mathematics

IMPACT FACTOR 2.3

CITESCORE 4.0

Review

Survey of Quantum Generative Adversarial Networks (QGAN) to Generate Images

Mohammadsaleh Pajuhanfard, Rasoul Kiani and Victor S. Sheng

MDPI

https://doi.org/10.3390/math12233852

mathematics

MDPI

Review

Survey of Quantum Generative Adversarial Networks (QGAN) to Generate Images

Mohammadsaleh Pajuhanfard, Rasoul Kiani and Victor S. Sheng *

Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA; ryan.pjuhanfard@ttu.edu (M.P.); rasoul.kiani87@yahoo.com (R.K.)
* Correspondence: victor.sheng@ttu.edu

Abstract: Quantum Generative Adversarial Networks (QGANs) represent a useful development in quantum machine learning, using the particular properties of quantum mechanics to solve the challenges of data analysis and modeling. This paper brings up a general analysis of five QGAN architectures, focusing on their evolution, strengths, weaknesses, and limitations in noisy intermediate-scale quantum (NISQ) devices. Primary methods like Entangling Quantum GAN (EQ-GAN) and Quantum state fidelity (QuGAN) concentrate on stability, convergence, and robust performance on small-scale datasets such as 2×2 grayscale images. Intermediate models such as Image Quantum GAN (IQ-GAN) and Experimental Quantum GAN (EXQGAN) provide new ideas like trainable encoders and patch-based sub-generators that are scalable to 8×8 datasets with increasing noise resilience. The most advanced method is Parameterized Quantum Wasserstein GAN (PQWGAN), which uses a hybrid quantum-classical structure to obtain high-resolution image processing for 28×28 grayscale datasets while trying to maintain parameter efficiency. This study explores, analyzes, and summarizes critical problems of QGANs, including accuracy, convergence, parameter efficiency, image quality, performance metrics, and training stability under noisy conditions. In addition, developing QGANs can generate and train parameters in quantum approximation optimization algorithms. One of the useful applications of QGAN is generating medical datasets that can generate medical images from limited datasets to train specific medical models for the recognition of diseases.

Keywords: Quantum Generative Adversarial Network (QGAN); generator; discriminator; QGAN architectures

MSC: 68T07; 81-05

Citation: Pajuhanfard, M.; Kiani, R.; Sheng, V.S. Survey of Quantum Generative Adversarial Networks (QGAN) to Generate Images. *Mathematics* **2024**, *12*, 3852. https://doi.org/10.3390/math12233852

Academic Editor: Tucker Carrington, Jr.

Received: 3 November 2024
Revised: 26 November 2024
Accepted: 4 December 2024
Published: 6 December 2024

1. Introduction

Generative Adversarial Networks (GANs) [1] represent an adversarial game such as police and banknote forgery that involve two neural networks. Such a network needs to have a lot of resources and data to train. On the other hand, according to Quantum Mechanics and Quantum physics, a new branch of computer science was born under Quantum computing, which, with the entry of quantum in this area, caused quantum GAN in 2018 to be born [2]. This birth provided quantum resources for GANs. Although there are many problems in this way such as making qubits, noise environments, decoherence, entanglements, and so on, scientists and researchers are trying to solve them.

In this paper, we collected advanced methods that improved some problems with using QGAN for generating images. We will introduce key terminology, background of mathematics and notation in quantum computing, generative adversarial networks, and Wasserstein GAN (WGAN) [3], which are crucial for the paper.

Motivated by these limitations, we arranged a survey about QGAN to resolve them. In this survey, collecting and a detailed analysis of all five pioneering articles in the field of Quantum Generative Adversarial Networks (QGANs) is presented. The first major

contribution we bring to this debate is a different categorization scheme of QGANs that helps provide a clearer and more exhaustive taxonomy of the extant literature. This new way of approaching categorization aids in understanding the principles, concepts, and methods that encompass the approach and the differences and improvements implemented in each category. After that, the results outlined in these initial papers have been compared. In this way, it was possible to systematically compare their results, which allowed the identification of each method's relative advantages and disadvantages.

The remainder of this paper is organized as follows: The introduction and fundamentals of QGAN are introduced in Section 2. Section 3 describes the existing QGAN architectures for generating synthetic images. Section 4 will discuss and compare the papers. The challenges and limitations are examined in Section 5. Sections 6 and 7 will address the conclusions and future studies, respectively.

1.1. Background on Quantum Computing

Hilbert space is a vector space that is defined by the inner product on it. The inner product of two vectors that say vector states, like ψ and ϕ, is represented by $\langle\psi|\phi\rangle$. The amount of overlap between two vectors that have the very definition in linear algebra and operators can show with matrices [4]. This major, according to the fundamentals of quantum mechanics, says information about the state of atoms and that each electron can according to its spin provide different states and with Heisenberg's uncertainty principle produce 0s and 1s simultaneously which is quantum Superposition. The superposition status caused a new concept; its name is qubit. Qubits include 0s and 1s simultaneously in superposition before being measured. Qubits can be in 0 or 1 mode as shown [5]:

$$Basis\ States: \begin{cases} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |0\rangle \\ \begin{pmatrix} 0 \\ 1 \end{pmatrix} = |1\rangle \end{cases} \tag{1}$$

Qubits can be in 0 or 1 mode simultaneously; therefore, a qubit is shown [6]:

$$|\psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \alpha|0\rangle + \beta|1\rangle \tag{2}$$

where $\sqrt{\langle\psi|\psi\rangle} = 1$ is a normalization condition. The tensor products of the matrices will be shown by [6]:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \otimes \begin{pmatrix} x & y \\ z & w \end{pmatrix} = \begin{pmatrix} a\begin{pmatrix} x & y \\ z & w \end{pmatrix} & b\begin{pmatrix} x & y \\ z & w \end{pmatrix} \\ c\begin{pmatrix} x & y \\ z & w \end{pmatrix} & d\begin{pmatrix} x & y \\ z & w \end{pmatrix} \end{pmatrix} = \begin{pmatrix} ax & ay & bx & by \\ az & aw & bz & bw \\ cx & cy & dx & dy \\ cz & cw & dz & dw \end{pmatrix} \tag{3}$$

This operator creates a combination system that the entanglement will be in.

