Recent Advances for Quantum Neural Networks in Generative Learning

Jinkai Tian, Xiaoyu Sun, Yuxuan Du, Shanshan Zhao, Qing Liu, Kaining Zhang, Wei Yi, Wanrong Huang, Chaoyue Wang, Xingyao Wu, Min-Hsiu Hsieh, Senior Member, IEEE, Tongliang Liu, Senior Member, IEEE, Wenjing Yang, and Dacheng Tao, Fellow, IEEE

Abstract—Quantum computers are next-generation devices that hold promise to perform calculations beyond the reach of classical computers. A leading method towards achieving this goal is through quantum machine learning, especially quantum generative learning. Due to the intrinsic probabilistic nature of quantum mechanics, it is reasonable to postulate that quantum generative learning models (QGLMs) may surpass their classical counterparts. As such, QGLMs are receiving growing attention from the quantum physics and computer science communities, where various QGLMs that can be efficiently implemented on near-term quantum machines with potential computational advantages are proposed. In this paper, we review the current progress of QGLMs from the perspective of machine learning. Particularly, we interpret these QGLMs, covering quantum circuit Born machines, quantum generative adversarial networks, quantum Boltzmann machines, and quantum variational autoencoders, as the quantum extension of classical generative learning models. In this context, we explore their intrinsic relations and their fundamental differences. We further summarize the potential applications of QGLMs in both conventional machine learning tasks and quantum physics. Last, we discuss the challenges and further research directions for QGLMs.

Index Terms—Generative learning, quantum generative learning, quantum machine learning, quantum computing.

I. INTRODUCTION

Deep generative learning models (GLM) have revolutionized the classical world during the past decade [1], including not only limited to computer vision [2], natural language processing [3], and drug discovery [4]. The unprecedented success of GLMs stems from the power of deep neural networks, which can effectively capture the underlying distribution of training data and then generate new samples from the same distribution. Celebrated by this property, GLMs have been recently exploited to tackle fundamental problems in quantum physics science. Namely, GLMs are used to address the ‘curse of dimensionality’ encountered in quantum physics [5]. Compared with conventional methods, GLMs generally ensure better performance as well as improved generalization ability. All of these characteristics contribute to physicists’ understanding of the mechanisms of nature.

In parallel with the design of advanced GLMs and the exploration of their potential applications, another critical line of research in artificial intelligence is seeking the next generation of GLMs with enhanced abilities. The current challenge is overcoming the computational overhead of GLMs as the limits of Moore’s law are approached [6]. To this end, a leading solution is executing GLMs on quantum computers, which have exhibited strong theoretical and experimental performance [7], [8]. In this respect, a great amount of effort has been made to design quantum generative learning models (QGLMs) that can be efficiently carried out on noisy intermediate-scale quantum (NISQ) machines [9] with computational advantages. To date, extensive studies have demonstrated the feasibility of QGLMs for different learning tasks, e.g., quantum state approximation [10], image generation [11], and drug design [12]. Some cornerstones are summarized in Fig. 1.
The rapid development of QGLMs necessitates a systematic review of existing works, which could benefit researchers from both the computer science and quantum physics communities. To this end, in this survey, we analyze the current progress of QGLMs through the lens of deep generative learning (see Fig. 2). Definitively, according to the typical protocols of GLMs, we categorize QGLMs into four types, which are quantum circuit Born machine (QCBM), quantum generative adversarial network (QGAN), quantum Boltzmann machine (QBM), and quantum variational autoencoder (QVAE). For each type of QGLM, we first introduce the pioneering work and its inherent relation with its classical counterparts, followed by elucidating its variants and potential applications in both conventional machine learning and quantum physics. To the best of our knowledge, this is the first review in the context of quantum generative learning. We believe that this survey can help audiences of varying backgrounds to understand the development of QGLMs.

The structure of this survey is as follows. In Section II, we introduce the basic knowledge of deep neural networks, typical classical generative learning models, quantum computing, and variational quantum algorithms. In Section III, we systematically review prior literature related to QGLMs and explain the relationship with their classical counterparts. According to the categorization of QGLMs, this section is comprised of four subsections, which orient QCBM, QGAN, QBM, and QVAE, respectively. In Section IV, we discuss the challenges and future directions of quantum generative learning.

A. Related Works

A few review articles have similar or inter-sectional topics with our survey, while their emphases and scopes are different. Schuld et al. [13], Biamonte et al. [14], and Ciliberto et al. [15] reviewed early works of quantum machine learning under fault-tolerant scenarios. Moving into the NISQ era, Bharti et al. [16], Cerezo et al. [17], and Massoli et al. [18] reviewed the fundamental building blocks of VQAs and their applications. Benedetti et al. [19] reviewed quantum neural network (QNNs) with parameterized quantum circuits (PQCs) as machine learning models for quantum supervised and unsupervised learning tasks. Although these NISQ-oriented review articles mention quantum generative learning models more or less, none of them provides a comprehensive review of modern QGLMs by elaborating on their fundamental mechanism and unveiling the intrinsic relations with GLMs as we did in this survey.

Some reviews work on the quantum extension of discriminative learning models. Li and Deng [20], Ablayev et al. [21], Aboshahima et al. [22], Li et al. [23] reviewed fault-tolerate quantum classifiers like quantum support vector machines [24] and variational quantum classifiers utilizing QNNs in NISQ era. Contrary to these studies concentrating on discriminative learning, our survey focuses on quantum generative learning models. Another branch of surveys is summarizing how classical generative learning models can be used in understanding fundamental problems in quantum physics [5], [25]. There are also
designed to implement machine learning algorithms to process various types of data, including images, text, and speech. Relying on the advent of relevant learning theory and hardware resources, we have witnessed the remarkable development of neural networks during the past decade. Nowadays, neural networks are extensively applied in various fields of artificial intelligence, including computer vision and natural language processing, which enable the research community to explore solutions for a plethora of challenging problems in the real world. In this subsection, we give a brief introduction to neural networks, beginning with the perceptron—a single-layer neural network.

Conceptually, the perceptron is a linear binary classifier, which has only one neuron. From the mathematical perspective, the perceptron is a linear function and thus can only process linearly separable data patterns. This limited capacity for modeling complex patterns hinders its real-world application. By concatenating a series of perceptrons and non-linear activation functions, the neural network, dubbed multi-layer perceptron (MLP) or feedforward neural network (FFNN), can achieve an enhanced capacity. Since MLP was proposed, it has been widely applied to different learning tasks, including image classification, machine translation, and speech recognition. The key components of an MLP are an input layer, an output layer, and some hidden layers. The input layer does not have learnable parameters, while the output and hidden layers contain some perceptrons and non-linear operations. An intuition of MLP is shown in the right subfigure of Fig. 2 (classical NN). The mathematical formulation of MLP yields

$$ y = f_w(x) = \mathcal{L}^{(L)} \circ \ldots \circ \mathcal{L}^{(1)}(x), $$

where $x$ and $y$ are the input and output, respectively, and $w = \{W^{(1)}, b^{(1)}, \ldots, W^{(L)}, b^{(L)}\}$ is the set of trainable parameters. More specifically, $W^{(i)} \in \mathbb{R}^{d_i \times d_{i-1}}$ and $b^{(i)} \in \mathbb{R}^{d_i}$ are trainable parameters of the $i$-th layer and $d_i$ refers to the dimension of the $i$-th layer. Each layer $\mathcal{L}^{(i)} : \mathbb{R}^{d_{i-1}} \rightarrow \mathbb{R}^{d_i}$ consists of a linear transformation followed by a non-linear activation function $\phi(\cdot)$, i.e.,

$$ x^{(i)} = \mathcal{L}^{(i)}(x^{(i-1)}) = \phi(W^{(i)} x^{(i-1)} + b^{(i)}), $$

where $i \in \{1, 2, \ldots, L\}$ and $x^{(0)} = x$. For the hidden layer, $\phi(\cdot)$ can be ReLU, sigmoid, or tanh function, while for the output layer, $\phi$ can be softmax for classification or sigmoid for regression. The weights are initialized randomly and the MLP can be trained with data through backpropagation. The objective function which represents the discrepancy between the predicted results and the true label is designed to provide gradient information for parameter updates. The universal approximation theorem [29] shows that an MLP with enough parameters can approximate an arbitrary function on a bounded subset of $\mathbb{R}^n$ to any degree of accuracy.

Apart from MLP, there are many other neural networks designed for specific data/tasks. For example, recurrent neural networks (RNN) [30] are exploited to process sequential data, like text data. convolutional neural networks (CNN) with the weight sharing architecture [31] are proposed for image data analysis. Recently, the Transformer [32] consisting of MLPs has...
dominated the natural language processing (NLP) area [33], [34] and has further been extensively studied in the computer vision community [35], [36]. Celebrated by the remarkable development of graphics processing units (GPUs), we can effectively train a neural network with over one billion parameters on a large amount of data. For example, in NLP, GPT [37] contains around 0.1 billion parameters and is trained on 5 GB of data. Following the same manner, GPT-2 [38] is designed with over 1 billion parameters, and GPT-3 [39] has more parameters, at over 100 billion, and is learned on very large-scale text data, of around 45 TB.

B. Classical Generative Learning Models

There are primarily three learning paradigms in machine learning: discriminative learning, generative learning, and reinforcement learning. As illustrated in Fig. 2, generative learning models aim to find the underlying distribution of the training dataset and generate new samples of this distribution. For ease of understanding the quantum generative learning models, this subsection separately introduces four representative classical generative learning protocols, i.e., a naive stochastic neural network, generative adversarial network, Boltzmann machine, and variational autoencoder.

1) A Naive Stochastic Neural Network: Stochastic neural networks (SNNs) are a class of generative learning models towards the discrete distribution estimation [40]. For clarity, here we only discuss a naive SNN, which is composed of an input layer and an output layer.

The construction of SNN is shown in Fig. 3. As a directed graphical model, the generative process of SNN starts from the given input $x$, interacts with the tunable weights, and generates output $y$. The output $y$ is a $d$-dimensional binary vector, i.e., $y \in \{0, 1\}^d$. Mathematically, $y$ is sampled according to the distribution $p(y|x) = \prod_{i=1}^{d} \sigma(W_i x + b)^y_i (1 - \sigma(W_i x + b))^{1-y_i}$, where $W$ and $b$ refer to tunable weights and $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function. In the training procedure, SNN aims to maximize the log-likelihood $\log p(y|x)$. Suppose that there are $n$ observation pairs $\{(x^{(j)}, y^{(j)})\}_{j=1}^{n}$ sampled from the distribution $D$. The objective function of SNN takes the form of $\max_{W,b} \mathcal{L}_{\text{SNN}}(W,b) = \frac{1}{n} \sum_{i=1}^{n} \log p(y^{(j)}|x^{(j)})$. Since the objective function is differentiable, the trainable parameters $W$ and $b$ can be effectively updated via gradient descent methods. In the inference stage, given a new input $x'$ sampled from $D$, the trained SNN can output the prediction of the corresponding $y'$.

The idea behind SNN has been employed as a common approach in many renowned deep learning models. An example of this is utilizing deep neural networks to represent a desired quantum state [5], [41]. In this protocol, deep neural networks act as discrete generative models, with the goal of approximating the discrete probability distribution of the target quantum state. The output of the deep neural network is probabilistic, representing the outcome of a single measurement of the quantum state.

2) Generative Adversarial Network: Since the seminal work of Goodfellow et al. [42], generative adversarial networks (GANs) have been extensively studied due to their powerful ability to simulate complex target distributions. Different from most generative models adopting the unsupervised learning manner, GAN provides a novel adversarial learning scheme to frame the problem as a minimax training problem by introducing a discriminator, the adversary of the generator. Benefiting from their superiority, GANs have been exploited in a myriad of fields, such as image-to-image translation tasks [43], [44], [45], image generation [46], and domain adaptation [47].

