A Generative Model Based on Variational Quantum Principal Component Analysis[*]

Haimeng Zhao[1†] and Weixiao Sun[2‡]
[1] *Zhili College, Tsinghua University, Beijing 100084, China*
[2] *Center for Quantum Information, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, China*

(Dated: April 5, 2022)

Generative learning is a major branch of machine learning which has yielded significant applications in a wide range of areas. Quantum generative learning utilizes the power of quantum computers to learn and generate quantum data, demonstrating potential supremacy over its classical counterpart. Here, we propose a simple yet powerful generative model based on variational quantum principal component analysis (G-qPCA), which extracts the leading eigenvalues and corresponding eigenvectors of a given quantum state via a parameterized quantum circuit, and generates data by preparing the state accordingly. Recognizing the linearity of quantum mechanics and optimality of qPCA in state discrimination, we further discover that three mainstream generative models in the classical literature can be unified in the quantum regime and reduced to our algorithm. Along the way, we propose a fully quantum formulation of variational autoencoder and introduce the quantum version of normalizing flow. Our algorithm is demonstrated with numerical simulations and experimentally implementable on noisy intermediate-scale quantum devices, while free from the use of quantum random access memory which is a caveat in most quantum machine learning algorithms.

I. INTRODUCTION

Generative modeling is an important task in machine learning. A generative model learns to model the distribution of the training data and generate new samples accordingly[1, 2]. Classical generative models can be divided into two categories: directed ones and undirected ones[1]. Traditional generative models like Bayesian networks, Markov random fields, deep belief networks and numerous variants of Boltzmann machines are examples of the undirected ones[1–3]. As for the directed ones, differential generator network based models are the spotlight of modern research for their expressiveness and flexibility[1]. There're three major types of such models: generative adversarial network (GAN)[4], variational autoencoder (VAE)[5] and normalizing flow (Flow)[6].

Recent advances in quantum machine learning exploit the power of quantum computers to accelerate generation tasks. Examples include factor graph based quantum generative models (QGM)[7] and quantum generative adversarial networks (qGAN)[8, 9]. These quantum extension of generative models have demonstrated supremacy in representation power, training and inference speed[7–9]. Moreover, quantum generative models by its nature circumvent[7] the use of quantum random access memory (QRAM)[10] which is a caveat in most quantum machine learning algorithms[11, 12].

However, the species of modern classical generative models still remain largely untouched. Traditional undirected generative models can be unified in the language of factor graph, with QGM as their natural quantization. Whether modern directed generative models, especially those based on differential generator networks, can be extended or even unified in the quantum regime is still unknown. In this article, we propose a simple yet powerful generative model based on variational quantum principal component analysis (qPCA), and show that the forementioned three major types of directed generative models can be unified in the quantum regime and reduced to our algorithm. The optimality of qPCA in state discrimination further indicates that a wide range of classical generative models, especially those based on differentiable generator networks, can be quantized in our framework and reduced to our algorithm.

Principal component analysis (PCA)[13] is the most simple yet widely applicable dimensionality reduction tool which extracts the leading eigenvalues and eigenvectors of the data's covariance matrix. Reversing the procedure of PCA is the most simple idea one can think of to do generation tasks, but is prohibited by its non-probabilistic nature. The probabilistic extension of PCA is capable of generating new samples, but is largely limited in expression power[14, 15] compared to modern approaches. The originally proposed quantum extension of principal component analysis (qPCA)[16], which tries to extract the leading eigenvalues and corresponding eigenvectors from a given density matrix representing the data distribution, in contrast is irreversible and cannot be used as a generation algorithm.

Nevertheless, a recent experiment proposed a variant of qPCA: the variational qPCA, which used parametrized quantum circuits to learn the algorithm of diagonalizing a density matrix[17]. It's not only suitable for noisy intermediate-scale quantum (NISQ) devices, but also reversible. Reversing this trained state diagonalization circuit yields a generative model, in which we only need to prepare states in the computational basis according to the eigenvalues and feed them into the reversed circuit.

[*] This is a course project for Quantum Artificial Intelligence given by Dongling Deng at Tsinghua.

