generator G updates $\boldsymbol{\theta}$ to minimize the classification accuracy. With this regard, the optimization of GAN follows

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\gamma}} \mathcal{L}(D_{\boldsymbol{\gamma}}(G_{\boldsymbol{\theta}}(\mathbf{z})), D_{\boldsymbol{\gamma}}(\mathbf{x})) := \mathbb{E}_{\mathbf{x} \sim \mathbb{Q}}[D_{\boldsymbol{\gamma}}(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim \mathbb{P}_{\boldsymbol{\theta}}}[(1 - D_{\boldsymbol{\gamma}}(G_{\boldsymbol{\theta}}(\mathbf{z})))] \quad (\text{A1})$$

where \mathbb{Q} is the distribution of training dataset, and $\mathbb{P}_{\boldsymbol{\theta}}$ is the probability distribution of the latent variable \mathbf{z} . In general, $G_{\boldsymbol{\theta}}$ and $D_{\boldsymbol{\gamma}}$ are constructed by deep neural networks, and their parameters are updated iteratively using gradient descent methods [103].

The key difference between GANs and QGANs is the way of implementing $G_{\boldsymbol{\theta}}$ and $D_{\boldsymbol{\gamma}}$. Particularly, in QGANs, either G , D , or both can be realized by variational quantum circuits instead of deep neural networks. The training strategy of QGANs is similar to classical GANs. In this study, we focus on QGANs with MMD loss, which can be treated as the quantum extension of MMD-GAN [77]. Unlike conventional GANs and QGANs, MMD-GAN and QGAN replace a trainable discriminator with MMD. In this way, the family of discriminators is substituted with a family \mathcal{H} of test functions, closed under negation, where the optimization of D can be completed with the analytical form. Therefore, the goal of QGANs is finding an estimator minimizing an unbiased MMD loss, i.e.,

$$\text{MMD}_U^2(\mathbb{P}_{\boldsymbol{\theta}} \parallel \mathbb{Q}) := \frac{1}{n(n-1)} \sum_{i \neq i'}^n k(\mathbf{x}^{(i)}, \mathbf{x}^{(i')}) + \frac{1}{m(m-1)} \sum_{j \neq j'}^m k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')}) - \frac{2}{nm} \sum_{i,j} k(\mathbf{x}^{(i)}, \mathbf{y}^{(j)}) \quad (\text{A2})$$

where $\mathbf{x}^{(i)} \sim \mathbb{P}_{\boldsymbol{\theta}}$ and $\mathbf{y}^{(j)} \sim \mathbb{Q}$.

Equivalence between QCBMs and QGANs when \mathbb{Q} is discrete. In accordance with the explanations in Refs. [45, 104], when QGAN is applied to estimate a discrete distribution \mathbb{Q} (e.g., quantum state approximation), the quantum generator aims to directly capture the distribution of the data itself. This violates the criteria of implicit generative models, where a stochastic process is employed to draw samples from the underlying data distribution after training. More specifically, when \mathbb{Q} is discrete, the output of the quantum circuit for both QCBM and QGAN takes the form $\mathbb{P}_{\boldsymbol{\theta}}(i) = \text{Tr}(\Pi_i \hat{U}(\boldsymbol{\theta}) \rho_0 \hat{U}(\boldsymbol{\theta})^\dagger)$ in Eq. (1). The concept ‘adversarial’ originates from the way of optimizing $\boldsymbol{\theta}$. Instead of using a deterministic distance measure (e.g., KL divergence) as in QCBMs, QGANs utilize a discriminator $D_{\boldsymbol{\gamma}}$, implemented by either trainable parameterized quantum circuit or a neural network, to maximally separate $\mathbb{P}_{\boldsymbol{\theta}}(i)$ from \mathbb{Q} . The behavior of simultaneously update $\boldsymbol{\theta}$ (to minimize the loss) and $\boldsymbol{\gamma}$ (to maximize the loss) is termed as quantum generative adversarial learning. With this regard, when we replace the trainable $D_{\boldsymbol{\gamma}}$ by the deterministic measure MMD, QGAN takes an equivalent mathematical form with QCBM.

SM B: Optimization of QGANs with MMD loss

For self-consistency, in this section, we introduce the elementary backgrounds of the optimization of QGLMs with MMD loss. See Ref. [72] for elaborations. A central concept used in MMD loss is kernels.

Definition 2 (Definition 2, [74]). Let \mathcal{X} be a nonempty set, called the input set. A function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{C}$ is called kernel if the Gram matrix \mathcal{K} with entries $\mathcal{K}_{m,m'} = k(\mathbf{x}^m, \mathbf{x}^{m'})$ is positive semi-definite.

MMD loss. Let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a Borel measurable kernel on \mathcal{X} , and consider the reproducing kernel Hilbert space \mathcal{H}_k associated with k (see Berlinet and Thomas-Agnan [2004]), equipped with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_k}$. Let $\mathcal{P}_k(\mathcal{X})$ be the set of Borel probability measures μ such that $\int_{\mathcal{X}} \sqrt{k(\mathbf{x}, \mathbf{x})} \mu(d\mathbf{x}) < \infty$. The kernel mean embedding $\Pi_k(\mu) = \int k(\cdot, y) \mu(dy)$, interpreted as a Bochner integral, defines a continuous embedding from $\mathcal{P}_k(\mathcal{X})$ into \mathcal{H}_k . The mean embedding pulls-back the metric on \mathcal{H}_k generated by the inner product to define a pseudo-metric on $\mathcal{P}_k(\mathcal{X})$ called the maximum mean discrepancy MMD: $\mathcal{P}_k(\mathcal{X}) \times \mathcal{P}_k(\mathcal{X}) \rightarrow \mathbb{R}_+$, i.e.,

$$\text{MMD}(\mathbb{P}_1 \parallel \mathbb{P}_2) = \|\Pi_k(\mathbb{P}_1) - \Pi_k(\mathbb{P}_2)\|_{\mathcal{H}_k}. \quad (\text{B1})$$

The MMD loss has a particularly simple expression that can be derived through an application of the reproducing property ($f(x) = \langle f, k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}_k}$), i.e.,

$$\begin{aligned} \text{MMD}^2(\mathbb{P}_1 \parallel \mathbb{P}_2) &:= \left\| \int_{\mathcal{X}} k(\cdot, \mathbf{x}) \mathbb{P}_1(d\mathbf{x}) - \int_{\mathcal{X}} k(\cdot, \mathbf{x}) \mathbb{P}_2(d\mathbf{x}) \right\|_{\mathcal{H}_k}^2 \\ &= \int_{\mathcal{X}} \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{y}) \mathbb{P}_1(d\mathbf{x}) \mathbb{P}_1(d\mathbf{y}) - 2 \int_{\mathcal{X}} \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{y}) \mathbb{P}_1(d\mathbf{x}) \mathbb{P}_2(d\mathbf{y}) + \int_{\mathcal{X}} \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{y}) \mathbb{P}_2(d\mathbf{x}) \mathbb{P}_2(d\mathbf{y}) \\ &= \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim \mathbb{P}_1}(k(\mathbf{x}, \mathbf{y})) - 2 \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_1, \mathbf{y} \sim \mathbb{P}_2}(k(\mathbf{x}, \mathbf{y})) + \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim \mathbb{P}_2}(k(\mathbf{x}, \mathbf{y})), \end{aligned} \quad (\text{B2})$$

which provides a closed form expression up to calculation of expectations.

Optimization of QCBMs with MMD loss. The goal of QCBMs is finding an estimator minimizing the loss function $\text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} \parallel \mathbb{Q})$ in Eq. (B2), where $\mathbb{P}_{\boldsymbol{\theta}}$ is defined in Eq. (1). The optimization is completed by the gradient based descent optimizer. The updating rule satisfies $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} \parallel \mathbb{Q})$ and η is the learning rate. Concretely, the partial derivative of the j -th entry satisfies

$$\begin{aligned} &\frac{\partial \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} \parallel \mathbb{Q})}{\partial \boldsymbol{\theta}_j} \\ &= \frac{\partial \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim \mathbb{P}_{\boldsymbol{\theta}}}(k(\mathbf{x}, \mathbf{y})) - 2 \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\boldsymbol{\theta}}, \mathbf{y} \sim \mathbb{Q}}(k(\mathbf{x}, \mathbf{y})) + \mathbb{E}_{\mathbf{x}, \mathbf{y} \sim \mathbb{Q}}(k(\mathbf{x}, \mathbf{y}))}{\partial \boldsymbol{\theta}_j} \\ &= \sum_{\mathbf{x}, \mathbf{y}} k(\mathbf{x}, \mathbf{y}) \left(\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{y}) \frac{\partial \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \boldsymbol{\theta}_j} + \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) \frac{\partial \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{y})}{\partial \boldsymbol{\theta}_j} \right) - 2 \sum_{\mathbf{x}, \mathbf{y}} k(\mathbf{x}, \mathbf{y}) \frac{\partial \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x})}{\partial \boldsymbol{\theta}_j} \mathbb{Q}(\mathbf{y}) \\ &= \sum_{\mathbf{x}, \mathbf{y}} k(\mathbf{x}, \mathbf{y}) \left(\mathbb{P}_{\boldsymbol{\theta}}(\mathbf{y}) (\mathbb{P}_{\boldsymbol{\theta} + \frac{\pi}{2} \mathbf{e}_j}(\mathbf{x}) - \mathbb{P}_{\boldsymbol{\theta} - \frac{\pi}{2} \mathbf{e}_j}(\mathbf{x})) + \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) (\mathbb{P}_{\boldsymbol{\theta} + \frac{\pi}{2} \mathbf{e}_j}(\mathbf{y}) - \mathbb{P}_{\boldsymbol{\theta} - \frac{\pi}{2} \mathbf{e}_j}(\mathbf{y})) \right) \\ &\quad - 2 \sum_{\mathbf{x}, \mathbf{y}} k(\mathbf{x}, \mathbf{y}) (\mathbb{P}_{\boldsymbol{\theta} + \frac{\pi}{2} \mathbf{e}_j}(\mathbf{x}) - \mathbb{P}_{\boldsymbol{\theta} - \frac{\pi}{2} \mathbf{e}_j}(\mathbf{x})) \mathbb{Q}(\mathbf{y}) \\ &= \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\boldsymbol{\theta} + \frac{\pi}{2} \mathbf{e}_j}, \mathbf{y} \sim \mathbb{P}_{\boldsymbol{\theta}}}(k(\mathbf{x}, \mathbf{y})) - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\boldsymbol{\theta} - \frac{\pi}{2} \mathbf{e}_j}, \mathbf{y} \sim \mathbb{P}_{\boldsymbol{\theta}}}(k(\mathbf{x}, \mathbf{y})) + \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\boldsymbol{\theta}}, \mathbf{y} \sim \mathbb{P}_{\boldsymbol{\theta} + \frac{\pi}{2} \mathbf{e}_j}}(k(\mathbf{x}, \mathbf{y})) - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\boldsymbol{\theta}}, \mathbf{y} \sim \mathbb{P}_{\boldsymbol{\theta} - \frac{\pi}{2} \mathbf{e}_j}}(k(\mathbf{x}, \mathbf{y})) \\ &\quad - 2 \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\boldsymbol{\theta} + \frac{\pi}{2} \mathbf{e}_j}, \mathbf{y} \sim \mathbb{Q}}(k(\mathbf{x}, \mathbf{y})) + 2 \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\boldsymbol{\theta} - \frac{\pi}{2} \mathbf{e}_j}, \mathbf{y} \sim \mathbb{Q}}(k(\mathbf{x}, \mathbf{y})), \end{aligned} \quad (\text{B3})$$

where the last second equality employs the parameter shift rule [105] to calculate the partial derivative $\partial \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{y}) / \partial \boldsymbol{\theta}_j$ and $\partial \mathbb{P}_{\boldsymbol{\theta}}(\mathbf{x}) / \partial \boldsymbol{\theta}_j$. According to Lemma 1, the six expectation terms in Eq. (B3) can be analytically and efficiently calculated when the $k(\cdot, \cdot)$ is quantum. In the case of classical kernels, the six expectation terms in Eq. (B3) are estimated by the sample mean.