A schematic workflow of GANs is exhibited in Fig. 4. There are two main components during the construction and training of a generative adversarial network: a generator $G$ and a discriminator $D$. The generator receives a random input vector $z \sim p_z(z)$ (i.e., $p_z$ refers to a prior distribution like the uniform distribution or the Gaussian distribution) and generates data $x = G(z)$ with learnable parameters $\theta_G$. The goal of the generator is to simulate the target data distribution $q$. The real data sampled from the target distribution and the generated data are fed into the discriminator $D$ with parameters $\theta_D$. The goal of the discriminator is to distinguish data from these two different sources by labelling the target data and the generated data with 1 and 0, respectively. In the learning procedure, $\theta_G$ is optimized to generate data in a pattern mimicking the target data distribution, and $\theta_D$ is optimized in the direction of assigning the right label for the target data $x \sim q(x)$ and the generated data $x \sim G(z)$, respectively.
Mathematically, the two-player minimax game takes the form \( \min_w \max_z C_G(w, z, D) \), where the explicit form of the objective function \( C_G(w, z, D) \) is

\[
E_{x \sim \mathcal{p}(x)}[\log D(x)] + E_{z \sim \mathcal{q}(z)}[\log(1 - D(G(z)))].
\]

When the training strategy reaches the Nash equilibrium, the discriminator cannot distinguish the generated data from the target data.

Optimizing such a minimax objective function could be a challenging task since the parameters of both generator and discriminator are dynamically changed. For this reason, the training of GANs is prone to model collapse and gradient vanish issues. Many variants of the vanilla GAN [42] are proposed to tackle specific problems with improved performance. For example, Deep Convolutional GAN (DCGAN) [48] implements the generator and the discriminator by using convolutional neural networks and de-convolutional neural networks respectively, which enhances the training stability. By involving extra information in the generator and the discriminator, e.g., class label, the vanilla GAN is transformed into the conditional GAN (CGAN) [49], which has the ability to control the outcome of the generator. By combining different objectives of GANs with the evolutionary strategy, Evolutionary GAN [50] not only alleviates the mode collapse issue but further improved the generation performance. For large-scale and high-quality image generation, Big Generative Adversarial Network (BigGAN) [51] improves previous works concerning the training strategy and network structures. For the recent improvements and trends of classical GANs, refer to Ref. [52], [53], [54].

3) Boltzmann Machine: A Boltzmann machine (BM) [55] is essentially a stochastic Ising model with an external field. Mathematically, a Boltzmann machine contains an array of visible units, \( v \), and an array of hidden units, \( h \). For simplicity, here, we only discuss the binary-valued BM, which means that all units \( z_i \in z = \{v, h\} \) are binary variables (i.e., \( z_i = \pm 1 \)). As a type of energy-based model (EBM) [56], the objective function of BM corresponds to its energy function,

\[
E_z = -\sum_{i<j} w_{ij} z_i z_j - \sum_i b_i z_i
\]

which can also be represented in a matrix product form as \( E_z = -z^T W z - b^T z \) with \( W \) \((w_{ij})\) and \( b \) \((b_i)\) being trainable parameters. According to the Boltzmann distribution, the probability of observing a visible state \( v \) can be calculated by summing over the hidden variables,

\[
p_{\mathcal{q}}(v) = Z^{-1} \sum_h \exp(-E(v, h)), \quad Z = \sum_z \exp(-E_z).
\]

Therefore, the parameters of the Boltzmann machine \( w = \{W, b\} \) could be optimized towards the direction in which the generated distribution \( p_{\mathcal{q}} \) is closer to the real distribution of dataset \( q \) by performing gradient ascent [55]. However, due to the expensive computational overhead, this learning strategy is usually impractical, especially when the number of hidden units is large. To overcome this issue, restricted Boltzmann machines (RBM), as shown in Fig. 5 (the middle) are introduced [57], where the connections between the hidden and visible units are allowed while the connections within the units, both visible and hidden, are restricted. The RBM can be optimized efficiently via contrastive divergence [58] with Gibbs sampling since the update of hidden and visible units can be implemented in parallel.

4) Variational Autoencoder: The variational autoencoder (VAE) model [59] is a highly influential example of a generative learning model. It takes the same approach as autoencoders (AEs) [60], where high-dimensional data is compressed into a lower-dimensional representation (known as the bottleneck) and then decoded to create patterns similar to the input. However, VAEs have the added advantage of being able to generate new data from the data distribution, which is not possible with AEs.

VAE leverages the variational Bayesian algorithm to learn the joint probability distribution \( p_{\mathcal{w}}(\mathbf{z}, x) \) for a directed graph model with the latent variable \( \mathbf{z} \) and observable variable \( x \) (see Fig. 6). The parametric distribution \( p_{\mathcal{w}}(\mathbf{z}, x) \) is represented by deep neural networks. The central aim of VAE is maximizing the marginal likelihood \( p_{\mathcal{w}}(x) = \int p_{\mathcal{w}}(x, \mathbf{z}) d\mathbf{z} \) which allows us to sample new data from the data distribution. However, this integral is generally intractable, which leads that the posterior \( p_{\mathcal{w}}(\mathbf{z}|x) = p_{\mathcal{w}}(x, \mathbf{z})/p_{\mathcal{w}}(x) \) is also intractable. In this regard, an inference model \( q_{\theta}(\mathbf{z}|x) \) with parameters \( \theta \) is introduced to estimate \( p_{\mathcal{w}} \), i.e., the probabilistic encoder constructed by deep
neural networks. The objective function of VAE is
\[
\log p_w(x) = C_{\theta, w}(x) + D_{KL}(q_\theta(\mathbf{z}|x)||p_\mathbf{z}(\mathbf{z}|x)),
\]
where \(C_{\theta, w}(x)\) is the evidence lower bound (ELBO). The inequality \(\log p_\theta(x) \geq C_{\theta, w}(x)\) always holds since the second term take a form of Kullback-Leibler (KL) divergence [62] that is non-negative.

The optimization of VAE is challenged since the gradient information \(\nabla_\theta C_{\theta, w}(x)\) has a high variance when using naive Monte Carlo (MC) sampling. One strategy to mitigate this issue is REINFORCE method [63]. A more efficient and widely adopted method is called reparameterization [59], [64], [65], expressing the latent variable \(\mathbf{z}\) as a deterministic and differentiable function \(\mathbf{z} = g_\theta(\mathbf{e}, x)\), where the randomness in \(\mathbf{e}\) is transferred to an auxiliary variable \(\mathbf{e} \sim p(\mathbf{e})\), so that \(\mathbb{E}_{\mathbf{z} \sim q_\theta}[f(\mathbf{z})] = \mathbb{E}_{\mathbf{e} \sim p(\mathbf{e})}[f(g_\theta(\mathbf{e}, x))].\) In this way, VAE can be trained by neural networks with respect to parameters \(\theta\) and \(w\) jointly.

In recent years, numerous variations of VAE have been developed and applied in various fields, including image generation [66], natural language processing [67], and molecular design [68]. For readers interested in further exploration, we refer to several notable examples, with a comprehensive review available in Refs. [69], [70], [71]. In particular, Ref. [72] introduced the Importance Weighted Autoencoder (IWAE) which increases the flexibility in approximating the true posterior distribution. The InfoVAE [73] and \(\beta\) VAE [74] incorporate regularizers in their Evidence Lower Bound (ELBO) objective functions, promoting disentanglement learning. The Vector Quantized VAE (VQ-VAE) (Ref. [75]) treats discrete latent variables as vectors, thereby addressing the issue of posterior collapse during training. To capture the discrete nature of training data, the Discrete VAE (DVAE) (Ref. [76]) was proposed, as illustrated in Fig. 6(b).

It is worth noting that many quantum VAEs are inspired by DVAE.

C. Quantum Mechanics and Quantum Computing

In this subsection, we introduce the background knowledge and notations of quantum mechanics, quantum computing, and quantum complexity theory. Refer to [77] for details.

1) Quantum Mechanics: Quantum mechanics provides a conceptual and mathematical framework for describing the physical world. In general, any isolated physical system can be completely represented by a normalized quantum state \(|\psi\rangle\), which is a complex unit vector in the system’s state space (Hilbert space) \(\mathcal{H}\). The evolution of such a quantum system can be described by a unitary operator \(U\), with \(|\psi'\rangle = U|\psi\rangle\) denoting the state after the evolution. In particular, the time evolution of a closed quantum system is characterized by the Schrödinger equation,
\[
i\hbar \frac{d|\psi(t)\rangle}{dt} = H|\psi(t)\rangle, \tag{7}
\]
where \(\hbar\) is Planck’s constant, \(|\psi(t)\rangle\) is the state at time \(t\) and \(H\) denotes the Hamiltonian of the quantum system. Given the state at some initial time \(t = 0\), if \(H\) is independent of time, the state of the system can be derived as \(|\psi(t)\rangle = e^{-iHt/\hbar}|\psi(0)\rangle\), with \(U(t) = e^{-iHt/\hbar}\) named as the unitary time evolution operator.

The density matrix provides an alternative formulation for describing quantum systems in mixed states. Suppose a quantum system is in state \(|\psi_i\rangle\) with probability \(p_i\), the density operator for this system is defined as \(\rho = \sum p_i|\psi_i\rangle\langle\psi_i|\), where \(|\psi_i\rangle\) is the conjugate transpose of \(|\psi_i\rangle\) and \(\rho\) is a normalized, positive semi-definite Hermitian operator. The evolution of a mixed state \(\rho\) is accordingly described as \(\rho' = U\rho U^\dagger\).

Information contained within a quantum system can be extracted through quantum measurements. The most typical one is the projective measurement, which is described by an orthonormal basis \(|i\rangle\). Performing this measurement on a quantum system \(\rho\) will collapse the system to state \(|i\rangle\) with a probability of \(|i|\rho|\rangle|\langle\rho|\rangle|\langle i|\). Physical quantities such as position, momentum, and energy can be described by observables \(O\), which act on the Hilbert space of the system. Since the observable is Hermitian, this implies, from the spectral decomposition theorem, that \(O = \sum \lambda_i P_i\), where \(P_i = |i\rangle\langle i|\) is the projector onto the eigenspace of \(O\) with a real eigenvalue of \(\lambda_i\). Note that \(|i\rangle\) is the corresponding eigenvector and \(\lambda_i\) also indicates the possible outcomes of the observable \(O\). The expectation value of \(O\) is quantified by \(\langle O \rangle_\rho = \text{tr}(\rho O)\). Generally, a wide class of observables are formed by the Pauli matrices
\[
\sigma_x \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.
\]
And the eigenstates of \(\sigma_z\) are \(|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}\), \(|1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}\).

2) Quantum Computing: Quantum computing is an interdisciplinary of quantum physics and computation theory that has recently gained much popularity due to its potential in defining the next generation of computers. The concept of quantum computing stems from the realization that simulating quantum evolution is infeasible for classical computers [78], since the computational resources required for classical simulation scale exponentially. Earlier work in the 1980s [79] brought forward a quantum mechanical model of the Turing machine, and promoted research on quantum computing. Although large fault-tolerant quantum computers are still absent, intermediate-scale quantum devices [9] which can handle practical applications will arise in the foreseeable future.