[†] haimengzhao@icloud.com

[‡] sunwx21@mails.tsinghua.edu.cn

The remainder of this article is structured as follows. First, we introduce the variational qPCA and construct our generative model based on it. Next, we introduce the fully quantum version of GAN, VAE and Flow, in which the latter two has not yet been proposed in the literature. We further reduce all the three algorithms to our model and discuss the reason why our model can unify them by exploiting the linearity of quantum mechanics. Then, we demonstrate the validity of our algorithm with numerical simulations of four qubits. Finally, the conclusions are presented.

II. ALGORITHM

As stated in the above section, our generative model is essentially the Hermitian conjugate of the variational qPCA proposed by [17]. So we first give a brief summary of the variational qPCA and then introduce our generative model.

A. Variational qPCA

Given a quantum state ρ, qPCA aims to find its leading eigenvalues and corresponding eigenstates. Suppose ρ's spectral decomposition is given by $\rho = \sum_{j=0}^{N-1} \lambda_j |\psi_j\rangle\langle\psi_j|$, where $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{N-1}$ are its eigenvalues and $\{|\psi_j\rangle\}$ the corresponding eigenstates. We want to find a unitary operation U that maps $|\psi_j\rangle$ to the computational basis $|j\rangle$, that is, diagonalize ρ as

$$\rho_c = U\rho U^\dagger = \sum_{j=1}^{N-1} \lambda_j |j\rangle\langle j|. \tag{1}$$

For a low rank state ρ, we can then perform measurements in the computational basis $|j\rangle$, and efficiently extract the leading λ_j from the statistics.

To variationally learn such a unitary operation, we can parametrize it with a parametrized quantum circuit $U(\theta)$ and obtain an approximation by minimizing a loss function $L(\theta)$. It's shown in [17] that for any non-degenerate Hermitian operator P that is diagonal in the computational basis, i.e. $P = \sum_{j=0}^{N-1} p_j |j\rangle\langle j|$, $p_0 < p_1 < \cdots < p_{N-1}$, a suitable loss function is provided by

$$L(\theta) = \mathrm{Tr}\left[U(\theta)\rho U(\theta)^\dagger \cdot P\right] \geq \sum_{j=1}^{N} \lambda_j p_j. \tag{2}$$

The inequality follows from the Birkhoff–von Neumann theorem and the rearrangement inequality, where equality holds only when ρ is diagonalized in the computational basis, i.e. when (1) holds.

A direct choice of P is a diagonal matrix with elements $\{0, 1, 2, \ldots, N-1\}$ normalized by the factor $N(N-1)/2$, or, in Pauli form,

$$P = \frac{1}{N(N-1)} \sum_{j=1}^{n} 2^{j-1}(\sigma_z^j + 1). \tag{3}$$

Note that P in (3) only involves local Pauli-Z measurements, which is friendly to experimental implementation.

In summary, we can learn a parametrized quantum circuit $U(\theta)$ to approximate the target U by minimizing the loss function $L(\theta)$. The resulting $U(\theta^*)$, where $\theta^* = \mathrm{argmin}_\theta L(\theta)$ can be regarded as a good approximation of the the diagonalization algorithm. Note that a different state diagonalizaion algorithm has been proposed in [18] which is equally applicable for our purpose, but less appealing to experimental implementation.

B. Generative qPCA

Now we introduce our generative model. Given a density matrix ρ representing the distribution of some data, a quantum generative model aims to find a quantum circuit to reproduce ρ, then measuring the generated ρ yields desired samples.

Recall that the above variational qPCA algorithm gives us a unitary operator $U(\theta^*)$ that diagonalizes ρ in the computational basis. For a low rank (e.g. rank r) ρ, we can measure the resulting $\rho_c = U(\theta^*)\rho U(\theta^*)^\dagger$ in the computational basis and efficiently extract the leading r eigenvalues $\{\lambda_j\}_{j=1}^r$ as the frequency of getting $\{|j\rangle\}_{j=1}^r$.

For the generation task, we prepare $|j\rangle$ with probability λ_j, and obtain the ensemble state $\tilde{\rho}_c = \sum_j \lambda_j |j\rangle\langle j|$. Feeding this state into the reversed circuit $U(\theta^*)^\dagger$, we obtain

$$\tilde{\rho} = U(\theta^*)^\dagger \left(\sum_{j=1}^{r} \lambda_j |j\rangle\langle j|\right) U(\theta^*) \approx \rho. \tag{4}$$

This completes the generation procedure. Note that the only part that needs training is the variational qPCA. The generator is as simple as its Hermitian conjugate. We call this algorithm the generative quantum principal component analysis, or G-qPCA for short.