Optimization of QGANs with MMD loss. We next derive the gradients of QGANs with respect to the ℓ -th entry. Since the evaluation of expectation is runtime expensive when \mathbb{Q} is continuous, QGANs employ an unbiased estimator of the MMD loss in Eq. (A2) to update $\boldsymbol{\theta}$. The updating rule at the t -th iteration is $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} \text{MMD}_U^2(\mathbb{P}_{\boldsymbol{\theta}} \parallel \mathbb{Q})$.

According to the chain rule, we have

$$\begin{aligned} & \frac{\partial \text{MMD}_U^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q})}{\partial \boldsymbol{\theta}_\ell} \\ &= \frac{\partial \frac{1}{n(n-1)} \sum_{i \neq i'}^n k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i')})) - \frac{2}{nm} \sum_{i,j} k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), \mathbf{y}^{(j)})}{\partial \boldsymbol{\theta}_\ell} \end{aligned} \quad (\text{B4})$$

$$= \frac{1}{n(n-1)} \sum_{i \neq i'}^n \frac{\partial k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i')}))}{\partial G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)})} \frac{\partial G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)})}{\partial \boldsymbol{\theta}_\ell} + \frac{\partial k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i')}))}{\partial G_{\boldsymbol{\theta}}(\mathbf{z}^{(i')})} \frac{\partial G_{\boldsymbol{\theta}}(\mathbf{z}^{(i')})}{\partial \boldsymbol{\theta}_\ell} -$$

$$\frac{2}{nm} \sum_{i,j} \frac{\partial k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), \mathbf{y}^{(j)})}{\partial G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)})} \frac{\partial G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)})}{\partial \boldsymbol{\theta}_\ell} \quad (\text{B5})$$

where the first equality uses $\partial \frac{1}{m(m-1)} \sum_{j \neq j'}^m k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')}) / \partial \boldsymbol{\theta}_\ell = 0$, each derivative $\partial k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i')})) / \partial G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)})$ can be easily computed for standard kernels, and the derivative $\partial G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}) / \partial \boldsymbol{\theta}_\ell$ for $\forall i \in n, j \in [m]$ can be computed via the parameter shift rule. Therefore, the gradients of QGANs with MMD loss can be achieved.

SM C: The comparison between QCBMs and GLMs for estimating discrete distributions

In this section, we emphasize a central question in QGLMs, i.e., when both variational quantum circuits and neural networks are used to implement $\mathbb{P}_{\boldsymbol{\theta}}$, which one can attain a lower $\inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q})$. The importance of this issue comes from Eq. (7), where the generalization error bound becomes meaningful when $\inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q})$ is small. In this respect, it is necessary to understand whether QCBMs allow a lower $\inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q})$ over (classical) GLMs.

In what follows, we analyze when QCBMs promise a lower $\inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q})$ over a typical GLM—restricted Boltzmann machine (RBM) [79]. To be more specific, consider that both QCBMs and RBMs are universal approximators with an exponential number of trainable parameters [106, 107] with $\inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q}) \rightarrow 0$, we focus on the more practical scenario in which the number of parameters polynomially scales with the feature dimension, the qudit count, and the number of visible neurons. Denote the space of the parameterized distributions formed by QCBMs (or RBM) as $\mathbb{P}_{\boldsymbol{\Theta}}^{\text{QCBM}}$ (or $\mathbb{P}_{\boldsymbol{\Theta}}^{\text{RBM}}$). The superiority of QCBMs can be identified by showing $\inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}}^{\text{QCBM}} || \mathbb{Q}) \leq \inf_{\boldsymbol{\theta} \in \Theta} \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}}^{\text{RBM}} || \mathbb{Q})$. This amounts to finding a distribution \mathbb{Q} satisfying

$$(\mathbb{Q} \in \mathbb{P}_{\boldsymbol{\Theta}}^{\text{QCBM}}) \wedge (\mathbb{Q} \notin \mathbb{P}_{\boldsymbol{\Theta}}^{\text{RBM}}). \quad (\text{C1})$$

According to the results in [108], there is a large class of quantum states meeting the above requirement. Representative examples include projected entangled pair states and ground states of k -local Hamiltonians.

SM D: Proof of Theorem 2 (generalization of QGANs)

The proof of Theorem 2 utilizes the following three lemmas. For clearness, we defer the proof of Lemmas 5 and 6 to SM D 1 and D 2, respectively.

Lemma 4 (McDiarmids inequality, [80]). *Let $f : \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_N \rightarrow \mathbb{R}$ and assume there exists $c_1, \dots, c_N \geq 0$ such that, for all $k \in \{1, \dots, N\}$, we have*

$$\sup_{\mathbf{x}_1, \dots, \mathbf{x}_k, \tilde{\mathbf{x}}_k, \dots, \mathbf{x}_N} |f(\mathbf{x}_1, \dots, \mathbf{x}_k, \dots, \mathbf{x}_N) - f(\mathbf{x}_1, \dots, \tilde{\mathbf{x}}_k, \dots, \mathbf{x}_N)| \leq c_k. \quad (\text{D1})$$

Then for all $\epsilon \geq 0$ and independent random variables ξ_1, \dots, ξ_N in \mathcal{X} ,

$$\Pr(|f(\xi_1, \dots, \xi_N) - \mathbb{E}(f(\xi_1, \dots, \xi_N))| \geq \epsilon) \leq \exp\left(\frac{-2\epsilon^2}{\sum_{n=1}^N c_n^2}\right). \quad (\text{D2})$$

Lemma 5. Following the notations in Theorem 2, define $\mathcal{G} = \{k(G_{\boldsymbol{\theta}}(\cdot), G_{\boldsymbol{\theta}}(\cdot)) | \boldsymbol{\theta} \in \Theta\}$ and $\mathcal{G}_+ = \{k(G_{\boldsymbol{\theta}}(\mathbf{z}), G_{\boldsymbol{\theta}}(\cdot)) | \boldsymbol{\theta} \in \Theta, \mathbf{z} \in \mathcal{Z}\}$. Given the set $\mathcal{S} = \{\mathbf{z}^{(i)}\}_{i=1}^n$, we have

$$\begin{aligned} & \mathbb{E} \left(\sup_{\boldsymbol{\theta} \in \Theta} \left| \mathbb{E}_{\mathbf{z}, \mathbf{z}'} (k(G_{\boldsymbol{\theta}}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}')))) - \frac{1}{n(n-1)} \sum_{i \neq i'} k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i')}))) \right| \right) \\ & \leq \frac{8}{n-1} + \frac{24\sqrt{d^{2k}(N_{ge} + N_{gt})}}{n-1} (1 + N \ln(441dC_3^2(n-1)N_{ge}N_{gt})) . \end{aligned} \quad (\text{D3})$$

Lemma 6. Following the notations in Theorem 2, define $\mathcal{W} = \{k(G_{\boldsymbol{\theta}}(\cdot), \cdot) | \boldsymbol{\theta} \in \Theta\}$ and $\mathcal{W}_+ = \{k(G_{\boldsymbol{\theta}}(\cdot), \mathbf{y}) | \boldsymbol{\theta} \in \Theta, \mathbf{y} \in \mathcal{Y}\}$. Given the set $\mathcal{S} = \{\mathbf{z}^{(i)}\}_{i=1}^n$ and the set $\{\mathbf{y}^{(j)}\}_{j=1}^m$, we have

$$\begin{aligned} & \mathbb{E} \left(\sup_{\boldsymbol{\theta} \in \Theta} \left| \mathbb{E}_{\mathbf{z}, \mathbf{y}} (k(G_{\boldsymbol{\theta}}(\mathbf{z}), \mathbf{y})) - \frac{1}{mn} \sum_{i \in [n], j \in [m]} k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), \mathbf{y}^{(j)}) \right| \right) \\ & \leq \frac{8}{n} + \frac{24\sqrt{d^{2k}(N_{ge} + N_{gt})}}{n} (1 + N \ln(441dC_3^2 n N_{ge} N_{gt})) . \end{aligned} \quad (\text{D4})$$

We are now ready to prove Theorem 2.

Proof of Theorem 2. Let $\mathcal{E}(\boldsymbol{\theta}) = \text{MMD}_U^2(\mathbb{P}_{\boldsymbol{\theta}}^n || \mathbb{Q}^m)$ and $\mathcal{T}(\boldsymbol{\theta}) = \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q})$. Note that the generalization error equals to

$$\mathfrak{R}^C = \mathcal{T}(\hat{\boldsymbol{\theta}}^{(n,m)}) - \mathcal{T}(\boldsymbol{\theta}^*). \quad (\text{D5})$$

In the remainder of the proof, when no confusion occurs, we abbreviate $\hat{\boldsymbol{\theta}}^{(n,m)}$ as $\hat{\boldsymbol{\theta}}$ for clearness. The above equation is upper bounded by

$$\mathfrak{R}^C = \mathcal{E}(\hat{\boldsymbol{\theta}}) - \mathcal{E}(\hat{\boldsymbol{\theta}}) + \mathcal{T}(\hat{\boldsymbol{\theta}}) - \mathcal{T}(\boldsymbol{\theta}^*) \leq \mathcal{E}(\boldsymbol{\theta}^*) - \mathcal{E}(\hat{\boldsymbol{\theta}}) + \mathcal{T}(\hat{\boldsymbol{\theta}}) - \mathcal{T}(\boldsymbol{\theta}^*) \leq |\mathcal{T}(\hat{\boldsymbol{\theta}}) - \mathcal{E}(\hat{\boldsymbol{\theta}})| + |\mathcal{T}(\boldsymbol{\theta}^*) - \mathcal{E}(\boldsymbol{\theta}^*)|, \quad (\text{D6})$$

where the first inequality employs the definition of $\hat{\boldsymbol{\theta}}$ with $\mathcal{E}(\boldsymbol{\theta}^*) \geq \mathcal{E}(\hat{\boldsymbol{\theta}})$ and the second inequality uses the property of absolute value function. In the following, we derive the probability for $\sup_{\boldsymbol{\theta} \in \Theta} |\mathcal{T}(\boldsymbol{\theta}) - \mathcal{E}(\boldsymbol{\theta})| < \epsilon$, which in turn can achieve the upper bound of \mathfrak{R}^C .