Analogous to the way that a classical computer stores information in binary classical bits, quantum computers adopt quantum bits, referred to as qubits, as their basic working units. A qubit is physically implemented by a two-level quantum system, described by a two-dimensional complex vector in the Hilbert space \(\mathcal{H}\). Following the Dirac notation (a.k.a. bra–ket notation), a general single-qubit system can be described as \(|\psi\rangle := \alpha|0\rangle + \beta|1\rangle\), where \(\alpha\) and \(\beta\) are complex numbers with the normalization condition that \(|\alpha|^2 + |\beta|^2 = 1\). Several qubits can be integrated into a multi-qubit quantum system via the tensor product. For example, an \(n\)-qubits system with each qubit in state \(|\alpha\rangle\) can be expressed as \(|\alpha\rangle^\otimes n = \sum_{\sigma^n} = \begin{pmatrix} 1, 0, \ldots, 0 \end{pmatrix}\).
Quantum computers are built from quantum circuits, which consist of wires and quantum gates to transmit and manipulate quantum information. As basic building blocks of quantum circuits, quantum gates manipulate the states of the qubits, affecting the evolution of the overall quantum system. Generally, a unitary evolution is carried out via a series of elementary quantum gates, which are formed by single-qubit and two-qubit gates. Typical elementary quantum gates include the Pauli gates, which can be represented by the Pauli matrices $\sigma_x$, $\sigma_y$, and $\sigma_z$, the single-qubit rotation gates ($RX$, $RY$, $RZ$), the Hadamard gates $H$, and the CNOT gates. Note that the single-qubit rotation gates are defined as $RX(\theta) = \exp(-i\sigma_x\theta/2)$, $RY(\theta) = \exp(-i\sigma_y\theta/2)$ and $RZ(\theta) = \exp(-i\sigma_z\theta/2)$. In addition, for the sake of brevity, a Pauli gate with a subscript describes a gate operated on an $n$-qubit system where only one particular qubit is operated.

Take $\sigma_x^2$, for example, $\sigma_x^2 \equiv I \otimes \cdots \otimes I \otimes \sigma_x \otimes I \otimes \cdots \otimes I$. In fact, the single-qubit rotation gates, together with the phase shift gate $P(\phi)$ and the CNOT gates, form a universal gate set. That is, any unitary operation $U$ of an arbitrary dimension can be implemented as a finite sequence of the gates from this set.

3) Quantum Complexity Theory: Quantum complexity theory is a subfield of computational complexity theory. Theoretical works [80], [81], [82], [83] have proven the advantage of quantum computing over a classical computer on some tasks from the perspective of computational complexity theory. Although there are many unsolved problems in computational complexity theory (e.g., a well-known open problem is whether P is equal to NP), the relation of some frequently used complexity classes is widely believed. Specifically, bounded-error quantum polynomial time (BQP) contains decision problems that are solvable by a quantum computer in polynomial time. While relationships between NP and BQP have not been mathematically proven, some problems are solvable in polynomial time by quantum algorithms and only verifiable in polynomial time classically, e.g., integer factorization. Shor’s algorithm [84] solves the integer factorization problem in polynomial time which is exponentially faster than the best-known classical algorithm. This quantum algorithm poses a real threat to the current crypto system. Grover’s algorithm [85], also known as the quantum search algorithm, searches an unstructured database with $N$ entries using only $\sqrt{N}$ queries. Amplitude amplification is a generalization of Grover’s algorithm and the application of amplitude amplification usually leads to quadratic speedups over classical algorithms. However, noisy environments and qubit limitations prevent the implementation of the aforementioned algorithms.

To fully utilize the currently available NISQ devices, a different algorithm design philosophy arose in recent years in the form of variational quantum algorithms.

D. Variational Quantum Algorithms

Variational quantum algorithms (VQAs) [17] are a class of quantum-classical hybrid algorithms whose mechanisms resemble deep neural networks. The applications of VQAs include combinatorial optimization [86], quantum chemistry [87], quantum error correction [88], and machine learning [14]. To fully leverage the power of both quantum and classical computers, VQAs implement parameterized quantum circuits (PQCs) on real quantum devices and use classical optimizers to update these trainable parameters. In this way, VQAs can readily adapt to the limitations of near-term quantum devices, which have restricted circuit depth and unavoidable gate noise. The superiority of VQAs stems from the fact that the state space represented by PQCs is unlikely to be efficiently simulated by classical computers. These properties ensure that VQAs are one of the most promising strategies to obtain quantum advantages in the NISQ era [16].

The schematic diagram of VQAs is outlined in the rightmost subfigure of Fig. 2. Analogous to NNs, VQAs are composed of three fundamental building blocks: the learning model, the optimizer, and the objective function. The key difference between classical machine learning algorithms and VQAs is in the way the learning model is implemented. Namely, the former adopts deep neural networks while the latter adopts the PQCs (or QNNs in the context of quantum machine learning), which is formed by the parameterized quantum gates and fixed quantum gates. Given a set of training data $\{(x_i, y_i)\}_{i=1}^N$, denote the prediction of QNN for the $i$-th example as $\hat{y}_i = h(x_i, \theta)$ for all $i \in [N]$. The optimization of VQAs amounts to finding the optimal parameters $\theta^*$ and minimizing the distance between the true labels and predictions according to a predefined objective function $C(\cdot, \cdot)$, i.e., $\theta^* = \arg \min_{\theta} \sum_{i=1}^N C(y_i, h(x_i, \theta))$. The possible choice of objective functions contains the mean square loss and the cross-entropy loss in discriminative learning and KL divergence in generative learning. The optimization can be completed via both gradient-free or gradient descent methods. The gradients can be acquired via the parameter shift rule [89].

Next, we elucidate the mathematical foundation of QNN $h(x_i, \theta)$, which is constructed by the initial state, the encoding unitary $U_x$, the Ansatz $U(\theta)$, and the measurement operator. Given the classical data, which can be the training example or the instance sampled from the prior distribution, QNN encodes it into the quantum state via the encoding unitary $U_x$. Note that the method used to achieve $U_x$ is flexible and can be via basis encoding, amplitude encoding, and gate encoding. For instance, when the gate encoding is adopted, the encoding unitary may take the form $U_x = \otimes_i RY(x_i)$. Another key component in QNN is Ansatz $U(\theta)$. As with deep neural networks, Ansatz obeys a multi-layer structure, i.e., $U(\theta) = \prod_{l=1}^L U_l(\theta_L)$. Particularly, the gate arrangement in each layer is diverse. As detailed below, a well-designed Ansatz can dramatically enhance the performance of VQAs for certain learning tasks. Let us denote the initial state as $|\Psi_\theta\rangle$ and the measurement operators as $\{O_j\}_{j=1}^d$ when the dimension of the prediction is $d$. There are two strategies to implement QNN. The first one only queries the data unitary once, i.e., $h(x_i, \theta)$ equals to $\{\text{tr}(O_1 U(\theta) U^{d} U^{d-1} U^{d-2} \cdots U_1 U^0)\}, \ldots, \text{tr}(O_d U(\theta) U^{d-1} U^{d-2} \cdots U_1 U^0)\}$. The second one is using $U_x$ and $U(\theta)$ multiple times, i.e., the $j$-th component of $h(x_i, \theta)$ for $j \in [d]$ becomes $\text{tr}(O_j U(\theta) U_x \cdots U_1 U^0 U^{d-1} U^{d-2} \cdots U_1 U^0)$. As mentioned above, the design of Ansätze is critical in VQAs. This is because some gates are costly to construct from
native gates. Furthermore, employing more gates hints at introducing more noise into the quantum system in light of the immaturity of current quantum devices. Therefore, choosing effective Ansätze depending on the given learning tasks can alleviate these issues. Based on whether the problem Hamiltonian is available, Ansätze of VQAs can be roughly classified into two categories, i.e., Hamiltonian-informed Ansätze and Hamiltonian-agnostic Ansätze.

Hamiltonian-Informed Ansätze: For most quantum physics problems where the Hamiltonian is provided, we could tailor the Ansatz by using the information of the Hamiltonian. Specifically, each unitary gate is the time evolution of a certain Hamiltonian \( H \) in the form of \( U(t) = e^{-iHt} \). Thus, we could control the size of the Ansatz by only keeping gates that are relevant to the problem Hamiltonian. For a Hamiltonian that is hard to simulate directly but can be expressed as the linear sum of easily-implemented Hamiltonians \( H = \sum_k \alpha_k h_k \), the Suzuki-Trotter expansion is commonly used to approximate the corresponding time evolution, i.e., \( e^{-iHt} = \lim_{m \to \infty} \prod_k e^{-\frac{\alpha_k h_k t}{m}} \).

Hamiltonian-Agnostic Ansätze: Hamiltonian-informed Ansätze may not ensure good VQA performance due to challenges including the constrained qubit connectivity, the restricted gate sets, the imperfect gate fidelities, and limited coherence times of current quantum devices. Moreover, for most conventional machine learning problems, less is known about how to construct the corresponding Hamiltonian. In this scenario, Hamiltonian-agnostic Ansätze are broadly employed in VQAs instead of Hamiltonian-informed Ansätze.

III. QUANTUM GENERATIVE LEARNING MODELS

In this section, we systematically review the current progress of quantum generative learning models (QGLMs), which can be interpreted as the quantum extension of classical generative learning models as illustrated in Fig. 2. In each subsection, we first elaborate on the seminar work of each class of QGLMs, followed by demonstrating their variants and potential applications. For comprehension, we summarize representative works of QGLMs in Table III and the available source codes of QGLMs in Table IV.

A. Quantum Circuit Born Machine

The quantum circuit Born machine (QCBM) was first proposed by Benedetti et al. [90] (which is also referred to as the data-driven quantum circuit learning framework). QCBM can be thought of as the quantum version of the SNN model described in Section II-B1, with the added feature that the randomness is an intrinsic property from quantum mechanics rather than being sampled after each layer. The key concept of a QCBM is leveraging a QNN to generate a tunable and discrete probability distribution. It inherits the Born rule naturally and can be efficiently implemented on NISQ devices due to the versatility of PQCs.

The correspondence of a QCBM and a SNN is illustrated in Table III. The first proposal of a QCBM is the KL divergence, and the gradient-free particle swarm optimizer (PSO) [114], respectively. More specifically, an initial Hamiltonian-agnostic Ansätze are broadly employed in VQAs instead of Hamiltonian-informed Ansätze.
the generated discrete distribution $p_\theta$ and the target discrete distribution $q$, i.e., $D_{KL}(q||p_\theta) = \sum_x q(x) \log(\frac{q(x)}{p_\theta(x)})$.

To better approximate $q$, the PSO is employed to continuously update $\theta$, and minimize $D_{KL}(q||p_\theta)$ until it converges.

Benedetti et al. [90] applied QCBM to complete three learning tasks on both numerical simulators and a trapped-ion quantum computer [115]. These learning tasks are BAS (2,2) image generation, 3-qubit Greenberger-Horne-Zeilinger (GHZ) state preparation [116], and coherent Boltzmann probability distributed thermal state preparation. The experimental results for the image generation confirmed that the performance of QCBM depends on the topology of the adopted entangling layers in PQC. That is, when the topology of PQC accommodates the hardware architectures, QCBM can attain high performance. On the task of GHZ state preparation, the trained QCBM recovered the most efficient and compact protocols on trapped-ion computers [116]. On the task of thermal state preparation, experimental results indicated that a deeper circuit may achieve better performance when the learning tasks become difficult.

1) Variants: Since Benedetti et al. [90], extensive studies exploring the enhancement of QCBM capabilities have been conducted. The related literature can be categorized into three classes, i.e., the design of different Ansätze, the selection of alternative objective functions, and the employment of powerful optimizers.