When measuring, the quantum state will collapse and be one of them (0 or 1). It should be known that the information for the transfer on bits and qubits is the same. However, it is interesting that n bits can transfer n 0s or 1s but n qubits will make Hilbert's space with 2^n superposition for the transfer. That means this difference is in the stream of qubits. For example, 2 qubits provide us 2^2 bits, and the dimensions of Hilbert's space increase exponentially [7].

$$\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \lambda|11\rangle \tag{4}$$

There is another display for a quantum state whose name is a density matrix of vectors. This presentation is appropriate for the trace of the result. This operator known as the outer product is the following:

$$\rho = |\psi\rangle\langle\psi| \tag{5}$$

The mixed quantum state is:

$$\sum_i \rho_i |\psi_i\rangle\langle\psi_i| \tag{6}$$

where $\sum p_i = 1$, and $p_i \geq 0$ [6,8], and there is a matrix which is called Unitary Matrix. It means $UU^\dagger = U^\dagger U = I$, where † is the conjugate transpose. The unitary matrix is important because it can normalize gates and be used to represent gates that cause reversible gates. Gate Models in quantum computing have two basic concepts, circuit and gate, which complement each other. Circuits, which are somehow qubits, are a sequence of blocks to understand rudimentary operations, and these blocks involve gates. Somehow, operations on qubits are encrypted by gates. When the number of qubits is more than one, gates can make the entanglement state. The simplest state to represent a single-qubit gate is a 2×2 matrix. The Pauli gates are one of the groups of single-qubit gates that each qubit can write a linear combination of Pauli gates. The three important Pauli gates are the x-gate, y-gate, and z-gate; they show rotations of 180 degrees around x- the x-axis, y-axis, and z-axis [8,9].

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{7}$$

Hadamard gate creates a superposition state. The matrix representation of the Hadamard gate [9] is as follows:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{8}$$

The Hadamard Gate has effects on $|0\rangle$, $|1\rangle$ and then changes into a superposition state. Interestingly, if we apply the Hadamard gate again on the previous Hadamard gate, then qubit comes out of the superposition [9].

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{9}$$

Another important gate is the CNOT gate, which as the control gate works on two qubits. The first bit is the control qubit, and the second bit is the target qubit. The CNOT gate flips the target qubit's state if and only if the control qubit is in state $|1\rangle$, but it is unchanged if the control qubit is in state $|0\rangle$. The CNOT gate is used to create entanglement between qubits [6].

$$|0,0\rangle \rightarrow |0,0\rangle, |0,1\rangle \rightarrow |0,1\rangle, |1,0\rangle \rightarrow |1,1\rangle, |1,1\rangle \rightarrow |1,0\rangle \tag{10}$$

The swap gate works on two qubits and exchanges the place of the first bit with the second one. This gate allows for the sorting of qubits in a quantum circuit [6].

$$|0,0\rangle \rightarrow |0,0\rangle, |0,1\rangle \rightarrow |1,0\rangle, |1,0\rangle \rightarrow |0,1\rangle, |1,1\rangle \rightarrow |1,1\rangle \tag{11}$$

This gate works on three qubits and can accomplish basic reversible logic operations if and only if both control qubits (the first and second qubits) are in state $|1\rangle$. Then, the Toffoli gate applies a NOT operation to a target qubit (the third qubit). Otherwise, the target qubit is unchanged. In other words, if both control qubits are in the state $|1\rangle$, the target qubit is changed. If one of the control qubits is in the state $|0\rangle$, the target qubit remains unchanged [6].

$$|1,1,0\rangle \rightarrow |1,1,1\rangle, |1,1,1\rangle \rightarrow |1,1,0\rangle \tag{12}$$

1.2. Generative Adversarial Network

This form of deep learning network consists of two distinct neural networks [1]. The generator (G) is designed to create fake data from a Gaussian noise vector (z). In contrast, the discriminator (D) is responsible for identifying and distinguishing real data from the

fake data produced by the generator [1]. Thus, it is imperative for this network to utilize real data. In this configuration, the discriminator takes in two inputs: one being real data and the other being the output generated by the generator. For example, the discriminator plays a role, like the police, in determining whether real banknotes are fake, and the generator plays the role of a forger. This game helps both improve in their job. There should be a loss function that calculates the performance of the discriminator and generator. Figure 1 shows this structure. However, such a function consists of two parts; one part is a loss function for the discriminator that is shown as follow [1]:

$$L_{D1} \log f(D(x)) \tag{13}$$

The f is a non-linearizer function. As usual, sigmoid is used. Therefore, the result will be [0, 1]. If the output of the sigmoid is 1 and around 1, it means the log will be zero, so there is no loss. For fake input G(z), if the D network can determine all G(z) are fake, it will return zero for such inputs. Therefore, one can write as follows [1]:

$$L_{D2} = \log(1 - f(D(G(z)))) \tag{14}$$

By summation two, the previous equations can follow [1]:

$$L_D = L_{D1} + L_{D2} = \log f(D(x)) + \log(1 - f(D(G(z)))) \tag{15}$$

L_D should maximize because the argument of the log is [0, 1], and for the log such values have negative value, so we must maximize that loss function to approach zero. On the other hand, the loss function of the generator should be minimized because it means the generator could create fake data that is very similar to real data. Consequently, we can update Equation (15) [1]:

$$\min_G \max_D \log f(D(x)) + \log(1 - f(D(G(z)))) \tag{16}$$

About predicted labels, it must be said that if the output of the discriminator, which is a probability function, is greater than 0.5, then the discriminator will label it real and otherwise will label it fake. We can update Formula (16). Let P_r be a probability distribution from real data and P_z be a probability distribution of fixed noise. The E is the expected value. Therefore, the Formula (4) will be updated similarly below [1]:

$$\min_G \max_D E_{x \sim P_r} \log f(D(x)) + E_{z \sim P_z} \log(1 - f(D(G(z)))) \tag{17}$$