According to the explicit form of the MMD loss, $\sup_{\boldsymbol{\theta} \in \Theta} |\mathcal{E}(\boldsymbol{\theta}) - \mathcal{T}(\boldsymbol{\theta})|$ satisfies

$$\begin{aligned} & \sup_{\boldsymbol{\theta} \in \Theta} |\text{MMD}_U^2(\mathbb{P}_{\boldsymbol{\theta}}^n || \mathbb{Q}^m) - \text{MMD}^2(\mathbb{P}_{\boldsymbol{\theta}} || \mathbb{Q})| \\ & = \sup_{\boldsymbol{\theta} \in \Theta} \left| \mathbb{E}_{\mathbf{z}, \mathbf{z}'} (k(G_{\boldsymbol{\theta}}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}'))) - 2 \mathbb{E}_{\mathbf{z}, \mathbf{y}} (k(G_{\boldsymbol{\theta}}(\mathbf{z}), \mathbf{y})) + \mathbb{E}_{\mathbf{y}, \mathbf{y}'} (k(\mathbf{y}, \mathbf{y}')) - \frac{1}{n(n-1)} \sum_{i \neq i'} k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i')})) \right. \\ & \quad \left. + \frac{2}{mn} \sum_{i \in [n], j \in [m]} k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), \mathbf{y}^{(j)}) - \frac{1}{m(m-1)} \sum_{j \neq j'} k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')}) \right| \\ & \leq \underbrace{\sup_{\boldsymbol{\theta} \in \Theta} \left| \mathbb{E}_{\mathbf{y}, \mathbf{y}'} (k(\mathbf{y}, \mathbf{y}')) - \frac{1}{m(m-1)} \sum_{j \neq j'} k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')}) \right|}_{T1} + \underbrace{\sup_{\boldsymbol{\theta} \in \Theta} \left| \mathbb{E}_{\mathbf{z}, \mathbf{z}'} (k(G_{\boldsymbol{\theta}}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}'))) - \frac{1}{n(n-1)} \sum_{i \neq i'} k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i')})) \right|}_{T2} \\ & \quad + \underbrace{2 \sup_{\boldsymbol{\theta} \in \Theta} \left| \mathbb{E}_{\mathbf{z}, \mathbf{y}} (k(G_{\boldsymbol{\theta}}(\mathbf{z}), \mathbf{y})) - \frac{1}{mn} \sum_{i \in [n], j \in [m]} k(G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}), \mathbf{y}^{(j)}) \right|}_{T3} \end{aligned} \quad (\text{D7})$$

where the inequality comes from the Jensen inequality.

We next separately derive the upper bounds of the terms $T1$, $T2$, and $T3$ in Eq. (D7).

Upper bound of $T1$. The upper bound of $T1$ only depends on the examples sampled from the target distribution \mathbb{Q} , which is independent of $\boldsymbol{\theta} \in \Theta$. With this regard, $T1$ can be taken out of the supremum and we apply the concentration inequality in Lemma 4 to derive the upper bound of $|\mathbb{E}_{\mathbf{y}, \mathbf{y}'} (k(\mathbf{y}, \mathbf{y}')) - \frac{1}{m(m-1)} \sum_{j \neq j'} k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')})|$.

Recall the precondition of employing Lemma 4 is finding the upper bound on $f(\cdot)$. Let the function $f(\cdot)$ be $\frac{1}{m(m-1)} \sum_{j \neq j'} k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')})$. For each $\ell \in \{1, \dots, m\}$, the desired upper bound yields

$$\begin{aligned} & \left| -\frac{1}{m(m-1)} \left(\sum_{j \neq j', j \neq \ell} k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')}) + \sum_{j' \neq \ell} k(\mathbf{y}^{(\ell)}, \mathbf{y}^{(j')}) \right) + \frac{1}{m(m-1)} \left(\sum_{j \neq j', j \neq \ell} k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')}) + \sum_{j' \neq \ell} k(\tilde{\mathbf{y}}^{(\ell)}, \mathbf{y}^{(j')}) \right) \right| \\ &= \left| \frac{1}{m(m-1)} \sum_{j' \neq \ell} \left(k(\mathbf{y}^{(\ell)}, \mathbf{y}^{(j')}) - k(\tilde{\mathbf{y}}^{(\ell)}, \mathbf{y}^{(j')}) \right) \right| \\ &\leq \frac{2C_2}{m}, \end{aligned} \quad (\text{D8})$$

where the inequality leverages the assumption that the kernel $k(\cdot, \cdot)$ is upper bounded by C_2 .

Given this upper bound, we obtain

$$\begin{aligned} & \Pr(T1 \geq \epsilon) \\ &= \Pr \left(\left| \mathbb{E}_{\mathbf{y}, \mathbf{y}'}(k(\mathbf{y}, \mathbf{y}')) - \frac{1}{m(m-1)} \sum_{j \neq j'} k(\mathbf{y}^{(j)}, \mathbf{y}^{(j')}) \right| \geq \epsilon \right) \\ &\leq \exp \left(-\frac{\epsilon^2}{8C_2^2} m \right) = \delta_{T1}, \end{aligned} \quad (\text{D9})$$

where the inequality exploits the results in Lemma 4.

Upper bound of T2. We next use the concentration inequality to quantify the upper bound of T2. The derivation is similar to that of $T1$. In particular, supported by the results of Lemma 4, we have

$$\Pr(|T2 - \mathbb{E}(T2)| \geq \epsilon) \leq \exp \left(-\frac{\epsilon^2}{8C_2^2} n \right) = \delta_{T2}. \quad (\text{D10})$$

Suppose that $\mathbb{E}(T2) \leq \epsilon_1$, an immediate observation is that

$$\Pr(T2 \geq \epsilon_1 + \epsilon) \leq \exp \left(-\frac{\epsilon^2}{8C_2^2} n \right) = \delta_{T2}. \quad (\text{D11})$$

In other words, the derivation of the upper bound of T2 amounts to analyzing the upper bound ϵ_1 .

Upper bound of T3. Following the same routine with the derivation of the upper bound of T2, we obtain

$$\Pr(|T3 - \mathbb{E}(T3)| \geq \epsilon) \leq \exp \left(-\frac{\epsilon^2}{8C_2^2} \frac{nm}{n+m} \right) = \delta_{T3}. \quad (\text{D12})$$

Suppose that $\mathbb{E}(T3) \leq \epsilon_2$. The above result hints that

$$\Pr(T3 \geq \epsilon_2 + \epsilon) \leq \exp \left(-\frac{\epsilon^2}{8C_2^2} \frac{nm}{n+m} \right) = \delta_{T3}. \quad (\text{D13})$$

Summing up Eqs. (D9), (D11), and (D13), the union bound gives

$$\begin{aligned} & \Pr \left(\sup_{\boldsymbol{\theta} \in \Theta} |\mathcal{E}(\boldsymbol{\theta}) - \mathcal{T}(\boldsymbol{\theta})| \geq \epsilon_1 + 2\epsilon_2 + 4\epsilon \right) \leq \delta_{T1} + \delta_{T2} + \delta_{T3} \\ & \Rightarrow \Pr \left(\sup_{\boldsymbol{\theta} \in \Theta} |\mathcal{E}(\boldsymbol{\theta}) - \mathcal{T}(\boldsymbol{\theta})| \geq \epsilon_1 + 2\epsilon_2 + 4\epsilon \right) \leq 3\delta_{T3} \\ & \Rightarrow \Pr \left(|\mathcal{E}(\hat{\boldsymbol{\theta}}) - \mathcal{T}(\hat{\boldsymbol{\theta}})| \geq \epsilon_1 + 2\epsilon_2 + 4\epsilon \right) \leq 3\delta_{T3}. \end{aligned} \quad (\text{D14})$$

This yields that with probability at least $1 - 3\delta_{T3}$,

$$2(\epsilon_1 + 2\epsilon_2 + 4\epsilon) \geq |\mathcal{E}(\hat{\boldsymbol{\theta}}) - \mathcal{T}(\hat{\boldsymbol{\theta}})| + |\mathcal{E}(\boldsymbol{\theta}^*) - \mathcal{T}(\boldsymbol{\theta}^*)| \geq |\mathcal{E}(\hat{\boldsymbol{\theta}}) - \mathcal{T}(\hat{\boldsymbol{\theta}}) - \mathcal{E}(\boldsymbol{\theta}^*) + \mathcal{T}(\boldsymbol{\theta}^*)| \geq |\mathcal{T}(\hat{\boldsymbol{\theta}}) - \mathcal{T}(\boldsymbol{\theta}^*)| = \mathfrak{R}^C. \quad (\text{D15})$$

According to the explicit forms of ϵ_1 and ϵ_2 achieved in Lemmas 5 and 6, with probability $1 - 3\delta_{T3}$, the generalization error of QGANs is upper bounded by

$$\begin{aligned} \mathfrak{R}^C &\leq 8\epsilon + 2 \left(\frac{8}{n-1} + \frac{24\sqrt{d^{2k}(N_{ge} + N_{gt})}}{n-1} (1 + N \ln(1764C_3^2(n-1)N_{ge}N_{gt})) \right) \\ &\quad + 4 \left(\frac{8}{n} + \frac{24\sqrt{d^{2k}(N_{ge} + N_{gt})}}{n} (1 + N \ln(1764C_3^2nN_{ge}N_{gt})) \right) \\ &\leq 8\sqrt{\frac{8C_2^2(n+m)}{nm} \ln(\frac{1}{3\delta_{T3}})} + 6 \left(\frac{8}{n-1} + \frac{24\sqrt{d^{2k}(N_{gt} + N_{ge})}}{n-1} (1 + N \ln(441dC_3^2nN_{ge}N_{gt})) \right). \end{aligned} \quad (\text{D16})$$

where the last inequality uses the relation between δ_{T3} and ϵ in Eq. (D13) with $\epsilon = \sqrt{8C_2^2(n+m) \ln(1/\delta_{T3})/(nm)}$, $1/n < 1/(n-1)$, and $n-1 < n$. ■

1. Proof of Lemma 5

Recall Lemma 5 aims to derive the upper bound of $\mathbb{E}(T2)$ in Eq. (D7). In Ref. [77], the authors utilize a statistical measure named Rademacher complexity to quantify these two terms. Different from the classical counterpart, here we adopt an another statistical measure, i.e., covering number, to derive the upper bound of $\mathbb{E}(T2)$. This measure allows us to identify how $\mathbb{E}(T2)$ scales with the qudit count N and the architecture of the employed Ansatz such as the trainable parameters N_{gt} and the types of the quantum gates. For self-consistency, we provide the formal definition of covering number and Rademacher as follows.