Ansätze Design: Several studies have investigated the possibility of tailoring the hardware-efficient Ansatz to improve the performance of QCBM. In particular, Liu and Wang [91] proposed a variant of hardware-efficient Ansatz whose entangling layers are realized by the Chow-Liu tree structure [117]. Meanwhile, a differentiable learning method, which is completed by the parameter shift rule, was employed to optimize QCBMs. The simulation results conducted on the BAS (3,3) dataset and the double Gaussian peak model validated the feasibility of their approach. Hamilton et al. [118] benchmarked the performance of QCBM implemented on IBMQ_Melbourne [119]. When the number of entangling layers is increased to 2, a QCBM with sparse connectivity in the entanglement layer attained a better performance than the denser one. In addition, they identified that the optimization may be stuck into local minima for certain configurations of the entanglement layer. Zhu et al. [92] experimentally studied the performance of QCBM executed on a customized programmable trapped-ion system [115], and the experimental results showed that the convergence behavior is closely related to both the optimizer and hardware connectivity. Leyton-Ortega et al. [120] performed a robust test of QCBM on the Rigetti Aspen superconductive quantum processor using a tailored hardware-efficient Ansatz, which is formed by RY gates and CZ gates. The abandonment of RZ gates allows for a reduced number of trainable parameters and thus facilitates accelerated optimization. Following the same routine, Hamilton et al. [121] conducted extensive simulations to investigate how different parameterizations (i.e., RY rotation gates versus arbitrary rotation gates) and varied settings for the sparsity of the entangling layers (i.e., periodic closure versus unitary 2-designs [122]) affect the performance of QCBM. By analyzing the loss landscape, the authors unveiled that although employing RY gates enables fast training, this strategy unfortunately induces rigidity into the circuits. On the other hand, employing arbitrary rotation gates benefits a deeper circuit and generally yields better performance, since it improves the connectivity of the loss landscape. Besides engineering the layout of the hardware-efficient Ansatz, Du et al. [93] enhanced the expressive power of QCBM by introducing ancillary qubits. Numerical simulations verified the effectiveness of their proposal. Recently, Gong et al. [123] attempted to integrate tensor-network-based Ansätze [124] with QCBM. The simulation results indicate that their proposal can attain a comparable generative capability with other QCBSMs while requiring fewer qubit resources and parameters.

Initial attempts have been conducted to combine QCBM with the Hamiltonian-informed Ansatz. Specifically, Coyle et al. [94] proposed the quantum circuit Ising Born machine (QCIBM). The authors argued that QCIBM could achieve quantum advantages when the employed objective function, i.e., total variation (TV) distance, is less than a small threshold. Experiments on the Rigetti platform exhibited the expressive power of QCIBM to learn random instances of target probability distributions. Meanwhile, the authors claimed that QCIBM can be used to perform a type of ‘weak’ quantum circuit compilation. Let us recall that the objective of quantum circuit compilation is to represent an arbitrary unitary $U$ as a sequence of native gates in a quantum device. With the limited quantum resources, ‘weak’ quantum circuit compilation aims at simulating the output of the target unitary instead of the unitary itself. They used QCIBM with QAOA [86] to learn the output distribution of an instantaneous quantum polynomial time (IQP) [125] circuit.

Objective Functions: Another line of research in the context of QCBM is designing different objective functions. Besides KL divergence applied in Refs. [120], [123], [126], [127], the maximum mean discrepancy (MMD) [128], [129] has also been widely utilized in Refs. [91], [93], [94], [118], [121], [123], [130], [131], [132] for QCBSMs. The mathematical form of MMD is $MMD[p,q] = \mathbb{E}_{x,x',y,y'} [k(x,x') - 2\mathbb{E}_{x,y,y'}k(x,y)] + \mathbb{E}_{y,y'}k(y,y')$, where $k(x,y)$ is the kernel function (e.g., Gaussian kernel [91] or quantum kernel [133], [134] applied in Refs. [94], [132]). Other alternative objective functions include the Stein discrepancy (SD) [135], the Sinkhorn divergence (SHD) [136], [137], and the Jensen-Shannon (JS) divergence [120] derived from KL divergence. Liu and Wang [91] pointed out that the MMD cost can be more efficiently applied in large-scale systems since KL divergence is usually inaccessible. Furthermore, Coyle et al. [94] claimed that SHD is superior to MMD in the measure of accuracy and convergence rate. Numerical simulation results certified the advantages of SHD on certain tasks.

Optimizers: Both gradient-free and gradient-based optimizers have been broadly used in QCBM. For gradient-free optimizers, PSO has been used in Refs. [90], [92], which enables faster training by using a small amount of memory. The covariance matrix adaptation evolution strategy (CMA-ES) is capable of optimizing non-linear and non-convex objective functions [138], and it has been exploited in Refs. [91], [126] to attain better performance than the simultaneous perturbation stochastic approximation (SPSA) method [139], but it has nevertheless,
underperformed in comparison to Adam when the quantum circuit noise is considered [91]. Bayesian optimization (BO), as a black-box derivative-free global optimization method, has been explored in Ref. [92]. It constructed a surrogate Bayesian statistical model for the same objective function and queried an acquisition function consisting of the posterior distribution to decide the next sampling direction. The BO method can be applied to problems involving a large number of parameters but the number of queries for evaluating objective functions is limited. Refs. [131], [140] adopted Genetic Algorithm (GA) [141] to optimize QCMBs, which can avoid some local minima in principle. Last, a newly proposed zeroth-order optimization method called ZOOpt toolbox [142] has been used in Ref. [120], which is designed for large-scale systems and is competent at suppressing the influence of noise. It attained the lowest cost value in the robust test in comparison to Adam and the Stochastic Variation of Hill-Climbing type algorithm (SVHC) [120].

The Gradient-Based optimizers can search local optima efficiently towards deep circuits and large parameter spaces. In this regard, the Adam optimizer has been broadly applied to optimize QCMB models [91], [93], [94], [118], [120], [123], [131]. The Adam optimizer generally requires a modest amount of computational resources and is well-adapted to most differentiable objective functions. The L-BFGS-B (bound limited memory BFGS) is another gradient-based optimizer adopted in Ref. [91]. It has a comparable efficiency to L-BFGS while its performance is superior when handling constrained problems [143]. Although it is not noise-tolerant, L-BFGS is particularly favored in large systems (e.g., \( \geq 1000 \) variables) due to its fast convergence rate. Liu and Wang [91] compared the performances of Adam and CMA-ES while treating the L-BFGS-B as a reference with infinite batch size. The simulation results indicated that CMA-ES is more vulnerable to noise and converges more slowly than Adam.

Although the selection of a suitable optimizer is typically learning task-dependent, it can be concluded that gradient-based methods are capable of dealing with a large-scale QNN and are less sensitive to system noise. Gradient-free optimizers are suitable for small-scale QNNs with fewer parameters and highly non-smooth objective functions.

2) Applications: QCMB and its variants have been applied in finance, such as learning empirical financial data distributions [126], [140], generating synthetic financial data [131], [140], and learning joint distributions [127], [131]. Some recent works applied QCMB to real-life financial datasets and have shown better (or at least equivalent) performance than classical learning models like the RBMs [126], [127], [131], [140], especially when the scale of the problems increases. Zhu et al. [127] reported an exponential advantage of QCMB with an SPSA optimizer over classical methods. Their approach utilized an Ansatz called ‘qupola’ and learned the joint probability distribution of two random variables of the stock market by modeling their copulas using the IonQ trapped-ion platform with up to 8 qubits. It is also possible to utilize QCMBs to efficiently prepare quantum states of interested financial data [144], [145] when certain conditions, e.g., the integrable property are satisfied.

B. Quantum Generative Adversarial Network

The concept of the quantum generative adversarial network (QGAN) was first conceived by Lloyd and Weedbrook [95]. The underlying mechanism of QGAN accords with classical GANs introduced in Section II-B2, where the generator and the discriminator are employed to carry out a two-player minimax game. QGAN can be used to estimate both discrete and continuous distributions. The key difference between QGAN and its classical counterparts is how the generator and discriminator are constructed. As depicted in Fig. 4, classical GANs generally employ deep neural networks to implement these two players, while QGANs may exploit QNNs. Celebrated by the strong power of quantum computers, such a construction rule may enable QGANs to gain certain computational advantages over classical GANs. In this regard, Ref. [95] discussed four types of QGANs, categorized by the learned tasks and the ways of implementing generators and discriminators. In what follows, we separately summarize these four types of QGANs and discuss their merits and limitations.

The first type of QGAN, abbreviated as QT-QGQD, orients to the Quantum Task and consists of both Quantum Generator and Quantum Discriminator. In this scenario, the generator \( G \) and the discriminator \( D \) refer to QNNs \( U(\theta_G) \) and \( U(\theta_D) \), respectively. The aim of \( G \) is to generate a quantum state \( \rho = U(\theta_G)p_0U(\theta_G)\dagger \), where \( p_0 \) is the initial input state of \( G \), to approximate the target state \( \sigma \). Conversely, the goal of \( D \) is to distinguish the target state \( \sigma \) from the generated state \( \rho \). The objective function \( C_{\text{QGAN}}(\theta_G, \theta_D) \) for QT-QGQD [146] can be defined as

\[
\text{tr}[\Pi U(\theta_D)\sigma U(\theta_D)\dagger]p_1 - \text{tr}[\Pi U(\theta_D)\rho U(\theta_D)\dagger]p_2, \tag{8}
\]

where \( \Pi \) is the predefined measurement operator, and \( p_1 \) and \( p_2 \) refer to prior of feeding \( \sigma \) and \( \rho \) to \( D \) with \( p_1 + p_2 = 1 \). The optimization of QGAN follows \( \min_{\theta_G} \max_{\theta_D} C_{\text{QGAN}}(\theta_G, \theta_D) \), which can be completed by using the methods introduced in Section II-D. Specifically, QGANs are trained in a two-step strategy: the discriminator is updated first to obtain a better judgement while the generator is fixed; then the generator is updated to be more deceptive while the discriminator is fixed. These two steps are conducted successively until \( C_{\text{QGAN}} \) converges. Following optimization, the discriminator \( D \) labels the input state as true if the posterior probability of \( \sigma \) is greater than that of \( \rho \) and vice versa.

The theoretical foundation of QT-QGQD is established in Ref. [95]. Due to Naimark’s dilation theorem, the role of the discriminator \( U(\theta_D)\Pi U(\theta_D)\dagger \) corresponds to a positive operator-valued measurement (POVM) \( \{\Pi_T, \Pi_F\} \) with \( \Pi_T + \Pi_F = 1 \). As such, for a fixed \( G \), the minimization of \( C_{\text{QGAN}} \) in (8) with respect to \( D \) can be achieved by the Helstrom measurement [147]. The probability that the input data is sampled from the target distribution (or produced by the generator) refers to \( \text{tr}(\Pi_T\sigma) \) (or \( \text{tr}(\Pi_T\rho) \)). Moreover, the Nash equilibrium \( C_{\text{QGAN}} = 0 \) can only be achieved when the generated state exactly recovers the target state with \( \rho = \sigma \). Moreover, as shown in Ref. [95], if we can acquire the gradient information of \( p(T|\sigma) \) or \( p(T|\rho) \) through the convex strategy
space at each iteration, the discriminator or generator can move directly towards the Nash equilibrium.

The second type of QGAN, abbreviated as QT-CGQD, targets the Quantum Task and consists of Classical Generator and Quantum Discriminator. Considering that some distributions generated by quantum circuits (e.g., the output of IQP circuit [81] and boson sampling [148]) cannot be efficiently simulated by classical machines, classical $G$ may not fit the quantum target distribution precisely. Consequently, following the theory of QT-QGQD, there exists a measurement for the discriminator to distinguish the target distribution from the generated distribution, and the Nash equilibrium can never be achieved.

The third type of QGAN, abbreviated as QT-CGCD, targets the Quantum Task and consists of both Classical Generator and Discriminator. Although the classical $G$ may not exactly recover the target quantum data, the classical $D$ may fail to detect the difference between the generated data from a classical generator and the complex quantum target data. To this end, the Nash equilibrium can still be achieved. Note that QT-CGCD is closely associated with the topic of using deep neural networks to study quantum physics problems, including quantum state tomography and the entanglement identification [5],[25].

The last type of QGAN discussed in Ref. [95], abbreviated as CT-QGQD, targets the Classical Task and consists of Quantum Generator and Classical Discriminator. CT-QGQD is designed for estimating continuous distributions, where the generated data is described by the expectation values of a set of observables. As claimed by the authors, CT-QGQD may possess computational advantages over classical GANs, warranted by the ability of quantum processors to represent $N$-dimensional vectors using $\log N$ qubits. In this way, advanced quantum algorithms can apply linear algebra on those vectors in time $O(poly(\log N))$ to optimize both the generator and discriminators. By contrast, classical GANs generally request $O(poly(N))$ runtime complexity in optimization.