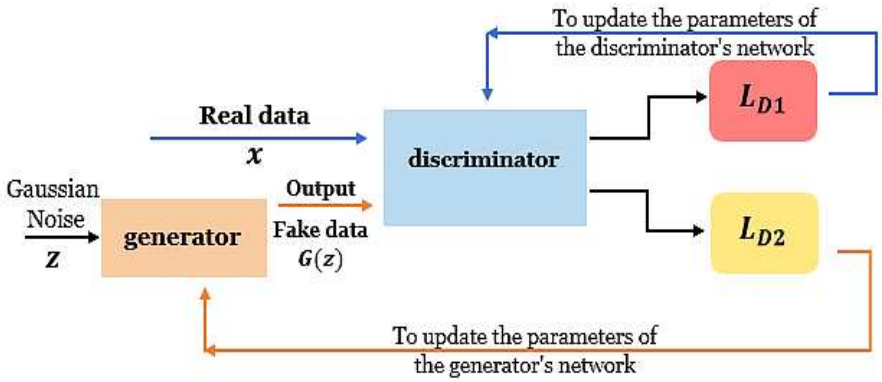

Figure 1. Operation of the GANs loss function.

Algorithm 1 indicates the way of function for the previous description [1].

Algorithm 1. Minibatch stochastic gradient descent to train GAN. The count steps are shown with k, which is a hyperparameter for applying the discriminator, and for achieving the least cost used k = 1

for the number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\left\{z^{(1)}, \ldots, z^{(m)}\right\}$ from noise prior $p_g(z)$.
- Sample minibatch of m examples $\left\{x^{(1)}, \ldots, x^{(m)}\right\}$ from data-generating distribution $p_{data}(x)$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log f\left(D\left(x^{(i)}\right)\right) + \log\left(1 - f\left(D\left(G\left(z^{(i)}\right)\right)\right)\right)\right].$$

end for

- Sample minibatch of m noise samples $\left\{z^{(1)}, \ldots, z^{(m)}\right\}$ from noise prior $p_g(z)$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - f\left(D\left(G\left(z^{(i)}\right)\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. In addition, the momentum is used in the algorithm.

1.3. Wasserstein GAN (WGAN)

The metrics defined in the original GANs to calculate distance and convergence between two probability distributions such as P_r and P_g are the Total Variation (TV) distance and the Kullback–Leibler (KL) divergence, shown in the following [3,10]:

$$\delta(P_r, P_g) = \sup_{\mathrm{a_i} \in each\ point\ in\ distributions} \left|P_r(a_i) - P_g(a_i)\right| \tag{18}$$

$$KL(P_r \| P_g) = \int \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x)\ \ d\mu(x) \tag{19}$$

where μ is the measure defined. The Jensen–Shannon (JS) divergence formula is the following that is popular to use and symmetrical [3,10]:

$$JS(P_r, P_g) = KL(P_r \| P_m) + KL(P_g \| P_m) \tag{20}$$

where $P_m = \frac{P_r + P_g}{2}$, and we can choose P_m as μ in (19). On the other hand, there is another metric named The Earth-Mover (EM) distance or Wasserstein-1 [3]:

$$W(P_r, P_g) = \inf_{\gamma \in \prod(P_r, P_g)} E_{(x,y)\sim\gamma}[\|x - y\|] \tag{21}$$

where $\prod(P_r, P_g)$ is the common distribution $\gamma(x, y)$ with fringe P_r and P_g, respectively. Therefore, the W can calculate the optimal cost better than the JS. The JS estimates distance point by point while the Wasserstein metric calculates the cost of transporting one probability distribution to another. In this situation, the discriminator gives a score that is the Wasserstein distance between P_r and P_G, such as a critic [3]. The critic's gradients have better results than the discriminator because the generator will train more easily and optimally. Since vanishing gradients are important, this method will guarantee that the gradient will not vanish. Since this formula is used from the 1-Lipschitz constraint, it was

suggested to cut the gradients of each critical parameter inside a specified range, such as $[-0.01, 0.01]$. However, such work, in a large gradient, reduces the rate of reaching optimism, but in small sizes, the gradient will vanish [3,11]. Gulrajani et al. [12] provided the WGAN-GP, which uses a gradient penalty (GP) for the 1-Lipschitz restriction. Thus, the previous formula was updated like follow as:

$$\min_{G} \max_{D \in \mathcal{D}} E_{x \sim P_r}\left(D(x)\right) - E_{z \sim P_z}(D(G(z))) - \lambda E_{\hat{x} \sim P_{\hat{x}}}\left(\left(\left\|\nabla_{\hat{x}} D(\hat{x})\right\|_2 - 1\right)^2\right) \quad (22)$$

where λ is a constant, and $P_{\hat{x}}$ is a distribution sampled uniformly in between P_r and P_G. This method is known as WGAN-GP. The WGAN-GP is capable of training random architectures to some minimum Inception score. It can train on 32 X 32 pixels with acceptable quality [13].

2. Introduction and Fundamentals of QGANs

In this section, we aim to introduce the structure of QGAN, which includes three parts.