Definition 3 (Covering number, [109]). *The covering number $\mathcal{N}(\mathcal{U}, \epsilon, \|\cdot\|)$ denotes the least cardinality of any subset $\mathcal{V} \subset \mathcal{U}$ that covers \mathcal{U} at scale ϵ with a norm $\|\cdot\|$, i.e.,*

$$\sup_{A \in \mathcal{U}} \min_{B \in \mathcal{V}} \|A - B\| \leq \epsilon. \quad (\text{D17})$$

Definition 4 (Rademacher, [109]). *Let μ be a probability measure on \mathcal{X} , and let \mathcal{F} be a class of uniformly bounded functions on \mathcal{X} . Then the Rademacher complexity of \mathcal{F} is*

$$\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_{\mu} \mathbb{E}_{\sigma_1, \dots, \sigma_n} \left(\frac{1}{\sqrt{n}} \sup_{f \in \mathcal{F}} \left| \sum_{i=1}^n \sigma_i f(\mathbf{x}^{(i)}) \right| \right), \quad (\text{D18})$$

where $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_n)$ is a sequence of independent Rademacher variables taking values in $\{-1, 1\}$ and each with probability $1/2$, and $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathcal{X}$ are independent, μ -distributed random variables.

Intuitively, the covering number concerns the minimum number of spherical balls with radius ε that occupies the whole space; the Rademacher complexity measures the ability of functions from \mathcal{F} to fit random noise. The relation between Rademacher complexity and covering number is established by the following Dudley entropy integral bound.

Lemma 7 (Adapted from [77, 110]). *Let $\mathcal{F} = \{f : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}\}$ and $\mathcal{F}_+ = \{h = f(\mathbf{x}, \cdot) : f \in \mathcal{F}, \mathbf{x} \in \mathcal{X}\}$ and $\mathcal{F}_+ \subset B(L_\infty(\mathcal{X}))$. Given the set $\mathcal{S} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(n)}\} \in \mathcal{X}$, denote the Rademacher complexity of \mathcal{F}_+ as $\mathfrak{R}_n(\mathcal{F}_+)$, it satisfies*

$$\mathfrak{R}_n(\mathcal{F}_+) \leq \inf_{\alpha > 0} \left(4\alpha + \frac{12}{\sqrt{n}} \int_{\alpha}^1 \sqrt{\ln(\mathcal{N}((\mathcal{F}_+)|_{\mathcal{S}}, \epsilon, \|\cdot\|_2))} d\epsilon \right), \quad (\text{D19})$$

where $(\mathcal{F}_+)|_{\mathcal{S}} = \{[f(\mathbf{x}, \mathbf{x}^{(i)})]_{i=1:n} : f \in \mathcal{F}, \mathbf{x} \in \mathcal{X}\}$ denotes the set of vectors formed by the hypothesis with \mathcal{S} .

Ref. [77] hinges on the term $\mathbb{E}(T2)$ with the Rademacher complexity, as stated in the following lemma.

Lemma 8 (Adapted from Lemma 1, [77]). *Following notations in Theorem 2 and Lemma 7, define $\mathcal{G} = \{k(G_{\boldsymbol{\theta}}(\cdot), G_{\boldsymbol{\theta}}(\cdot)) | \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$ and $\mathcal{G}_+ = \{k(G_{\boldsymbol{\theta}}(\mathbf{z}), G_{\boldsymbol{\theta}}(\cdot)) | \boldsymbol{\theta} \in \boldsymbol{\Theta}, \mathbf{z} \in \mathcal{Z}\}$. Given the set $\mathcal{S} = \{\mathbf{z}^{(i)}\}_{i=1}^n$, we have*

$$\mathbb{E}(T2) \leq \frac{2}{\sqrt{n-1}} \mathfrak{R}_{n-1}(\mathcal{G}_+),$$

where $\mathfrak{R}_{n-1}(\mathcal{G}_+)$ refers to the Rademacher's complexity of \mathcal{G}_+ .

In conjunction with the above two lemmas, the term $\mathbb{E}(T2)$ is upper bounded by the covering number of \mathcal{G}_+ . As such, the proof of Lemma 5 utilizes the following three lemmas, which are used to formalize the relation of covering number of two metric spaces, quantify the covering number of variational quantum circuits, and evaluate the covering number of the space living in N -qudit quantum states, respectively.

Lemma 9 (Lemma 5, [111]). *Let (\mathcal{H}_1, d_1) and (\mathcal{H}_2, d_2) be metric spaces and $f : \mathcal{H}_1 \rightarrow \mathcal{H}_2$ be bi-Lipschitz such that*

$$d_2(f(\mathbf{x}), f(\mathbf{y})) \leq Kd_1(\mathbf{x}, \mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{H}_1, \quad (\text{D20})$$

and

$$d_2(f(\mathbf{x}), f(\mathbf{y})) \geq kd_1(\mathbf{x}, \mathbf{y}), \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{H}_1 \text{ with } d_1(\mathbf{x}, \mathbf{y}) \leq r. \quad (\text{D21})$$

Then their covering numbers obey

$$\mathcal{N}(\mathcal{H}_1, 2\epsilon/k, d_1) \leq \mathcal{N}(\mathcal{H}_2, \epsilon, d_2) \leq \mathcal{N}(\mathcal{H}_1, \epsilon/K, d_1), \quad (\text{D22})$$

where the left inequality requires $\epsilon \leq kr/2$.

Lemma 10 (Lemma 2, [66]). *Define the operator group as*

$$\mathcal{H}_{circ} := \left\{ \hat{U}(\boldsymbol{\theta}) \Pi_j \hat{U}(\boldsymbol{\theta})^\dagger \mid \boldsymbol{\theta} \in \Theta \right\}. \quad (\text{D23})$$

Suppose that the employed encoding Ansatz $\hat{U}(\boldsymbol{\theta})$ containing in total N_g gates, each gate $\hat{u}_i(\boldsymbol{\theta})$ acting on at most k qudits, and $N_{gt} \leq N_g$ gates in $\hat{U}(\boldsymbol{\theta})$ are trainable. The ϵ -covering number for the operator group \mathcal{H}_{circ} with respect to the operator-norm distance obeys

$$\mathcal{N}(\mathcal{H}_{circ}, \epsilon, \|\cdot\|) \leq \left(\frac{7N_{gt}\|\Pi_j\|}{\epsilon} \right)^{d^{2k}N_{gt}}, \quad (\text{D24})$$

where $\|\Pi_j\|$ denotes the operator norm of Π_j .

Lemma 11. *Define the input state group as $\mathcal{B} := \left\{ \rho_z := \hat{U}(z)^\dagger (|0\rangle\langle 0|) \hat{U}(z) \mid z \in \mathcal{Z} \right\}$. Suppose that the employed quantum circuit $\hat{U}(z)$ containing in total N_{ge} parameterized gates to load z and each gate $\hat{u}_i(z)$ acting on at most k qudits. The ϵ -covering number for \mathcal{B} with respect to the operator-norm distance obeys*

$$\mathcal{N}(\mathcal{B}, \epsilon, \|\cdot\|) \leq \left(\frac{7N_{ge}}{\epsilon} \right)^{d^{2k}N_{ge}}. \quad (\text{D25})$$

Proof of Lemma 11. The proof is identical to that presented in Lemma 2 of Ref. [66]. ■

We are now ready to prove Lemma 5.

Proof of Lemma 5. Recall the aim of Lemma 5 is to obtain the upper bound of $\mathbb{E}(T2)$. In conjunction with Lemmas 7 and 8, we obtain

$$\mathbb{E}(T2) \leq \mathbb{E} \frac{2}{\sqrt{n-1}} \inf_{\alpha > 0} \left(4\alpha + \frac{12}{\sqrt{n-1}} \int_\alpha^1 \sqrt{\ln(\mathcal{N}((\mathcal{G}_+)_{|\mathcal{S}}, \epsilon, \|\cdot\|_2))} d\epsilon \right), \quad (\text{D26})$$

where $\mathcal{G}_+ = \{k(G_{\boldsymbol{\theta}}(z), G_{\boldsymbol{\theta}}(\cdot)) \mid \boldsymbol{\theta} \in \Theta, z \in \mathcal{Z}\}$ and \mathcal{S} denotes the set $\{z^{(1)}, \dots, z^{(n-1)}\}$ sampled from the prior distribution $\mathbb{P}_{\mathcal{Z}}$, and $(\mathcal{G}_+)_{|\mathcal{S}} = \{[k(G_{\boldsymbol{\theta}}(z), G_{\boldsymbol{\theta}}(z^{(i)}))]_{i=1:n-1} : \boldsymbol{\theta} \in \Theta, z \in \mathcal{Z}\}$ denotes the set of vectors formed by the hypothesis with \mathcal{S} . In other words, the upper bound of $\mathbb{E}(T2)$ is quantified by the covering number $\mathcal{N}((\mathcal{G}_+)_{|\mathcal{S}}, \epsilon, \|\cdot\|_2)$.

We next follow the definition of covering number to quantify how $\mathcal{N}((\mathcal{G}_+)_{|\mathcal{S}}, \epsilon, \|\cdot\|_2)$ depends on the structure information of the employed Ansatz and the input quantum states. Denote \mathcal{Q}_{ϵ_1} as an ϵ_1 -covering of the set $\mathcal{Q}_1 = \{G_{\boldsymbol{\theta}}(z) \mid \boldsymbol{\theta} \in \Theta\}$ and \mathcal{Q}_{ϵ_3} as an ϵ_3 -covering of the set $\mathcal{Q} = \{G_{\boldsymbol{\theta}}(z) \mid z \in \mathcal{Z}\}$. Then, the covering number $\mathcal{N}((\mathcal{G}_+)_{|\mathcal{S}}, \epsilon, \|\cdot\|_2)$

can be upper bounded by $\mathcal{N}((\mathcal{Q}_1)_{|\mathcal{S}}, \epsilon_1, \|\cdot\|_2)$ and $\mathcal{N}((\mathcal{Q}_3)_{|\mathcal{S}}, \epsilon_3, \|\cdot\|_2)$. Mathematically, according to the explicit expression of $(\mathcal{G}_+)_{|\mathcal{S}}$, we have for any $(\boldsymbol{\theta}, \mathbf{z})$ and $(\boldsymbol{\theta}', \mathbf{z}')$