1) Variants: The development of powerful QGANs has attracted great attention in the past three years. To elucidate, here we follow the method used in Ref. [95] to categorize QGANs. Specifically, according to whether quantum resources are involved in the learning task or the construction of learning models, previous literature related to QGANs can be cast into eight groups, as summarized in Table V. Considering that the main focus of this work is quantum learning models, we exclude the discussion about CT-CGQD and QT-CGCD.

**Table V**

| Generator | Discriminator | Quantum | Classical |
|-----------|---------------|---------|-----------|
| Quantum (C) | CT-QGQD | CT-QGQD |
| Quantum (Q) | QT-QGQD | QT-QGQD |
| Classical (C) | CT-CGQD | CT-CGQD |
| Classical (Q) | QT-CGQD | QT-CGQD |

QT-QGQD: The potential of QT-QGQD has been extensively explored from both algorithmic and experimental aspects. In particular, Dallaire-Demers and Killoran [96] proposed the quantum extension of conditional GANs [49], dubbed QuGAN, and claimed that the proposed model has a higher representation power than their classical counterparts. Potential applications of QuGAN include performing quantum chemistry calculations on quantum computers and compressing time evolution gate sequences. Benedetti et al. [149] utilized QGAN to estimate an unknown quantum pure state on near-term quantum computers. They suggested using resilient back-propagation as the optimizer and employing entanglement entropy as the stopping criterion. Relating to the same task, Hu et al. [10] experimentally demonstrated the first proof-of-principle QGAN on superconducting quantum devices to approximate a single-qubit state. Then, Huang et al. [150] implemented QGAN on a 5-qubit superconducting quantum device to learn multi-qubit quantum states and the classical XOR gate. Yin et al. [151] proposed a quantum adversarial solver to address quantum information processing tasks. By restricting the expressive power of the quantum generator, their proposal can efficiently detect the bipartite entanglement of quantum states. Experimental results on the linear optical network indicated the effectiveness of the proposal and exhibited potential quantum advantages. On par with designing different implementations of QGANs, some studies advised using different objective functions during training to attain robust performance. Chakrabarti et al. [112] studied the smoothness of the original objective function and proposed the Wasserstein semi-metric for quantum data, based on which they constructed the quantum WGAN model. Stein et al. [107] proposed a quantum-based objective function, called Quantum State Fidelity, to reduce the number of parameters.

QT-QGCD & QT-CGQD: A large body of work falls into the regime of QT-QGQD and QT-CGQD. In particular, Zoufal et al. [97] proposed a type of QGAN that aims to efficiently learn and load random distributions. Experiments conducted on the IBMQ cloud validated the effectiveness of their proposal. Zeng et al. [110] presented a QGAN model, as an adversarial training scheme for QCBM. Compared with the original QCBM, their proposal quadratically reduces the sample complexity of estimating the objective function. The proposed model can be applied to infer the unobserved data conditioned on partial observations such as image in-painting. Nakaji and Yamamoto [152] proposed the quantum semi-supervised GAN (QSGAN). As with the classical semi-supervised GANs that train the discriminator as a good classifier [153], the main focus of QSGAN is training a good classical discriminator rather than the quantum generator. The simulation results showed that QSGAN can achieve comparable performance to classical semi-supervised GANs by using a fewer number of parameters. Situ et al. [154] designed a type of QGAN and demonstrated its potential to avoid the vanishing gradients problem encountered in classical GANs when generating discrete data distributions. Romero and Aspuru-Guzik [155] proposed a variational quantum generator to model continuous classical probability distributions. Li et al. [105] demonstrated how to use a quantum generator to significantly reduce the number of required parameters compared
to its classical counterpart in the regime of drug discovery. Tsang et al. [156] proposed a model to bridge the gap between classical and quantum GANs by generating high dimensional data in the form of images. Zhou et al. [157] proposed a remapping method to simplify the data distribution.

C. Quantum Boltzmann Machine

The quantum Boltzmann machine (QBM) is first proposed by Amin et al. [98]. As the quantum generalization of the classical Boltzmann machine (BM) introduced in Section II-B3, it exploits quantum devices to prepare the Boltzmann distribution estimating the discrete target distributions. The main difference between QBM and classical BMs is that the units in BMs are replaced by qubits, and naturally, the energy term that appeared in (5) is replaced by a Hamiltonian. For a transverse-field Ising model, the Hamiltonian takes the form

\[ H_\theta = -\sum_{i,j} w_{ij} \sigma_i^x \sigma_j^x - \sum_i b_i \sigma_i^z - \sum_i \sum_j \Gamma_i \sigma_i^z, \]

where \( \theta = \{ w, b, \Gamma \} \) are the trainable parameters. The implementation of a semi-restrict QBM (semi-RQBM) is depicted in Fig. 5 (the rightmost), where the internal connections between hidden units are disallowed. For a general QBM, the probability of observing the state \( |v\rangle \) is \( p_\rho (|v\rangle) = \text{tr}(\Lambda_v \rho) = Z^{-1} \text{tr}(\Lambda_v e^{-H_\theta}) \), where \( Z = \text{tr}(e^{-H}) \) is the partition function, \( \rho = Z^{-1} e^{-H_\theta} \) is the density matrix. The diagonal matrix \( \Lambda_v = |v\rangle \langle v| \otimes 1_h \) is a standard projective measurement. To approximate the target distribution \( q \), QBM updates its trainable parameters \( \theta \) to minimize the negative log-likelihood, i.e.,

\[ C_{QBM} = -\sum_q q(v) \log p (v). \]

The gradients of \( C_{QBM} \) take the form

\[ \partial QBM = \sum_q q(v) \left( \frac{\text{tr}(\Lambda_v e^{-H})}{\text{tr}(e^{-H})} - \frac{\text{tr}(\partial_\theta e^{-H})}{\text{tr}(e^{-H})} \right), \tag{9} \]

where the second term (negative phase) refers to the Boltzmann average and is equivalent to \( \langle \partial_\theta H \rangle = \text{tr}(\rho \partial_\theta H) \). Note that since the exact sampling of \( \rho \) is NP-hard [168], the calculation of the negative phase becomes intractable for large-scale problems. To this end, instead of exact sampling, quantum analogues of the classical constructive divergence method [58] have been proposed [169], [170]. While the problem posted by the first term (positive phase) is the non-commutative between \( \Lambda_v \) and \( H \) due to the non-diagonal part in the quantum Hamiltonian. By leveraging the Golden-Thompson inequality [171], [172], Amin et al. [98] argued that the training cost of QBM can be replaced by its upper bound \( C_{QBM} \), i.e.,

\[ C_{QBM} \leq \sum_q q(v) \log \frac{\text{tr}(e^{-H_v})}{\text{tr}(e^{-H})}, \]

where \( H_v = \langle v | H | v \rangle \) is the clamped Hamiltonian. In this way, the trainable parameters \( \{ b, w \} \) can be updated efficiently. However, this method does not allow the training of the transverse filed \( \Gamma \), which has to be preset as a hyper-parameter in this model.

Amin et al. [98] further applied two types of QBMs, i.e., the fully visible QBM and the semi-restricted QBM (semi-RQBM), to simulate a Bernoulli mixture distribution. Experiments have shown that QBMs achieve better trainability than classical ones. A supervised generative task was also performed on a fully visible model, which takes an 8-qubit input and provides a 3-qubit output, where the input \( x \) and the output \( y \) are treated jointly as the visible pairs \( v = (x, y) \). The numerical result indicated that QBM trained with a joint distribution \( p(x, y) \) obtained a much lower cost over classical BMs. The authors
also discussed the potential of employing a quantum annealer as the processor to implement QBMs.

Many references proposed novel methods for sampling the two phases of (9). Regarding the positive phase, it could be numerically estimated using exact diagonalization or quantum Monte Carlo (QMC) techniques, or more efficiently, be approximated via the Golden-Thompson inequality as been applied in Refs. [98], [173], [174]. However, it prevents the training of the transverse field and hence that has to be treated as a hyper-parameter. Employing POVM [99] mitigates this issue, but the price is the demand for prior information on the data distribution, which is not generally available. There might be other approximation methods that yield better performance like quantum mean-field methods [174]. The negative phase sampling is a computationally NP-hard problem where the exact diagonalization needs exponential complexity. Multiple works have been proposed to accelerate the sampling of the Gibbs state. The general approach to approximate thermal state sampling is by applying QMC techniques, including simulated quantum annealing (SQA) [175] and the state-of-art continuous-time quantum cluster Monte Carlo (CT-QCMC) [174]. With no known theoretical bound complexity, heuristic algorithms for thermodynamics or other physics systems can hopefully find more efficient solutions. For instance, Ref. [173] adopted a combination system based on the eigenstate thermalization hypothesis, while it is not applicable for general interaction systems. Ref. [101] proposed a method using VarQITE and experimentally implemented QNN on quantum hardware, which provides a framework of quantum circuits with complex structures.

1) Variants: The variants of QBMs can be divided into three classes, depending on the training method. The first class is bound-based training in which the original objective function is replaced by its upper bound [61], [98], [99], [173]. The second class is the state-based training by minimizing the relative entropy [99], [176], [177], [178]. The last class is the circuit-based training [101], which utilizes variational quantum imaginary time evolution (VarQITE) methods [179], [180], [181].

Bound-Based Training: The key concept of bound-based training is using Golden-Thompson inequality to impose an upper bound on the objective function. This allows an efficient calculation of the gradient information. Kieferová and Wiebe [99] proposed a POVM-based training scheme which is a generalization of Ref. [98], where the measurements $\Lambda_{v}$ in (9) are generalized to POVM. As such, the training in the transverse field can proceed. To explore potential quantum advantages, Ref. [99] applied QBMs to solve Hamiltonians with the fermionic form since the sign problem [182], [183] prevents the efficient implementation of QMC techniques. Simulation results have indicated that QBMs can provide a significantly enhanced learning performance over classical BMs.

Puškarov and Cubero [184] argued that the generalized Gibbs ensembles [185], [186] could be leveraged as a basis for QBMs. In doing so, the gradient information can be effectively obtained due to the commutativity of the conserved charge in the integrable Hamiltonian by learning the optimal effective temperatures. Simulation results showed that their proposal could achieve a reasonably low error rate with much fewer parameters than RBM. Ansuecht and Cao [173] proposed an efficient method to approximately sample the thermal states of the QBM negative phase, which is supported by the eigenstate thermalization hypothesis. The numerical simulation results illustrated that their proposal has a comparable performance with the exact QBM and is superior to the classical RBM. Meanwhile, it is also robust against noise. Xiao et al. [174] proposed a general training scheme for various QBMs, which includes general transverse-field Ising models and the stochastic $\kappa$-local Hamiltonians. The authors also verified the efficiency of their proposal through simulations on small-scale multi-mode Bernoulli distribution [173] and large-scale MNIST dataset.

State-Based Training: Kieferová and Wiebe [99] first illustrated a state-based training scheme for QBM without hidden units. The model is trained to minimize the relative entropy between a known quantum state $\rho$ prepared by an oracle and the generated observable state with density matrix $\sigma$, i.e., $S(\rho|\sigma) = \text{tr}(\rho \log \rho) - \text{tr}(\rho \log \sigma)$. This protocol provides an alternative technique to quantum state tomography (QST) and can further generate copies of the learned quantum distributions. Based on the fully visible QBM model, the tomographic reconstructions of Haar-random pure states and mixed two-qubit states show a high level of accuracy using the state-based training. Wiebe and Wossnig [177] proposed two training methods for QBM to minimize the relative entropy with the presence of hidden units. The first approach is to propose a variational upper bound of the objective function, which is based on a constrain-form Hamiltonian with commutative hidden units. The second approach is adaptive to general forms of Hamiltonians and is more up-to-date.