C. Complexity Analysis

Finally we give a brief analysis of our algorithm's complexity. First of all, the PAC (probably approximately correct) learnability of our poly(n) size circuits is guaranteed by theorems in [19] with polynomially bounded sample complexity. As for the training procedure, whether we can reach such global minimum efficiently is contrained by the optimization algorithm and loss landscape, an open problem even for classical machine learning. For a given number of optimization steps N_{step}, each measure of the loss function to precision ϵ with probability

$1-\delta$ requires $O(1/(\epsilon^2\delta))$ runs of the circuit. Therefore, by computing gradients using the parameter shift rule[20, 21] for poly(n) parameters, the total complexity of training is $O(\text{poly}(n)N_{\text{step}}/(\epsilon^2\delta))$. In the generation process, for a rank r state ρ, we need $O(r/(\epsilon^2\delta))$ runs of the circuit to estimate the leading r eigenvalues to precision ϵ with probability $1-\delta$. After that, we prepare state $|j\rangle$ with probability λ_j and feed it into the reversed circuit to obtain generated samples on the fly. In a word, our algorithm can efficiently learn and generate samples, holding an exponential speedup compared to its classical counterpart.

III. UNIFYING GAN, VAE AND FLOW IN THE QUANTUM REGIME

The crux of generation tasks is to model the probability distribution of the data. In the machine learning literature, there are three major types of generative models that are based on differentiable generator networks[1]: generative adversarial network (GAN)[4], variational autoencoder (VAE)[5] and normalizing flow (Flow)[6].

For generator based networks, the main problem is how to model the probabilities given only information of samples. As we will show below, due to the linear nature of quantum mechanics, this problem no longer exists. Thus all the three models are essentially the same in the quantum regime and have an explicit optimal solution which is exactly provided by our G-qPCA.

In the following, we briefly introduce these models and their quantum extension one by one. For each model, we discuss how to reduce them to G-qPCA. At the end, we give some intuition behind this unification. A summary can be found in Fig.1.

A. Quantum Generative Adversarial Network

Generative adversarial network (GAN)[4] turns the problem of modeling probability distribution into a minimax two-player game. The discriminator D tries to tell the generated samples from the real ones while the generator G aims to fool the discriminator. In Nash equilibrium, the generator recovers the target distribution and the discriminator cannot tell the difference any more. In the training procedure, the discriminator essentially provide the generator with a learned distance measure in the probability distribution space, although it has only seen samples.

In its quantum extension (qGAN)[8, 9], the discriminator and generator are parameterized quantum circuits, and the probability distributions are density matrices. Now the task of the discriminator is to tell the difference between these two states. This problem has been extensively studied in the field of quantum state discrimination[22] and the optimal discriminator of two states under ℓ_2 distance is known to be a Helstrøm measurement[23].

The Helstrøm measurement can be efficiently implemented using qPCA for low rank states[16] and this strategy has already been suggested in classification tasks[24]. Here in the context of qGAN, qPCA naturally fits in the place of the discriminator. Therefore, it seems that only the generator needs training.

However, as our G-qPCA shows, once we've trained a variational qPCA, we simultaneously obtain a generator by reversing $U(\theta)$. That means we don't need to train another generator at all and qGAN is essentially reduced to G-qPCA. See Fig.1a for a sketch.

B. Quantum Autoencoder

Different from GAN, variational autoencoder (VAE)[5] uses an encoder-decoder structure to model the target probability distribution. Data samples x are fed into the encoder $\mathcal{E}$ and transformed into latent variables z. The decoder $\mathcal{D}$ then tries to recover the samples from latent variables. The latent variables are set to follow some relatively simple prior distribution, e.g. a Gaussian. By minimizing the evidence lower bound (ELBO), the model implicitly learns the target distribution. Note that just as the situation in GAN, the encoder and decoder only deals directly with samples, but are trying to learn the probability distribution. The difference is how they achieve that.

There exist many attempts to quantize VAE, using a quantum model to generate latent variables[25] or parameterized quantum circuits as encoder-decoder[26]. However, the most natural way is to treat both the probability distribution and the encoder-decoder quantum, i.e. regard quantum VAE as a quantum circuit encoding and decoding density matrices. In this perspective, the architecture is essentially the same as a quantum autoencoder for compressing quantum data[27], but differently used.

