Quantum Generative Adversarial Network: A Survey

Tong Li[1], Shibin Zhang[1, *] and Jinyue Xia[2]

Abstract: Generative adversarial network (GAN) is one of the most promising methods for unsupervised learning in recent years. GAN works via adversarial training concept and has shown excellent performance in the fields image synthesis, image super-resolution, video generation, image translation, etc. Compared with classical algorithms, quantum algorithms have their unique advantages in dealing with complex tasks, quantum machine learning (QML) is one of the most promising quantum algorithms with the rapid development of quantum technology. Specifically, Quantum generative adversarial network (QGAN) has shown the potential exponential quantum speedups in terms of performance. Meanwhile, QGAN also exhibits some problems, such as barren plateaus, unstable gradient, model collapse, absent complete scientific evaluation system, etc. How to improve the theory of QGAN and apply it that have attracted some researcher. In this paper, we comprehensively and deeply review recently proposed GAN and QAGN models and their applications, and we discuss the existing problems and future research trends of QGAN.

Keywords: Quantum machine learning, generative adversarial network, quantum generative adversarial network, mode collapse.

1 Introduction

Artificial intelligence has been a hot topic in recent years. Statistical machine learning has been the hottest branch of artificial intelligence. Deep learning has been one of the most popular subfields in statistical machine learning. With continuously increasing the variety and amount of data in the industry. Main IT companies compete with each other in the information market depend on the technology products of data, machine learning and data mining.

As early as 1990s, statistical machine learning has become mainstream. The algorithms involved in statistical machine learning are divided into the following major parts. The first part is classification and regression algorithms, mainly include perceptron learning algorithm, k-nearest neighbors (KNN) algorithm, Naive Bayes algorithm, Logistic Regression, maximum entropy model, support vector machines (SVM) [Li (2012)], GDBT [Friedman (2001)], XGboost [Chen and Guestrin (2016)], Prophet [Taylor and

[1] School of Cybersecurity, Chengdu University of Information Technology, Chengdu, 610225, China.

[2] International Business Machines Corporation, New York, 14201, USA.

[*]Corresponding Author: Shibin Zhang. Email: cuitzsb@cuit.edu.cn.

Received: 09 March 2020; Accepted: 31 March 2020.

CMC. doi:10.32604/cmc.2020.010551 www.techscience.com/journal/cmc

Letham (2018)], etc. The second part is the algorithm of association analysis, mainly include Apriori algorithm [Agrawal and Srikant (1994)], FP-Growth algorithm [Han, Pei and Kamber (2011)], PrefixSpan algorithm [Han, Pei, Mortazavi-Asl et al. (2001)], etc. The third part is the clustering algorithms, such as K-means [MacQueen (1967)], Gaussian mixture model [Stauffer and Grimson (1999)], etc.

By the beginning of the 21st century, emerging connectionism and deep learning has a tremendous progress. The deep learning algorithm rely on a large amount of training data and strong computing power, ensuring the extraction of complex high-dimensional features to better express the nature of the data and draw the rich information in the massive data, so that the accuracy of task is significantly improved. However, because of the insufficient innovation of deep learning theory and limited application scope, deep learning hasn't abandoned statistical machine learning. Typical deep learning algorithms include Convolutional Neural Network (CNN) [Krizhevsky, Sutskever and Hinton (2012)], Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber (1997)], Gated Recurrent Unit (GRU) [Cho, Merriënboer, Gulcehre et al. (2014)], GAN [Goodfellow, Pouget-Abadie, Mirza et al. (2014)], Variational Autoencoder (VAE) [Kingma and Welling (2013)], Graph Neural Networks (GNN) [Battaglia, Hamrick, Bapst et al. (2018)], Bidirectional Encoder Representation from Transformers (BERT) [Devlin, Chang, Lee et al. (2018)], XLNet [Yang, Dai, Yang et al. (2019)], etc. As well as reinforcement learning and semi-supervised learning algorithms, such as "AlphaGo Zero" [Silver, Schrittwieser, Simonyan et al. (2017)], semi-supervised GAN [Salimans, Goodfellow, Zaremba et al. (2016)], etc. These algorithms has shown excellent performance in many tasks such as computer vision, speech recognition, natural language processing, driverless, machine translation, spam classification, medical diagnosis, etc.

Meanwhile, quantum computing also has a tremendous progress. Main IT companies, such as Google, IBM, Alibaba, Baidu, XANADU, Origin Quantum, domestic and foreign scientific research institutions have been working on the development of quantum computers. They are committed to reduce the computational complexity of traditional computers in processing tasks and the spatial complexity of storing data, to solve the problems of quantum chemistry, quantum physics and other machine learning. These IT companies and research institutes have developed corresponding toolkits and Internet interfaces to simulate small quantum computers. For example, PennyLane [Bergholm, Izaac, Schuld et al. (2018)] provides a bridge between classical and quantum computations, making it easy to build and optimize hybrid computations, Strawberry Fields [Killoran, Izaac, Quesada et al. (2019)] is a fullstack Python library for designing, simulating, and optimizing continuous variable quantum optical circuits. QuTiP [Johansson, Nation and Nori (2013)] is open-source software for simulating the dynamics of open quantum systems, as well as the cloud platform of quantum computer of Alibaba and Origin Quantum, the quantum Software Development Kit (SDK) QPanda for quantum computing service. Researcher used quantum algorithms can solve some problems, such as the over-fitting of machine learning [Mitarai, Negoro, Kitagawa et al. (2018)], and can reduce the time complexity of machine learning algorithms [Lloyd, Mohseni and Rebentrost (2014); Rebentrost, Mohseni and Lloyd (2013)], etc. Researcher also used machine learning to solve some problems in the quantum physics field. For example, Gao et al. [Gao, Qiao, Jiao et al. (2018)] solved the problem of quantum state classification, Gao et al. designed the quantum state classifier trained by machine learning algorithm in 2018.

Up to now, many researchers combine quantum computing with classical computing, especially with machine learning. On the one hand, they use properties of the quantum physics to accelerate the costly part of traditional algorithm, or propose better new algorithms, to solve the problems that classical computing cannot solve. On the other hand, some researchers use classical machine learning to solve quantum physics problems. Therefore, we believe that the quantum classical hybrid computing architecture can solve problems that classical computing cannot solve.

The rest of this paper is organized as follows. In Section 2, we survey the research involving QGAN. In Section 3, we survey the mechanism, applications, advantages, and challenge of GAN algorithm. In Section 4, we systematically survey the mechanism, application, experiment of QGAN algorithm, these QGAN are further evaluated from loss function, applicability aspects. In Section 5, we summarize advantages, some existing problems of QGAN and discuss the potential future research topics.

2 The research involving QGAN

The research of QGAN mainly involving: GAN, QML, quantum-classical hybrid model.

2.1 The status of research on GAN

GAN is a generative model. For the history and development of GAN, Goodfellow et al. [Goodfellow, Pouget-Abadie, Mirza et al. (2014)] firstly proposed GAN model in June 2014, which widely used in unsupervised learning and semi-supervised learning field, has become one of the most discussed topics in machine learning. GAN has many advantages, it doesn't require complex Markov chains, and it also doesn't need to define explicit probability density functions, and it can generate samples in parallel. However, GAN exhibited some disadvantages, such as gradient unstable issue, mode collapse [Arjovsky and Bottou (2017)]. After the GAN framework was proposed, researchers continuously proposed variants to improve GAN. For example, in November 2014, Mirza et al. [Mirza and Osindero (2014)] proposed the conditional generative adversarial nets (cGAN). Compared with the original GAN, cGAN used the label of some data points as conditions, which greatly improved the quality of image generation. In November 2015, Radford et al. [Radford, Metz and Chintala (2015)] proposed deep convolutional generative adversarial networks (DCGAN), which combined GAN with convolution networks to solve the gradient unstable issue of GAN training. In November 2016, Mao et al. [Mao, Li, Xie et al. (2017)] used the least squares loss function to replace the loss function of GAN, and proposed the least squares generating adversarial networks (LSGAN), in some extent, which solved the vanishing gradient problem, the quality of poor image generation and mode collapse. In January 2017, Arjovsky et al. [Arjovsky, Chintala and Bottou (2017)] theoretically analyzed the lack of diversity of the original GAN in generating samples, which is mode collapse. They proposed several improvement opinion. According to those opinions, the Wasserstein GAN algorithm [Arjovsky, Chintala and Bottou (2017)] was proposed. The Wasserstein GAN (WGAN) algorithm introduced the Wasserstein distance from the loss function, which could solve the vanishing gradient problem and solve the problem of unstable training and generated diverse results. In March 2017, Gulrajani et al. [Gulrajani, Ahmed, Arjovsky et al.

(2017)] found that forced cut the updated weight of WGAN's, which easily leads to exploding or vanishing Gradients. Therefore, Gulrajani et al. [Gulrajani, Ahmed, Arjovsky et al. (2017)] used the gradient penalty to satisfy the Lipschitz continuity condition and proposed the WGAN-GP algorithm. WGAN-GP faster converged than standard WGAN, and the quality of the generated samples was higher. In March 2017, Berthelot et al. [Berthelot, Schumm and Metz (2017)] also proposed boundary equilibrium generative adversarial networks (BEGAN). The BEGAN discriminator uses the architecture of the auto-encoder to make the training process fast and stable, which better controlled the balance between the discriminator and the generator, balanced the diversity of the generated images and the image quality, and provided an indicator that whether BEGAN converge. In addition, there are many GAN variants, such as ORGAN [Guimaraes, Sanchez-Lengeling, Outeiral et al. (2017)] for music generation, Sequence Generative Adversarial Nets (SeqGAN) [Yu, Zhang, Wang et al. (2017); Lin, Li, He et al. (2017)] model for text generation, domain adaptation, medical image field, semi-supervised learning field, defense the attack of adversarial examples [Jin, Shen, Zhang et al. (2019)], etc. For reading more detailed information on GAN, please to see a review of GAN variants written by Creswell et al. [Creswell, White, Dumoulin et al. (2018)] in 2018 and Hong et al. [Hong, Hwang, Yoo et al. (2019)] in 2019.

2.2 The status of research on QML

QML is an emerging cross-cutting field combining quantum mechanics and machine learning [Biamonte, Wittek, Pancotti et al. (2017)]. For the history and development of QML, Feynman [Feynman (1982)] firstly proposed that computer based on quantum mechanical to simulate quantum systems in 1982. Compared to classical computer, quantum computers show an exponential speedup in simulation capability. In 1994, Shor [Shor (1994)] proposed Shor algorithm, which used to find the integer factors of a large number with exponential speed-up. As the size of integer factorization problem increases, the Shor algorithm can complete integer factorization in polynomial time. In 1997, Grover [Grover (1996)] proposed a quantum search algorithm. Compared to the traditional unordered database search algorithm, the Grover algorithm solve unstructured search problems with a quadratic speedup. In 2009, Harrow et al. [Harrow, Hassidim and Lloyd (2009)] proposed the Harrow-Hassidim-Lloyd (HHL) algorithm. In certain cases, the HHL algorithm provides exponential quantum speedups for solving linear equations. Since then, researcher proposed many QML algorithms based on the improved HHL algorithm and Grover algorithm. The research of QML was getting hotter. For example, in 2012, Wiebe et al. [Wiebe, Braun and Lloyd (2012)] proposed a quantum version of the least squares fit based on the HHL algorithm [Harrow, Hassidim and Lloyd (2009)] and amplitude amplication algorithm [Brassard, Hoyer, Mosca et al. (2002)]. Compared with the traditional algorithm, it has the efficiency of exponential quantum speedups in certain cases. In 2014, Lloyd et al. [Lloyd, Mohseni and Rebentrost (2014)] proposed Quantum Principal Component Analysis (QPCA) algorithm based on Quantum random access memory (QRAM). Rebentrost et al. [Rebentrost, Mohseni and Lloyd (2013)] proposed Quantum Support Vector Machine (QSVM) algorithm. The same year, Low et al. [Low, Yoder and Chuang (2014)] proposed a quantum inference based on Bayesian networks, which provides a quadratic speedup compared to classical bayesian inference network. In July 2018, Duan et al. [Duan, Yuan, Liu et al. (2018)] proposed a quantum version of the singular value threshold algorithm, which achieved exponential quantum speedups for the singular value threshold algorithm. Compared to traditional machine learning algorithms, these algorithms was used to solve some problem with speed-up, greatly reduced the time complexity of algorithm, but the classification accuracy of traditional algorithms won't be improved.

More and more researchers combined quantum computing with machine learning, continuously explored and experimented, and improved the theoretical system of QML to solve quantum chemistry problems, quantum physics problems, optimization problems, machine learning problems and other traditional problems.

2.3 The status of research on quantum-classical hybrid model

Due to small and unreliable near-term quantum computers, hardware requirements of many algorithms far beyond the capability of near-term quantum computers. Therefore, most companies and researchers committed to the quantum-classical hybrid algorithms, hoped to explore the potential quantum supremacy of the hybrid model, and believed that the hybrid model will be the most promising approach in the next few years. For the history and development of quantum-classical hybrid model. In July 2014, Peruzzo et al. [Peruzzo, McClean, Shadbolt et al. (2014)] proposed the Variational Quantum Eigensolver (VQE) algorithm to find the eigenvalues of a large Hamiltonian matrix, which utilize both quantum and classical resources to design a first hybrid quantum-classical algorithm. VQE was a heuristic search algorithm for finding eigenvalues, and used the Nelder-Mead optimization method, the quantum program runs within the classical program by minimizing the objective function and continuously optimize until VQE convergence. Compared with the phase estimation algorithm [Nielsen and Chuang (2010)], the VQE algorithm doesn't require long-term coherent evolution. Subsequently, in September 2014, Farhi et al. [Farhi, Goldstone and Gutmann (2014)] proposed another quantum-classical hybrid algorithm, a quantum approximation optimization algorithm (QAOA) that solves the combinatorial optimization problem, which applied to the MaxCut on regular graphs. In September 2018, Mitarai et al. [Mitarai, Negoro, Kitagawa et al. (2018)] proposed a parameterized quantum circuit learning (QCL), a quantum-classical hybrid algorithm that combined low-depth quantum circuits with classical computational computer, which implemented on near-term quantum computers. QCL could also approximate any nonlinear function, and it could solve the problem of over-fitting of machine learning tasks by using the property of the unitary operator.