Quantum Circuit Based Training: Zoufal et al. [101] proposed a variational QBM. Using the VarQITE method, the approximated Gibbs states are prepared by quantum circuits, and the analytic gradients are computed using the automatic differentiation method. The effectiveness of the VarQITE method is confirmed on quantum hardware by examining the closeness of the prepared Gibbs states and the target ones, and the experimental results on generative and discriminative tasks also demonstrated the feasibility and the noise-resilient property of variational QBMs.

2) Applications: Crawford et al. [100] employed RDBM, deep Boltzmann machine (DBM), and QBM models of clamped Hamiltonians to accomplish reinforcement learning (RL) tasks of Maze Traversal problems [187], which are Markov decision processes and are widely adopted to benchmark RL algorithms. As it is argued that the simulated quantum annealing (SQA) [188] could simulate the quantum annealing process due to its behavioral similarity with quantum tunnelling and entanglement [189], it is utilized to generate samples approximating Boltzmann distributions. In the numerical simulations, the SQA model is applied to both the classical Hamiltonian of DBM and the quantum Hamiltonian of QBM. The results showed that the DBM outperforms RDBM as the number of training samples increases, whereas the QBM has the best performance. Wiebe et al. [178] exploited a Fock-space representation to encode linguistic structures in vector spaces. In this way, the
authors illustrated how QBM could be applied to solve language processing tasks.

D. Quantum Variational Autoencoder

The first quantum variational autoencoders (QVAEs) was proposed by Khoshaman et al. [61]. QVAEs utilize RBM as the prior distribution $p_w(z)$ instead of RBM presented in DVAE. In doing so, the explicit form of the prior is $p_w(z) = \frac{\text{tr}(Ae^{-H_w})}{Z_w}$ and $Z_w = \text{tr}(e^{-H_w})$ with the Hamiltonian described in Section III-C. Such substitution introduces similar issues to those encountered in the use of QBMs. Specifically, the quantum transverse field prohibits efficient sampling of the positive phase during gradient calculations, while exact sampling of the negative phase is computationally intractable using conventional methods, as described in (9). To overcome this limitation, QVAE employs the Golden-Thompson inequality in conjunction with a clamped Hamiltonian, resulting in a lower bound of ELBO referred to as Q-ELBO.

The authors compared the performance of DVAE, QVAE with ELBO, and QVAE with Q-ELBO on generating hand-written images in MNIST dataset. The negative phase of DVAE is computed by persistent contrastive divergence (PCD) [190] while it in QVAE is calculated by continuous-time quantum Monte Carlo (CT-QMC) [191] with population annealing [192]. The positive phase of DVAE and QVAE with Q-ELBO is computed classically, while the positive phase in QVAE with ELBO is computed with quantum distributions. The simulation results show that QVAE can be trained effectively with either Q-ELBO or ELBO, but their performances deteriorate with increasing transverse fields. The experiment results indicate that quantum annealers provide a promising and realistic solution for implementing QVAE, that it acts as a physical sampler to return samples of the Boltzmann distribution with tunable parameters of the QBM Hamiltonian.

1) Variants: Despite the growing body of research in the field of QGLMs, the study of QVAEs remains relatively under-explored. Variants of existing QVAEs can be roughly classified into two main categories based on their implementation methods: annealer-based QVAEs [193] and gate-based QVAEs [194].

Annealer-Based QVAEs: For QVAE with QBM, an early application of the annealer-based QVAE was realized by Gao et al. [193], applied in the earth science field to search for high-dimensional similarity from satellite datasets. The QVAE assisted by D-Waves achieves competitive performance with simulated QVAE and DVAE with RBM, and it enables a continuous tuning of the transverse field. However, increasing the transverse field could narrow the latent distribution and deteriorate the expressivity of RBM/QBM priors [195], [196]. The authors argued that all three models attain tremendous speedups and significantly reduce memory costs compared with some state-of-art searching methods. The annealer-based QVAE is proposed to gain computational advantages when the complexity of simulated data increases, yet advanced annealing techniques with reduced noise and control error are still demanded to pursue quantum advantages in real devices.

Gate-Based QVAEs: Li and Ghosh [194] proposed a class of quantum gate-based QVAE for drug discovery. Specifically, the baseline QVAE (BQ-VAE) and the scalable QVAE (SQ-VAE). In their proposal, the encoder is composed of amplitude embedding and the expectation output; the decoder is composed of angle embedding and measurements, and the latent space is realized by a PQC with the repeated layer structure. Simulation on low-dimensional QM9 drug molecules is performed with BQ-VAE, and a modest learning speedup compared with classical VAE is reported. Meanwhile, high-dimensional PDBbind molecules are simulated by SQ-VAE with patched circuits that greatly increase the latent space dimensions, and the authors argue that SQ-VAEs can attain better drug properties than classical VAEs.

IV. CONCLUSIONS AND DISCUSSIONS

In this survey, we provide a comprehensive review of numerous QGLMs and elaborate on their potential to benefit both conventional machine learning tasks and quantum physics from the perspective of computational efficiency and model expressivity. Despite the astonishing progress, quantum generative learning is still in its infancy, and many critical issues still remain unexplored. In the remainder of this section, we discuss these unsolved issues and present insights into the future prospects of QGLMs.

Different types of QGLMs have their own unique characteristics and benefits. Table VI summarizes the types and scales of data that are competent for each model and summarized their compatible quantum device. In particular, QGANs can work with both continuous data and discrete data. On the contrary, QCBMs, QMs, and QVAEs are intrinsically designed for discrete distributions. QCBMs and QGANs are well-adopted for gate-based quantum computers, whereas QBMs and QVAEs are better to be assisted with quantum annealers. According to the current development stage of quantum hardware, QCBMs and QGANs can only handle small-scale datasets, while QBMs and QVAEs assisted with quantum annealers are adaptive for large-scale datasets. As a result, comparing the performance of QGLMs running on different types of quantum devices is challenging, and how to exemplify quantum advantage in these generative models is still an open question. Besides, despite possible advantages, all of these QGLMs have the training

| TABLE VI |
| IDENTIFYING WHICH TASKS CAN BE ACCOMPLISHED BY FOUR QGLMS (BASED ON MAJORITY REFERENCES). THE ROW LABELED BY ‘DATA’ REFERS TO THE TYPE OF DATA THAT A SPECIFIED MODEL CAN HANDLE (CONTINUOUS, DISCRETE, OR BOTH). THE ROW LABELED BY ‘SCALE’ REFERS TO THE DATASET SCALE THAT A QGLM IS COMPETENT WITH (SMALL, LARGE, OR BOTH). THE ROW LABELED BY ‘DEVICE’ REFERS TO THE QUANTUM DEVICE THAT IS MOST COMPATIBLE WITH EACH MODEL (GATE-BASED QUANTUM DEVICES OR QUANTUM ANNEALERS). FOR G’ IN QBM, IT REFERS TO THE VARIATIONAL QBM VARIANT (SEE SECTION III-C1 FOR REF. [101]) |
| QCBM | QGAN | QBM | QVAE |
| data | D | C&D | D | D |
| scale | S | SkL | SkL | L |
| device | G | G | G* & A | A |
efficiency issue.Ultimately, the choice of QGLM to use in practice depends on the specific task scenarios and each model has its own research value.

A. Challenges

Read-In & Read-Out Bottleneck: A common caveat of QGLMs presenting in estimating high-dimensional continuous distribution tasks is the read-in and read-out bottleneck [197], the process of encoding data into a quantum state and measuring the output of a quantum computation, respectively. Particularly, both QGAN (i.e., QT-CGQD and CT-CGQD) and gate-based QVAE encounter the read-in bottleneck when the training data is classical and sampled from a continuous distribution, and the amplitude encoding is specified. That is, how to efficiently load the example $x \in \mathbb{R}^{2^N}$ into an $N$-qubit state $|x\rangle = \frac{1}{\sqrt{2^N}} \sum_{i=1}^{2^N} |x_i\rangle |i\rangle$ remains unknown. The read-out bottleneck concerns how to extract quantum information into classical registers. In QGAN, when CT-QGCD is applied to estimate the continuous distribution, it suffers from the read-out bottleneck. For example, in the task of image generation, the features of the generated image are stored in the probability amplitudes of quantum states. To extract these features into the classical world, QST is required, where the computational complexity scales with the feature dimension. To address issues with the read-in and read-out speed of data, one can use quantum-classical hybrid models. These models can take different forms, such as alternating between quantum circuits and classical layers [198], connecting quantum circuits with classical neural networks (e.g., QGANs [97]), or using PCA to pre-process data before encoding it. Nevertheless, there are several challenges associated with these hybrid models. For instance, batch training is not an option when using hybrid models, making it difficult to efficiently train classical models at scale. More importantly, it is difficult to determine which component of the model is providing the most value in terms of performance. As a result, current QGLMs are inferior at processing high-dimensional continuous distribution tasks. The design of efficient read-in and read-out strategies can dramatically improve the capabilities of QGLMs in the regime of many conventional machine learning tasks.

Learning Machines: Another key challenge in the regime of QGLMs is exploring their learnability from the perspective of generalization, expressivity, and trainability. A deep theoretical understanding of these topics cannot only benefit the evaluation of the performance of different QGLMs but also provides guidelines to design more advanced QGLMs with computational advantages. For instance, a recent study [199] argued to take generalization as a measure to quantify the capability of GLMs and QGLMs in generating unseen and diverse data. Nevertheless, contrary to quantum discriminative learning [200], [201], [202], [203], [204], [205], [206], very few studies have attempted to unveil the learnability of QGLMs, caused by the following reasons.

The main difficulty in studying the generalization and the expressivity of QGLMs is that, unlike (quantum) discriminative models, the definition of generalization in (quantum) generative learning is vague under different model settings. As a result, prior literature only focuses on the specified QGLMs. Concretely, Ref. [207] proved a theoretical proof of expressivity enhancement from Bayesian networks to their quantum counterpart. Moreover, Du et al. [208] analyzed the generalization of QCBMs and QGANs under the maximum mean discrepancy loss and demonstrated the potential advantages of these QGLMs. However, less is known about the generalization of QGLMs in terms of different objective functions (e.g., KL divergence) and the varied protocols (e.g., QRBMs and QVAE).

The hardness in investigating the trainability of QGLMs originates from the non-convex loss landscape for the objective function. When the number of trainable parameters becomes large, the analysis for the universal convergence behavior toward the global minima is intractable. This further incurs two unsolved but crucial issues. First, how to mitigate the barren plateaus phenomenon of QGLM in the training procedure? Concisely, barren plateaus refer that the gradients of QNN could exponentially vanish when the number of qubits and the circuit depth are linearly increased. Although Kieferova et al. [209] proved that unbounded loss may enable the absence of barren plateaus for QBMs and Unitary QNNs, the strategies to alleviate barren plateaus for other QGLMs with varied objective functions are rather unclear. Second, how to ensure the optimization can locate the optimal or near-optimal result? A large discrepancy between the estimated and the optimal results hint at the large difference between the estimated and the target distribution, which may make the potential advantages of QGLMs ambiguous.

Noise: The performance of QGLMs can be significantly degraded by quantum hardware imperfections such as gate noise and measurement error. As discussed by Stilck França and Garcia-Patron [210], when the level of noise exceeds a threshold, the power of near-term quantum computers is inferior to the classical computers. Moreover, quantum system noise may result in diverged optimization [211]. Consequently, the unavoidable noise forces us to redesign QGLMs, i.e., how to improve the robustness of QGLMs in NISQ devices while keeping their superiorities? The conventional approach is adopting the hardware-efficient Ansätze with a shallow circuit depth to avoid the negative effects of quantum noise, while the price to pay is suppressing the power of QGLMs and diminishing the computational advantages of QGLMs. More clever methods are demanded.