2.1. Introduction to QGAN

In GANs, scientists assume while training that the generators and discriminators have an infinity capacity, which means they can encode every function and distribution, but classical computers cannot support such ideas. Therefore, scientists try to employ quantum computing properties to improve the capacity of the GANs. Here there are two neural networks, the generator (G) and the discriminator (D), and the input data. We can define four models of QGANs that were adopted from the idea of quantum computing in the following Figure 2 to comprehend better [14]. The general idea of QGANs, which is shown in Figure 3, is the same as GANs; however, there is a difference in that at least part of this network is quantum mode such as input data, the generator, or the discriminator [14]. We can discuss the three models of QGAN. They are (1) Quantum Data, Quantum generator, and Quantum discriminator; (2) Quantum Data, Quantum or Classical discriminator, and Classical generator; and (3) Classical data, Quantum generator, and discriminator.

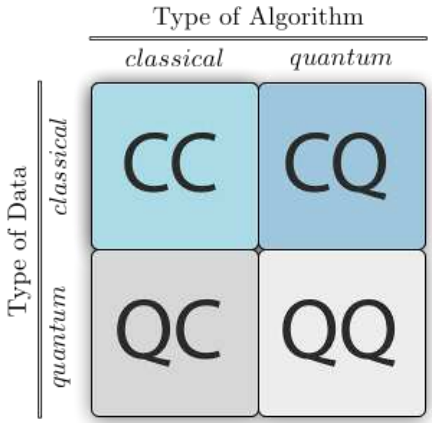

Figure 2. The CC means the data and the algorithms are classic, but the quantum concept, methods, or process has helped improve the classical algorithms. The CQ means the data is classic and the algorithms are quantum. The QC means the data is quantum (such as chemistry data) and the algorithms are classic. The QQ means the data and the algorithms are quantum. https://commons.wikimedia.org/wiki/File:Qml_approaches.tif?page=1 (accessed on 2 November 2024).

Figure 3. The view of QGAN.

2.2. Structures of QGAN

As mentioned before, we aim to discuss the structures of the QGAN in this part.

2.2.1. Quantum Data, Quantum Generator, and Quantum Discriminator

In this case, we suppose real data, the generator, and the discriminator are quantum. Let us provide some notations to comprehend better; therefore, we assign a label $|\lambda\rangle$ to real data (R) such that output will be a density matrix (ρ_λ^R) [15]:

$$R(|\lambda\rangle) = \rho_\lambda^R \tag{23}$$

As we know, the generator wants to generate a similar sample to the real data. The quantum systems perform that by variational quantum circuits (VQCs) that are constructed by gates and parametrized by vector θ_G. The generator receives two quantum states as input; one of them is the label $|\lambda\rangle$, and another is the Gaussian noise $|Z\rangle$ (as usual) to create the following quantum state:

$$G(\theta_G,\ |\lambda, Z\rangle) = \rho_\lambda^G(\theta_G, Z) = |\psi_\lambda(Z)\rangle\langle\psi_\lambda(Z)| \tag{24}$$

where ρ_λ^G and create a unitary fixed state for each λ, $|Z\rangle$ as output is similar to real data [15]. The discriminator must realize which data entered is real or fake. It is useful to know that first, the discriminator attempts to develop its approach to discriminate between real data and fake data; then through feedback from the discriminator, the generator will update to get better. Figure 4 shows this structure. This minimax game will be played between the discriminator and the generator; the main type of input data is statistical distributions that are encoded quantum. This game will continue until the probability of finding real data from fake data is $\frac{1}{2}$ in the discriminator; then this game will finish [2]. The discriminator is parameterized by a vector θ_D. The discriminator has two inputs. One of them is from a real data label $|\lambda\rangle$; another is output from the generator that we can show by $\rho_\lambda^{\frac{R}{G}}$ notation. Here there is a qubit to measure [15].

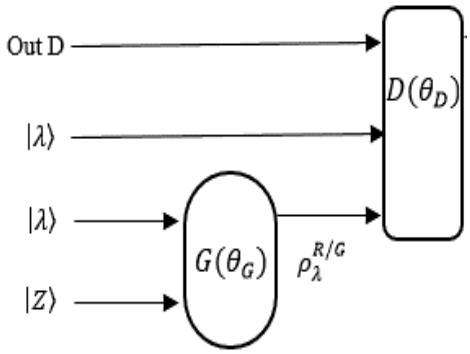

Figure 4. The general structure of QGAN.

The next step is the measurement and optimization we can define a cost function as [15]:

$$\min_{\theta_G}\max_{\theta_D} V(\theta_G, \theta_D) = \min_{\theta_G}\max_{\theta_D} \frac{1}{\Lambda}\sum_{\lambda=1}^{\Lambda} \Pr((D(\theta_D, |\lambda\rangle),\ R(|\lambda\rangle) = |Real\rangle) \cap (D(\theta_D, |\lambda\rangle),\ G(\theta_G, |\lambda, Z\rangle) = |fake\rangle)\ ) \tag{25}$$

where the labels are countable, and Λ is the cardinality of them. The expectation value is equivalent to the following that will appear on the measurement qubits of the discriminator [15]:

$$Z \equiv |real\rangle\langle real| - |fake\rangle\langle fake| \tag{26}$$

2.2.2. Quantum Data, Classical Generator, and Quantum or Classical Discriminator

The main part of GAN systems is to train based on statistical distributions and probabilities. Suppose we have quantum real data (such as chemistry quantum data) that is generated by a quantum system and measured by $P_{true}\ (x)$ for outcomes x. Since the generator is classic and the data are quantum, because of the quantum advantage such a generator will not be able to generate such data P_g that exactly fits the real data unless it has high-scale resources. Thus, we can have the minimum error in the positive part of $P_{true} - P_g$. This occurrence shows us the advantage of quantum, and it means there is still no classical algorithm that can compete with quantum states except exponential [2].

Lloyd and Weedbrook say in [2] "If the discriminator has access to a quantum information processor to adjust her measurement strategy, then we conjecture that she can find the optimal measurement to discriminate between the quantumly generated data and the classically generated data. If the discriminator only has access to classical information processing, then we conjecture that she can't determine the optimal measurement".