$$\begin{aligned}
& \left\| [k(G_{\boldsymbol{\theta}}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}))]_{i=1:n-1} - [k(G_{\boldsymbol{\theta}'}(\mathbf{z}'), G_{\boldsymbol{\theta}'}(\mathbf{z}^{(i)}))]_{i=1:n-1} \right\|_2 \\
&= \left\| [k(G_{\boldsymbol{\theta}}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}))]_{i=1:n-1} - [k(G_{\boldsymbol{\theta}'}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}))]_{i=1:n-1} + [k(G_{\boldsymbol{\theta}'}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}))]_{i=1:n-1} \right\|_2 \\
&\quad - [k(G_{\boldsymbol{\theta}'}(\mathbf{z}), G_{\boldsymbol{\theta}'}(\mathbf{z}^{(i)}))]_{i=1:n-1} + [k(G_{\boldsymbol{\theta}'}(\mathbf{z}), G_{\boldsymbol{\theta}'}(\mathbf{z}^{(i)}))]_{i=1:n-1} - [k(G_{\boldsymbol{\theta}'}(\mathbf{z}'), G_{\boldsymbol{\theta}'}(\mathbf{z}^{(i)}))]_{i=1:n-1} \right\|_2 \\
&\leq \left\| [k(G_{\boldsymbol{\theta}}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}))]_{i=1:n-1} - [k(G_{\boldsymbol{\theta}'}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}))]_{i=1:n-1} \right\|_2 + \left\| [k(G_{\boldsymbol{\theta}'}(\mathbf{z}), G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)}))]_{i=1:n-1} \right. \\
&\quad \left. - [k(G_{\boldsymbol{\theta}'}(\mathbf{z}), G_{\boldsymbol{\theta}'}(\mathbf{z}^{(i)}))]_{i=1:n-1} \right\|_2 + \left\| [k(G_{\boldsymbol{\theta}'}(\mathbf{z}), G_{\boldsymbol{\theta}'}(\mathbf{z}^{(i)}))]_{i=1:n-1} - [k(G_{\boldsymbol{\theta}'}(\mathbf{z}'), G_{\boldsymbol{\theta}'}(\mathbf{z}^{(i)}))]_{i=1:n-1} \right\|_2 \\
&\leq C_3 \left(\sqrt{n-1} \|G_{\boldsymbol{\theta}}(\mathbf{z}) - G_{\boldsymbol{\theta}'}(\mathbf{z})\| + \left\| [\|G_{\boldsymbol{\theta}'}(\mathbf{z}^{(i)}) - G_{\boldsymbol{\theta}}(\mathbf{z}^{(i)})\|]_{i=1:n-1} \right\|_2 + \sqrt{n-1} \|G_{\boldsymbol{\theta}'}(\mathbf{z}) - G_{\boldsymbol{\theta}'}(\mathbf{z}')\| \right), \quad (\text{D27})
\end{aligned}$$

where the first inequality uses the triangle inequality and the last inequality exploits C_3 -Lipschitz property of the kernel. Following the definition of covering number, the above relationship indicates that if for any $\boldsymbol{\theta}$ there exists $\boldsymbol{\theta}'$ such that $\|G_{\boldsymbol{\theta}}(\mathbf{z}) - G_{\boldsymbol{\theta}'}(\mathbf{z})\| \leq \epsilon_1$ holds for every \mathbf{z} , and for any \mathbf{z} there exists \mathbf{z}' such that $\|G_{\boldsymbol{\theta}}(\mathbf{z}) - G_{\boldsymbol{\theta}}(\mathbf{z}')\| \leq \epsilon_3$ holds for every $\boldsymbol{\theta}$, the composition of the covering sets \mathcal{Q}_{ϵ_1} and \mathcal{Q}_{ϵ_3} forms the covering set of $(\mathcal{G}_+)_{|\mathcal{S}}$. That is, the covering number of $(\mathcal{G}_+)_{|\mathcal{S}}$ is upper bounded by

$$\mathcal{N}((\mathcal{G}_+)_{|\mathcal{S}}, C_3 \sqrt{n-1} (2\epsilon_1 + \epsilon_3), \|\cdot\|_2) \leq \mathcal{N}(\mathcal{Q}_1, \epsilon_1, \|\cdot\|_2) \times \mathcal{N}(\mathcal{Q}_3, \epsilon_3, \|\cdot\|_2). \quad (\text{D28})$$

In other words, to quantify the ϵ -covering of $(\mathcal{G}_+)_{|\mathcal{S}}$, it is equivalent to deriving the upper bound of $\mathcal{N}(\mathcal{Q}_1, \epsilon/(3C_3\sqrt{n-1}), \|\cdot\|_2)$ and $\mathcal{N}(\mathcal{Q}_3, \epsilon/(3C_3\sqrt{n-1}), \|\cdot\|_2)$, respectively. We next separately derive these two quantities.

The upper bound of $\mathcal{N}(\mathcal{Q}_1, \epsilon/(3C_3\sqrt{n-1}), \|\cdot\|_2)$. Let \mathcal{Q}_4 be an $\frac{\epsilon}{3C_3d^N\sqrt{n-1}}$ -cover of \mathcal{H}_{circ} in Eq. (D23). Then, for any $\boldsymbol{\theta}$, there exists $\boldsymbol{\theta}'$ such that $\|\hat{U}(\boldsymbol{\theta})\Pi_j \hat{U}(\boldsymbol{\theta}) - \hat{U}(\boldsymbol{\theta}')\Pi_j \hat{U}(\boldsymbol{\theta}')\| \leq \frac{\epsilon}{3C_3d^N\sqrt{n-1}}$ for every j with $\hat{U}(\boldsymbol{\theta}')\Pi_j \hat{U}(\boldsymbol{\theta}') \in \mathcal{Q}_4$. This leads that for any \mathbf{z} , we have

$$\begin{aligned}
& \|G_{\boldsymbol{\theta}}(\mathbf{z}) - G_{\boldsymbol{\theta}'}(\mathbf{z})\|_2 \\
&= \left\| [\text{Tr}(\hat{U}(\boldsymbol{\theta})\Pi_j \hat{U}(\boldsymbol{\theta})^\dagger \rho_{\mathbf{z}}) - \text{Tr}(\hat{U}(\boldsymbol{\theta}')\Pi_j \hat{U}(\boldsymbol{\theta}')^\dagger \rho_{\mathbf{z}})]_{j=1:d^N} \right\|_2 \\
&\leq \left\| [\|\hat{U}(\boldsymbol{\theta})\Pi_j \hat{U}(\boldsymbol{\theta})^\dagger - \hat{U}(\boldsymbol{\theta}')\Pi_j \hat{U}(\boldsymbol{\theta}')^\dagger\|]_{j=1:d^N} \right\|_2 \\
&\leq d^N \frac{\epsilon}{3C_3d^N\sqrt{n-1}} \\
&= \frac{\epsilon}{3C_3\sqrt{n-1}}, \quad (\text{D29})
\end{aligned}$$

where the first inequality comes from the Cauchy-Schwartz inequality and the second inequality follows the definition of covering number.

The above observation means that the covering set of $\mathcal{N}(\mathcal{Q}_1, \epsilon/(3C_3\sqrt{n-1}), \|\cdot\|_2)$ is independent with \mathbf{z} and its covering number is upper bound by $\mathcal{N}(\mathcal{H}_{circ}, \frac{\epsilon}{3C_3d^N\sqrt{n-1}}, \|\cdot\|_2)$. Then, by leveraging the results in Lemma 10, we obtain

$$\mathcal{N}\left(\mathcal{Q}_1, \frac{\epsilon}{3C_3\sqrt{n-1}}, \|\cdot\|_2\right) \leq \mathcal{N}\left(\mathcal{H}_{circ}, \frac{\epsilon}{3C_3d^N\sqrt{n-1}}, \|\cdot\|_2\right) \leq \left(\frac{21C_3d^N\sqrt{n-1}N_{gt}\|\Pi_j\|}{\epsilon}\right)^{d^{2k}N_{gt}}. \quad (\text{D30})$$

The upper bound of $\mathcal{N}(\mathcal{Q}_3, \epsilon/(3C_3\sqrt{n-1}), \|\cdot\|_2)$. Let \mathcal{Q}_5 be an $\frac{\epsilon}{3C_3d^N\sqrt{n-1}}$ -cover of \mathcal{B} in Eq. (D25). Then, for any encoding state $\rho_{\mathbf{z}} \in \mathcal{B}$, there exists $\rho_{\mathbf{z}'} \in \mathcal{Q}_5$ with $\|\rho_{\mathbf{z}} - \rho_{\mathbf{z}'}\| \leq \frac{\epsilon}{3C_3d^N\sqrt{n-1}}$. By expanding the term $\|G_{\boldsymbol{\theta}'}(\mathbf{z}) - G_{\boldsymbol{\theta}'}(\mathbf{z}')\|$,

we obtain the following result, i.e., for any $\boldsymbol{\theta}'$,

$$\begin{aligned}
& \|G_{\boldsymbol{\theta}'}(\mathbf{z}) - G_{\boldsymbol{\theta}'}(\mathbf{z}')\| \\
&= \|[\text{Tr}(\Pi_j \hat{U}(\boldsymbol{\theta}') \rho_{\mathbf{z}} \hat{U}(\boldsymbol{\theta}')^\dagger) - \text{Tr}(\Pi_j \hat{U}(\boldsymbol{\theta}') \rho_{\mathbf{z}'} \hat{U}(\boldsymbol{\theta}')^\dagger)]_{j=1:d^N}\| \\
&= \|[\text{Tr}(\hat{U}(\boldsymbol{\theta}')^\dagger \Pi_j \hat{U}(\boldsymbol{\theta}') (\rho_{\mathbf{z}} - \rho_{\mathbf{z}'}))]_{j=1:d^N}\| \\
&\leq \left\| [\|\rho_{\mathbf{z}} - \rho_{\mathbf{z}'}\|]_{j=1:d^N} \right\| \\
&\leq d^N \frac{\epsilon}{3C_3 d^N \sqrt{n-1}} \\
&= \frac{\epsilon}{3C_3 \sqrt{n-1}}, \tag{D31}
\end{aligned}$$

where the first inequality uses $\text{Tr}(AB) \leq \text{Tr}(A)\|B\|$ when $0 \preceq A$ and $\text{Tr}(\hat{U}(\boldsymbol{\theta}')^\dagger \Pi_j \hat{U}(\boldsymbol{\theta}')) = \text{Tr}(\Pi_j) = 1$ for $\forall j \in [d^N]$, and the last inequality follows the definition of covering number. The achieved relation means that the covering set of $\mathcal{N}(\mathcal{Q}_3, \epsilon/(3C_3 \sqrt{n-1}), \|\cdot\|_2)$ does not depend on $\boldsymbol{\theta}$ and its covering number is upper bounded by $\mathcal{N}(\mathcal{B}, \frac{\epsilon}{3C_3 d^N \sqrt{n-1}}, \|\cdot\|_2)$.