Most QGANs are a quantum-classical hybrid algorithm, which was born under the development environment of GAN, QML and Quantum-Classical Hybrid Model. In July 2018, Lloyd et al. [Lloyd and Weedbrook (2018)] proposed the Quantum Generative Adversarial Learning (QGAL) protocol, which analyzed three possible adversarial learning scenarios from a theoretical perspective and analyzed that QGAN may exhibit the potential quantum supremacy. In July 2018, Dallaire-Demers et al. [Dallaire-Demers and Killoran (2018)] designed a conditional QGAN framework, the data of which was quantum data. The generator, discriminator and gradient calculations were parameterized quantum circuits. In October 2018, Situ et al. [Situ, He, Wang et al. (2018)] proposed that

QGAN generated discrete data by using quantum Born rules to solve the problem that classical GAN cannot generate discrete data, and complement the classical GAN theory. In January 2019, Luyan Sun team of Center for Quantum Information of Tsinghua University, implemented QGAN in a superconducting quantum circuit [Hu, Wu, Cai et al. (2018)], the fidelity between the quantum state generated by the generator and the real quantum state was 98.8%. Compared to the classical GAN, which exhibits the potential "exponential" advantage. In January 2019, Romero et al. [Romero and Aspuru-Guzik (2019)] proposed a QGAN framework for generating continuous data distribution based on quantum circuit learning [Mitarai, Negoro, Kitagawa et al. (2018)]. The discriminator was a classical neural networks or variational quantum circuit (VQC). The classical automatic differentiation tool implemented the optimization process and QGAN was implemented on near-term quantum computers. In April 2019, Benedetti et al. [Benedetti, Grant, Wossnig et al. (2019)] designed a QGAN framework for learning quantum mechanical wave functions. Since the framework generated quantum data, and learned along with the phase of the wave function, the learning difficulty was also greatly increased. In May 2019, Zeng et al. [Zeng, Wu, Liu et al. (2019)] designed a quantum-classical hybrid model QGAN, explored the effective direct sampling ability of quantum circuits by using the born rule. The framework learn to generate data which is classical discrete data. The generator is a VQC. The discriminator is a classical neural network. The optimization process uses Adam optimization algorithm, and numerical experiments were carried out by using Luo et al. [Luo, Liu, Zhang et al. (2019)]. In April 2019, Du et al. [Du, Hsieh and Tao (2019)] proposed the quantum generative adversarial learning framework for online learning, quantum multiplicative matrix weight algorithm (QMMW). The QMMW framework idea combines the QGAL framework and the online learning algorithm Matrix Multiplicative Weight framework [Arora, Hazan and Kale (2012)], the task of entanglement test is well done for pure state by QMMW. In July 2019, Barbeau et al. [Barbeau and Garcia-Alfaro (2019)] proposed QGAN to fake the Navigation Data of a Micro Aerial Vehicle and explore QGAN's application. In September 2019, Shrivastava et al. [Shrivastava, Puri, Gupta et al. (2019)] proposed the OpticalGAN to generate energy eigenstates and coherent states based on a quantum computer with continuous variable. In October 2019, Chakrabarti et al. [Chakrabarti, Huang, Li et al. (2019)] propose the Quantum Wasserstein generative adversarial network (qWGAN) to generate mixed states and pure states. In November 2019, Zoufal et al. [Zoufal, Lucchi and Woerner (2019)] proposed a QGAN framework to learn and load random distributions onto n qubits by using $O(poly(n))$ gates. Compared with current state-of-the-art techniques [Plesch and Brukner (2011)], which need to use $O(2^n)$ gates. The efficiency of loading random distribution for efficient quantum state preparation was greatly improved.

With the development of Noisy Intermediate-Scale Quantum (NISQ) technology, which will continuously promote the development of QML theory, including QGAN. With the quantum supremacy of quantum-classical hybrid algorithms show on near-term quantum computers, which will continuously promote the development of QML.

3 Research on GAN

In this section, the mechanism, advantages, disadvantages, and applications of GAN algorithm will be introduced. The GAN framework is inspired by minimax two-player game. The two players are the generator and the discriminator. The generator tries to create statistics for data that mimics those of a true data set, while a discriminator tries to discriminate between the true and fake data. The learning process for generator and discriminator can be thought of as an adversarial game, and under reasonable assumptions, the game converges to the point where the generator generates the same statistics as the true data and the discriminator is unable to discriminate between the true and the generated data. Eventually, the GAN converges to a "Nash equilibrium" [Nash (1951)] point.

3.1 The structure of GAN

GAN proposed by Goodfellow et al. [Goodfellow, Pouget-Abadie, Mirza et al. (2014)] in 2014, which is a neural network of unsupervised learning, and generator (G) and discriminator (D) play each other. Different from the generative model, such as Boltzmann machine, belief network, auto-encoder, etc. GAN is an implicit generative model. It isn't restricted by network structure.

Fig. 1 shows the structure of GAN network, which uses a simple neural network structure. The generator uses a full connection, the input Z is a randomly initialized, and the output $G(Z)$ is the fake picture data generated by the generator. Then, the generated picture and the real picture as input of discriminator for discrimination.

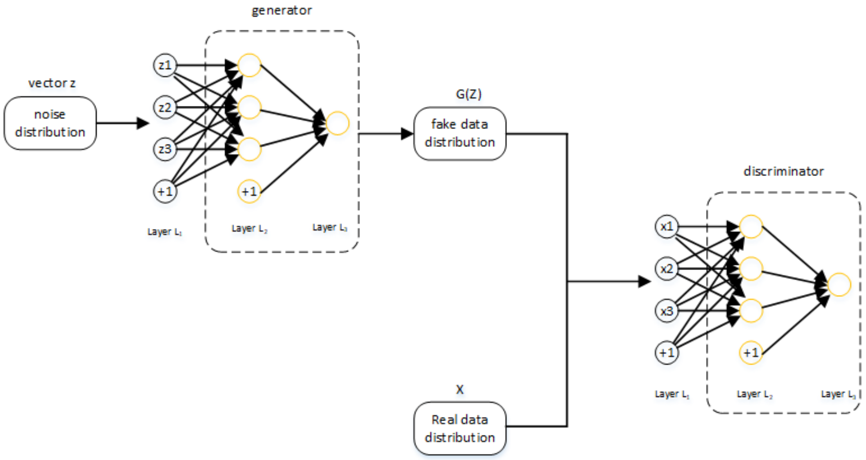

Figure 1: Simple structure of GAN

3.1.1 Original GAN

GAN consists of two parts. (1) Generator: Randomly sample Z from the latent space, such as Gaussian random noise as input of generator, which try to imitate the real sample in the training set to fool discriminator. (2) Discriminator: The real sample or generated sample from generator as input of discriminator, which try to identify the fake generated samples from generator as much as possible, and determine whether the input data is real data. These processes be described using Fig. 2.

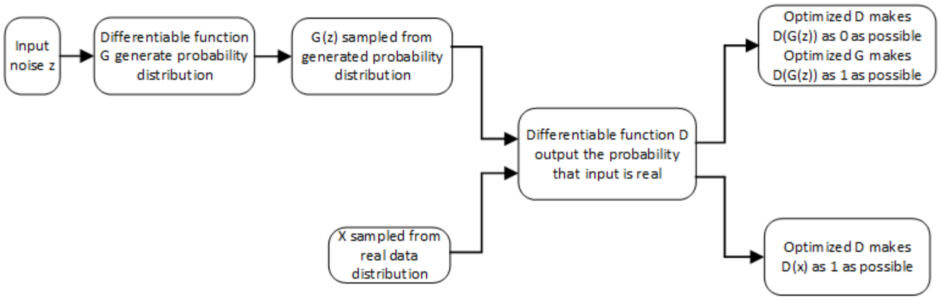
Input noise z
Differentiable function G generate probability distribution
G(z) sampled from generated probability distribution
X sampled from real data distribution
Differentiable function D output the probability that input is real
Optimized D makes D(G(z)) as 0 as possible Optimized G makes D(G(z)) as 1 as possible
Optimized D makes D(x) as 1 as possible

Figure 2: Game process of GAN

The GAN algorithms perform a *k*-step iteration to optimize the discriminator D, and then perform one-step to optimize the generator G. Eventually, the objective function converges to a saddle point, which is at the "Nash Equilibrium" [Nash (1951)] point.

3.1.2 DCGAN

The generator D and the discriminator G may be the same or different structure. Such as DCGAN [Radford, Metz and Chintala (2015)]. Fig. 3 shows the entire structure of DCGAN.

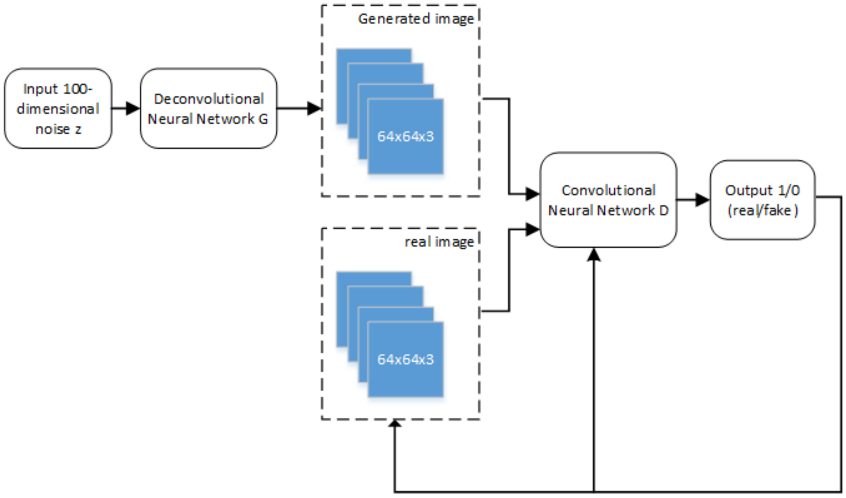
Input 100-dimensional noise z
Deconvolutional Neural Network G
Generated image
64x64x3
real image
64x64x3
Convolutional Neural Network D
Output 1/0 (real/fake)

Figure 3: Structure of DCGAN

By adding a convolutional layer, DCGAN became better excellent than original GAN to generate image data. Generator receive the initially random vector z, and output a 64×64×3 image by multiple de-convolution operations.

3.1.3 Conditional GAN

Another GAN variants is the conditional GAN (cGAN) [Mirza and Osindero (2014)] network with supervised information. The cGAN by adding some supervised information Y to guide the model output. Different from the previous model, the input part is not only a simple random noise Z, but also add label vector Y.

Fig. 4 shows the network structure of cGAN, the condition variable Y is the tag category or text.

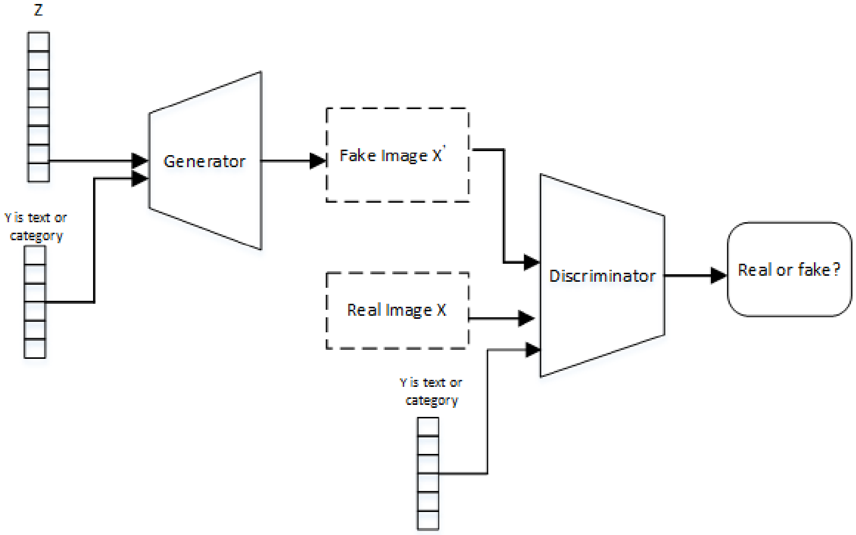

Figure 4: Structure of cGAN

If the given Y is category, the category Y be encoded into the tag vector Y by using one-hot coding, and then conditional GAN is trained. If the given Y is a text, the model [Reed, Akata, Yan et al. (2016)] for generating an image from the text $\psi(t)$, the model encodes the text as $\psi(t)$ concatenated with noise z as input, and perform the deconvolution operation to generate a picture, which related to text. The discriminator model extracts the picture features concatenated with $\psi(t)$, then output a real value or fake value. In addition to being text, Y can also be a picture.

Except for these network frameworks, there are other more network frameworks, such as hierarchical frameworks improve the performance of the GAN model to some extent.

3.2 The loss of GAN

3.2.1 GAN loss function

Since the generator competes with the discriminator in the GAN network, the generator G and the discriminator D play minimax games. The value function $V(\theta_d, \theta_g)$ of GAN network is as follows:

$$\min_{\theta_d} \max_{\theta_g} V(\theta_d, \theta_g) = E_{x\sim P_{data}(x)}[logD(x;\theta_d)] + E_{z\sim P_z(z)}[log(1 - D(G(z;\theta_g);\theta_d)) \quad (1)$$

where, $P_{data}(x)$ represents real data distribution, the $P_z(z)$ represents noise distribution, x in $D\ (x;\theta_d)$ come from the real data distribution, $D\ (x;\ \theta_d)$ is the probability that the discriminator function D classifies the correct data, $D(G(z;\theta_g);\ \theta_d)$ is the probability that the discriminator function D classifies the fake data. By fixing G, GAN algorithms maximize the value function to train discriminator D:

$$\max_{\theta_d} V(\theta_d,\theta_g) = E_{x\sim P_{data}(x)}[log\ \ D(x;\theta_d)] + E_{z\sim P_z(z)}[log(1-D(G(z;\theta_g);\theta_d))] \tag{2}$$

While by fixing D, GAN algorithms minimize the value function to train generator G:

$$\min_{\theta_g} V(\theta_d,\theta_g) = E_{z\sim P_z(z)}[log(1-D(G(z;\theta_g);\theta_d))] \tag{3}$$

3.2.2 Research on the improvement of GAN loss function

Though GAN has achieved great success, there are still some problems. In fact, the minimizing value function $E_{x\sim P_g(x)}[1-log(D(x;\theta_d))]$ may not be able to provide enough gradient information for G to learning. At the beginning of training, when the discriminator D is trained very well, and the generator G is trained very poor, the generated data by the generator G is significantly different from the training data, and D rejects the generated samples with high confidence. At this time, the optimal discriminator of the GAN network $D_G^*(x)$ is:

$$D_G^*(x) = \frac{P_{data}(x)}{P_{data}(x)+P_g(x)} \tag{4}$$

Merging Eq. (4) into the Eq. (1), we obtain:

$$\begin{aligned}\min_{\theta_g} V(\theta_d,\theta_g) &= E_{x\sim P_{data}(x)}\left[log\left(\frac{P_{data}(x)}{\frac{1}{2}\cdot[P_{data}(x)+P_g(x)]}\right)\right] + E_{x\sim P_g(x)}\left[log\left(\frac{P_g(x)}{\frac{1}{2}\cdot[P_{data}(x)+P_g(x)]}\right)\right] - 2\,log\,2 \\ &= 2JS(P_{data}||P_g) - 2\,log\,2\end{aligned} \tag{5}$$

At this time, if $P_{data} \neq 0$, $P_g = 0$ or if the support sets of P_{data} and P_g is a low-dimensional manifold in a high-dimensional space, which will cause $JS(P_{data}||Pg) = log2$ to be constant [Arjovsky, Chintala and Bottou (2017)]. The loss of the generator has saturated, which is always constant, so that the gradient is always zero. Therefore, maximizing function $E_{x\sim P_g(x)}[log(D(x;\theta_d))]$ is better than the minimizing function $E_{x\sim P_g(x)}[1-log(D(x;\theta_d))]$ to train the generator G, which makes the dynamic fixed points of G and D the same, and in the early stage of training, the objective function can provide a more powerful gradient [Goodfellow (2016)]. Therefore, maximizing $E_{x\sim P_g(x)}[log(D(x;\theta_d))]$ is equivalent to minimizing $(P_g||P_{data}) - 2JS(P_{data}||P_g)$.

However, although the vanishing gradient problem is solved, the gradient of the training process is unstable, and the loss function after the replacement has a gradient of fluctuations. At the same time, the problem of insufficient diversity of generated samples is introduced, which is a mode collapse problem [Arjovsky, Chintala and Bottou (2017)].

Therefore, many scholars have proposed GAN variants such as LSGAN [Mao, Li, Xie et al. (2017)], WGAN [Arjovsky, Chintala and Bottou (2017)], WGAN-GP [Gulrajani, Ahmed, Arjovsky et al. (2017)], BEGAN [Berthelot, Schumm and Metz (2017)], etc. Some models have changed the original GAN network structure, but these models almost improve the objective function, we describe several examples as follow.