Recognizing the structural similarity, we can certainly train the quantum VAE just as [27] did for quantum autoencoders. However, we immediately see that a quantum VAE can be reduced to G-qPCA. Specifically, $U(\theta)$ and $U(\theta)^\dagger$ play the role of encoder and decoder. The resulting eigenvalues $\{\lambda_j\}_{j=1}^r$ is the latent variable and can be efficiently manipulated with a classical computer for further processing. Therefore, we only need to train the variational qPCA to obtain the quantum VAE. Fig.1b shows the scheme.

Moreover, the compression task is also within the capacity of G-qPCA. It compresses a low rank state by diagonalizing it and simply keep the largest r eigenvalues and eigenvectors, which can then be fed into some classical lossless or lossy compression algorithm. In some sense, this variant of our algorithm is a "quantum algorithm for quantum tomography", in that the information of eigenvalues is retrieved and stored as classical information $\{\lambda_j\}_{j=1}^r$ while that of eigenvectors is stored in the

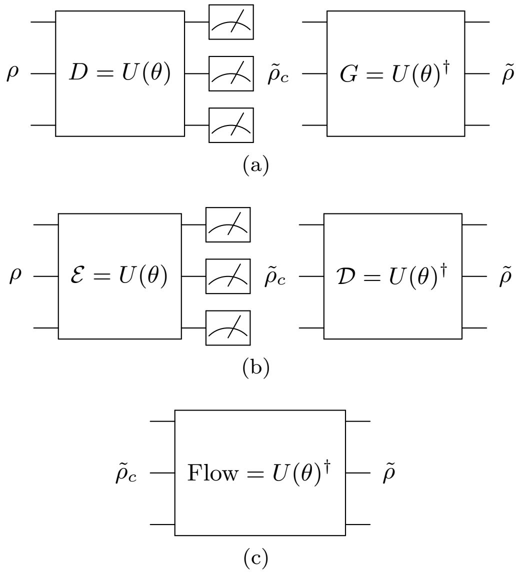

FIG. 1: GAN (1a), VAE (1b) and Flow (1c) reduced to G-qPCA.

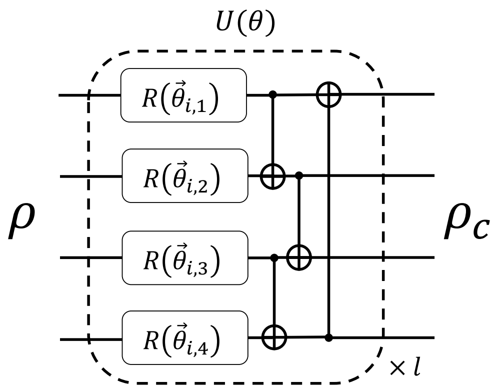

FIG. 2: The implemented $l = n^2$ layers parametrized quantum circuit $U(\theta)$. In each layer, a rotation on each qubit and a CNOT gate on each nearest neighbors are executed.

circuit $U(\theta^*)$.

C. Quantum Normalizing Flow

Classical normalizing flow[6] aims to learn a complicated probability density function $p(x)$ by fitting an invertible map of samples $f : z \to x$, which induces a map of probability density from a relatively simple density $r(z)$ (similar to VAE), to the targeted density $p(x)$. That is, for any observable A, its expectation value

$$\int A \mathrm{d}\mu = \int A(x) q(x) \mathrm{d}x = \int A(x(z)) r(z) \mathrm{d}z, \tag{5}$$

where $\mathrm{d}\mu$ is the probability measure and $q(x)$ is the transformed density whose distance with $p(x)$ is the optimization target. Recall the change-of-variable formula

$$q(x) = r(z) \left| \det \frac{\partial z}{\partial x} \right| = r(z) \left| \det \frac{\partial f}{\partial z} \right|^{-1}. \tag{6}$$

As long as we can keep track of the Jacobian $\det \frac{\partial f}{\partial z}$, we can calculate the density $q(x)$. Then we can find the optimal map f^* by minimizing the distance between $q(x)$ and the target density $p(x)$ (e.g. the Kullback–Leibler divergence). When generating new samples, we only need to sample $z \sim r(z)$ and map it to $x = f^*(z)$. Similar to GAN and VAE, the map f only deals directly with samples rather than probabilities, but here we explicitly track the density and minimize its distance with the target density.