B. Prospects

There are several promising directions for the future study of quantum generative learning.

Read-In & Read-Out Bottleneck: The query of efficient state preparation methods is ubiquitous in quantum computing, which can tackle the read-in bottleneck of QGLMs as well as other quantum algorithms. Recently, Zhang et al. [212] analyzed the optimal circuit depth in quantum state preparation. Namely, given an exponential number of ancillary qubits, any $N$-qubit quantum state can be prepared with $\Theta(N)$ depth circuit using only single- and two-qubit gates. Moreover, when the state is sparse with $d$ non-zero entries, the circuit depth can be reduced to
$O(\log(Nd))$ with $O(Nd\log d)$ ancillary qubits. It is intriguing to generalize this result into the NISQ regime, i.e., under the tolerable error, is any more efficient amplitude encoding method to load the classical data into quantum states?

As for mitigating the read-out bottleneck of QGLMs, there are two potential solutions. The first one is extending the ideas developed in the fault-tolerant regime to the NISQ regime [213]. Another one is integrating random measurement techniques [214], e.g., shadow tomography [215] and measurement grouping [216] into QGLMs, where the target data can be extracted by the post-processing effectively.

**Learning Machines:** To unveil the learnability of QGLM in terms of expressivity, generalization, and trainability, we can follow the path of quantum discriminative learning. For instance, Du et al. [208] have utilized the tool in statistical learning — covering numbers, to quantify the generalization of QGANs under MMD loss. Concisely, they demonstrate how the number of training data, the architecture of QNNs, and the encoding methods impact the generalization of QGANs. Notably, such a tool has also been utilized in analyzing the expressivity and the generalization of QNNs in quantum discriminative learning [202]. It is intriguing to exploit many other advanced statistical tools to understand the capabilities and limitations of QGLMs.

A philosophy in devising powerful underparameterized QGLMs is controlling the expressivity to be moderate. That is, the optimal solution is covered by the hypothesis space represented by QGLMs, while the size of this space is restricted to be relatively small. As such, the estimated solution can well approximate the optimal solution with both low empirical risk and low generalization error. Two leading paradigms in controlling the expressivity of QGLMs are quantum circuit architecture search [217] and injecting prior knowledge such as invariance in Ansatz design [218]. In a nutshell, the former continuously discards the unintended hypothesis functions in the training stage [217], [219], [220], [221], [222], [223], while the latter discards the unintended hypothesis functions in the initialization stage [218], [224], [225]. In this regard, an interesting future direction is combining these two paradigms to further enhance the power of QGLMs. Meanwhile, the slimmed parameter space contributes to the alleviation of the barren plateaus phenomenon.

Besides delving into underparameterized QGLMs, it is non-trivial to unveil the abilities of overparameterized QGLMs. Recent studies have shown the optimal training behavior of the overparameterized quantum discriminative learning models [205]. Nevertheless, whether overparameterized QGLMs can preserve this desired property or not deserves to be further explored. Meanwhile, whether the initialization strategies designed for underparameterized quantum discriminative learning models are suitable for overparameterized QGLMs is unclear [226], [227], [228], [229]. A new pivotal line of research is the creation of novel Quantum Graphical Learning Machines (QGLMs). For example, Gao et al. [230] proposed a unique QGLM and proved that for certain cases, it can offer a quantum advantage in its representation capabilities compared to classical generative models. As their proposal requires the development of parent Hamiltonians for many-body entangled states and recursive quantum phase estimation, its implementation is resource-intensive and not applicable for near-term quantum computers. All in all, the developed theories related to the learnability of QGLM should indicate how different learning models, objective functions, and optimizers determine the performance of QGLMs and contribute to pursuing computational advantages.

**Noise:** To suppress the imperfection of the NISQ machine, two possible ways are developing error mitigation techniques for QGLMs [231], [232], [233], [234], [235] and designing problem-specific and hardware-oriented Ansätze [217], [236], [237], [238]. The former aims to extract noise-free results from the noisy ones, which could help compensate for the errors induced by readout and the quantum circuit itself [130]. The latter aims to design problem-oriented and hardware-specific Ansätze to suppress the effects of noise. The effectiveness of these two approaches has been validated in the regime of quantum discriminative learning, whereas little is known about QGLMs.

**REFERENCES**

[1] J. Goodfellow et al., Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[2] A. Voulodimos et al., “Deep learning for computer vision: A brief review,” Comput. Intell. Neurosci., vol. 2018, 2018, Art. no. 7068349.
[3] D. W. Otter, J. R. Medina, and J. K. Kalita, “A survey of the usages of deep learning for natural language processing,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 2, pp. 604–624, Feb 2021.
[4] J. Vanamethevan et al., “Applications of machine learning in drug discovery and development,” Nature Rev. Drug Discov., vol. 18, no. 6, pp. 463–477, 2019.
[5] G. Carleo et al., “Machine learning and the physical sciences,” Rev. Modern Phys., vol. 91, no. 4, 2019, Art. no. 045002.
[6] N. C. Thompson et al., “The computational limits of deep learning,” 2020, arXiv: 2007.05558.
[7] S. Bravyi et al., “Quantum advantage with noisy shallow circuits,” Nature Phys., vol. 16, no. 10, pp. 1040–1045, 2020.
[8] F. Arute et al., “Quantum supremacy using a programmable superconducting processor,” Nature, vol. 574, no. 7797, pp. 505–510, 2019.
[9] J. Preskill, “Quantum computing in the NISQ era and beyond,” Quantum, vol. 2, 2018, Art. no. 097.
[10] I. Liu et al., “Quantum generative adversarial learning in a superconducting quantum circuit,” Sci. Adv., vol. 5, no. 1, 2019, Art. no. eaax2761.
[11] H.-L. Huang et al., “Experimental quantum generative adversarial networks for image generation,” Phys. Rev. Appl., vol. 16, no. 2, 2021, Art. no. 024051.
[12] Y.-X. Jin et al., “Quantum deep learning for mutant COVID-19 strain prediction,” 2022, arXiv:2203.03556.
[13] M. Schuld et al., “An introduction to quantum machine learning,” Contemp. Phys., vol. 56, no. 2, pp. 172–185, 2015.
[14] J. Biamonte et al., “Quantum machine learning,” Nature, vol. 549, no. 7671, pp. 195–202, 2017.
[15] C. Ćiliberto et al., “Quantum machine learning: A classical perspective,” Proc. Roy. Soc. A: Math. Phys. Eng. Sci., vol. 474, no. 2209, 2018, Art. no. 20170551.
[16] K. Bhati et al., “Noisy intermediate-scale quantum algorithms,” Rev. Modern Phys., vol. 94, no. 1, 2022, Art. no. 015004.
[17] M. Cerezo et al., “Variational quantum algorithms,” Nature Rev. Phys., vol. 3, no. 9, pp. 625–649, 2021.
[18] F. V. Massoli et al., “A leap among quantum computing and quantum neural networks: A survey,” ACM Comput. Surv., vol. 55, 2022, Art. no. 98.
[19] M. Benedetti et al., “Parameterized quantum circuits as machine learning models,” Quantum Sci. Technol., vol. 4, no. 4, 2019, Art. no. 043001.
[20] W. Li and D.-L. Deng, “Recent advances for quantum classifiers,” Sci. China Phys. Mechanics Astron., vol. 65, no. 2, pp. 1–23, 2022.
A. Radford et al., “Unsupervised representation learning with deep energy functions,” in Proc. Int. Conf. Learn. Representations, 2019.

G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” Neural Computation, vol. 14, no. 8, pp. 1771–1800, 2002.

D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 307–392, 2019.

D. J. Rezende et al., “Stochastic backpropagation and approximate inference in deep generative models,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 1278–1286.

K. Gregor et al., “DRAW: A recurrent neural network for image generation,” in Proc. Int. Conf. Mach. Learn., 2015, pp. 1462–1471.

S. Kullback and R. A. Leibler, “On information and sufficiency,” Ann. Math. Statist., vol. 22, no. 1, pp. 79–86, 1951.

R. Gómez-Bombarelli et al., “Automatic chemical design using a data-driven continuous representation of molecules,” ACS Central Sci., vol. 4, no. 2, pp. 268–276, 2018.

B. Dai and D. Wipf, “Diagnosing and enhancing VAE models,” in Proc. Int. Conf. Learning Representations, 2019.

A. Kamar et al., “Regularized autoencoders via relaxed injective probability flow,” in Proc. Int. Conf. Artif. Intell. Statist., 2020, pp. 4292–4301.

A. Vahdat and J. Kautz, “NVAE: A deep hierarchical variational autoencoder,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 19667–19679.

Y. Burda et al., “Importance weighted autoencoders,” in Proc. 4th Int. Conf. Learn. Representations, 2016.

S. Zhao et al., “InfoVAE: Balancing learning and inference in variational autoencoders,” in Proc. 33rd AAAI Conf. Artif. Intell. 31st Innov. Appl. Artif. Intell. Conf. 9th AAAI Symp. Educ. Adv. Artif. Intell., AAAI Press, 2019, Art. no. 722.

I. Higgins et al., “β-VAE: Learning basic visual concepts with a constrained variational framework,” in Proc. 5th Int. Conf. Learn. Representations, 2017.

A. Van Der Oord et al., “Neural discrete representation learning,” in Proc. 3rd ICLR Conf. Artif. Intell. 31st Innov. Appl. Artif. Intell. Conf. 9th AAAI Symp. Educ. Adv. Artif. Intell., AAAI Press, 2019, Art. no. 722.

J. T. Rolfe, “Discrete variational autoencoders,” arXiv:1609.02200.

M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, 10th Anniversary ed. Cambridge, U.K.: Cambridge Univ. Press, 2010.

R. P. Feynman, “Simulating physics with computers,” in Feynman and Computation. Boca Raton, FL, USA: CRC Press, 2018, pp. 133–153.
[138] L. Rios and N. Sahinidis, “Derivative-free optimization: A review of algorithms and comparison of software implementations,” J. Glob. Optim., vol. 56, pp. 1247–1293, 2013.

[139] J. C. Spall, “Implementation of the simultaneous perturbation algorithm for stochastic optimization,” IEEE Trans. Aerosp. Electron. Syst., vol. 34, no. 3, pp. 817–823, Jul. 1998.

[140] A. Kondratyev, “Non-differential learning of quantum circuit Born machine with genetic algorithm,” Wilmott, vol. 2021, no. 114, pp. 50–61, 2021.

[141] J. H. Holland, “Genetic algorithms,” Sci. Amer., vol. 267, no. 1, pp. 66–73, 1992.

[142] Y.-R. Liu et al., “ZOOpt: Toolbox for derivative-free optimization,” 2017, arXiv: 1801.00329.

[143] K. Nakaji and N. Yamamoto, “Quantum semi-supervised generative adversarial network for enhanced data classification,” J. Math. Phys., vol. 6, no. 11, pp. 1812–1813, 1965.

[144] E. R. Anschuetz and Y. Cao, “Realizing quantum Boltzmann machines with hidden units,” 2019, arXiv: 1905.09902.

[145] N. Stamatopoulos et al., “Option pricing using quantum computers,” Wilmott, vol. 4, 2020, Art. no. 291.

[146] S. Aaronson and A. Arkhipov, “The computational complexity of linear optics,” J. Statist. Phys., vol. 128, no. 3, pp. 305–328, 2007.

[147] T. Xiao et al., “Quantum Boltzmann machine algorithm with dimension-expanded equivalent Hamiltonian,” Phys. Rev. A, vol. 101, no. 3, 2020, Art. no. 032304.

[148] M. Pistoia et al., “Adversarial quantum circuit learning for pure state approximation,” New J. Phys., vol. 21, no. 4, 2019, Art. no. 043023.

[149] M. Benedetti et al., “Adversarial circuit learning in quantum physics,” New J. Phys., vol. 21, no. 4, 2019, Art. no. 043023.