2.2.3. Classical Data, Discriminator, and Generator Quantum

QGAN is widely favored due to the prevalence of classic datasets, including images, voices, finance, and similar categories. For the QGAN type, we need to have an embedding from classical data to quantum states $|v_i\rangle$. The important benefit of such QGAN is the power of the processor for N-dimensional vector using log N qubits while classical uses N bits. Also, the time process is for QGANs, O (poly(log N)), and the classical model is O(N^2) [2,16]. To reproduce real data, we need to know the covariance matrix that can be put forward by $C = \frac{1}{M}\sum_i v_i\, v_i^{\dagger}$ formula where M is the number of the normalized vector v_i that equates to the density matrix $\rho = \frac{1}{M}\sum_i |v_i\rangle\langle v_i|$ [2,16].

3. Architectures and Algorithms

This section outlines several methods and algorithms suggested by scientists and provides a comparative assessment of their respective merits. In this context, we highlight the contributions of Murphy et al. [17] published in 2021, which presents a groundbreaking architecture known as "Entangling Quantum Generative Adversarial Networks (EQ-GANs)". This framework represents an innovative approach to Quantum Generative Adversarial Networks (QGANs) [17]. This architecture says some restrictions of existing QGANs, such as non-convergence, fluctuations, and model collapse, by using quantum entanglement to improve adversarial training. EQ-GAN allows the discriminator to entangle true and generated quantum data, enabling more accurate fidelity measurements and guaranteeing convergence to a Nash equilibrium. The architecture covers a parameterized, unsuitable ancilla-free swap test to evaluate the similarity between true and generated quantum states.

In their 2021 work, "Quantum state fidelity," Samuel A. Stein et al. [18] introduced a Quantum Generative Adversarial Network (QGAN) that utilizes quantum modes for both the generator and the discriminator. A new measurement method, the SWAP test, was used to measure the fidelity of two quantum states as input. It is designed such that the architecture can beat classical GAN problems like mode collapse and vanishing gradients or clean up the mess or high computational cost that follows it by pushing the limits of computer science with quantum computing. QuGAN is fully quantum, and both the generator and discriminator are implemented as quantum neural networks that operate only on quantum circuits and then use quantum state fidelity-based loss functions. The generator generates quantum states from latent variables, and the discriminator measures them to real quantum states using a SWAP test. This approach ensures both parts operate in the quantum state, and it optimizes the process through gradient-based updates from fidelity measurements. Cheng Chu et al. [19] in 2023 developed the "Image Quantum GAN (IQGAN)," which represents a significant enhancement of Quantum Generative Adversarial Networks (QGANs) tailored for the processing of image data. This work introduces an innovative trainable multiqubit encoding alongside a novel quantum circuit that

successfully removes the necessity for costly operations like CNOT. It utilizes alternative gates, namely CRX, CROT, and ISWAP, for the creation of entanglement, offering a more cost-effective solution compared to traditional CNOT gates. IQGAN includes dynamic encoding and less costly entanglement gates, which help enhance the data representation of quantum computing and its efficiency on quantum computers. Hence, more experiments reveal that IQGAN is superior to the standard QGAN, while still noting that it is quite sensitive to the trade-off between the model's added layers and its exposure to quantum noise. In summary, IQGAN presents enhancements to enhance the performance of QGANs in quantum computers sensibly.

In their 2021 publication "Experimental Quantum Generative Adversarial Networks for Image Generation," He-Liang Huang et al. [19] propose two methodologies, namely Quantum Patch GAN (QPGAN) and Quantum Batch GAN (QBGAN). These methodologies enhance the efficiency of quantum resource usage, enabling improved functionality in the context of near-term quantum devices. QPGAN operates with a restricted set of qubits to generate image patches, utilizing several quantum sub-generators in conjunction with a classical discriminator. Conversely, QBGAN employs a more extensive qubit array and capitalizes on quantum parallelism to achieve efficient training through a fully quantum architecture. The experimental results indicate that although QPGAN generally outperforms its counterpart, QBGAN demonstrates superior efficiency in terms of time. Both methodologies highlight the significant potential of quantum GANs, particularly when deployed on NISQ computers, for practical applications in image processing.

The research conducted by Shu Lok Tsang et al. [20] in 2023 introduces the Parameterized Quantum Wasserstein Generative Adversarial Network (PQWGAN), a novel approach that integrates both quantum and classical computational techniques to improve the generation of high-resolution images. PQWGAN combines a quantum generator with a classical critic, reflecting the trend in many quantum applications that utilize both quantum and classical computing to enhance outcomes, similar to the methodologies employed in patch strategy QGAN and WGAN-GP [12]. The classical critic, represented by a neural network, assesses these images while guiding the generator for future improvements. Consequently, the training framework is structured using a patch strategy alongside WGAN-GP to ensure both stability and efficacy. Evaluations conducted on the MNIST and Fashion MNIST datasets indicate that the PQWGAN produces images that closely resemble authentic ones.