In the quantum regime, probability densities are represented by the matrix elements $r(z, z') = \langle z|\rho|z'\rangle$ of the density matrix ρ. Similar to the classical change of variable, a unitary basis transformation $U : |z\rangle \to |x\rangle$ will change the probability density to $q(x, x') = \langle x|\rho|x'\rangle$. Intuitively speaking, it's just a change of matrix elements under basis transformation. More rigorously, and parallel to its classical counterpart, any observable A's expectation value reads[28]

$$\begin{aligned} \mathrm{Tr}(\rho A) &= \int \langle x|\rho A|x\rangle \, \mathrm{d}x = \iint \langle x'|A|x\rangle \, q(x, x') \mathrm{d}x \mathrm{d}x' \\ &= \int \langle z|\rho A|z\rangle \, \mathrm{d}z = \iint \langle z'|A|z\rangle \, r(z, z') \mathrm{d}z \mathrm{d}z'. \end{aligned} \tag{7}$$

Therefore, the quantum version of normalizing flow is nothing but a unitary transformation $U(\theta)$ (or a general quantum channel which is equivalent to a unitary with ancillary qubits) parameterized by θ, by which we aim to minimize the distance between $q(x, x')$ and some target density $p(x, x')$.

Then we readily see that our G-qPCA is exactly a quantum flow. We can certainly imitate the classical way to train the flow: prepare an prior state, apply the flow and minimize the distance between the resulting density matrix and the target density. But since we've recognized the advantage of qPCA as the optimal state discriminator, we can just train a variational qPCA and reverse it to obtain a quantum flow. Thus the training of a quantum flow is essentially reduced to our G-qPCA algorithm. See Fig.1c for a visualization.

It's quite surprising that such a straightforward quantization of normalizing flow has not yet been mentioned in the literature to our best knowledge, except that [29] introduced a similar but classical algorithm, where quantum flow is also identified as a unitary transformation but is much more complicated due to its classical implementation.

D. Why The Unification?

We end this section with a discussion of why G-qPCA can unify GAN, VAE, and Flow in the quantum regime. The intuition lies in the linear nature of quantum mechanics.

In the classical machine learning literature, generative models can be divided into two categories: directed ones and undirected ones[1]. GAN, VAE and Flow are examples of the former, while the Bayesian nets, Markov random fields, deep belief nets and numerous variants of Boltzmann machines are examples of the latter. For undirected generative models, a natural quantum extension and unification called quantum generative model (QGM) is introduced in [7] using the quantum version of factor graph. Here we aim to show that in the other category of directed generative models, those models based on differentiable generator networks can be essentially reduced to our G-qPCA in the quantum regime.

The main feature of generator network based generative models is that they are "sample oriented" rather than "probability oriented". They essentially learn a transformation of samples using a differentiable generator network rather than directly model the probability distribution. Therefore, their major concern is how to track the probability distributions and measure their distance given only information of samples. Their differences only lie in the techniques and approximations used: implicit or explicit probability density, distance measure, etc., as mentioned earlier.[1, 30]

Things are much different in the quantum regime: samples are pure states and probability distributions are density matrices which are nothing but the coherent superposition of samples. That's the key difference between classical and quantum cases: they all live in the space of states. We treat quantum probability distributions in the very same way we treat quantum samples. Moreover, this space is convex and linear leading to a canonical way to discriminate between two states under ℓ_2 distance called the Helstrøm measurement and can be implemented efficiently via qPCA[24]. These facts, along with the reversible nature of quantum circuits, give birth to a canonical construction of generator network based quantum generative models: G-qPCA.

IV. NUMERICAL EXPERIMENTS

We demonstrate our algorithm with the PennyLane software framework[31] on $n = 4$ qubits. Fig.2 shows the implemented parametrized quantum circuit $U(\theta)$, which is composed of $l = n^2$ layers of the same structure: a rotation on each qubit and a CNOT gate on each nearest neighbors.

For a given target state ρ, we apply $U(\theta)$ and measure the observable P. The loss function $L(\theta)$ is calculated and minimized via stochastic gradient descent (SGD)[1] or the Adam optimizer[32] using Tensorflow[33]. In practice, ADAM converges to the global minimum much faster than SGD, probably because of its momentum update strategy. See Fig.3 for a comparison.