In the LSGAN model, LSGAN replaces the sigmoid cross-entropy loss of objective function with the least squares loss, which solve the problem of low quality and unstable training process to some extent, the objective function is improved to:

$$\begin{cases} \min\limits_{\theta_d} J^{(D)} = \frac{1}{2} E_{x\sim P_{data}(x)}[D(x;\theta_d) - b]^2 + \frac{1}{2} E_{z\sim P_z(z)}\left[D\left(G\left(z;\theta_g\right);\theta_d\right) - a\right]^2 \\ \min\limits_{\theta_g}\ J^{(G)} = \frac{1}{2} E_{z\sim P_z(z)}[D(G(z;\theta_g) - c;\theta_d)^2] \end{cases} \tag{6}$$

In the WGAN model, which can solve the problem of gradient disappearance to some extent, solves the problem of instability in training, and makes generated samples diverse. The improved objective function is as follow:

$$\begin{cases} \min\limits_{\theta_g} J^{(D)} = -E_{x\sim P_{data}(x)}[D(x;\theta_d)] + E_{z\sim P_Z(z)}[(1 - D(G(z;\theta_g);\theta_d))] \\ \min\limits_{\theta_d} J^{(G)} = -E_{x\sim P_g(x)}[D(x;\theta_d)] \end{cases} \tag{7}$$

In the BEGAN model, which can balance the abilities of the discriminator and proposes a hyper-parameter that can make a balance between the image diversity and the generation quality. The improved objective function in optimization of step t is as follows, for updating θ_D, θ_G, k_t separately :

$$\begin{cases} \min\limits_{\theta_D} J^{(D)} = E_{x\sim P_{data}(x)}[D_{AE}(x;\theta_d)] - k_t \cdot E_{z\sim P_Z(z)}\left[D_{AE}\left(G\left(z;\theta_g\right);\theta_d\right)\right] \\ \min\limits_{\theta_D} J^{(G)} = E_{z\sim P_Z(z)}\left[D\left(G\left(z;\theta_g\right);\theta_d\right)\right] \\ k_{t+1} = k_t + \lambda_k\left(\gamma E_{x\sim P_{data}(x)}[D_{AE}(x;\theta_d)] - E_{z\sim P_Z(z)}\left[D_{AE}\left(G\left(z;\theta_g\right);\theta_d\right)\right]\right) \end{cases} \tag{8}$$

In Conditional GAN, which add a conditional information, the new objective function is as follows:

$$\min_{\theta_g} \max_{\theta_d} V(\theta_d, \theta_g) = E_{x\sim P_{data}(x)}[logD(x|y;\theta_d)] + E_{z\sim P_z(z)}[log(1 - D(G(z|y;\theta_g);\theta_d))] \tag{9}$$

Of course, except for these examples about loss function, the researcher proposes more improvements for the loss function in other models. Therefore, we can carefully design the loss function of GAN and its variants to make better performance of GAN.

3.3 GAN's application

GAN no need to explicitly model any data distribution to generate samples, GAN generate samples through latent space *z*. Therefore, GAN has a wide range of applications in many academic and engineering fields such as image, text, speech, etc.

3.3.1 Images

In the field of images. Image translation is one of the applications of GAN. Image translation is that one image translates to another, which transform source domain X to target domain Y. After the image translation process, the source domain image content remains unchanged, some the style or the property of source domain image X translate to the target domain Y, such as the style migration using CycleGAN [Zhu, Park, Isola et al. (2017)]. Multi-domain image translation can generate multi-domain images using GAN. For example, a normal face image as the input of StarGAN [Choi, Choi, Kim et al. (2018)] to generate an image of angry, happy, fearful facial expressions. Super-resolution can convert a low-resolution image into a high-resolution image and reconstruct texture details. Such as SRGAN [Ledig, Theis, Huszár et al. (2017)] and ESRGAN [Wang, Yu, Wu et al. (2018)], the super-resolution medical images have a great auxiliary effect on the accurate judgment of doctors in clinical pathology. Target detection can extract the interesting target in the picture or video. For the guide system, all kinds of vehicles, pedestrians, traffic signs, and traffic lights are attention objects. Such as A-Fast-RCNN [Wang, Shrivastava and Gupta (2017)] can generate occlusion and deformation picture samples to train the detection network, and image recognition has achieved good results for A-Fast-RCNN. Image inpainting reconstruct the missing and damaged parts of the image and video. Such as Context Encoders [Pathak, Krahenbuhl and Donahue (2016)] uses the surrounding image information to infer missing images and fills in missing areas in the image. Video generation focuses on the prediction of the next frame image. Such as Dual Video Discriminator GAN (DVD-GAN) [Clark, Donahue and Simonyan (2019)] depend on the current frame pose and past pose features to predict the motion information of the next frame, which apply in video synthesis and video prediction tasks, it is current state-of-the-art techniques of GAN. Attention prediction is that when people look at a picture, they tend to focus on specific parts. Such as SalGAN [Pan, Sayrol, Nieto et al. (2017)] can predict the hotspot regions that human concerned, which not only has higher accuracy, but also contains regions that many models could not predict, it is possible to predict that whether the placed advertisement area is the hotspot of people's attention.

3.3.2 Speech and text

In the field of sequence data generation, GAN also has some applications. By introducing the policy-gradient algorithm in reinforcement learning into GAN, which can solve some sequence generation problems. For example, GANSynth [Engel, Agrawal, Chen et al. (2019)] can generate music, by independently controlling pitch and tone, to insert smoother sound between instruments, and finally generate high fidelity audio. In language generation for natural language processing, GAN can generate sentences and speech.

3.3.3 Other application

In the field of semi-supervised learning, GAN can automatically mark image data tags, learn and predictive tags. In the field of transfer learning, generators of GAN can transform source domain data features into target domain data features. In the field of image steganography, the generator of SSGAN [Shi, Dong, Wang et al. (2017)] generates a picture with steganographic information, and the discriminator can discriminate whether the picture has steganographic secret information. In the field of cryptography, CipherGAN [Gomez, Huang, Zhang et al. (2018)] can decipher Caesar cipher and Vignere cipher algorithms. In addition, there are some works in the fields of medical image segmentation and continual learning.

3.4 GAN's advantages and challenges

3.4.1 Advantages

By continuously developing mechanism of GAN and their variants, GAN includes some advantages as follows 1) GANs produce a sample in one shot and greatly reduce the runtime of algorithms. 2) Except for noise distribution, GAN have few prior assumptions. GAN haven't any hypothesis about data distribution. 3) The dimensionality of the latent space z isn't restricted. 4) GAN can generate any distributions if the discriminator fits perfectly. 5) GAN no need to pre-design the complex function model.

3.4.2 Challenges

To develop faster and better GAN, the future research of GAN mainly focuses on two directions: 1) How to improve the performance and theory of GAN. 2) How to find the killer application of GAN. Therefore, GAN combined with other algorithms has a more extensive prospect.

For the first research direction. We summarize GAN and their Variants in several aspect: How to improve the diversity and quality of generative samples, which need to establish suitable evaluative indicator involving diversity, accuracy, over-fitting degree, generated images visual quality, etc., to scientifically and comprehensively reflect the performance of GAN. How to solve the mode collapse problem and how to solve the problem of vanishing and unstable gradients. Although, there have some improvement method and made some breakthroughs, such as weights pruning [Arjovsky, Chintala and Bottou (2017)], weights regularization [Gulrajani, Ahmed, Arjovsky et al. (2017)], designed loss functions [Mao, Li, Xie et al. (2017)], further improvement is still necessary. How to design GAN to deal scenarios of discrete variables, so that the generator outputs discrete data. In fact, GAN works with reinforcement learning can deal with discrete variables by using policy gradient algorithms of reinforcement learning, it is still necessary to widen scope of application. Therefore, for solving these problems, we need to consider model architecture, the loss function, training techniques for specific model or task, such as one-sided label smoothing, Virtual batch normalization, balancing G and D, etc.

For the second research direction, how to apply GAN to solve more specific application problems in the specific science and engineering area or interdisciplinary area, such as transfer learning, reinforcement learning, semi-supervised field, domain adaptation, continual learning, Internet of Things (IoT), driverless, image, speech, text, medical image segmentation, steganography, network security, quantum physics, quantum chemistry, etc. The different fields generate complex attributes data. For example, GAN combines some method of IoT [Ren, Zhu, Sharma et al. (2020)] to generate richer IoT data when IoT data are not sufficient. Those is also a meaningful research work.

4 Research on QGAN

Recently, how to improve the performance of generative models through quantum computing, which has become a hot topic. For example, Benedetti et al. [Benedetti, Garcia-Pintos, Perdomo et al. (2019)] trained a shallow parameterized quantum circuit to generate Greenberger-Horne-Zeilinger (GHZ) states, thermal coherent states, and Bars and Stripes (BAS) images. The QGAN is a parameterized quantum-classical hybrid generative model, which is a new QML algorithm that can demonstrate quantum advantages on near-term quantum computers. QGAN's theory is similar to the classical GAN model, involving data types, algorithm models, objective functions, distance measure, gradient calculations, optimization processes, which combines classical GAN and quantum computing theory. Compared with the classical GAN, QGAN has the potential of exponential quantum speedups in some aspects, but the quantum advantage doesn't have sufficient theoretical and experimental support. QGAN's exponential advantage is still in the exploration stage, but QGAN have achieved quantum advantage by experiment in some tasks, such as learning and loading random distributions [Zoufal, Lucchi and Woerner (2019)]. This advantage is reflected in quantum gradient calculation [Schuld, Bergholm, Gogolin et al. (2019)], convergence process [Du, Hsieh and Tao (2019)]. When optimizing the QGAN model, we can use Adam [Benedetti, Grant, Wossnig et al. (2019); Kingma and Ba (2014)], Nelder-Mead [Peruzzo, McClean, Shadbolt et al. (2014)], etc. This section summarizes the QGAN model in recent year, including the theory, network architecture, objective functions, distance measure, optimization methods, Experiment and application.

4.1 The structure of QGAN

Inspired by GAN, quantum researchers propose the quantum version of QGAN. Lloyd et al. [Lloyd and Weedbrook (2018)] firstly propose the theory of QGAN based on the inherent probabilistic nature of quantum system. In QGAN model, N-dimensional vectors represented by $logN$ qubits, QGAN algorithms perform sparse and low-rank matrices in $O(poly(logN))$ time. When quantum generators generate very high-dimensional measurement statistics, the QGAN can exhibit quantum advantages over the classical GAN. Therefore, QGAN can converge faster or require fewer physical resources. Several adversarial learning situations of QGAN show as follows.

The first situation of QGAN is that data, discriminator, generator both are quantized. The real data is the ensembles of quantum states σ. The quantum generator generates the ensemble ρ that tries to match the real data σ, where σ and ρ are density matrices. The optimization process of QGAN is convex optimization, such as linear programming. At the beginning of QGAN training optimization, the generator is fixed, the positive-operator valued measure (POVM) strategy of discriminator D adjusted by gradient descent to finds the minimum error measurement by maximizing the function $p(T|\sigma) = Tr(T\sigma)$. Then, when the POVM strategy is fixed, the generator G adjusted by gradient descent to imitate real data σ as much as possible by maximizing function $p(T|\rho\ ) = Tr(T\rho)$. Finally, QGAN achieve "Nash equilibrium" point, the generator can accurately generate σ, the discriminator can't distinguish σ and ρ, and $p(T|\sigma) = p(T|\rho) = 1/2$. The Kullback-Leibler (KL) divergence can also measure the distance between ρ and σ. But there have some open questions: whether the convergence rate of the random gradient descent is $O(ploy(logN))$ time when searching for large quantum circuit $O(ploy(logN))$ parameters, Du et al. [Du, Hsieh and Tao (2019)] propose a solution.

The second situation of QGAN is that the data, discriminator is quantized, generator is classical, or the data is quantized, discriminator and generator is classical. For former, generator will generate a data distribution $p_g(x)$ to effectively match $p_{true}(x)$, it is a difficult problem due to generator G hasn't exponentially scaled resources. For latter, the discriminator of classical neural network still can't demonstrate quantum advantages under reasonable computational complexity assumptions due to generator of exponential resource costs. Therefore, QGAN's generator can't be designed a classical neural network.

The third situation of QGAN is that the data is classical, the discriminator D and generator G are quantized or the data and discriminator D is classical, the generator G is quantized. For former, if classical data is encoded as quantum data in $O(ploy(logN))$ time, which is first situation of QGAN. For latter, generators has exponential resource costs, which will demonstrate quantum advantages in some task, such as Situ et al. [Situ, He, Wang et al. (2018)] generates classical discrete data, Romero et al. [Romero and Aspuru-Guzik (2019)] generates classical continuous data, Zoufal et al. [Zoufal, Lucchi and Woerner (2019)] propose the learning and loading random distributions by using $O\ (poly\ (n))$ quantum gates. Compared with the current state-of-the-art complexity of $O\ (2^n)$ quantum gates [Plesch and Brukner (2011)], which greatly improves the efficiency of loading general probability distributions, and can combine with HHL algorithms [Harrow, Hassidim and Lloyd (2009)] or quantum amplitude estimation [Brassard, Hoyer, Mosca et al. (2002)] to promote quantum advantages, these algorithms are stable against small errors in input states.

In particular, the first situation of QGAN is the most studied. Such as, Du et al. [Du, Hsieh and Tao (2019)] proposes QMMW to complete the task of entanglement test, the computational complexity of which is $O(N^3T^4)$, which is proportional to the number of training rounds T and the input size Chakrabarti et al. [Chakrabarti, Huang, Li et al. (2019)] propose the qWGAN to improve the scalability and robustness of model, which define quantum Wasserstein semi-metrics (metrics without triangle inequalities) and design specific quantum circuits of loss function.