3.1. Entangling Quantum Generative Adversarial Networks

EQ-GAN [17] presents an innovative architecture for Quantum Generative QGANs designed to address particular challenges, including issues of non-convergence, oscillations, and mode collapse. EQ-GAN uses quantum entanglement in the discriminator to improve adversarial training, making sure of convergence to a Nash equilibrium and developing robustness against errors in noisy quantum devices. Unlike old QGANs, the discriminator in EQ-GAN entangles real data σ and fake data. $\left(\rho\left(\theta_g\right)\right)$, using fidelity-based measurements such as:

$$D_{\sigma}^{fid}\left(\rho\left(\theta_g\right)\right) = \left(Tr\sqrt{\sigma^{\frac{1}{2}}\rho\left(\theta_g\right)\sigma^{\frac{1}{2}}}\right)^2 \tag{27}$$

which guarantees superior convergence properties. The EQ-GAN training is related to a minimax optimization problem that the generator and the discriminator are optimized by using the cost function $\min_{\theta_g}\max_{\theta_d} V\left(\theta_g, \theta_d\right) = \min_{\theta_g}\max_{\theta_d}\left[1 - D_{\sigma}\left(\theta_d, \rho\left(\theta_g\right)\right)\right]$ where $D_{\sigma}\left(\theta_d, \rho\left(\theta_g\right)\right)$ is the parameterization of the swap test result. Since the loss function is not optimal, assume, the discriminator measures $|0\rangle$ states in output, if θ_d^{opt} achieves a SWAP test $\left(D_{\sigma}\left(\theta_d^{opt}, \rho\left(\theta_g\right)\right) = \frac{1}{2} + \frac{1}{2}D_{\sigma}^{fid}\left(\rho\left(\theta_g\right)\right)\right)$ then D_{σ} helps the discriminator to approach the optimal point [17,21]. A SWAP test needs two double-qubit gates such that if we have n-qubit states, 2n qubits will require more overhead. Hence, the discriminator uses a parameterized destructive ancilla-free swap test [17,21].

The EQ-GAN development increases state generation $\rho(\theta_g)$ and fidelity measurement D_σ. in an adversarial strategy. Provided the new cost function that is following, it guarantees a Nash equilibrium. Let a SWAP test with $U(\theta_d) = \exp(-i\theta_d CSWAP)$ that has control through an angle θ_d [17,22].

The input states are $\rho_{in} = |\psi\rangle\langle\psi|$ and $\sigma = |\zeta\rangle\langle\zeta|$ for the discriminator that changes into the following formula [17,22]:

$$equal HU(\theta_d)H|0\rangle_a|\psi\rangle|\zeta\rangle = \frac{i\sin\theta_d}{2}|1\rangle_a[|\zeta\rangle|\psi\rangle - |\psi\rangle|\zeta\rangle] + \frac{1}{2}|0\rangle_a\left[\left(e^{-i\theta_d} + \cos\theta_d\right)|\psi\rangle|\zeta\rangle - i\sin\theta_d|\zeta\rangle|\psi\rangle\right] \quad (28)$$

According to circuit ansatz in (29), the maximum angle between two arbitrary states in the SWAP test will be $\theta = \frac{\pi}{2}$. For example, to calculate the probability of state $|0\rangle$, the parameters of the SWAP test are determined by the angle θ.

$$D_\sigma(\theta_d,\ \rho(\theta_g)) = \frac{1}{2}\left[1 + \cos^2\theta_d + \sin^2\theta_d\ D_\sigma^{fid}(\rho(\theta_g))\right] \quad (29)$$

Choosing unsuitable parameters can lead to divergences in the loss function and the inability to optimize gradient descent. Instead of adversarially training the parametric swap test employed as a discriminator in EQ-GAN, a frozen discriminator might apply a perfect swap test with each iteration. This may also lead the generator circuit to converge on the real data. Because EQ-GAN is unknown by the specific parameter of a perfect swap test, a suitable ansatz can learn to fix coherent problems found on quantum hardware. For example, parametric gates such as conditional Z phase, single qubit Z phase, and swap angles in two-qubit entangling gates may change and fluctuate with time. The big differences in single-qubit and two-qubit Z rotation angles will substantially decrease by entering extra single-qubit Z phase compensations. The discriminator is generated with flexible Z rotation angles to minimize CZ gate errors. This allows the generator to be closer to the real data state by adjusting X and Z rotation angles [17,23].

Using frozen swap tests on EQ-GAN results in superior performance than those of traditional supervised methods on Google's Sycamore quantum processor. It finds that the state fidelity error for the perfect swap test is $(2.4 \pm 0.5) \times 10^{-4}$, while EQ-GAN achieves $(0.6 \pm 0.2) \times 10^{-4}$ in noisy environments. Additionally, EQ-GAN appeared to train reliably on data, avoiding the behaviors of excess variability seen in QuGANs. In practical applications in quantum machine learning, it uses this architecture, for example, to generate approximate Quantum Random Access Memory (QRAM). This is shown by training a quantum neural network (QNN) on EQ-GAN-generated QRAM on a two-peak Gaussian dataset, which achieves 69% accuracy compared to 53% accuracy for exact data sampling. In noiseless setups, EQ-GAN also handles challenges with vanishing gradients that have commonly been encountered when employing perfect swap tests. Equipped with dynamically optimizing the discriminator, EQ-GAN can sidestep gradient issues such that the generator converges on the true data state. Nevertheless, these architectures are presented to be scalable, yet they face limited scalability in terms of circuit depth in current quantum hardware.

To summarize, our EQ-GAN framework shows how to resolve difficult convergence and robustness challenges in quantum generative modeling while providing practical utility in quantum machine learning.

3.2. Quantum State Fidelity

Samuel A. Stein et al. [18] provide a new method for quantum generative adversarial networks (GANs) using quantum state fidelity as a metric for loss functions. The generator and discriminator in QuGAN use quantum circuits and quantum state fidelity to guide the adversarial learning process. This structure uses quantum state fidelity computed via the swap test [17] as the loss metric. Deep learning in a quantum system is often represented by groups of quantum gates that perform specialized data manipulations [24]. Parameterized gates modify the quantum state in specified methods [25]. Ref. [18] says key challenges

in classical GANs, such as mode collapse, vanishing gradients, and high computational requirements, and offers a new approach to GANs through quantum computing, namely QuGAN, achieving improved stability and efficiency. The generator and discriminator in QuGAN are implemented as quantum neural networks operating entirely on quantum circuits and guided by quantum state fidelity in the adversarial learning process. First, two states $|\psi,\rangle$ and $|\phi\rangle$ as input enter a system that measures their fidelity, and then there is an ancilla qubit that is in a $|+\rangle$ state passed to the system during this process. The result follows as [18]:

$$\begin{aligned} state_{befor} &= \tfrac{1}{\sqrt{2}}(|0,\ \phi,\psi\rangle + |1,\phi,\psi\rangle) \\ State_{after} &= \tfrac{1}{\sqrt{2}}(|0,\ \phi,\psi\rangle + |1,\psi,\phi\rangle) \end{aligned} \tag{30}$$