[150] K. Huang et al., “Realizing a quantum generative adversarial network using a programmable superconducting processor,” 2020, arXiv: 2009.12827.

[151] X.-F. Yin et al., “Efficient bipartite entanglement detection scheme with a quantum adversarial solver,” Phys. Rev. Lett., vol. 128, no. 11, Art. no. 110501.

[152] K. Nakaji and N. Yamamoto, “Quantum semi-supervised generative adversarial network for enhanced data classification,” Sci. Rep., vol. 11, no. 1, pp. 1–10, 2021.

[153] J. Statist. Phys., vol. 1, no. 2, pp. 231–249, 1999.

[154] S. Aaronson and A. Arkhipov, “The computational complexity of linear optics,” in Prog. 43rd Annu. ACM Symp. Theory Comput., 2011, pp. 333–342.

[155] N. Wiebe et al., “Quantum language processing,” 2019, arXiv: 1902.05162.

[156] W. Magnus, “On the exponential solution of differential equations for a linear operator,” Commun. Pure Appl. Math., vol. 7, no. 4, pp. 649–673, 1954.

[157] S. McArdle et al., “Variational Ansatz-based quantum simulation of imaginary time evolution,” npj Quantum Inf., vol. 5, no. 1, pp. 1–6, 2019.

[158] X. Yuan et al., “Theory of variational quantum simulation,” Quantum, vol. 3, 2019, Art. no. 191.

[159] R. Arbes et al., “Machine learning algorithms based on generalized Gibbs ensembles,” J. Statist. Mechanics: Theory Experiment, vol. 2018, no. 10, 2018, Art. no. 103102.

[160] M. Rigol et al., “Relaxation in a completely integrable many-body quantum system: An ab initio study of the dynamics of the highly excited states of 1D lattice hard-core bosons,” Phys. Rev. Lett., vol. 98, no. 5, 2007, Art. no. 050405.

[161] T. Barthel and U. Schollwöck, “Dephasing and the steady state in quantum many-particle systems,” Phys. Rev. Lett., vol. 100, no. 10, 2008, Art. no. 100601.

[162] R. S. Sutton, “Integrated architectures for learning, planning, and reacting based on approximating dynamic programming,” in Proc. 7th Int. Conf. Mach. Learn., B. W. Porter and R. J. Mooney, Eds. San Francisco, CA, USA, 1990, pp. 216–224.

[163] E. Crosson and A. W. Harrow, “Simulated quantum annealing can be exponentially faster than classical simulated annealing,” in Proc. IEEE 57th Annu. Symp. Found. Comput. Sci., 2016, pp. 714–723.

[164] S. V. Isakov et al., “Understanding quantum tunneling through quantum Monte Carlo simulations,” Phys. Rev. Lett., vol. 117, no. 18, 2016, Art. no. 180402.

[165] T. Tieleman, “Training restricted Boltzmann machines using approximations to the likelihood gradient,” in Proc. 25th Int. Conf. Mach. Learn., USA, 2008, pp. 1064–1070.

[166] H. Rieger and N. Kawashima, “Application of a continuous time cluster expansions to the likelihood gradient,” Eur. Phys. J. B-Condens. Matter Complex Syst., vol. 9, no. 2, pp. 233–236, 1999.

[167] K. Hukushima and Y. Iba, “Population annealing and its application to a spin glass,” in Proc. AIP Conf., Amer. Inst. Phys., College Park, MD, USA, 2003, pp. 200–206.

[168] N. Gao et al., “High-dimensional similarity search with quantum-assisted variational algorithms,” J. Phys. A: Math. Gen., vol. 34, no. 4, 2001, Art. no. 2000003.

[169] E. Crosson and A. W. Harrow, “Simulated quantum annealing can be exponentially faster than classical simulated annealing,” in Proc. IEEE 57th Annu. Symp. Found. Comput. Sci., 2016, pp. 714–723.

[170] S. V. Isakov et al., “Understanding quantum tunneling through quantum Monte Carlo simulations,” Phys. Rev. Lett., vol. 117, no. 18, 2016, Art. no. 180402.

[171] T. Tieleman, “Training restricted Boltzmann machines using approximations to the likelihood gradient,” in Proc. 25th Int. Conf. Mach. Learn., USA, 2008, pp. 1064–1070.

[172] H. Rieger and N. Kawashima, “Application of a continuous time cluster expansions to the likelihood gradient,” Eur. Phys. J. B-Condens. Matter Complex Syst., vol. 9, no. 2, pp. 233–236, 1999.

[173] K. Hukushima and Y. Iba, “Population annealing and its application to a spin glass,” in Proc. AIP Conf., Amer. Inst. Phys., College Park, MD, USA, 2003, pp. 200–206.

[174] N. Gao et al., “High-dimensional similarity search with quantum-assisted variational algorithms,” J. Phys. A: Math. Gen., vol. 34, no. 4, 2001, Art. no. 2000003.

[175] E. Crosson and A. W. Harrow, “Simulated quantum annealing can be exponentially faster than classical simulated annealing,” in Proc. IEEE 57th Annu. Symp. Found. Comput. Sci., 2016, pp. 714–723.
[196] M. D. Hoffman and M. J. Johnson, “ELBO surgery: Yet another way to carve up the variational evidence lower bound,” in Proc. Workshop Adv. Approx. Bayesian Inference, 2016.

[197] S. Aaronson, “Read the fine print,” *Nature Phys.*, vol. 11, no. 4, pp. 291–293, 2015.

[198] M. Henderson et al., “Quantum convolutional neural networks: Powering image recognition with quantum circuits,” *Quantum Mach. Intell.*, vol. 2, no. 1, pp. 1–9, 2020.

[199] K. Gali et al., “Evaluating generalization in classical and quantum generative models,” 2022, arXiv:2201.08770.

[200] A. Abbas et al., “The power of quantum neural networks,” *Nature Comput. Sci.*, vol. 1, no. 6, pp. 403–409, 2021.

[201] L. Banchi et al., “Generalization in quantum machine learning: A quantum information standpoint,” *PRX Quantum*, vol. 2, no. 4, 2021, Art. no. 040321.

[202] Y. Du et al., “Efficient measure for the expressivity of variational quantum algorithms,” *Phys. Rev. Lett.*, vol. 128, no. 8, 2022, Art. no. 080506.

[203] H.-Y. Huang et al., “Information-theoretic bounds on quantum advantage in machine learning,” *Phys. Rev. Lett.*, vol. 126, no. 19, 2021, Art. no. 190505.

[204] Y. Qian et al., “The Dilemma of quantum neural networks,” 2021, arXiv:2106.04975.

[205] J. Liu et al., “An analytic theory for the dynamics of wide quantum neural networks,” 2022, arXiv:2203.16711.

[206] M. C. Caro et al., “Generalization in quantum machine learning from few training data,” 2021, arXiv:2111.05292.

[207] X. Gao et al., “Enhancing generative models via quantum correlations,” *Phys. Rev. X*, vol. 12, no. 2, 2022, Art. no. 021037.

[208] Y. Du et al., “Theory of quantum generative learning models with maximum mean discrepancy,” 2022, arXiv:2205.04730.

[209] M. Kieferova et al., “Quantum generative training using Rényi divergences,” 2021, arXiv:2106.09567.

[210] D. Stilck França and R. García-Patron, “Limitations of optimization algorithms on noisy quantum devices,” *Nature Phys.*, vol. 17, no. 11, pp. 1221–1227, 2021.

[211] Y. Du et al., “Learnability of quantum neural networks,” *PRX Quantum*, vol. 2, no. 4, 2021, Art. no. 040337.

[212] X.-M. Zhang et al., “Quantum Gram-Schmidt processes and their application to efficient state readout for quantum algorithms,” *Phys. Rev. Res.*, vol. 3, no. 4, 2021, Art. no. 043095.

[213] A. Elben et al., “The randomized measurement toolbox,” 2022, arXiv:2203.11374.

[214] H.-Y. Huang et al., “Predicting many properties of a quantum system from very few measurements,” *Nature Phys.*, vol. 16, no. 10, pp. 1050–1057, 2020.

[215] P. J. O’Malley et al., “Scalable quantum simulation of molecular energies,” *Phys. Rev. X*, vol. 6, no. 3, 2016, Art. no. 031007.

[216] D. Stilck França and R. García-Patron, “Limitations of optimization algorithms on noisy quantum devices,” 2021, arXiv:2106.04975.

[217] Y. Du et al., “Quantum circuit architecture search: Error mitigation and trainability enhancement for variational quantum solvers,” 2020, arXiv:2010.10217.

[218] M. Larocca et al., “Group-invariant quantum machine learning,” 2022, arXiv:2205.02261.

[219] K. Linghui et al., “Quantum circuit architecture search on a superconducting processor,” 2022, arXiv:2201.06934.

[220] H. R. Grimsley et al., “An adaptive variational algorithm for exact molecular simulations on a quantum computer,” *Nature Commun.*, vol. 10, no. 1, pp. 1–9, 2019.

[221] D. Chivilikhin et al., “MoG-VQE: Multibjective genetic variational quantum eigensolver,” 2020, arXiv:2007.04424.

[222] L. Li et al., “Quantum optimization with a novel Gibbs objective function and Ansatz architecture search,” *Phys. Rev. Res.*, vol. 2, no. 2, 2020, Art. no. 023074.

[223] M. Ostaszewski et al., “Structure optimization for parameterized quantum circuits,” *Quantum*, vol. 5, 2021, Art. no. 391.

[224] A. Skolik et al., “Equiariant quantum circuits for learning on weighted graphs,” 2022, arXiv:2205.06109.

[225] J. J. Meyer et al., “Exploiting symmetry in variational quantum machine learning,” 2022, arXiv:2205.06217.

[226] E. Grant et al., “An initialization strategy for addressing barren plateaus in parameterized quantum circuits,” *Quantum*, vol. 3, Dec. 2019, Art. no. 214.

[227] K. Mitarai et al., “Quadratic Clifford expansion for efficient benchmarking and initialization of variational quantum algorithms,” 2020, arXiv:2011.09927.

[228] J. Dotor et al., “Matrix product state pre-training for quantum machine learning,” *Quantum Sci. Technol.*, vol. 7, 2022, Art. no. 035014.

[229] K. Zhang et al., “Gaussian initializations help deep variational quantum circuits escape from the barren plateau,” 2022, arXiv:2203.09376.

[230] X. Gao et al., “A quantum machine learning algorithm based on generative models,” *Sci. Adv.*, vol. 4, no. 12, 2018, Art. no. eaat9004.

[231] K. Temme et al., “Error mitigation for short-depth quantum circuits,” *Phys. Rev. Lett.*, vol. 119, no. 18, 2017, Art. no. 180509.

[232] Y. Li and S. C. Benjamin, “Efficient variational quantum simulator incorporating active error minimization,” *Phys. Rev. X*, vol. 7, no. 2, 2017, Art. no. 021050.

[233] N. Olch et al., “Extending the lifetime of a quantum bit with error correction in superconducting circuits,” *Nature*, vol. 536, no. 7617, pp. 441–445, 2016.

[234] C. K. Andersen et al., “Repeated quantum error detection in a surface code,” *Nature Phys.*, vol. 16, no. 8, pp. 875–880, 2020.

[235] S. Endo et al., “Hybrid quantum-classical algorithms and quantum error mitigation,” *J. Phys. Soc. Jpn.*, vol. 90, no. 3, 2021, Art. no. 032001.

[236] M. Bilikis et al., “A semi-agnostic ansatz with variable structure for quantum machine learning,” 2021, arXiv:2103.06712.

[237] H. L. Tang et al., “Qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient Ansätze on a quantum processor,” *PRX Quantum*, vol. 2, no. 2, 2021, Art. no. 020310.

[238] D. Amaro et al., “Filtering variational quantum algorithms for combinatorial optimization,” *Quantum Sci. Technol.*, vol. 7, no. 1, 2022, Art. no. 015002.