Then, based on the results in Lemma 11, we have

$$\mathcal{N}\left(\mathcal{Q}_3, \frac{\epsilon}{3C_3 \sqrt{n-1}}, \|\cdot\|_2\right) \leq \mathcal{N}\left(\mathcal{B}, \frac{\epsilon}{3C_3 d^N \sqrt{n-1}}, \|\cdot\|_2\right) \leq \left(\frac{21C_3 d^N \sqrt{n-1} N_{ge}}{\epsilon}\right)^{d^{2k} N_{ge}}. \tag{D32}$$

Combining Eqs. (D30) and (D32), the covering number $\mathcal{N}((\mathcal{G}_+)|_{\mathcal{S}}, \epsilon, \|\cdot\|_2)$ in Eqn. (D28) is upper bounded by

$$\begin{aligned}
& \mathcal{N}((\mathcal{G}_+)|_{\mathcal{S}}, \epsilon, \|\cdot\|_2) \\
&\leq \mathcal{N}\left(\mathcal{H}_{circ}, \frac{\epsilon}{3C_3 2^{N-1} \sqrt{n-1}}, \|\cdot\|_2\right) \times \mathcal{N}\left(\mathcal{B}, \frac{\epsilon}{3C_3 2^{N-1} \sqrt{n-1}}, \|\cdot\|_2\right) \\
&\leq \left(\frac{21C_3 d^N \sqrt{n-1} N_{gt} \|\Pi_j\|}{\epsilon}\right)^{d^{2k} N_{gt}} \left(\frac{21C_3 d^N \sqrt{n-1} N_{ge}}{\epsilon}\right)^{d^{2k} N_{ge}} \\
&= (21C_3 d^N \sqrt{n-1} N_{gt})^{d^{2k} N_{gt}} (21C_3 d^N \sqrt{n-1} N_{ge})^{d^{2k} N_{ge}} \left(\frac{1}{\epsilon}\right)^{d^{2k} (N_{ge} + N_{gt})}. \tag{D33}
\end{aligned}$$

Denote $C_5 = (21C_3 d^N \sqrt{n-1} N_{gt})^{\frac{d^{2k} N_{gt}}{d^{2k} (N_{ge} + N_{gt})}} (21C_3 d^N \sqrt{n-1} N_{ge})^{\frac{d^{2k} N_{ge}}{d^{2k} (N_{ge} + N_{gt})}}$. Using Lemma 7, we obtain

$$\begin{aligned}
\mathbb{E}(T2) &\leq \frac{2}{\sqrt{n-1}} \inf_{\alpha>0} \left(4\alpha + \frac{12}{\sqrt{n-1}} \int_{\alpha}^1 \sqrt{\ln\left(\left(\frac{C_5}{\epsilon}\right)^{d^{2k} (N_{ge} + N_{gt})}\right)} d\epsilon \right) \\
&= \frac{2}{\sqrt{n-1}} \inf_{\alpha>0} \left(4\alpha + \frac{12\sqrt{d^{2k} (N_{ge} + N_{gt})}}{\sqrt{n-1}} \int_{\alpha}^1 \sqrt{\ln\left(\frac{C_5}{\epsilon}\right)} d\epsilon \right) \\
&\leq \frac{2}{\sqrt{n-1}} \inf_{\alpha>0} \left(4\alpha + \frac{12\sqrt{d^{2k} (N_{ge} + N_{gt})}}{\sqrt{n-1}} \left(\epsilon + \epsilon \ln\left(\frac{C_5}{\epsilon}\right)\right) \Big|_{\epsilon=\alpha}^1 \right). \tag{D34}
\end{aligned}$$

For simplicity, we set $\alpha = 1/\sqrt{n-1}$ in Eq. (D34) and then $\mathbb{E}(T2)$ is upper bounded by

$$\begin{aligned}
\mathbb{E}(T2) &\leq \frac{2}{\sqrt{n-1}} \left(\frac{4}{\sqrt{n-1}} + \frac{12\sqrt{d^{2k} (N_{ge} + N_{gt})}}{\sqrt{n-1}} \left(\epsilon + \epsilon \ln\left(\frac{C_5}{\epsilon}\right)\right) \Big|_{\epsilon=\alpha}^1 \right) \\
&\leq \frac{8}{n-1} + \frac{24\sqrt{d^{2k} (N_{ge} + N_{gt})}}{n-1} (1 + \ln C_5). \tag{D35}
\end{aligned}$$

Since the two exponent terms in C_5 are no larger than 1, we have $C_5 \leq (21C_3 d^N \sqrt{n-1} N_{gt})(21C_3 d^N \sqrt{n-1} N_{ge})$. This relation further simplifies the upper bound $\mathbb{E}(T2)$ as

$$\mathbb{E}(T2) \leq \frac{8}{n-1} + \frac{24\sqrt{d^{2k} (N_{ge} + N_{gt})}}{n-1} (1 + N \ln(441dC_3^2(n-1)N_{ge}N_{gt})). \tag{D36}$$

■

2. Proof of Lemma 6

Proof of Lemma 6. The proof of Lemma 6 is very similar to the one of Lemma 5 and thus we skip it here. \blacksquare

SM E: QGLMs in parameterized Hamiltonian learning

Here we first explain how QGLMs advance GLMs in the task of parameterized Hamiltonian learning (PHL) from the theoretical view. Then we conduct numerical simulations to apply QGANs to tackle parameterized Hamiltonian learning problems. Recall an N -qubit parameterized Hamiltonian is defined as $H(\mathbf{a})$, where \mathbf{a} is the interaction parameter (e.g., the strength of the transverse magnetic field) sampled from a prior distribution \mathbb{D} (e.g., uniform distribution). Denote $|\phi(\mathbf{a})\rangle$ as the ground state of $H(\mathbf{a})$. PHL aims to use m training samples $\{\mathbf{a}^{(i)}, |\phi(\mathbf{a}^{(i)})\rangle\}_{i=1}^m$ to approximate the distribution of the ground states for $H(\mathbf{a})$ with $\mathbf{a} \sim \mathbb{D}$, i.e., $|\phi(\mathbf{a})\rangle \sim \mathbb{Q}$. If a trained learning model can well approximate \mathbb{Q} , then it can prepare the ground state of $H(\mathbf{a}')$ for an unseen parameter $\mathbf{a}' \sim \mathbb{D}$.

1. Proof of Lemma 3

The proof of Lemma 3 is established on the results of quantum random circuits, which is widely believed to be classically computationally hard and in turn can be used to demonstrate quantum advantages on NISQ devices [86, 87]. The construction of a random quantum circuit is as follows. Denote $\mathbb{H}(N, s)$ as the distribution over the quantum circuit \mathcal{C} under 2D-lattice ($\sqrt{N} \times \sqrt{N}$) structure, where \mathcal{C} is composed of s two-qubit gates, each of them is drawn from the 2-qubit Haar random distribution, and s is required to be greater than the number of qubits N . For simplicity, here we choose $s = 2N^2$ to guarantee the hardness of simulating the distribution $\mathbb{H}(N, s)$. The operating rule for the i -th quantum gate satisfies:

- If $i \leq N$, the first qubit of the i -th gate is specified to be the i -th qubit and the second is selected randomly from its neighbors;
- If $i > N$, the first qubit is randomly selected from $\{1, 2, \dots, N\}$ and randomly select the second qubit from its neighbors.

Following the same routine, Ref. [88] proposed a Heavy Output Generation (HOG) problem detailed below to separate the power between classical and quantum computers on the distribution of the output quantum state after performing a circuit \mathcal{C} sampled from $\mathbb{H}(N, s)$ on the initial state $|0\rangle^{\otimes N}$.

Definition 5 (HOG, [88]). *Given a random quantum circuit $\mathcal{C} \sim \mathbb{H}(N, s)$ for $s \geq N^2$, generate k binary strings z_1, z_2, \dots, z_k in $\{0, 1\}^N$ such that at least $2/3$ fraction of z_i 's are greater than the median of all probabilities of the circuit outputs $\mathcal{C}|0\rangle^{\otimes N}$.*

Concisely, under the quantum threshold assumption, there do not exist classical samples that can spoof the k samples output by a random quantum circuit with success probability at least 0.99 [88]. In addition, they prove that quantum computers can solve the HOG problem with high success probability. For completeness, we introduce the quantum threshold assumption as follows.

Assumption 1 (quantum threshold assumption, [88]). *There is no polynomial-time classical algorithm that takes a random quantum circuit \mathcal{C} as input with $s \geq N^2$ and decides whether 0^n is greater than the median of all probabilities of \mathcal{C} with success probability $1/2 + \Omega(2^{-N})$.*

We are now ready to prove Lemma 3.

Proof of Lemma 3. The core idea of the proof is to show that there exists a ground state $|\phi(\mathbf{a})\rangle$ of a Hamiltonian $H(\mathbf{a})$, which can be efficiently prepared by quantum computers but is computationally hard for classical algorithms. To achieve this goal, we connect $|\phi(\mathbf{a})\rangle$ with the output state of random quantum circuits.

With the quantum threshold assumption in Assumption 1, Aronson and Chen [88] proved that there exists a quantum state $|\Phi\rangle$ generated from a random circuit \mathcal{C} sampling from $\mathbb{H}(N, 2N^2)$ such that HOG problem on this instance is classically hard, with success probability at least 0.99. Conversely, $|\Phi\rangle$ can be efficiently prepared by the parameterized quantum circuit $\hat{U}(\boldsymbol{\theta}^*)$ whose topology is identical to \mathcal{C} . Moreover, due to the fact that quantum adiabatic algorithms with 2-local Hamiltonians can implement universal quantum computational tasks [112], the quantum state $|\Phi\rangle \equiv |\phi(\mathbf{a}^*)\rangle = \hat{U}(\boldsymbol{\theta}^*)|0\rangle^{\otimes N}$ must correspond to the ground state of a certain Hamiltonian $H(\mathbf{a}^*)$.

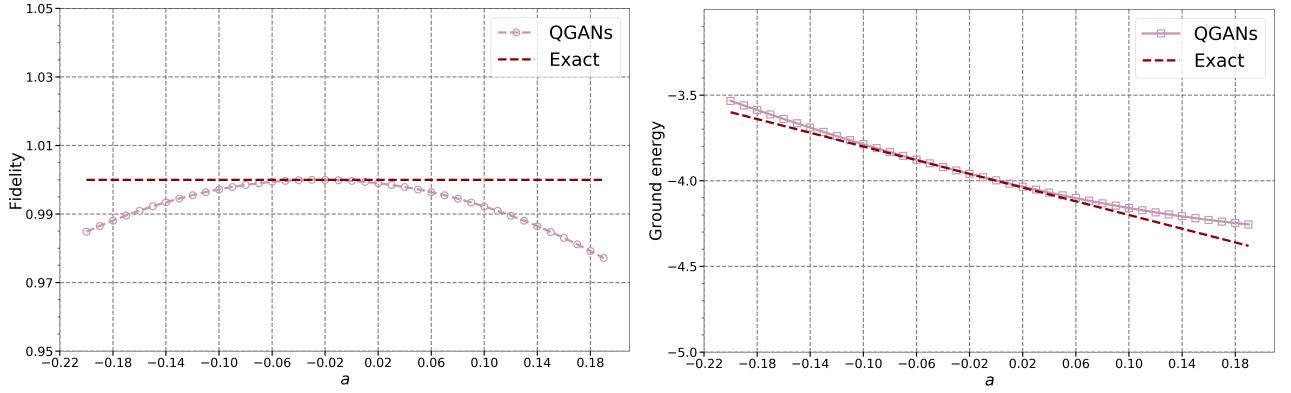


FIG. 5: Simulation results of QGANs in parameterized Hamiltonian learning. The left panel shows the fidelity of the approximated ground states and the exact ground states of Hamiltonian $H(\mathbf{a})$ with varied \mathbf{a} . The right panel exhibits the estimated and exact ground energies of a class of Hamiltonians $H(\mathbf{a})$.