Next, the ancilla qubit will go by from a Hadamard gate so that the result follows as:

$$\begin{aligned} state_{befor} &= \tfrac{1}{\sqrt{2}}(|0,\ \phi,\psi\rangle + |1,\phi,\psi\rangle) \\ state = \tfrac{1}{2}(|0,\ \phi,\psi\rangle + |1,\phi,\psi\rangle + |0,\ \psi,\phi\rangle - |1,\psi,\phi\rangle) &= \\ \tfrac{1}{2}|\ 0\rangle(|\ \phi,\psi\rangle + |\psi,\phi\rangle) + \tfrac{1}{2}|1\rangle(|\ \phi,\psi\rangle - |\psi,\phi\rangle) & \end{aligned} \tag{31}$$

To measure the probability of $|0\rangle$ state, we need to square the coefficient of $|0\rangle$. Also, we talked about the fidelity of $|\psi\rangle$ and $|\phi\rangle$ states which is a number in [0.5, 1] that when two states are perpendicular to each other their inner product is 0 and their fidelity will be 0.5. (if their inner product is 1, their fidelity will be 1); then according to the descriptions, we have the following to measure the $|0\rangle$ state [18]:

$$P_{Measure}(|0\rangle) = \frac{1}{2} + \frac{1}{2}|\langle\phi|\psi\rangle|^2 \tag{32}$$

To continue, we describe how such a fidelity model works. Here are two classical and quantum computers that on the classical computer were designed quantum circuits. It will pass the parameterized quantum circuits from the classical computer to the quantum computer and the end returns calculated fidelity. First, classical data will be embedded in quantum mode with an angle encoding. Figure 5 pictures it. After that, we enter the step of parameterizing that will be performed by the quantum circuits that the first layer was utilized by $R_Y(\theta)$ gates which are single qubit gates. The second layer was applied by $R_{YY}(\theta)$ gates which are dual qubits. The third layer, the entanglement layer was run through by $CR_Y(\theta)$ (Control $R_Y(\theta)$). In this step, quantum circuits load in a quantum computer. It should be highlighted that the presence of an ancilla qubit is crucial for the fidelity calculation to ensure the result is stored. In this step, three positions will occur on the quantum computer whether optimizing the generator or optimizing the discriminator or in the end step sample G state.

The quantum circuits attempt to realize which data are real or fake by measuring the loss function of the discriminator and the generator that follows the formula for loss function. After calculating loss functions, it is time to optimize by loss functions. The common method to optimize is gradient descent, and this architecture follows similarly to classical GAN [26]. When calculating the gradient of parameters, the loss function of the discriminator with regard to the generator and real data will update itself. The loss functions for the discriminator and generator are [18]:

$$D_{loss} = E\left[\log\left(D\left(|\langle\xi,\delta\rangle|^2\right)\right)\right] + E\left[\log\left(1 - D\left(|\langle\gamma,\delta\rangle|^2\right)\right)\right] \tag{33}$$

$$G_{loss} = E\left[log\left(D\left(|\langle\gamma,\delta\rangle|^2\right)\right)\right] \tag{34}$$

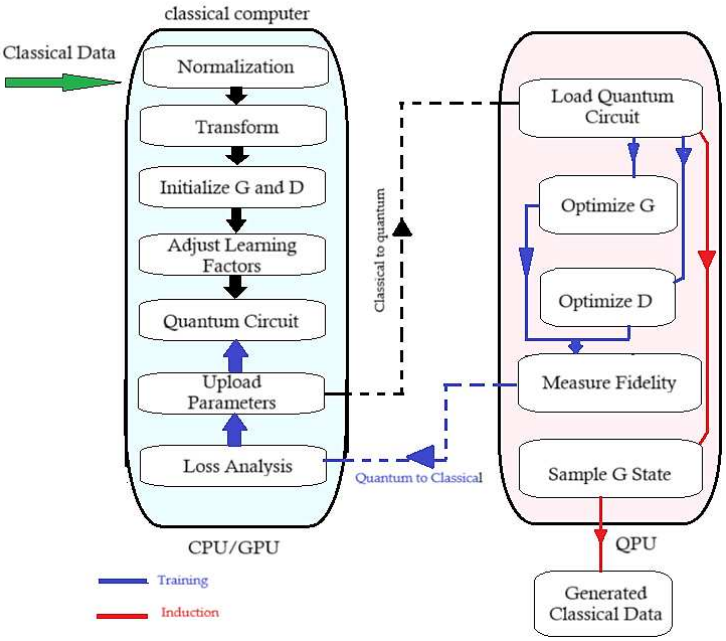

Figure 5. The structure of Quantum state fidelity.