4.1.1 QGAN's structure: quantum generator, quantum discriminator, quantum data

Dallaire-Demers et al. [Dallaire-Demers and Killoran (2018)] propose conditional QGAN, which inspired by conditional GAN [Mirza and Osindero (2014)]. Firstly, in this model, the generator $U_G(\vec{\theta}_G)$ parameterized by a vector $\vec{\theta}_G$, the label |λ> and the additional quantum noise state |*z*> as inputs of generator, then generator output $G(\vec{\theta}_G, |\lambda, z>) = U_G(\vec{\theta}_G)\rho_\lambda^0 U_G^\dagger(\vec{\theta}_G) = \rho_\lambda^G(\vec{\theta}_G, z)$. If randomly changed unstructured *z*, for each λ, generator will produce a different quantum state, but if one control structured *z*, you can change the properties of the generated quantum data, and these properties aren't captured by the tag λ. Then, discriminator $U_D(\vec{\theta}_D)$ parameterized by a vector $\vec{\theta}_D$. If the ture state $R(|\lambda>) = U_R\rho_\lambda^0 U_R^\dagger = \rho_\lambda^R$ is discriminated, the discriminator output $U_D(\vec{\theta}_D)\rho_\lambda^G(\vec{\theta}_G, z)U_D^\dagger(\vec{\theta}_D) = \rho_\lambda^{DG}(\vec{\theta}_D, \vec{\theta}_G, z)$. If the fake state $\rho_\lambda^G(\vec{\theta}_G, z)$ is discriminated, the discriminator output $U_D(\vec{\theta}_D)\rho_\lambda^G(\vec{\theta}_G, z)U_D^\dagger(\vec{\theta}_D) = \rho_\lambda^{DG}(\vec{\theta}_D, \vec{\theta}_G, z)$. For discriminator D, if quantum state comes from ρ_λ^R, then $<\sigma_Z>$*out* output |*real*>, otherwise it outputs |*fake*>. For generator G, G fool D to make D output |real>. The expected value $<\sigma_Z>$*out* is proportional to the probability $\frac{<\sigma_Z>+1}{2}$, where $\sigma_Z \equiv$|*real*><*real*| − |*fake*><*fake*|. This process be described using Fig. 5:

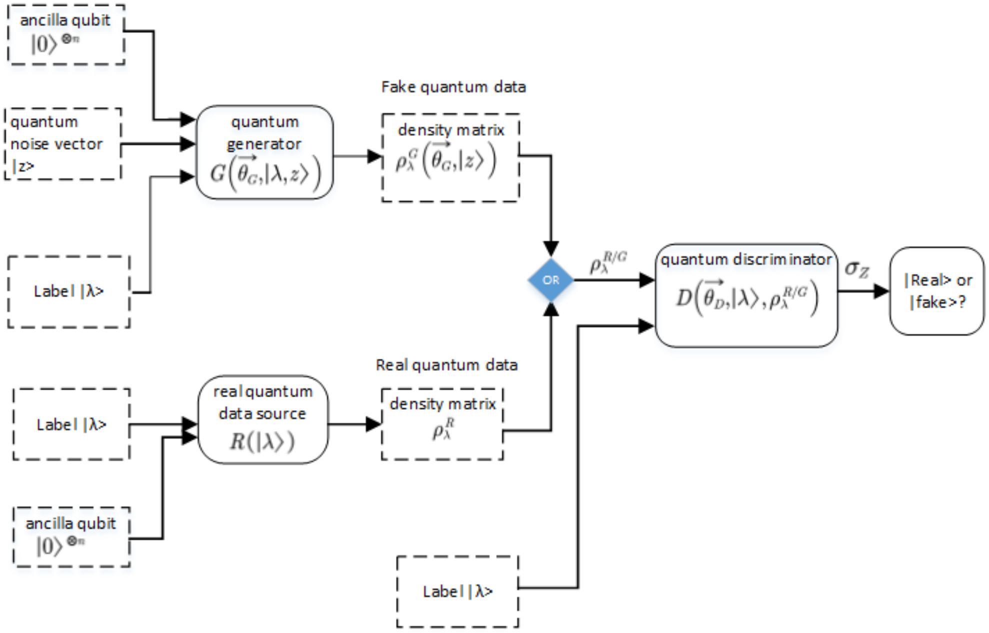
ancilla qubit $|0\rangle^{\otimes n}$
quantum noise vector |z>
Label |λ>
quantum generator $G\left(\overrightarrow{\theta_G},|\lambda,z\rangle\right)$
Fake quantum data
density matrix $\rho_\lambda^G\left(\overrightarrow{\theta_G},|z\rangle\right)$
OR
$\rho_\lambda^{R/G}$
quantum discriminator $D\left(\overrightarrow{\theta_D},|\lambda\rangle,\rho_\lambda^{R/G}\right)$
σ_Z
|Real> or |fake>?
Real quantum data
density matrix ρ_λ^R
Label |λ>
real quantum data source $R(|\lambda\rangle)$
ancilla qubit $|0\rangle^{\otimes n}$
Label |λ>

Figure 5: Structure of conditional QGAN

Conditional QGAN can implemented semi-supervised learning as long as the number of conditional tags increase to $\Lambda+1$ and some generated data samples are provided.

For unconditional information QGAN, Hu et al. [Hu, Wu, Cai et al. (2019)] propose QGAN, the fake data is generated by a superconducting quantum circuit, procedure of optimizing GAN using gradient descent algorithm. The structure of model be described using in Fig. 6:

The generator G randomly performs $\{U(\theta,\varphi)|g\rangle, U(\pi-\theta,\varphi+\pi)|g\rangle\}$ by selecting random probability $\{r,1-r\}$ to generate quantum data $\rho(r,\theta,\varphi)$, where $U(\theta,\varphi)=e^{i\varphi\sigma_z/2}e^{i\theta\sigma_x/2}$. The discriminator D discriminates the real quantum data by $p_\sigma=\mathrm{tr}\mathcal{M}\sigma$ and the generated quantum data by $p_\rho=\mathrm{tr}\mathcal{M}\rho$, where $\mathcal{M}=U^\dagger(\beta,\gamma)|g><g|U(\beta,\gamma)$, $|g>$ is the ground state.

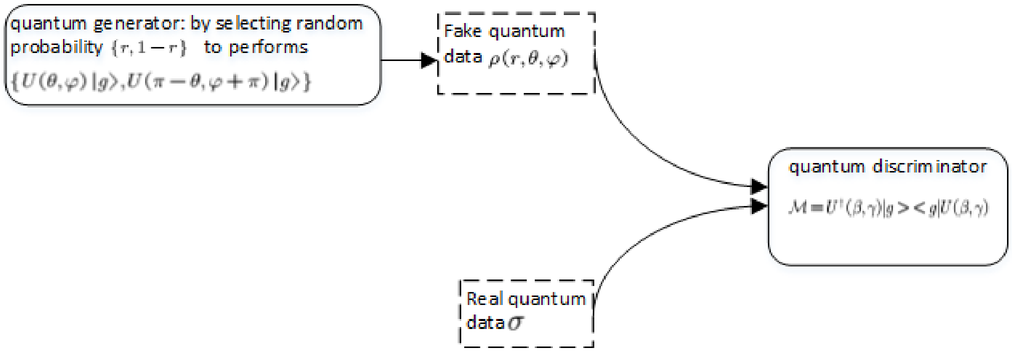
quantum generator: by selecting random probability $\{r,1-r\}$ to performs $\{U(\theta,\varphi)|g\rangle, U(\pi-\theta,\varphi+\pi)|g\rangle\}$
Fake quantum data $\rho(r,\theta,\varphi)$
quantum discriminator $\mathcal{M}=U^\dagger(\beta,\gamma)|g><g|U(\beta,\gamma)$
Real quantum data σ

Figure 6: Structure of QGAN

For learning all information of quantum states, including phase information. Benedetti et al. [Benedetti, Grant, Wossnig et al. (2019)] propose QGAN to learn pure state approximation. By constructing a generator circuit G generate a wave function $|\psi_g\rangle$ to imitate the state $|\psi_t\rangle$ generated by the target circuit as similar as possible, which define by $\min_{\theta_l}\frac{1}{2}\||\psi_g\rangle\langle\psi_g| - |\psi_t\rangle\langle\psi_t|\|$. Then, D discriminate the generated state of quantum circuit T and G by measuring the POVM of the auxiliary bits. Similar to QGAN in a superconducting quantum circuit [Hu, Wu, Cai et al. (2019)]. The discriminator selects the target state $|\psi_t\rangle$ and the generated state $|\psi_g\rangle$ with probability $P(t)$, $P(g)$ to perform discrimination. Implementing such QGAN structure be described using in Fig. 7:

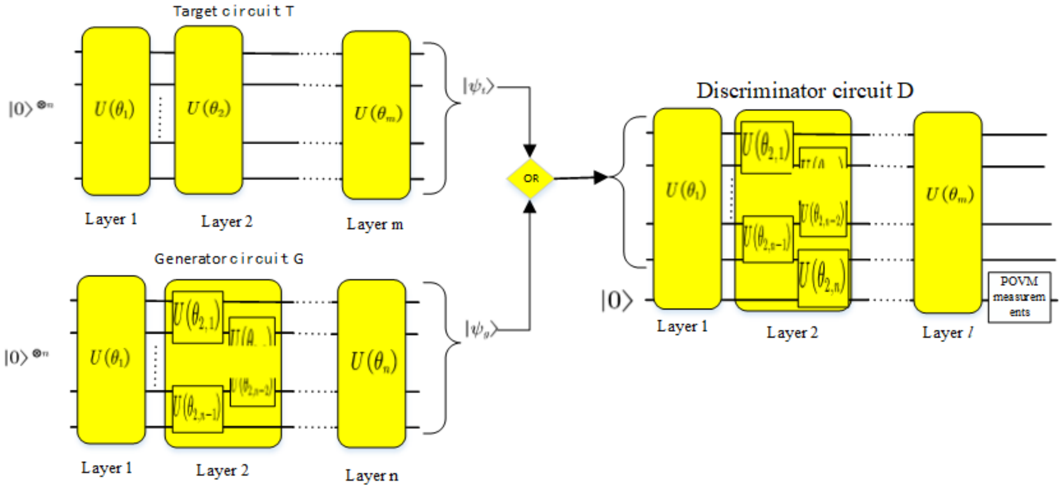

Figure 7: Structure of QGAN to approximate pure state

Researchers wish to generate higher dimensional separable state. Du et al. [Du, Hsieh and Tao (2019)] propose QMMW, the convergence rate of algorithm is $O\left(\sqrt{\frac{N}{T}}\right)$. The QMMW overcomes the non-convergent problem that is the "local minima" trap, and which can seamlessly embed into other optimization methods used in QGAN. For entanglement test tasks. Firstly, the entanglement test standard need to be designed by using the "constraint" step when updating QGAN. It is necessary to ensure that the no-regret attribute isn't destroyed, and it can be effectively implemented by quantum operations. Then, the separability rule of QGAN based on the multiplicative weight training method is reflected by the training loss. If after a certain number of training rounds T, the training loss can't converge to the Nash equilibrium state below the threshold, the given state is the entangled state. At this time, by adjusting the structure of the quantum circuit, the expression ability of the quantum generator circuit is limited, so that the generator only generates a separable state.

There are some work that generating mixed states and pure states. Chakrabarti et al. [Chakrabarti, Huang, Li et al. (2019)] propose qWGAN based on quantum Wasserstein semi-metric, and design a regularized quantum Wasserstein GAN, so that the loss function and gradient estimation can be effectively implemented on quantum processing unit (QPU) and central processing unit (CPU), as shown in Fig. 8.

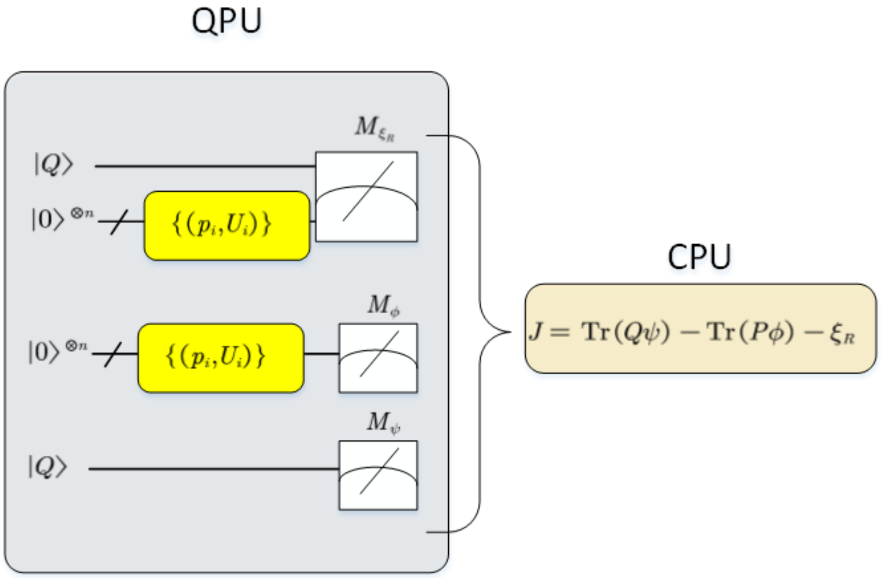

Figure 8: Regularized quantum Wasserstein GAN

where, $\{(p_i, U_i)\}$ refers to a parameterized generator and ϕ, ψ, ξ_R refers to a discriminator, $\xi_R = \frac{\lambda}{e}\mathrm{Tr}(exp(\frac{log\ (P\otimes Q) - C - \phi\otimes \mathrm{I}_\mathrm{Y} + \mathrm{I}_\mathrm{X}\otimes\psi}{\lambda}))$. The Fig. 8 shows how to evaluate the objective function J by post-processing measurements ϕ, ψ, ξ_R on the generated state P and the real state Q, and how to calculate J on the CPU, where the diagonal element of the density operators P, Q represents a classical probability distribution.

4.1.2 QGAN's structure: quantum generator, classical discriminator, classical data

Classical GAN can't work in discrete scenarios. Situ et al. [Situ, He, Wang et al. (2018)] propose QGAN to generate classical discrete data, such as BAS dataset. There are many types of generator structures. Such as, the first type generator is that each layer $U(\theta_i^l)$ consists of a single-qubit rotated gate and a controlled phase gate, the total parameter is $5NL$, N is the number of qubits, L is the number of layers of the circuit. The scend type generator is the matrix product state (MPS) quantum circuit [Huggins, Patil, Mitchell et al. (2019)]. For each node, input and output is $V + 1$ bits, measuring one qubit after each node, generator output a samples x by measuring N bits. Each node contains L≥1 layers, the layout structure of each layer is same as the first type generator. The total parameter of the MPS quantum circuit is $5NL(V + 1)$.

When measuring the quantum state generated by generator, which produce classical discrete data. Since the discriminator D is a shallow feedforward neural network, the dimension of the input layer is same as the sample dimension, the input layer accepts the real sample and the generated sample. Discriminator has one hidden layer, and output 0 or 1. Detailed architecture of QGAN be described using Fig. 9.

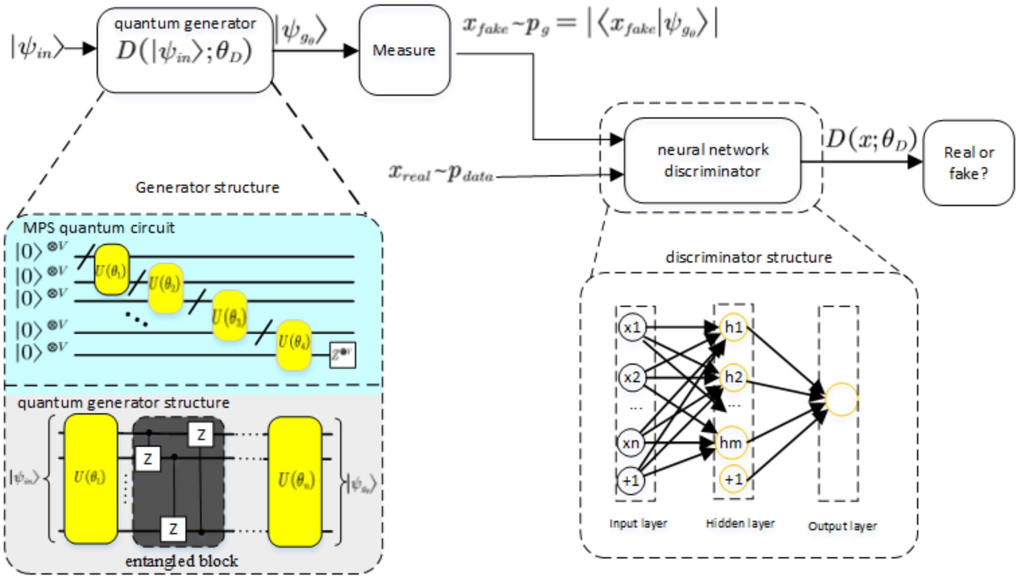

Figure 9: Structure of QGAN to generate discrete data

According to architecture of Fig. 9, Zeng et al. [Zeng, Wu, Liu et al. (2019)] designed a QGAN to learn BAS dataset and infer missing data, the generator of QGAN is a quantum circuit born machines (QCBM) [Liu and Wang (2018)], and structure of QCBM is similar to the first type generator. Generator G contains a single qubit rotated layer and two qubit entangled layers. The single qubit rotated layer is $U(\theta_{i,j}^{l}) = R_z(\theta_{i,1}^{l})R_x(\theta_{i,2}^{l})R_z(\theta_{i,3}^{l})$, where $l \in (0, L)$ represents the layer index, $i \in (0, N-1)$ represents the qubit index. The two-qubit entanglement layer is *C-NOT* gate, the last layer be a single-qubit rotation layer. The generated quantum pure state express the classical probability distribution $|\psi\rangle = \sum_x \sqrt{p(x)}\,|x>$, which is implicit, we obtain data from projection measurements sampling. The total parameter is $(3L+1)N$, where L is the maximum circuit depth. The discriminator has two hidden layers, each with 64 leaky ReLU [Maas, Hannun and Ng (2013)] activation functions, the output layer is the sigmoid function, and outputs 0, 1. For infering missing data, using the Amplitude Amplification algorithm [Brassard, Hoyer, Mosca et al. (2002)] can quickly infer the unobserved value q based on a part of the observed value e.