We now leverage the above result to design a parameterized Hamiltonian learning task that separates the power of classical and quantum machines. In particular, we restrict the target distribution \mathbb{Q} , or equivalently \mathbb{D} , as the delta distribution, where the probability of sampling $|\Phi\rangle \equiv |\phi(\mathbf{a}^*)\rangle = \hat{U}(\boldsymbol{\theta}^*)|0\rangle^{\otimes N}$ equals to one and the probability of sampling other ground states of $H(\mathbf{a}')$ with $\mathbf{a}' \neq \mathbf{a}^*$ is zero. In this way, the Hamiltonian learning task is reduced to using QGLM or GLM to prepare the quantum state $|\Phi\rangle$. This task can be efficiently achieved by QGML but is computationally hard for GLMs. ■

2. Numerical simulation details

We apply QGANs introduced in the main text to study the parameterized Hamiltonian learning problem. In particular, the parameterized Hamiltonian is specified as the XXZ spin chain, i.e.,

$$H(\mathbf{a}) = \sum_{i=1}^N (X_i X_{i+1} + Y_i Y_{i+1} + \mathbf{a} Z_i Z_{i+1}) + \eta \sum_{i=1}^N Z_i. \quad (\text{E1})$$

In all numerical simulations, we set $N = 2$ and $\eta = 0.25$. The distribution \mathbb{D} for the parameter \mathbf{a} is uniform ranging from -0.2 to 0.2 . In the preprocessing stage, to collect m referenced samples $\{\mathbf{a}^{(j)}, |\phi(\mathbf{a}^{(j)})\rangle\}_{j=1}^m$, we first uniformly sample $\{\mathbf{a}^{(j)}\}_{j=1}^m$ points from \mathbb{D} and calculate the corresponding eigenstates of $\{H(\mathbf{a}^{(j)})\}_{j=1}^m$ via the exact diagonalization.

The setup of QGAN is as follows. The prior distribution $\mathbb{P}_{\mathcal{Z}}$ is set as \mathbb{D} . The encoding unitary is $U(\mathbf{a}) = \text{CNOT}(\text{RZ}(\mathbf{a})\text{RY}(\mathbf{a})) \otimes (\text{RZ}(\mathbf{a})\text{RY}(\mathbf{a}))$. The hardware-efficient Ansatz is used to implement $U(\boldsymbol{\theta}) = \prod_{l=1}^L U_l(\boldsymbol{\theta})$ with $U_l(\boldsymbol{\theta}) = \text{CNOT}(\text{RZ}(\boldsymbol{\theta}_{l,1})\text{RY}(\boldsymbol{\theta}_{l,2})) \otimes (\text{RZ}(\boldsymbol{\theta}_{l,3})\text{RY}(\boldsymbol{\theta}_{l,4}))$. The number of blocks is $L = 4$. The number of training and referenced examples is set as $n = m = 9$. The Adagrad optimizer is used to update $\boldsymbol{\theta}$. The total number of iterations is set as $T = 80$. We employ 5 different random seeds to collect the statistical results. To evaluate the performance of the trained QGAN, we apply it to generate in total 41 ground states of $H(\mathbf{a})$ with $\mathbf{a} \in [-0.2, 0.2]$ and compute the fidelity with the exact ground states.

The simulation results are shown in Fig. 5. Specifically, for all settings of \mathbf{a} , the fidelity between the approximated ground state output by QGANs and the exact ground state is above 0.97. Moreover, when $\mathbf{a} \in [-0.06, 0.06]$, the fidelity is near to 1. The right panel depicts the estimated ground energy using the output of QGANs, where the maximum estimation error is 0.125 when $\mathbf{a} = 0.2$. These observations verify the ability of QGANs in estimating the ground states of parameterized Hamiltonians.

SM F: More details of numerical simulations

1. Hyper-para and metrics

RBF kernel. The explicit expression of the radial basis function (RBF) kernel is $k(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2})$, where σ refers to the bandwidth. In all simulations, we set σ^{-2} as $\{0.25, 4\}$ for QCBMs and $\{-0.001, 1, 10\}$ for QGANs,

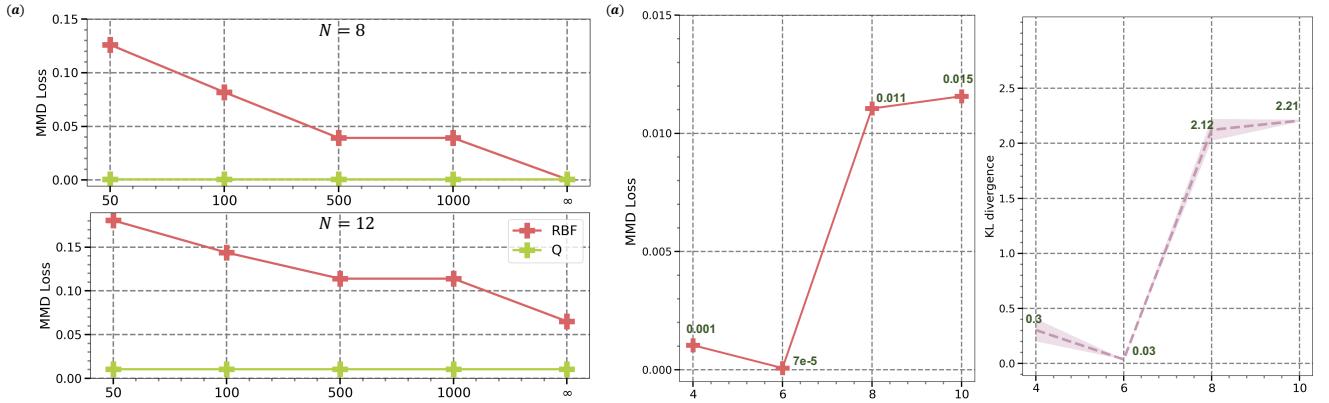


FIG. 6: **Generating discrete Gaussian with varied settings of QCBMs.** (a) The simulation results with the varied number samples n (corresponding to x -axis). The upper and lower panels demonstrate the achieved MMD loss for $N = 8, 12$, respectively. (b) The simulation results of QCBM with $N = 12$, the quantum kernel, and $n \rightarrow \infty$ for the varied circuit depth (corresponding to x -axis). The left and right panels separately show the achieved MMD loss and the KL divergence between the generated and the target distributions.

respectively.

KL divergence. We use the KL divergence to measure the similarity between the generated distribution \mathbb{P}_{θ} and true distribution \mathbb{Q} . In the discrete setting, its mathematical expression is $\text{KL}(\mathbb{P}_{\theta} \parallel \mathbb{Q}) = \sum_i \mathbb{P}_{\theta}(i) \log(\mathbb{Q}(i)/\mathbb{P}_{\theta}(i)) \in [0, \infty)$. In the continuous setting, $\text{KL}(\mathbb{P}_{\theta} \parallel \mathbb{Q}) = \int \mathbb{P}_{\theta}(d\mathbf{x}) \log(\mathbb{Q}(d\mathbf{x})/\mathbb{P}_{\theta}(d\mathbf{x})) d\mathbf{x} \in [0, \infty)$. When the two distributions are exactly matched with $\mathbb{P} = \mathbb{Q}$, we have $\text{KL}(\mathbb{P} \parallel \mathbb{Q}) = 0$.

State fidelity. Suppose that the generated state is $|\Psi(\theta)\rangle$ and the target state is $|\Psi^*\rangle$. The state fidelity [89] for pure states is defined as $F = |\langle\Psi(\theta)|\Psi^*\rangle|^2$.

Optimizer. For QCBMs, the classical optimizer is assigned as L-BFGS-B algorithm and the tolerance for termination is set as 10^{-12} . For QGANs, the classical optimizer is assigned as Adam with default parameters.

Hardware parameters. All simulation results in this study are completed by the classical device with Intel(R) Xeon(R) Gold 6267C CPU @ 2.60GHz and 128 GB memory.

Data and code availability The source code for conducting all numerical simulations will be available in Github repository <https://github.com/yuxuan-du/QGLM-Theory>.

2. Simulation results related to the task of discrete Gaussian approximation

Training loss of QCBMs. Fig. 6(a) plots the last iteration training loss of QCBMs. All hyper-parameter settings are identical to those introduced in the main text. The x -axis stands for the setting of n used to compute MMD_U in Eq. (A2). The simulation results indicate that the performance QCBM with RBM kernel is steadily enhanced with the increased n . When $n \rightarrow \infty$, its performance approaches to the QCBM with quantum kernel. These phenomena accord with Theorem 1.

Effect of circuit depth. We explore the performance of QCBMs with quantum kernels by varying the employed Ansatz. Specifically, we consider the case of $N = 12$ and set L_1 in Fig. 3(a) as 4, 6, 8, 10. The collected simulation results are shown in Fig. 6(b). In conduction with Fig. 3(c), QCBM with $L_1 = 6$ attains the best performance over all settings, where the achieved MMD loss is 7×10^{-5} and the KL divergence is 0.03. This observation implies that properly controlling the expressivity of Ansatz, which effects the term C_1 in Theorem 1, contributes to improve the learning performance of QCBM.

3. More simulation results related to the task of GHZ state approximation

The approximated GHZ states of QGANs with different random seeds discussed in Fig. 3 are depicted in Fig. 7. Specifically, the difference between the approximated state and the target GHZ state becomes apparent with the decreased number of examples n and the increased number of qubits N . These observations echo with the statement of Theorem 1.