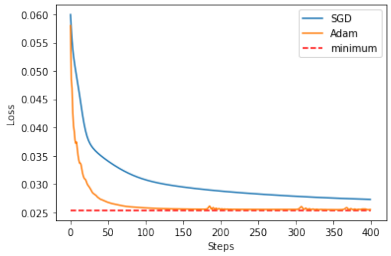

FIG. 3: Comparison of SGD and Adam on minimizing the loss function. The Red dashed line represents the theoretical minimum of loss.

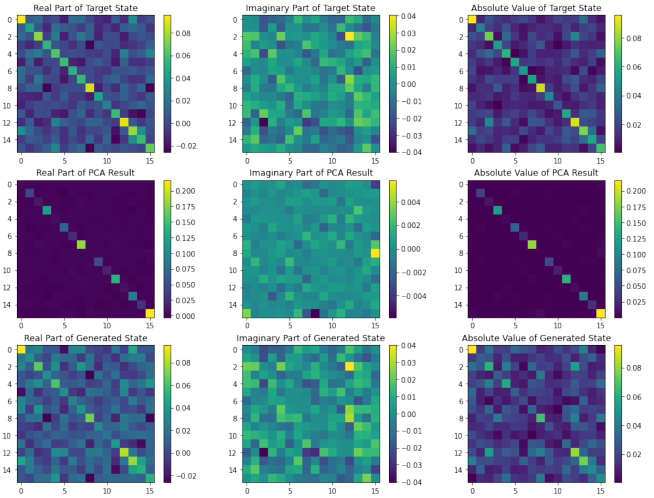

FIG. 4: Visualization of the target state ρ (top), the PCA resulted state ρ_c (middle) and the generated state $\tilde{\rho}$ (bottom) with $r = 4$.

After the circuit converges, the mean squared error (MSE) of the eigenvalues λ_j drops to about 10^{-6}. We further examined the resulted density matrix ρ_c. As shown in Fig.4, our algorithm successfully diagonalized the target state.

To carry out the generation task, we measure ρ_c and obtain the largest $r = 4$ eigenvalues $\{\lambda_j\}_{j=1}^{r}$. Then we renormalize the eigenvalues $\lambda'_j = \lambda_j / \sum_{i=1}^{r} \lambda_i$ and prepare the pure state $|j\rangle$ with probability λ'_j, obtaining the ensemble state ρ_c. Feeding it into the reversed qPCA $U(\theta)^\dagger$ yields the desired state $\tilde{\rho}$. In practice, the MSE of $\tilde{\rho}$, with ρ, is about 10^{-4} when $r = 4$, 10^{-5} when $r = 8$ and 10^{-6} when $r = 2^n = 16$. In Fig.4 we visualize the generated state $\tilde{\rho}$ with only four eigenvalues. It's readily seen that our algorithm recovers ρ with good quality.

V. CONCLUSION

In this article, we introduce a simple yet powerful generative model based on variational quantum principal component analysis (G-qPCA). It's PAC learnable with polynomially bounded sample complexity, and can efficiently learn and generate samples, holding an exponential speedup compared to its classical counterpart.

We also discover that the three mainstream generative models that are based on differentiable generator networks in classical machine learning literature: GAN, VAE and Flow, can be unified in the quantum regime and further reduced to G-qPCA. The unification lies in the linearity of quantum mechanics and the very same nature of samples and probability distributions in quantum machine learning. Along the way, we propose a fully quantum formulation of VAE and introduce the quantum version of Flow, the latter being identified as simply a parameterized unitary transformation. These variants of G-qPCA indicate its wider application in quantum tomography, data compression, etc.

We demonstrate the validity of our algorithm with numerical simulations of four qubits. The generated distribution recovers the original one with good quality. Furthermore, our algorithm's variational nature enables possible experimental implementations on NISQ devices, while its generative nature circumvents the use of QRAM which is a caveat in most quantum machine learning algorithms.

H.Z. and W.S. contributed equally to this work.

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) `http://www.deeplearningbook.org`.

[2] T. Jebara, *Machine learning: discriminative and generative*, Vol. 755 (Springer Science & Business Media, 2012).

[3] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer-Verlag, Berlin, Heidelberg, 2006).

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, Advances in neural information processing systems **27** (2014).