According to architecture of Fig. 9, Zoufal et al. [Zoufal, Lucchi and Woerner (2019)] propose QGAN to learn and load random distributions. In model, the structure of the n-qubit k-layers generator is that each layer consists of a single-qubit rotated gate RY and a controlled phase gate. Generator input $|\psi_{in}\rangle$ and the circuit depth k has a key impact on the complexity of the quantum generator. $|\psi_{in}\rangle$ is the discrete uniform distribution, normal or random distributions. $|\psi_{g_\theta}\rangle$ is the log-normal distribution, triangular distribution, and bimodal distribution. The Kolmogorov-Smirnov test and the relative entropy test can determine whether the generated distribution matches the target distribution. The structure of discriminator is 50 input nodes, 20 hidden nodes, and one output node, hidden node uses the Leaky ReLU activation function and the output node uses the sigmoid function.

4.1.3 QGAN's structure: quantum generator, quantum discriminator, classical data

How to generate continuous distributions by using QGAN, such as image and sound. Romero et al. [Romero and Aspuru-Guzik (2019)] firstly propose QGAN to generate continuous distributions. In model, the variational quantum generator (VQG) of QGAN framework comprises two quantum circuit components. The first component of VQG is quantum encoder that encode a classical random variable z from the latent space into a quantum state $R(z)|0^{\otimes r}\rangle = |\phi(z)\rangle$, encoding scheme includes amplitude encoding, variational encoding, etc. The second component of VQG simulate the target classical probability distribution, VQC generate the state $G(\Theta_g)|\phi(x)\rangle = |\psi(z,\Theta_g)\rangle$, and then by using the measurement decoding scheme obtain a set of vectors $P = [\langle P_1\rangle, \langle P_2\rangle, \cdots, \langle P_M\rangle]$, $\langle P_i\rangle$ is a expected value that measurement operators P_{i} measure state $|\psi(z,\Theta_g)\rangle$, then P as input of the classical post-processing function $f_g(P;\Omega_g)$, so that $f_g(x) = h(W \cdot P + b)$, and generating M-dimensional continuous data $x_{Fake} \sim f_g(x)$. The discriminator of QGAN is a classical neural network or a VQC. Finally, the optimization process of the model realized by using an automatic differentiation tool, which implement on near-term quantum computer. Detailed architecture of QGAN be described using Fig. 10.

Therefore, the model structure of QGAN has a great impact on the complexity and performance of the algorithm, such as Hierarchical structure [Grant, Benedetti, Cao et al. (2018)] and Universal topology structure [Chen, Wossnig, Severini et al. (2018)]. It is very important to study the model structure of QGAN.

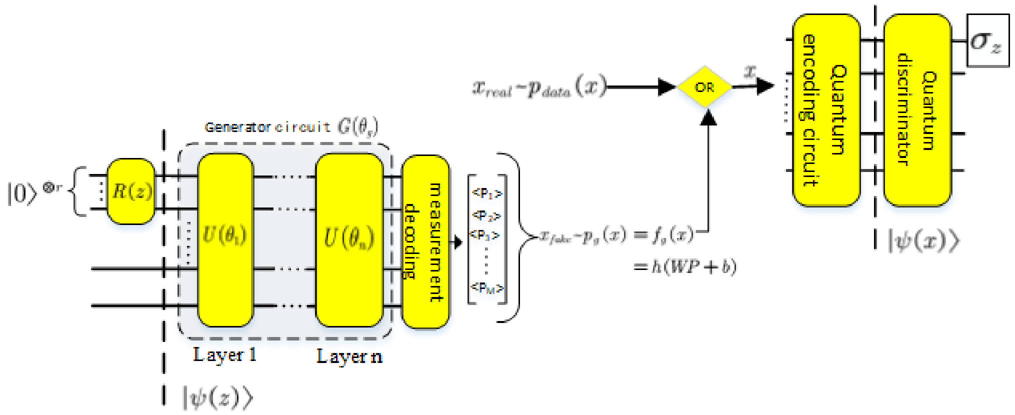

Figure 10: Structure of QGAN to generate continuous distributions

4.2 The objective function of QGAN

QGAN objective function is designed by measuring the distance between two distributions, such as Jensen-Shannon (JS) divergence, Total Variance and Wasserstein distance, etc. Such as, JS divergence applied in Situ et al. [Situ, He, Wang et al. (2018); Zeng, Wu, Liu et al. (2019); Zoufal, Lucchi and Woerner (2019); Romero and Aspuru-Guzik (2019)]. Total Variance applied in Dallaire-Demers et al. [Dallaire-Demers and Killoran (2018); Hu, Wu, Cai et al. (2019); Benedetti, Grant, Wossnig et al. (2019); Du, Hsieh and Tao (2019)]. Wasserstein distance applied in Chakrabarti et al. [Chakrabarti, Huang, Li et al. (2019)]. There existing many ways to measure the two distributions, different distance measure can derive different objective functions, which have different properties. The objective function of QGAN is summarized in detail below, the summary is shown in Tab. 1:

Table 1: a summary of QGAN objective function

QQGAN algorithm	data type	G	D	objective functions
Conditional QGAN [Dallaire-Demers and Killoran (2018)]	quantum	Yes	Yes	$\frac{1}{2}+\frac{1}{4\Lambda}\sum_{\lambda=1}^{\Lambda}(tr(Z\rho_{\lambda}^{DR})-tr(Z\rho_{\lambda}^{DG})))$
QGAN [Situ, He, Wang et al. (2018); Zeng, Wu, Liu et al. (2019); Zoufal, Lucchi and Woerner (2019)]	classical	No	Yes	$\frac{1}{2}E_{x\sim p_{data}(x)}[logD(x;\phi_D)]$ $+\frac{1}{2}E_{x\sim P_g(x)}[log(1-D(x;\phi_D))$
QGAN [Romero and Aspuru-Guzik (2019)] for continuous distributions	classical continuous data	Yes	Yes/No	$\frac{1}{2}E_{x\sim P_{data}(x)}[logD(x;\theta_d)]$ $+\frac{1}{2}E_{z\sim P_z(z)}[log(1-D(G(z;\theta_g);\theta_d))$
QGAN [Benedetti, Grant, Wossnig et al. (2019)] for pure state approximation.	quantum	Yes	Yes	$\frac{1}{2}(tr(Z\rho^{DR})-tr(Z\rho^{DG}))$
QMMW [Du, Hsieh and Tao (2019)]	quantum	Yes	Yes	$\frac{1}{2}\left(Tr\left(\sigma_D^{(t)}\rho\right)-Tr\left(\sigma_D^{(t)}\sigma_G^{(t)}\right)\right)+\frac{1}{2}$
qWGAN [Chakrabarti, Huang, Li et al. (2019)]	quantum	Yes	Yes	$Tr(Q\psi)-Tr(P\phi)-\xi_R$

In Tab. 1, trace distance [Nielsen and Chuang (2010)] $d=\frac{1}{2}|p_\rho-p_\sigma|_1$ corresponds to the total variation distance. The quantum Wasserstein semimetric [Chakrabarti, Huang, Li et al. (2019)] corresponds to wasserstein distance.

In addition, the fidelity $F(\sigma,\rho)=tr\sqrt{\sqrt{\sigma}\rho\sqrt{\sigma}}$ [Nielsen and Chuang (2010)] also can measure the distance between ρ and σ.

4.2.1 Quantum gradient of QGAN

In order to obtain gradient information in general quantum circuits. Supposing that Quantum circuits are specified by a sequence of gates $U(\vec{\theta})=U_N(\theta_N)U_{N-1}(\theta_{N-1})\cdots U_i(\theta_i)\cdots U_1(\theta_1)$. If $U(\vec{\theta})$ operates on ρ_0 and then measure with observable operator *P*, the expected value of observable operator *P* is Eq. (10):

$$\langle P(\vec{\theta})\rangle=tr(\rho_0 U^{\dagger}(\vec{\theta})PU(\vec{\theta})) \qquad (10)$$

Then, the derivative of function with respect to a parameter θ_j is:

$$\frac{\partial}{\partial\theta_j}\langle P(\vec{\theta})\rangle = -\frac{i}{2}tr(\rho_0 U^{\dagger}_{1:j}[U^{\dagger}_{j+1:N}PU_{N:j+1}, h_j]U_{j:1}) \tag{11}$$

where the commutator *[A, B] = [AB-BA]*.

In order to obtain gradient information. We can design a quantum circuit that calculates the derivative of function with respect to a parameter θ_j or directly use analytic gradient [Schuld, Bergholm, Gogolin et al. (2019)]. For former, the gradient estimate of each parameter make multiple measurements by observable operator *Z*, and then calculated result by summation average. For latter, time complexity of the quantum gradient calculation is *O (1)*.

4.2.2 The objective function of QGAN based on total variance

In more detail speaking, for Conditional QGAN [Dallaire-Demers and Killoran (2018)], the optimization goal is:

$$\begin{aligned}\min_{\vec{\theta}_G}\max_{\vec{\theta}_D} V\left(\vec{\theta}_G,\vec{\theta}_D\right) &= \frac{1}{\Lambda}\sum_{\lambda=1}^{\Lambda} Pr\left(\left(D\left(\vec{\theta}_D, |\lambda>, R(|\lambda>)\right) = |real>\right)\cap\left(D\left(\vec{\theta}_D, |\lambda>, G\left(\vec{\theta}_G, |\lambda, z>\right)\right) = |fake>\right)\right)\\ &= \frac{1}{2}+\frac{1}{2\Lambda}\sum_{\lambda=1}^{\Lambda}(cos^2(\phi)tr(Z\rho_\lambda^{DR}(\vec{\theta}_D)) - sin^2(\phi)tr(Z\rho_\lambda^{DG}(\vec{\theta}_D,\vec{\theta}_G,z)))\end{aligned} \tag{12}$$

If $\phi=\pi/4$, showing that the QGAN framework fairly chooses to discriminate between the real data source R and generating the data source $G\left(\vec{\theta}_G\right)$. In order to optimize the value function. Therefore, we need to calculate the gradient of the discriminator, as follow:

$$\frac{\partial}{\partial\theta_{Dj}}V(\vec{\theta}_D,\vec{\theta}_G) = -\frac{i}{8\Lambda}\sum_{\lambda=1}^{\Lambda} tr\,((\rho_\lambda^R - \rho_\lambda^G(\vec{\theta}_G, z))U^{\dagger}_{D,1:j}[U^{\dagger}_{D,j+1:N_D}ZU_{D,N_D:j+1}, h_j^D]U_{D,j:1}) \tag{13}$$

The gradient of the generator is:

$$\frac{\partial}{\partial\theta_{Gj}}V(\vec{\theta}_D,\vec{\theta}_G) = \frac{i}{8\Lambda}\sum_{\lambda=1}^{\Lambda} tr\,(\rho_\lambda^0(z)U^{\dagger}_{G,1:j}[U^{\dagger}_{G,j+1:N_G}U^{\dagger}_D(\vec{\theta}_D)ZU_D(\vec{\theta}_D)U_{G,N_G:j+1}, h_j^G]U_{G,j:1}) \tag{14}$$

Next, we need to use the gradient descent method to optimize the parameter of discriminator and generator.

Hu et al. [Hu, Wu, Cai et al. (2019)] propose to measure the distance between two density matrix by the trace distance and fidelity. In this QGAN architecture, the algorithm updates the discriminator D by maximizing the trace distance, the parameters β,γ is updated. By minimizing the trace distance $d = \frac{1}{2}|p_\rho - p_\sigma|_1$ to update the generator G, the parameters r,θ,φ is updated, where the d is calculated by measuring the quantum states ρ and σ 10,000 times. However, when optimizing the QGAN, the gradient information is obtained by digital estimation of the classical average, which isn't accurate value, so the convergence process is not fast with the quantum data dimension increasing.

Benedetti et al. [Benedetti, Grant, Wossnig et al. (2019)] define the objective function of the discriminator and generator is as follows:

$$\min_{\theta}\max_{\varphi} V(\theta,\varphi) = tr[E_0 D(|\psi_t\rangle\langle\psi_t| \otimes |0\rangle\langle 0|)D^{\dagger}]P(t) - tr[E_0 D(G|0\rangle\langle 0|G^{\dagger} \otimes |0\rangle\langle 0|)D^{\dagger}]P(g) \tag{15}$$

According to objective function. The gradient of the function $V(\theta,\varphi)$ with respect to a parameter θ_l is:

$$\frac{\partial V(\theta,\varphi)}{\partial \theta_l} = -\frac{P(g)}{2}\{tr\left[E_0 D(G_{l+}|0\rangle\langle 0|G_{l+}^{\dagger} \otimes |0\rangle\langle 0|)D^{\dagger}\right] - tr[E_0 D(G_{l-}|0\rangle\langle 0|G_{l-}^{\dagger} \otimes |0\rangle\langle 0|)D^{\dagger}]\} \quad (16)$$

where, $G_{l\pm} = G_L \cdots G_{l+1} G_l(\theta_l \pm \pi/2) G_{l-1} \cdots G_1$.

The gradient of the function $V(\theta,\varphi)$ with respect to a parameter φ_k is

$$\frac{\partial V(\theta,\varphi)}{\partial \phi_k} = \frac{P(t)}{2}\{tr[E_0 D_{k+}(|\psi_t\rangle\langle\psi_t| \otimes |0\rangle\langle 0|)D_{k+}^{\dagger}] - tr[E_0 D_{k-}(|\psi_t\rangle\langle\psi_t| \otimes |0\rangle\langle 0|)D_{k-}^{\dagger}]\} - \frac{P(g)}{2}\{tr[E_0 D_{k+}(G|0\rangle\langle 0|G^{\dagger} \otimes |0\rangle\langle 0|)D_{k+}^{\dagger}] - tr[E_0 D_{k-}(G|0\rangle\langle 0|G^{\dagger} \otimes |0\rangle\langle 0|)D_{k-}^{\dagger}]\} \quad (17)$$

where, $D_{k\pm} = D_K \cdots D_{k+1} D_k(\phi_k \pm \pi/2) D_{k-1} \cdots D_1$, which is similar to analytic gradient calculation [Schuld, Bergholm, Gogolin et al. (2019)].