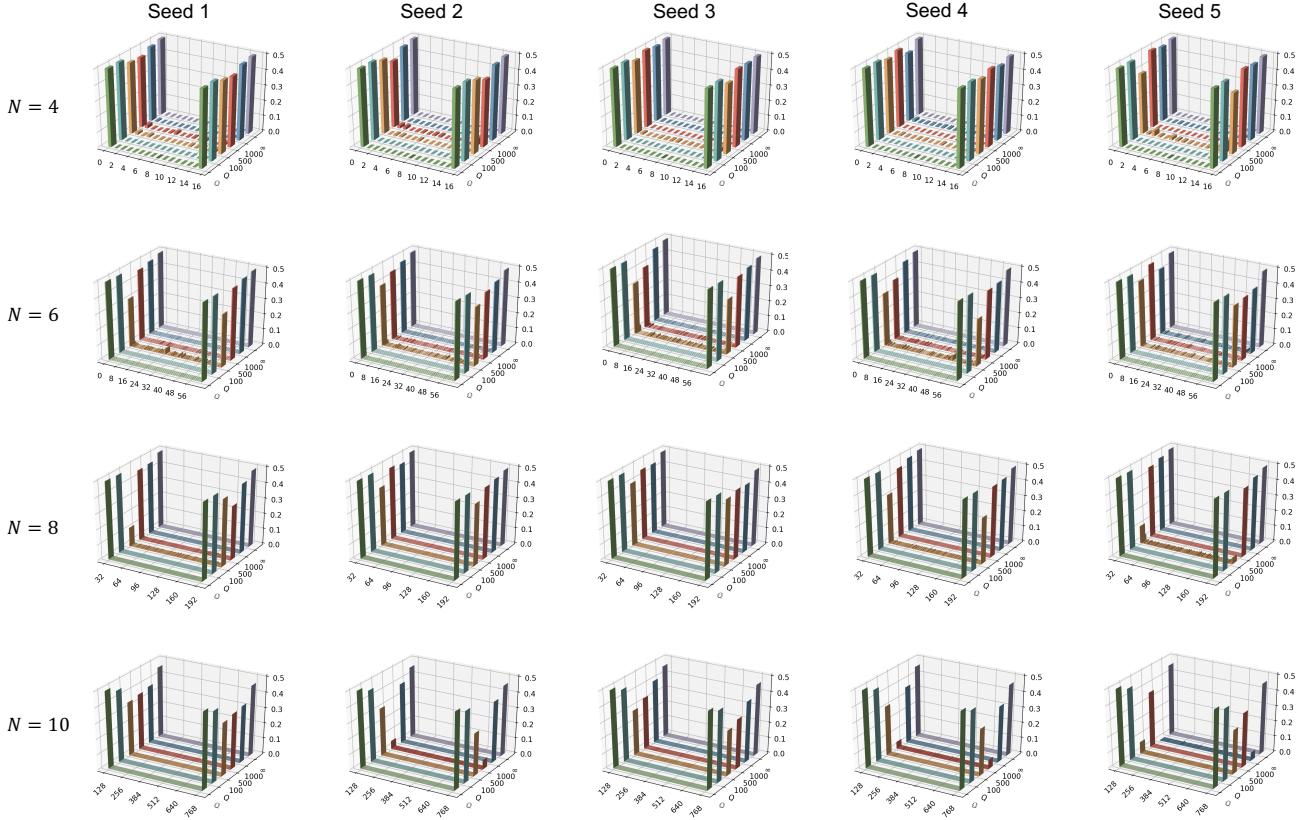


FIG. 7: The approximated GHZ state with the varied number of qubits. The label follows the same meaning explained in Fig. 3.

4. More simulation results related to the task of 3D Gaussian approximation

a. Implementation of the modified style-QGAN

The construction details of the modified style-QGAN, especially for $U(\mathbf{z})$ and $U(\boldsymbol{\theta})$, are illustrated in Fig. 8. Particularly, the circuit layout of $U(\mathbf{z})$ and the l -th layer of $\hat{U}(\boldsymbol{\theta})$ is the same. Mathematically, $U(\mathbf{z}) = U_E(\gamma_4)(\otimes_{i=1}^3 U(\gamma_i))$, where $U(\gamma_i) = R_Z(\mathbf{z}_3) R_Y(\mathbf{z}_2) R_Z(\mathbf{z}_2) R_Y(\mathbf{z}_1), \forall i \in [3]$ and $U_E(\gamma_4) = (\mathbb{I}_2 \otimes CR_Y(\mathbf{z}_2))(CR_Y(\mathbf{z}_1) \otimes \mathbb{I}_2)$ refers to the entanglement layer. Similarly, for the l -th layer of $\hat{U}(\boldsymbol{\theta})$, its mathematical expression is $\hat{U}_l(\boldsymbol{\theta}) = U_E(\gamma_4)(\otimes_{i=1}^3 U(\gamma_i))$, where $U(\gamma_i) = R_Z(\boldsymbol{\theta}_3) R_Y(\boldsymbol{\theta}_2) R_Z(\boldsymbol{\theta}_2) R_Y(\boldsymbol{\theta}_1), \forall i \in [3]$. When l is odd, the entanglement layer takes the form $U_E(\gamma_4) = (\mathbb{I}_2 \otimes CR_Y(\boldsymbol{\theta}_2))(CR_Y(\boldsymbol{\theta}_1) \otimes \mathbb{I}_2)$; otherwise, its implementation is shown in the lower right panel of Fig. 8.

The optimization of the modified style-QGAN follows an iterative manner. At each iteration, a classical optimizer leverages the batch gradient descent method to update the trainable parameters $\boldsymbol{\theta}$ minimizing MMD loss. After T iterations, the optimized parameters are output as the estimation of the optimal results. The Pseudo code of the

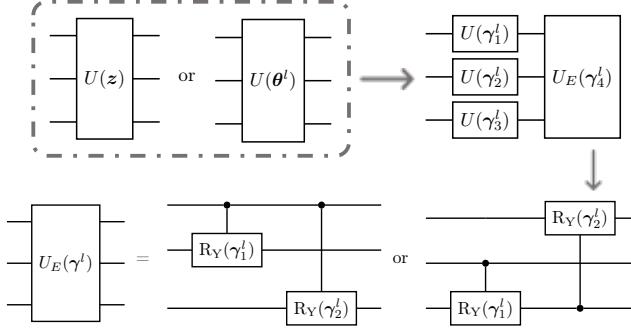


FIG. 8: The implementation detail of the modified style-QGANs. The implementation of QGANs when the number of qubits is $N = 3$. $U(z)$ refers to the encoding circuit to load the example z . The meaning of ' $L_1 - 1$ ' is identical to the one explained in Fig. 4. The gate $U(\theta_i^l)$ refers to the $R_Y R_Z$ gates applied on the i -th qubit in the l -th layer of $\hat{U}(\theta)$. The circuit architecture of $U(z)$ and the l -th layer of $\hat{U}(\theta)$ is identical, which is depicted in the upper right panel. The lower panel plots the construction of the entanglement layer $U_E(\gamma^l)$.

modified style-QGAN is summarized in Alg. 1.

Algorithm 1: The modified style-QGAN

Data: Training set $\{\mathbf{y}^i\}_{i=1}^m$, number of examples n , learning rate η , iterations T , MMD loss;
Result: Output the optimized parameters.

- 1 Randomly divide $\{\mathbf{y}^{(i)}\}_{i=1}^m$ into m_{mini} mini batches with batch size b ;
- 2 Initialize parameters θ ;
- 3 **while** $T > 0$ **do**
- 4 Regenerate noise inputs $\{z^{(i)}\}_{i=1}^n$ every r iterations;
- 5 **for** $j \leftarrow 1, m_{\text{mini}}$ **do**
- 6 Generate $\{\mathbf{x}^{(i)}\}_{i=1}^n$ with $\mathbf{x}^{(i)} = G_\theta(z^{(i)})$;
- 7 Compute the b 'th minibatch's gradient $\nabla \text{MMD}_U^2(\mathbb{P}_\theta^n \parallel \mathbb{Q}^b)$;
- 8 $\theta \leftarrow \theta - \eta \nabla \text{MMD}_U^2(\mathbb{P}_\theta^n \parallel \mathbb{Q}^b)$;
- 9 **end**
- 10 $T \leftarrow T - 1$;
- 11 **end**

b. More simulation results

We next examine how the number of examples m and the number of trainable gates N_g effect the generalization of QGANs. The experimental setup is identical to those introduced in the main text. To attain a varied number of N_g , the circuit depth of Ansatz in Fig. 4(a) is set as $L = 2, 4, 6, 8$. Other hyper-parameters are fixed with $T = 800$, $n = m = 5000$, batch size $b = 64$. We repeat each setting with 5 times to collect the simulation results.

Effect of the number of examples. Let us first focus on the setting $L = 2$ and $m = 5000$. In conjunction with the simulation results in the main text (i.e., $L = 2$ and $m = 2, 10, 200$), the simulation results in Fig. 9 indicate that an increased number of n and m contribute to a better generalization property. Specifically, at $t = 120$, the averaged empirical MMD loss (MMD_U) of QGANs is 0.0086, which is comparable with other settings discussed in the main text. The averaged expected MMD loss is 0.0045, which is similar to the setting with $m = 200$. In other words, when the number of examples m and n exceeds a certain threshold, the generalization error of QGANs is dominated by other factors instead of m and n .

Effect of the number of trainable gates. We next study how the number of trainable gates effects the generalization error of QGANs. Following the structure of the employed Ansatz shown in Fig. 4(a), varying the number of trainable gates amounts to varying the number of blocks L_1 . The results of QGANs with varied L_1 are illustrated in Fig. 9. For all setting of QGANs, their empirical MMD loss fast converges after 40 iterations. Nevertheless, their expected MMD loss is distinct, where a larger L_1 (or equivalently, a larger number of trainable gates N_g) implies a higher expected MMD loss and leads to a worse generalization. These observations accord with the result of Theorem 2 in the sense that an Ansatz with the overwhelming expressivity may incur a poor generalization ability of QGANs.

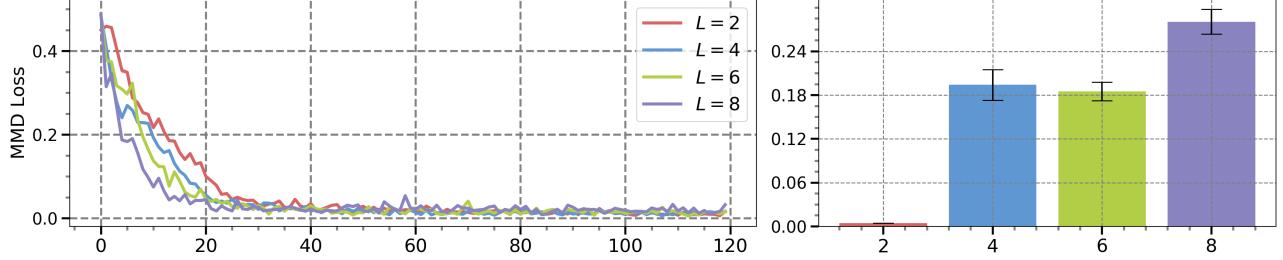


FIG. 9: **The simulation results of QGANs with the varied number of quantum gates.** The left panel shows the MMD loss of QGANs during 120 iterations. The label ‘ $L = a$ ’ refers to set the block number as $L_1 = a + 1$. The right panel evaluates the generalization property of trained QGANs by calculating the expected MMD loss. The x-axis refers to L .