[5] D. P. Kingma and M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114 (2013).

[6] D. Rezende and S. Mohamed, Variational inference with normalizing flows, in *International conference on machine learning* (PMLR, 2015) pp. 1530–1538.

[7] X. Gao, Z.-Y. Zhang, and L.-M. Duan, A quantum machine learning algorithm based on generative models, Science advances **4**, eaat9004 (2018).

[8] S. Lloyd and C. Weedbrook, Quantum generative adversarial learning, Physical review letters **121**, 040502 (2018).

[9] P.-L. Dallaire-Demers and N. Killoran, Quantum generative adversarial networks, Physical Review A **98**, 012324 (2018).

[10] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum random access memory, Physical review letters **100**, 160501 (2008).

[11] S. Aaronson, Read the fine print, Nature Physics **11**, 291 (2015).

[12] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: a classical perspective, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **474**, 20170551 (2018).

[13] C. Ding and X. He, K-means clustering via principal component analysis, in *Proceedings of the twenty-first international conference on Machine learning* (2004) p. 29.

[14] M. E. Tipping and C. M. Bishop, Probabilistic principal component analysis, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **61**, 611 (1999).

[15] B. Ghojogh, A. Ghodsi, F. Karray, and M. Crowley, Factor analysis, probabilistic principal component analysis, variational inference, and variational autoencoder: Tutorial and survey (2021), arXiv:2101.00734 [stat.ML].

[16] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum principal component analysis, Nature Physics **10**, 631 (2014).

[17] T. Xin, L. Che, C. Xi, A. Singh, X. Nie, J. Li, Y. Dong, and D. Lu, Experimental quantum principal component analysis via parametrized quantum circuits, Physical Review Letters **126**, 110502 (2021).

[18] R. LaRose, A. Tikku, É. O'Neel-Judy, L. Cincio, and P. J. Coles, Variational quantum state diagonalization, npj Quantum Information **5**, 1 (2019).

[19] H. Cai, Q. Ye, and D.-L. Deng, Sample complexity of learning parametric quantum circuits (2022), arXiv:2107.09078 [quant-ph].

[20] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Physical Review A **98**, 10.1103/physreva.98.032309 (2018).

[21] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Physical Review A **99**, 10.1103/physreva.99.032331 (2019).

[22] J. Bae and L.-C. Kwek, Quantum state discrimination and its applications, Journal of Physics A: Mathematical and Theoretical **48**, 083001 (2015).

[23] C. W. Helstrom, Quantum detection and estimation theory, Journal of Statistical Physics **1**, 231 (1969).

[24] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, Quantum embeddings for machine learning, arXiv preprint arXiv:2001.03622 (2020).

[25] A. Khoshaman, W. Vinci, B. Denis, E. Andriyash, H. Sadeghi, and M. H. Amin, Quantum variational autoencoder, Quantum Science and Technology **4**, 014001 (2018).

[26] S. Pramanik and M. G. Chandra, On a possible quantum variational autoencoder circuit, in *2021 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2021) pp. 1–6.

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

[28] Note that here a change of basis only affects the matrix elements, not the abstract operator ρ, just as in the clas-

sical case, a change of variable only affects the density, not the probability measure. A more rigorous correspondence and a definition of quantum probability measure can be found in [34, 35] or other texts on quantum probability theory.

[29] K. Cranmer, S. Golkar, and D. Pappadopulo, Inferring the quantum density matrix with machine learning, arXiv preprint arXiv:1904.05903 (2019).

[30] L. Mi, M. Shen, and J. Zhang, A probe towards understanding gan and vae models, arXiv preprint arXiv:1812.05676 (2018).

[31] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, K. McKiernan, J. J. Meyer, Z. Niu, A. Száva, and N. Killoran, Pennylane: Automatic differentiation of hybrid quantum-classical computations (2020), arXiv:1811.04968 [quant-ph].

[32] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2017), arXiv:1412.6980 [cs.LG].

[33] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, Tensorflow: A system for large-scale machine learning, in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (2016) pp. 265–283.

[34] J. M. Swart, Introduction to quantum probability (2020), `http://staff.utia.cas.cz/swart/lecture_notes/qua20_04_27.pdf`, Last accessed on 2022-1-7.

[35] K. Landsman, *Foundations of quantum theory: from classical concepts to operator algebras* (Springer Nature, 2017).