Next, by calculated gradient, the parameters of the generator and discriminator can be adjusted using Newton's iteration method, iRprop$^-$ algorithm [Riedmiller and Braun (1993)], heuristic stopping criterion algorithm or swap test method [Buhrman, Cleve, Watrous et al. (2001)], until QGAN achieve convergence. Where heuristic stopping criterion uses bipartite entanglement entropy (BEE) to quantify the entanglement between the main and auxiliary registers and using the scaled direct inversion (SDI) method [Schmied (2016)] to estimate BEE.

Du et al. [Du, Hsieh and Tao (2019)] propose QMMW, which defines the objective function in step t:

$$J(\sigma_G^{(t)}, \sigma_D^{(t)}) = \frac{1}{2}(Tr(\sigma_D^{(t)}\rho) - Tr(\sigma_D^{(t)}\sigma_G^{(t)})) + \frac{1}{2} \quad (18)$$

In this model, the gradient can be obtained by analytic gradient [Schuld, Bergholm, Gogolin et al. (2019)].The multiplicative weight training method is used to train and optimize QGAN model, which no need optimize four hyper-parameters, the total number of training rounds T, the number of internal iterations K, the learning rate α, and the scaling parameter η. When updating the discriminator, the QMMW algorithm will assign more weight to fool the discriminator.

4.2.3 The objective function of QGAN based on JS divergence

Situ et al. [Situ, He, Wang et al. (2018)] propose to generate classical discrete data, the QGAN optimization goal is:

$$\min_{w_d} \max_{\vec{\theta}} V(w_d, \vec{\theta}) = \frac{1}{2}E_{x\sim p_{data}(x)}[logD(x; w_d)] + \frac{1}{2}E_{x\sim P_g(x)}[log(1 - D(x; w_d)) \quad (19)$$

According to optimization goal. A binary cross entropy loss function for discriminator to calculate the average loss is:

$$J(x,y) = -\frac{1}{2*n_d}\sum_i y_i \, log \;\, D(x_i) + (1 - y_i)log\,(1 - D(x_i)) \quad (20)$$

where, n_d is the number of samples for discriminator in one mini-batch, $(x_i, y_i) \in (X, Y)$ represents the *i-th* sample x_i and Label y_i. If y_i=$D(x_i)$=1, then x_i is true data, otherwise x_i is fake data.

When the generator uses the non-saturating loss function based on optimization goal, the derivative of function $J(x,y)$ with respect to a generator parameter θ_i is:

$$\frac{\partial J(x,y)}{\partial \theta_i} = -\sum_{x\in\{0,1\}^N} log\, D(x)\frac{\partial P_{\vec{\theta}}(x)}{\partial \theta_i} \quad (21)$$

where $P_{\vec{\theta}}(x)$ is the probability distributions of obtaining the measurement result x from the $\vec{\theta}$ parameterized quantum circuit generator. Since $P_{\vec{\theta}}(x)$ is continuous and differentiable, then:

$$\frac{\partial P_{\vec{\theta}}(x)}{\partial \theta_i} = \frac{1}{2}\left(P_{\vec{\theta}^{+}}(x) - P_{\vec{\theta}^{-}}(x)\right) \tag{22}$$

where $\vec{\theta}^{\pm} = \vec{\theta} \pm \frac{\pi}{2}\mathbf{e}^{i}$, $\mathbf{e}^{i}$ is the *i-th* unit vector in the parameter space, and $\theta_i^{\pm} \leftarrow \theta_i \pm \frac{\pi}{2}$, the angles of other parameters are unchanged, which is similar to analytic gradient calculation [Schuld, Bergholm, Gogolin et al. (2019)]. So

$$\begin{aligned}\frac{\partial J(x,y)}{\partial \theta_i} &= -\frac{1}{2}\sum_{x\in\{0,1\}^N}\left(\boldsymbol{P}_{\vec{\boldsymbol{\theta}}^{+}}(x) - P_{\vec{\boldsymbol{\theta}}^{-}}(x)\right) log\, D(x) \\ &= \frac{1}{2}\sum_{x\in\{0,1\}^N} P_{\vec{\theta}^{-}}(x)\, log\, D(x) - \frac{1}{2}\sum_{x\in\{0,1\}^N} P_{\vec{\theta}^{+}}(x)\, log\, D(x) \\ &= \frac{1}{2}\mathbb{E}_{x\sim P_{\vec{\theta}^{-}}(x)}\, log\, D(x) - \frac{1}{2}\mathbb{E}_{x\sim P_{\vec{\theta}^{+}}(x)}\, log\, D(x)\end{aligned} \tag{23}$$

Then, the training optimization uses the gradient descent method, until QGAN achieve convergence.

Zeng et al. [Zeng, Wu, Liu et al. (2019)] based on QCBM [Liu and Wang (2018)] generator, the value function of the discriminator and generator is as follows:

$$\min_{\theta_G}\max_{\phi_D} V(\phi_D, \theta_G) = E_{x\sim p_{data}(x)}[lnD(x;\phi_D)] + E_{x\sim P_g(x)}[ln(1 - D(x;\phi_D)) \tag{24}$$

The generator uses a non-saturating heuristic loss function $J^G(\phi_D, \theta_G) = -\mathrm{E}_{x\sim P_g(x)}[ln\ D_{\boldsymbol{\phi}}(x)]$ to solve the vanishing gradient problem. Then, minimizing the loss function of discriminator and generator, and calculate the gradient information, which is similar to calculation of Situ et al. [Situ, He, Wang et al. (2018)]. After calculating the gradient, the optimization algorithm is used to update the parameters, which can use the classical optimization algorithm Adam [Kingma and Ba (2014)], Stochastic gradient descent (SGD), Simultaneous perturbation stochastic approximation (SPSA), etc. The optimization strategy is similar to the VQE [Peruzzo, McClean, Shadbolt et al. (2014)] and QAOA [Farhi, Goldstone and Gutmann (2014)] algorithms. Meanwhile, QGAN generate continuous distributions [Romero and Aspuru-Guzik (2019)] and a QGAN for learning and loading random distributions [Zoufal, Lucchi and Woerner (2019)] whose objective function, gradient calculation and optimization process are the same as [Liu and Wang (2018)] on the QCBM generator and Situ et al. [Situ, He, Wang et al. (2018)].

4.2.4 The objective function of QGAN based on Wasserstein distance

Chakrabarti et al. [Chakrabarti, Huang, Li et al. (2019)] use advantage of the continuity, smoothness and robustness of Wasserstein distance, to propose a quantum Wasserstein semi-metric to measure the distance between two distributions. It will solve the problem of the KL divergence, JS divergence, total variance divergence and other distance measurement methods are not sensitive to the QGAN model, and solve training problems as quantum systems expand.

Let qW(P, Q) denote the quantum Wasserstein semimetric between $P \in \mathrm{D(X)}$, $Q \in \mathrm{D(Y)}$. $qW(.,.)$ forms a semimetric over the set of density matrices *D(X)* over any space *X*. Quantum Wasserstein semimetric is defined by: $\mathrm{qW}(\mathcal{P}, \mathcal{Q}) = \min_{\pi} \mathrm{Tr}(\pi C)$ subject to $\mathrm{Tr_Y}(\pi) = P \in \mathrm{D(X)}, \mathrm{Tr_X}(\pi) = Q \in \mathrm{D(Y)}, \pi \in \mathrm{D(X \otimes Y)}$, where C is a matrix over $X \otimes Y$ that should refer to some cost-type function.

The Wasserstein distance metric adding the regular term is:

$$qW(P,Q) = \min_{\pi} Tr(\pi C) + \lambda Tr(\pi log\,(\pi) - \pi log\,(P \otimes Q)) \tag{25}$$

where, $Tr_{\mathrm{Y}}(\pi) = P \in \mathrm{D(X)}$, $\mathrm{Tr_X}(\pi) = Q \in \mathrm{D(Y)}$, $\pi \in \mathrm{D(X \otimes Y)}, C = \frac{1}{2}(\mathrm{I_{X \otimes Y}} - \mathrm{SWAP})$. *qW (P, Q)* doesn't satisfy the triangle inequality, C should be some kind of cost function. In order to optimize more efficiently, the dual problem transformed from this SDP problem about objective function is as follow:

$$qW(P,Q) = \max_{\phi,\psi} Tr(Q\psi) - Tr(P\phi) - \xi_R \tag{26}$$

where, $\mathrm{I_X} \otimes \psi - \phi \otimes \mathrm{I_Y} \preceq C, \phi \in \mathrm{H(X)}, \psi \in \mathrm{H(Y)}$,H (X), H (Y) represent the set of Hermitian matrices on spaces X and Y, $\xi_R = \frac{\lambda}{e}\mathrm{Tr}(exp(\frac{log\,(P \otimes Q) - C - \phi \otimes \mathrm{I_Y} + \mathrm{I_X} \otimes \psi}{\lambda}))$, objective function $J = \mathrm{Tr}(Q\psi) - \mathrm{Tr}(P\phi) - \xi_R$.

According to objective function, the gradient of the loss function *J* with respect to the generator parameters p_i and $\theta_{i,j}$ is:

$$\begin{cases} \frac{\partial J}{\partial p_i} = -Tr[U_i \vec{e}_0 \vec{e}_0^{\dagger} U_i^{\dagger} \phi] - Tr[(U_i \vec{e}_0 \vec{e}_0^{\dagger} U_i^{\dagger} \otimes Q)\xi_R] \\ \frac{\partial J}{\partial \theta_{i,j}} = \frac{\partial Tr[\phi(U_i \rho_0 U_i^{\dagger})]}{\partial \theta_{i,j}} - \frac{\partial Tr[\xi_R(U_i \rho_0 U_i^{\dagger} \otimes Q)]}{\partial \theta_{i,j}} \end{cases} \tag{27}$$

The gradient of the loss function *J* with respect to the discriminator parameters α_k and β_l is:

$$\begin{cases} \frac{\partial J}{\partial \alpha_k} = -Tr[PA_k] - Tr[(P \otimes Q)\frac{(A_k \otimes I_Y)\xi_R}{\lambda}] \\ \frac{\partial J}{\partial \beta_l} = Tr[QB_l] - Tr[(P \otimes Q)\frac{(I_X \otimes B_l)\xi_R}{\lambda}] \end{cases} \tag{28}$$

The gradient term $\frac{\partial Tr\left[\phi\left(U_i \rho_0 U_i^{\dagger}\right)\right]}{\partial \theta_{i,j}}$ and $\frac{\partial Tr\left[\xi_R\left(U_i \rho_0 U_i^{\dagger} \otimes Q\right)\right]}{\partial \theta_{i,j}}$ can be obtained by analytic gradient Schuld et al. [Schuld, Bergholm, Gogolin et al. (2019)], the objective function can be optimized by a gradient-based training method that trains the qWGAN model.

Therefore, by different distance measures, it can determine different objective functions and design specific optimization algorithms. Therefore, it is important to explore the impact of the objective function on the performance of the QGAN model and the impact of QGAN data generation.

4.2.5 Evaluation performance of QGAN

In addition to improving the loss function and network structure to improve QGAN performance, we need to study evaluation indicators. Traditional GAN measure the performance of GAN in many ways, such as Inception Score [Salimans, Goodfellow, Zaremba et al. (2016)], Wasserstein distance [Arjovsky, Chintala and Bottou (2017)], Mode Score [Che, Li, Jacob et al. (2016)], and kernel MMD [Li, Chang, Cheng et al. (2017)]. But, how to comprehensively evaluate the performance of QGAN, such as the diversity and quality of generated data, it is a problem.

Especially when QGAN's data dimension is very high, how to find an objective and quantifiable evaluation index to evaluate the performance of QGAN generated data, which is still a worth exploring problem.

4.3 QGAN's application

4.3.1 Preparing quantum states

In quantum chemistry, marked physical property is condition information, Conditional QGAN can prepare a list of VQE [Peruzzo, McClean, Shadbolt et al. (2014)] states of a molecule. Generator will generate new molecular states, which have the same physical properties as the original molecular state. In fact, Conditional QGAN has be successfully implemented by numerical simulation to generate quantum states.

For generating unknown quantum pure state, QGAN [Benedetti, Grant, Wossnig et al. (2019)] can find tensor network representations of complex target states. Through numerical simulation, the four qubits QGAN can approximate the entangled target state by using iRprop$^-$ optimization algorithm, which show that the higher the complexity of the generator and discriminator, the better the quality of the approximate target state.

Meanwhile, qWGAN [Chakrabarti, Huang, Li et al. (2019)] has better stability by numerical experiments, which can approximates a complex quantum state by a simple quantum state, and realizes the generation of 8-qubit pure states and 3-qubit mixed states. The qWGAN also has a certain ability to resist noise based on an ion trap using a noisy operation. QGAN can also generate quantum data in condensed matter physics and quantum chemistry, such as drug design.

4.3.2 Compressing quantum circuit

In quantum simulation, Conditional QGAN can be used to compress time-based evolution gate sequences of different time steps.

QGAN [Hu, Wu, Cai et al. (2019)] can run on NISQ devices [Preskill (2018)] with high fidelity, which use the gradient descent+linear search+normalization optimization method to generate the single qubit state, the BELL state, the GHZ state, the W state. Moreover, it may have far-reaching effects in solving the quantum many-body problem [Carleo and Troyer (2017)], which can directly extend to the optimal control [Li, Yang, Peng et al. (2017)] and self-guided quantum tomography [Chapman, Ferrie and Peruzzo (2016)]. For the n-dimensional quantum system, the number of QGAN parameters is $O(n^2)$, which showing a potential "exponential" advantage to some extent.

Moreover, the qWGAN only uses about 50 quantum gates to approximate 3-qubit 1-dimensional Hamiltonian simulation of the Heisenberg model, which achieve an average fidelity of 0.9999. Such a 1-dimensional 3-qubit Heisenberg model using standard techniques to simulate the Hamiltonian, which requires approximately 10,000 gates [Childs, Maslov, Nam et al. (2018)]. It is also possible to "compile" known gate sequences into gate sequences of different quantum computing systems. qWGAN can improve the robustness and scalability of the quantum generative model, implemented on NISQ [Preskill (2018)] equipment.

QGAN [Zoufal, Lucchi and Woerner (2019)] can load a random distribution, $|\psi_{in}\rangle$ is a discrete uniform distribution, normal distribution and random distribution as inputs of QGAN, QGAN output target distribution, QGAN use $O\ (poly\ (n))$ quantum gates. Finally, by using relative entropy and K-S test to verify QGAN, which can achieve a good performance.

4.3.3 Generating probability distributions

When original GAN generate discrete data, the discriminator D can't pass the gradient update information to G well. Therefore, QGAN can easily generate discrete data. For example, Situ et al. [Situ, He, Wang et al. (2018); Zeng, Wu, Liu et al. (2019)] design the QGAN to generate discrete data distributions $|\psi\rangle = \sum_x \sqrt{p(x)}\ |x >$ by using the inherent probabilistic properties of quantum mechanics. It shows the advantage of the expression ability and effective direct sampling capability of quantum circuits, and obtain good performance by accuracy, KL divergence and loss function value to verify the quality of the generated data. Those results can apply in natural language processing and drug design.

In the numerical simulation experiment on Yao.jl [Luo, Liu, Zhang et al. (2019)], this QGAN model can also infer the unobserved value q based on the partial observation value e by using the BAS dataset. The specific process is that QGAN output $|\psi\rangle = \sqrt{p(e)}|q,e\rangle + \sqrt{1-p(e)}|\overline{q,e}\rangle$, then the Amplitude Amplification of quadratic speedup algorithm [Brassard, Hoyer, Mosca et al. (2002)] to infer missing data q for image inpainting. The time complexity of the operation is $\mathcal{O}(1/\sqrt{p(e)})$, but the inferred data is still in the real dataset. How to infer that data is not in the real dataset, which is the direction of future research.

Zoufal et al. [Zoufal, Lucchi and Woerner (2019)] have designed QGAN to learn and load random distributions, which apply in the spot price S_T of an asset at maturity T underlying a European call option. The fair price of the option is estimated by combining the resulting distribution with a Quantum amplitude estimation (QAE) algorithm [Brassard, Hoyer, Mosca et al. (2002)]. In more detail, QAE estimate function $\mathbb{E}[max\{0,S_T - K\}] = \mathbb{P}[|1\rangle](2^n - K - 1)$, where K is the spot price, and the spot price S_T is sampled from the generated distribution $|g_\theta\rangle$, which can promote the pricing of financial derivatives. A simple Black-Scholes model was implemented to price European options, where European call options the spot price S_T of model at maturity T subject to log-normal distribution.

Of course, QGAN also generate continuous probability distributions [Romero and Aspuru-Guzik (2019)] by using numerical simulations with PyQuil and PyTorch tools or the NISQ [Preskill (2018)] device, such as images, but whether it can provide quantum advantages needs further study. In fact, the non-convergent oscillation behavior and barren plateaus [McClean, Boixo, Smelyanskiy et al. (2018)] is obvious, for solving this problem, which need add a regularization term based on the real data [Mescheder, Geiger and Nowozin (2018); Roth, Lucchi, Nowozin et al. (2017)] to the discriminator loss function. How to solve barren plateaus, which needs further study. Moreover, the QGAN model is also compatible with online learning and incremental learning.

4.3.4 Other application

Du et al. [Du, Hsieh and Tao (2019)] propose QMMW, which is the $\mathrm{O}\left(\sqrt{\frac{N}{T}}\right)$ convergence rate that means QGAN rapidly converges to Nash equilibrium. QMMW is the $\mathrm{O}(N^3T^4)$ computational complexity. For entanglement tests task, it needs to design effective separability standards to distinguish whether a particular quantum state is entangled or separable. When the input state is separable, training loss value fits well with Eq. (29).

$$|J(\overline{\sigma}_G,\overline{\sigma}_D)-J(\sigma_G^*,\sigma_D^*)|\leq 3\sqrt{\frac{N}{T}} \tag{29}$$

When the input state is entangled states, the training loss value oscillates around 0.85, and fidelity between target state and generated state is less than 0.25. By quantum state tomography to reconstruct the classical density matrix, the existing entanglement test standards [Doherty, Parrilo and Spedalieri (2004); Horodecki (1997)] will exponentially increase the device run time. QMMW also can directly manipulating quantum data to avoid time-consuming quantum state tomography. There, for any quantum information-processing task, it needs to be reconstructed into a quantum-generative adversarial learning language.

In addition, Shrivastava et al. [Shrivastava, Puri, Gupta et al. (2019)] propose OpticalGAN to generate energy eigenstates and coherent states. Barbeau et al. [Barbeau and Garcia-Alfaro (2019)] propose QGAN to fake micro-aircraft navigation data by using PennyLane simulation.

Anschuetz et al. [Anschuetz and Zanoci (2019); Wilson, Vandal, Hogg et al. (2019)] propese the QAAN, improve GAN Performance with Quantum Boltzmann Machine (QBM) and use Quantum Monte Carlo to train QBM. Compared associative adversarial networks (ANN) [Arici and Celikyilmaz (2016)], QANN can generate the higher quality of MNIST and CIFAR-10 quality, which is evaluated by Inception score and Fr'echet Inception distance. But, QANN can't show potential quantum advantages, because quantum Boltzmann machines only assist GAN training, QANN uses a quantum Boltzmann machine as the associative memory of the GAN to learn the high-layer feature distribution of the discriminator, which lies in the middle layer of the discriminator.

4.3.5 Application summary

The combination of quantum computing and machine learning has many advantages, such as the sampling task [Arute, Arya, Babbush et al. (2019)], which shows good performance. GAN is a kind of adversarial algorithm. QGAN is a quantum-classical hybrid adversarial algorithm. Generator or discriminator of QGAN is quantum circuit, quantum gradient calculations is either quantum circuits or a one-step calculation [Schuld, Bergholm, Gogolin et al. (2019)]. This sub-subsection compares QGAN from quantum supremacy, whether the data is quantified, whether the generator is quantized, whether the discriminator is quantized, whether it has numerical simulation experiments, and model applications aspects. For details, see Tab. 2.

Table 2: QGAN comparison

QGAN algorithm	quantum supremacy	data type	G	D	experiment	application
QGAL [Lloyd and Weedbrook (2018)]	No	quantum	No	Yes/ No	No	none
QGAL [Lloyd and Weedbrook (2018)]	Yes	quantum /classical	Yes	Yes	No	Learning quantum data
Conditional QGAN [Dallaire-Demers and Killoran (2018)]	Yes	quantum	Yes	Yes	Yes	Generate molecular states and compress gate sequences
QGAN [Situ, He, Wang et al. (2018); Zeng, Wu, Liu et al. (2019); Zoufal, Lucchi and Woerner (2019)]	Yes	classical	No	Yes	Yes	Generate classical discrete probability distributions, natural language processing, drug design, inference unknown data, European option pricing
QGAN in a superconducting quantum circuit [Hu, Wu, Cai et al. (2019)].	Yes	quantum	Yes	Yes	Yes	Learning quantum data
QGAN [Romero and Aspuru-Guzik (2019)] for continuous distributions	unknown	classical continuous data	Yes	Yes/ No	Yes	Generate a classical continuous probability distribution
QGAN [Benedetti, Grant, Wossnig et al. (2019)] for pure state approximation.	Yes	quantum	Yes	Yes	Yes	Find tensor representations of complex target states, compress known gate sequences to different quantum systems or simpler gate sequences
QMMW [Du, Hsieh and Tao (2019)]	Yes	quantum	Yes	Yes	Yes	Online Learning
qWGAN [Chakrabarti, Huang, Li et al. (2019)]	Yes	quantum	Yes	Yes	Yes	Compress the gate sequence of the Hamilton simulation to generate quantum data
OpticalGAN [Shrivastava, Puri, Gupta et al. (2019)]	Yes	quantum	Yes	Yes	Yes	Eigenstates and coherent states

QGAN is a very new quantum machine-learning algorithm that can exhibit quantum advantages in recent quantum devices, which can implemente on a fault-tolerant NISQ device [Preskill (2018)].

5 Conclusion and prospects

5.1 Conclusion

QGAN is a kind of quantum-classical hybrid architecture that developed in recent years. QGAN exhibits the potential exponential quantum speedups over the traditional GAN. Many researcher haven't make deeper experimental verification and theoretically exponential quantum speedups. It's the same as the classical GAN, there are many problems with quantum GAN, such as QGAN's instability in the training process, the problem of gradient disappearance, the mode collapse problem. How to introduce the classical GAN improvement methods, such as feature matching, minibatch discrimination, historical averaging, One-sided label smoothing, virtual batch normalization and other training techniques into the QGAN to improve the QGAN performance, which is worth studying. Tab. 3 summarizes the problems in QGAN.

Table 3: QGAN needs to solve problems and potential solutions in the future

QGAN algorithm	problems	Potential solutions
QGAL [Lloyd and Weedbrook (2018)]	How to search the parameter space to explore learning algorithms that quickly converge to Nash equilibrium? Barren plateaus [McClean, Boixo, Smelyanskiy et al. (2018)] problem?	QMMW [Du, Hsieh and Tao (2019)], QNG [Stokes, Izaac, Killoran et al. (2019)], an initialization strategy for addressing barren plateaus [Grant, Wossnig, Ostaszewski et al. (2019)].
QGAL [Lloyd and Weedbrook (2018)]	How to encode classical real data into quantum data with O $(poly$ $(logN))$ complexity?	QGAN for loading random probability distributions [Zoufal, Lucchi and Woerner (2019)].
QGAL [Lloyd and Weedbrook (2018)]	How to explore quantum convolutional GAN, conditional GAN, bidirectional GAN, semi-supervised GAN?	Conditional QGAN scheme [Dallaire-Demers and Killoran (2018)].
Conditional QGAN [Dallaire-Demers and Killoran (2018)]	How to stabilize the training process and how to solve model collapses problem? How to guide $G\left(\vec{\theta}_G\right)$ of training?	Through feature matching after truncation and decomposition $D\left(\vec{\theta}_D\right)$
QGAN for discrete data distributions [Situ, He, Wang et al. (2018)]	How to generate higher-dimensional data, how to design the structured quantum generator and discriminator?	QGAN [Zoufal, Lucchi and Woerner (2019)], hierarchical structure [Grant, Benedetti, Cao et al. (2018)], topology structure [Chen, Wossnig, Severini et al. (2018)].
QGAN for discrete data distributions [Situ, He, Wang et al. (2018); Zeng, Wu, Liu et al. (2019)]	How to make input state |z> of the generator generated from the prior distribution *p(z)* to capture a richer probability distribution?	QGAN for discrete probability distributions [Zoufal, Lucchi and Woerner (2019)].
GAN for discrete data distributions [Zoufal, Lucchi and Woerner (2019)]	How to study QGAN's ability to represent continuous or non-equidistant data distributions?	none
QGAN [Benedetti, Grant, Wossnig et al. (2019)] for pure state approximation.	How to find better stopping criteria for heuristic optimization algorithms? How to explore better Rprop$^-$ optimization algorithm and some training strategies? How to design a detailed numerical	QGAN in a superconducting quantum circuit [Hu, Wu, Cai et al. (2019)].

	benchmark to compare the performance of different QGAN models? How to evaluate the quality of model generation?	
QGAN [Benedetti, Grant, Wossnig et al. (2019)] for pure state approximation.	How to distinguish between two quantum states?	qWGAN [Chakrabarti, Huang, Li et al. (2019)]
QMMW [Du, Hsieh and Tao (2019)]	How to design advanced online learning methods that improve QGAN training performance?	none
qWGAN [Chakrabarti, Huang, Li et al. (2019)]	How to build richer Quantum Wasserstein metric theory? How noise affects the model? Can QGAN models run on larger quantum systems?	none
QGAN [Romero and Aspuru-Guzik (2019)] for continuous distributions	How to generate images and sounds better than traditional image and sound generation?	none

In the theory, there has some problems need to be solved. Such as how to reduce the time and space complexity of QGAN. How to choose the gate type, how to initialize the parameters, how to explore the latent space and let the input state $|z >$ of the generator generate from a prior distribution $p(z)$, so that the probability distribution generated by the generator is $p_{\boldsymbol{\theta}}(x) = \sum_z p\,(z)|\langle x|U_{\boldsymbol{\theta}}|z\rangle|^2$, the QGAN model can generate a richer probability distribution. How to apply QGAN in quantum chemistry, quantum many-body problem [Carleo, Giuseppe and Matthias (2017)]. How to explore a good Rprop$^-$ optimization algorithm and some training strategies, including using a large step size factor, or after remembering the last appropriately large, the sign of the gradient, take a step in that direction when the current gradient is zero. How to explore QGAN's quantum advantages in completing those learning tasks, such as the identification of quantum correlation tasks. All these issues are worth studying to explore the deeper QGAN theory.

In the real experimental environment, there are also hardware problems in the actual quantum equipment, such as the decoherence process of qubits, the limited measurement accuracy, the non-ideal gradient calculation, the noise of quantum devices, etc.

5.2 Prospects

QML is a cross-cutting field combining quantum mechanics and machine learning. QGAN is a quantum-classical hybrid algorithm, which involve (GAN), QML and quantum-classical hybrid model. How to apply QGAN to solve some complex problems. Let everyone see the potential value of quantum computers, which is a very meaningful research work in the future.

On the one hand, it is also worthy to explore the theory of QGAN. Here are some important problems. How to effectively initialize parameters and design the circuit structure based on a specific problem is a major research direction in the future. How to use quantum unique distance measurement method to design the objective function of QGAN, how to quantify the effect of QGAN generation, how to introduce the original GAN training skills into QGAN, to improve the performance of the QGAN and explore more applications. In QGAN, how to solve these problems is the future topic.

On the other hand, it is also worthy to explore the application of QGAN in the fields of quantum chemistry, quantum physics, quantum simulation, quantum finance, cryptography. Such as generating molecular states of the same physical properties and compressing gate sequences based on time evolution.

In conclusion, we research QGAN can further promote QML.

Funding Statement: This work is supported by the National Natural Science Foundation of China (No. 61572086, No. 61402058), the Key Research and Development Project of Sichuan Province (Nos. 20ZDYF2324, 2019ZYD027 and 2018TJPT0012), the Innovation Team of Quantum Security Communication of Sichuan Province (No. 17TD0009), the Academic and Technical Leaders Training Funding Support Projects of Sichuan Province (No. 2016120080102643), the Application Foundation Project of Sichuan Province (No. 2017JY0168), the Science and Technology Support Project of Sichuan Province (Nos. 2018GZ0204 and 2016FZ0112).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Agrawal, R.; Srikant, R. (1994): Fast algorithms for mining association rules in large databases. *Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487-499. Morgan Kaufmann Publishers Inc., USA.

Anschuetz, E. R.; Zanoci, C. (2019): Near-term quantum-classical associative adversarial networks. *Physical Review A*, doi: 10.1103/PhysRevA.100.052327.

Arici, T.; Celikyilmaz, A. (2016): Associative adversarial networks.

https://arxiv.org/abs/1611.06953.

Arjovsky, M.; Bottou, L. (2017): Towards principled methods for training generative adversarial networks. https://arxiv.org/abs/1701.04862.

Arjovsky, M.; Chintala, S.; Bottou, L. (2017): Wasserstein gan.

https://arxiv.org/abs/1701.07875.

Arora, S.; Hazan, E.; Kale, S. (2012): The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, vol. 8, no. 1, pp. 121-164.

Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin J. et al. (2019): Quantum supremacy using a programmable superconducting processor. *Nature*, vol. 574, no. 7779, pp. 505-510.

Barbeau, M.; Garcia-Alfaro, J. (2019): Faking and discriminating the navigation data of a micro aerial vehicle using quantum generative adversarial networks. https://arxiv.org/abs/1907.03038.

Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V. et al. (2018): Relational inductive biases, deep learning, and graph networks.

https://arxiv.org/abs/1806.01261.

Benedetti, M.; Garcia-Pintos, D.; Perdomo, O.; Leyton-Ortega, V.; Nam, Y. et al. (2019): A generative modeling approach for benchmarking and training shallow quantum circuits. *NPJ Quantum Information*, vol. 5, no. 1, pp. 1-9.

Benedetti, M.; Grant, E.; Wossnig, L.; Severini, S. (2019): Adversarial quantum circuit learning for pure state approximation. *New Journal of Physics*.

Berthelot, D.; Schumm, T.; Metz, L. (2017): Began: boundary equilibrium generative adversarial networks. https://arxiv.org/abs/1703.10717.

Bergholm, V.; Izaac, J.; Schuld, M.; Gogolin, C.; Alam, M. S. et al. (2018): Pennylane: automatic differentiation of hybrid quantum-classical computations. https://arxiv.org/abs/1811.04968.

Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N. et al. (2017): Quantum machine learning. *Nature*, vol. 549, no. 7671, pp. 195-202.

Brassard, G.; Hoyer, P.; Mosca, M.; Tapp, A. (2002): Quantum amplitude amplification and estimation. *Contemporary Mathematics*, vol. 305, pp. 53-74.

Buhrman, H.; Cleve, R.; Watrous, J.; Wolf, R. D. (2001): Quantum fingerprinting. *Physical Review Letters*.

Carleo, G.; Troyer, M. (2017): Solving the quantum many-body problem with artificial neural networks. *Science*, vol. 355, no. 6325, pp. 602-606.

Chakrabarti, S.; Huang, Y. M.; Li, T. Y.; Feizi, S.; Wu, X. D. (2019): Quantum wasserstein generative adversarial networks. *Advances in Neural Information Processing Systems*, pp. 6778-6789.

Chapman, R. J.; Ferrie, C.; Peruzzo, A. (2016): Experimental demonstration of self-guided quantum tomography. *Physical Review Letters*.

Che, T.; Li, Y.; Jacob, A. P.; Bengio, Y.; Li, W. J. (2016): Mode regularized generative adversarial networks. https://arxiv.org/abs/1612.02136.

Chen, H.; Wossnig, L.; Severini, S.; Neven, H.; Mohseni, M. (2018): Universal discriminative quantum neural networks. https://arxiv.org/abs/1805.08654.

Chen, T.; Guestrin, C. (2016): Xgboost: a scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794. Association for Computing Machinery, USA.

Childs, A. M.; Maslov, D.; Nam, Y.; Ross, N. J.; Su, Y. (2018): Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, vol. 115, no. 38, pp. 9456-9461.

Cho, K.; Merriënboer, V. B.; Gulcehre, C.; Bougares, F.; Schwenk, H. et al. (2014): Learning phrase representations using RNN encoder-decoder for statistical machine translation. https://arxiv.org/abs/1406.1078.

Choi, Y.; Choi, M.; Kim, M.; Ha, J. W.; Kim, S. et al. (2018): StarGAN: unified generative adversarial networks for multi-domain image-to-image translation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pp. 8789-8797.

Clark, A.; Donahue, J.; Simonyan, K. (2019): Efficient video generation on complex datasets. https://arxiv.org/abs/1907.06571.

Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B. et al. (2018): Generative adversarial networks: an overview. *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53-65.

Dallaire-Demers, P. L.; Killoran, N. (2018): Quantum generative adversarial networks. *Physical Review A*.

Devlin, J.; Chang, M. W.; Lee, K.; Toutanova, K. (2018): Bert: pre-training of deep bidirectional transformers for language understanding. https://arxiv.org/abs/1810.04805.

Doherty, A. C.; Parrilo, P. A.; Spedalieri, F. M. (2004): Complete family of separability criteria. *Physical Review A*.

Du, Y.; Hsieh, M. H.; Tao, D. (2019): Efficient online quantum generative adversarial learning algorithms with applications. https://arxiv.org/abs/1904.09602.

Duan, B.; Yuan, J.; Liu, Y.; Li, D. (2018): Efficient quantum circuit for singular-value thresholding. *Physical Review A*.

Engel, J.; Agrawal, K. K.; Chen, S.; Gulrajani, I.; Donahue, C. et al. (2019): Gansynth: adversarial neural audio synthesis. https://arxiv.org/abs/1902.08710.

Farhi, E.; Goldstone, J.; Gutmann, S. (2014): A quantum approximate optimization algorithm. https://arxiv.org/abs/1411.4028.

Feynman, R. P. (1982): Simulating physics with computers. *International Journal of Theoretical Physics*, vol. 21, no. 6/7, pp. 467-488.

Friedman, J. H. (2001): Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, vol. 29, no. 5, pp. 1189-1232.

Gao, J.; Qiao, L. F.; Jiao, Z. Q.; Ma, Y. C.; Hu, C. Q. et al. (2018): Experimental machine learning of quantum states. *Physical Review Letters*.

doi: 10.1103/PhysRevLett.120.240501.

Gomez, A. N.; Huang, S.; Zhang, I.; Li, B. M.; Osama, M. (2018): Unsupervised cipher cracking using discrete gans. https://arxiv.org/abs/1801.04883.

Goodfellow, I. (2016): NIPS 2016 tutorial: generative adversarial networks.

https://arxiv.org/abs/1701.00160.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D. et al. (2014): Generative adversarial nets. *International Conference on Neural Information Processing Systems*, pp. 2672-2680. MIT Press, USA.

Grant, E.; Benedetti, M.; Cao, S.; Hallam, A.; Lockhart, J. et al. (2018): Hierarchical quantum classifiers. *NPJ Quantum Information*, vol. 4, no. 1, pp. 1-8.

Grant, E.; Wossnig, L.; Ostaszewski, M.; Benedetti, M. (2019): An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*.

Grover, L. K. (1996): A fast quantum mechanical algorithm for estimating the median. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 212-219. Association for Computing Machinery, USA.

Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A. (2017): Objective-reinforced generative adversarial networks for sequence generation models. https://arxiv.org/abs/1705.10843.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. C. (2017): Improved training of Wasserstein gans. *Advances in Neural Information Processing Systems*, pp. 5767-5777.

Han, J.; Pei, J.; Kamber, M. (2011): *Data Mining: Concepts and Techniques*. Elsevier, Netherlands.

Han, J.; Pei, J.; Mortazavi-Asl, B.; Pinto, **H.** (2001): Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth. *Proceedings of the 17th International Conference on Data Engineering*, pp. 215-224. *IEEE*, Germany.

Harrow, A. W.; Hassidim, A.; Lloyd, S. (2009): Quantum algorithm for linear systems of equations. *Physical Review Letters*. doi: 10.1103/PhysRevLett.103.150502.

Hochreiter, S.; Schmidhuber, J. (1997): Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735-1780.

Hong, Y.; Hwang, U.; Yoo, J.; Yoon, S. (2019): How generative adversarial networks and their variants work: an overview. *ACM Computing Surveys*, vol. 52, no. 1, pp. 1-43.

Horodecki, P. (1997): Separability criterion and inseparable mixed states with positive partial transposition. *Physics Letters A*, vol. 232, no. 5, pp. 333-339.

Hu, L.; Wu, S. H.; Cai, W.; Ma, Y.; Mu, X. et al. (2019): Quantum generative adversarial learning in a superconducting quantum circuit. *Science Advances*.

Huggins, W.; Patil, P.; Mitchell, B. Whaley, K. B.; Stoudenmire, E. M. (2019): Towards quantum machine learning with tensor networks. *Quantum Science and Technology*. doi: 10.1088/2058-9565/aaea94.

Jin, G.; Shen, S.; Zhang, D.; Dai, F.; Zhang, Y. (2019): Ape-gan: adversarial perturbation elimination with GAN. *ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3842-3846.

Johansson, J. R.; Nation, P. D.; Nori, F. (2013): QuTiP 2: A Python framework for the dynamics of open quantum systems. *Computer Physics Communications*, vol. 184, no. 4, pp. 1234-1240.

Killoran, N.; Izaac, J.; Quesada, N.; Bergholm, V.; Amy, M. et al. (2019): Strawberry fields: a software platform for photonic quantum computing. *Quantum*.

Kingma, D. P.; Ba, J. (2014): Adam: a method for stochastic optimization.

https://arxiv.org/abs/1412.6980.

Kingma, D. P.; Welling, M. (2013): Auto-encoding variational bayes.

https://arxiv.org/abs/1312.6114.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012): ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pp. 1097-1105.

Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A. et al. (2017): Photo-realistic single image super-resolution using a generative adversarial network.

*Proceedings of the IEEE Conference on Computer Vision and Pattern Recogniti*on, pp. 105-114.

Li, C. L.; Chang, W. C.; Cheng, Y.; Yang, Y. M.; Poczos, B. (2017): MMD GAN: towards deeper understanding of moment matching network. *Advances in Neural Information Processing Systems*, pp. 2203-2213.

Li, H. (2012): *Statistical Learning Method.* Tsinghua University Press, China.

Li, J.; Yang, X.; Peng, X.; Sun, C. P. (2017): Hybrid quantum-classical approach to quantum optimal control. *Physical Review Letters*.

Lin, K.; Li, D.; He, X.; Zhang, Z.; Sun, M. T. (2017): Adversarial ranking for language generation. *Advances in Neural Information Processing Systems*, pp. 3155-3165.

Liu, J. G.; Wang, L. (2018): Differentiable learning of quantum circuit born machines. *Physical Review A*. doi: 10.1103/PhysRevA.98.062324.

Lloyd, S.; Mohseni, M.; Rebentrost, P. (2014): Quantum principal component analysis. *Nature Physics*, vol. 10, no. 9, pp. 108-113.

Lloyd, S.; Weedbrook, C. (2018): Quantum generative adversarial learning. *Physical Review Letters*. doi: 10.1103/PhysRevLett.121.040502.

Low, G. H.; Yoder, T. J.; Chuang, I. L. (2014): Quantum inference on bayesian networks. *Physical Review A*. doi: 10.1103/PhysRevA.89.062315.

Luo, X. Z.; Liu, J. G.; Zhang, P.; Wang, L. (2019): Yao.jl: extensible, efficient framework for quantum algorithm design. https://arxiv.org/abs/1912.10877.

Maas, A. L.; Hannun, A. Y.; Ng, A. Y. (2013): Rectifier nonlinearities improve neural network acoustic models. *Proceedings of the International Conference on Machine Learning*, vol. 30.

MacQueen, J. (1967**)**: Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 14, pp. 281-297.

Mao, X.; Li, Q.; Xie, H.; Lau, R.; Wang, Z. et al. (2017): Least squares generative adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794-2802.

McClean, J. R.; Boixo, S.; Smelyanskiy, V. N.; Babbush, R.; Neven, H. (2018): Barren plateaus in quantum neural network training landscapes. *Nature Communications*, vol. 9, no. 1, pp. 1-6.

Mescheder, L.; Geiger, A.; Nowozin, S. (2018): Which training methods for GANs do actually converge? *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 3478-3487.

Mirza, M.; Osindero, S. (2014): Conditional generative adversarial nets.

https://arxiv.org/abs/1411.1784.

Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. (2018): Quantum circuit learning. *Physical Review A*. doi: 10.1103/PhysRevA.98.032309.

Nash, J. (1951): Non-cooperative games. *Annals of Mathemat*ics, vol. 54, no. 2, pp. 286-295.

Nielsen, M. A.; Chuang, I. L. (2010): *Quantum Computation and Quantum Information*. Cambridge University Press, UK.

Pan, J.; Sayrol, E.; Nieto, X. G.; Ferrer, C. C.; Torres, J. et al. (2017): Salgan: visual saliency prediction with adversarial networks. https://arxiv.org/abs/1701.01081.

Pathak, D.; Krahenbuhl, P.; Donahue, J.; Darrell, T.; Efros, A. A. (2016): Context encoders: feature learning by inpainting. *Proceedings of the IEEE Conference on Computer Cision and Pattern Recognition*, pp. 2536-2544.

Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M. H.; Zhou, X. Q. et al. (2014): A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, vol. 5, pp. 4213.

Plesch, M.; Brukner, Č. (2011): Quantum-state preparation with universal gate decompositions. *Physical Review A*. doi: 10.1103/PhysRevA.83.032302.

Preskill, J. (2018): Quantum computing in the NISQ era and beyond. *Quantum*.

Radford, A.; Metz, L.; Chintala, S. (2015): Unsupervised representation learning with deep convolutional generative adversarial networks. https://arxiv.org/abs/1511.06434.

Rebentrost, P.; Mohseni, M.; Lloyd, S. (2013): Quantum support vector machine for big data classification. *Physical Review Letters*. doi: 10.1103/PhysRevLett.113.130503.

Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Lee, H. et al. (2016): Generative adversarial text to image synthesis. *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, pp. 1060-1069.

Ren, Y.; Zhu, F.; Sharma, P. K.; Wang, T.; Wang, J. et al. (2020): Data query mechanism based on hash computing power of blockchain in Internet of things. *Sensors*.

Riedmiller, M.; Braun, H. (1993): A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *Proceedings of the IEEE International Conference on Neural Networks*, vol. 1, pp. 586-591.

Romero, J.; Aspuru-Guzik, A. (2019): Variational quantum generators: generative adversarial quantum machine learning for continuous distributions.

https://arxiv.org/abs/1901.00848.

Roth, K.; Lucchi, A.; Nowozin, S.; Hofmann, T. (2017): Stabilizing training of generative adversarial networks through regularization. *Advances in Neural Information Processing Systems*, pp. 2018-2028.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A. et al. (2016): Improved techniques for training gans. *Advances in Neural Information Processing Systems*, pp. 2234-2242.

Schmied, R. (2016): Quantum state tomography of a single qubit: comparison of methods. *Journal of Modern Optics*, vol. 63, no. 18, pp. 1744-1758.

Schuld, M.; Bergholm, V.; Gogolin, C.; Izaac, J.; Killoran, N. (2019): Evaluating analytic gradients on quantum hardware. *Physical Review A*.

Shi, H.; Dong, J.; Wang, W.; Qian, Y.; Zhang, X. et al. (2017): SSGAN: secure steganography based on generative adversarial networks. *Pacific Rim Conference on Multimedia, Lecture Notes in Computer Science*, vol. 10735, pp. 534-544.

Shor, P. W. (1994): Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124-134.

Shrivastava, N.; Puri, N.; Gupta, P.; Krishnamurthy, B.; Verma, S. (2019): OpticalGAN: generative adversarial networks for continuous variable quantum computation. https://arxiv.org/abs/1909.07806.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A. et al. (2017): Mastering the game of go without human knowledge. *Nature*, vol. 550, no. 7676, pp. 354-359.

Situ, H.; He, Z.; Wang, Y. Y.; Li, L.; Zheng, S. (2018): Quantum generative adversarial network for generating discrete data. https://arxiv.org/abs/1807.01235.

Stauffer, C.; Grimson, W. E. L. (1999): Adaptive background mixture models for real-time tracking. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246-252.

Stokes, J.; Izaac, J.; Killoran, N.; Carleo, G. (2019): Quantum natural gradient. https://arxiv.org/abs/1909.02108.

Taylor, S. J.; Letham, B. (2018): Forecasting at scale. *The American Statistician*, vol. 27, no. 1, pp. 37-45.

Wang, X.; Shrivastava, A.; Gupta, A. (2017): A-fast-rcnn: hard positive generation via adversary for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2606-2615.

Wang, X.; Yu, K.; Wu, S.; Gu, J. J.; Liu, Y. H. et al. (2019): Esrgan: enhanced super-resolution generative adversarial networks. *Proceedings of the European Conference on Computer Vision, Lecture Notes in Computer Science*, vol. 11133, pp. 63-79.

Wiebe, N.; Braun, D.; Lloyd, S. (2012): Quantum algorithm for data fitting. *Physical Review Letters*. doi: 10.1103/PhysRevLett.109.050505.

Wilson, M.; Vandal, T.; Hogg, T.; Rieffel E. (2019): Quantum-assisted associative adversarial network: applying quantum annealing in deep learning.

https://arxiv.org/abs/1904.10573.

Yang, Z. L.; Dai, Z. H.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. et al. (2019): Xlnet: generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, pp. 5754-5764.

Yu, L.; Zhang, W.; Wang, J.; Yu, Y. (2017): Seqgan: sequence generative adversarial nets with policy gradient. *Thirty-First AAAI Conference on Artificial Intelligence*.

Zeng, J.; Wu, Y.; Liu, J. G.; Wang, L.; Hu, J. (2019): Learning and inference on generative adversarial quantum circuits. *Physical Review A*. **Zoufal, C.; Lucchi, A.; Woerner, S.** (2019): Quantum generative adversarial networks for learning and loading random distributions. *NPJ Quantum Information*, vol. 5, no. 1, pp. 1-9.

Zhu, J. Y.; Park, T.; Isola, P.; Efros, A. A. (2017): Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223-2232.