Then the generator, according to the discriminator, will update its performance. The differential equation that follows helps calculate the gradient [18].

$$\frac{\delta f}{\delta \theta} = \frac{1}{2}\left(f\left(\theta + \frac{\pi}{2}\right) - f\left(\theta - \frac{\pi}{2}\right)\right) \tag{35}$$

In [18], QuGAN has four main components: The generator circuit, discriminator circuit, data loading circuit, and an ancilla qubit for fidelity measurements are also present. Measurement on noisy intermediate-scale quantum (NISQ) devices is simplified by the main qubit ancilla. The resulting design also provides robust performance. It evaluated the model on the MNIST dataset, with dimensionality reduced from 784 to 4 by Principal Component Analysis (PCA) because of the limitation of the current quantum hardware. Inverse PCA was used to transform transformed back-generated quantum states into images for visualization. The performance of the model was evaluated with Hellinger distance, a probability distribution comparison metric. It also shows that classical GANs with small parameter counts can achieve the same level of generalization as QuGAN (Hellinger Distance 0.1951). For instance, classical GAN with 20 parameters can have a distance of 0.4448, but QuGAN with 10 parameters for the generator and discriminator. QuGAN has better stability and consistent convergence compared to other quantum GANs such as Qi-GAN and TFQ-GAN. QiGAN learns neither consistently nor reliably, with less than 0.5% improvement, TFQ-GAN still less than 48.33% AVD (Hellinger Distance), and QuGAN has reached a minimum of 48.33% AVD (Hellinger Distance) over 25 epochs. Experiments on the IBM-Q Melbourne quantum processor further validated the stability of QuGAN. Real quantum hardware introduces inherent noise, which QuGAN still performed robustly, achieving a Hellinger distance of 0.280 in simulation and 0.337 on the actual quantum processor due to hardware noise.

Per experiment, unlike classical GANs, QuGAN reduces parameter counts by 94.98%, with similar or better performance. Another virtue of this quantum GAN architecture is that it also helps avoid the instability and inefficiency of existing quantum GAN architectures. It illustrates the model's capability for effective learning of complex data distributions by generating MNIST samples that converged to recognizable digits. Nevertheless, the paper notes limitations associated with present quantum hardware constraints, like qubit limits and noise.

3.3. Image Quantum GAN (IQGAN)

Some works on QNN inspire GANs [27–29], while Chu et al. [30] bring up that their method is focused on where there is classical data while the generator and the discriminator are in quantum mode. The dataset is the MNIST with NISQ. They claim that standard quantum encoding put forward by the previous methods declines potentially generating QGAN; therefore, they provide multiqubit, more trainable encoding [17,18]. Then, they put forward a generator circuit ansatz that decreases CNOT gates (circuit depth). These improvements help implement NISQ devices and have superior performance on IBM computers as well. The previous method of classical data like x was a vector as input that was embedded in a quantum state like $|\psi_\sigma\rangle$, then the generator generates synthetic $G(\theta_g)$ from $|\psi_\rho(\theta_g)\rangle$ through quantum circuits. Next, the discriminator with $D(|\psi_\sigma\rangle, |\psi_\rho\rangle)$, and the SWAP test will calculate the fidelity of real data $|\psi_\sigma\rangle$ and fake data $|\psi_\rho\rangle$. Meanwhile, PCA [31] was used for dimension reduction in the previous method because of a double qubit as input and a single qubit in the circuit block, some problems in unfit should be adjusted. The quantum fidelity quality of images was low through angle encoding because of the existence of periodic sine and cosine periodic functions that cause distorted images [18]. IQGAN has two novel ideas the first one is about encoding data with modification $\arcsin(x.\theta_s)$ to embed to a quantum state unlike [32] that use *arctan* and the other one is the use of a specific circuit without cost CNOT or double qubit to perform entanglement. The first novel idea is about embedding data that the previous method used from fixed encoding ($\arcsin\sqrt{x}$) while in this method was proposed dynamic encoding and trainable that has a theta parameter that appeared in $\arcsin(x.\theta_s)$ function as commonly. The set theta includes parameters that were trained before where the clusters of data were separated. The encoder has been trained using a preparing dataset CIFAR10 [33] which $T = \{(x_i,\ y_i)|0 \le i \le N-1\}$, where x_i is an n-dimensional vector and y_i is the label. They create a quantum set $\sigma_{y_k} = \frac{1}{N_k}\sum_{j=0}^{N_k-1} |\psi_\sigma(x_j)\rangle\langle\psi_\sigma(x_j)|$ by randomly choosing N_k inputs from class y_k, $\left(T_k = \left\{(x_j,\ y_k)\middle|0 \le j \le N_k - 1\right\}\right)$, and sending them into the encoder. Then it trains the encoder to find the optimum values θ_s^* to maximize the distance of σ_{y_k} and σ_{y_m} at $k \ne m$. After that, IQGAN employs trained values for $(\arcsin x.\theta_s^*)$ as starting parameters. As a test dataset to maximize separability among data clusters, θ_s is pre-trained on datasets such as CIFAR10. Such an adaptive approach improves the expressivity of quantum neural networks (QNNs) and also provides a robust initialization for training GANs. The second novel idea is this method using parametric gates like CRX, CROT$(\varphi, \theta,\ \omega)$, and a fixed ISWAP instead of CNOT to create entanglement because as we know CNOT gate is very costly for us whether hardware or implement. IQGAN devises the compact quantum generator that does away with complex two-qubit entanglement gates, such as CNOT gates. IQGAN replaces these gates with simpler or no entanglement gates so that the circuit depth is greatly reduced, and hardware is significantly less required without degrading performance. The fidelity achieved by this compact generator is experimentally compared with that of more complex designs, and hardware costs are reduced by 6x. The generator and discriminator of IQGAN are trained with a fidelity-based loss function. The training objective minimizes the difference between real and generated data states, measured by quantum fidelity:

$$L(\theta_g) = \min_{\theta_g}\left[1 - \left\langle \psi\sigma \middle| \psi\rho(\theta g)\right\rangle^2\right] \tag{36}$$

where $\left\langle \psi\sigma \middle| \psi\rho(\theta g)\right\rangle^2$ quantifies the similarity between the real state ψ_σ and generated state ψ_ρ. The results show using IBM's 5-qubit quantum processor (IBM Quito) [34] and PennyLane simulators [35] high performance of IQGAN compared to old quantum GANs, such as QuGAN 2021 and EQ-GAN. Compared to previously developed methods, IQGAN produces MNIST images that are more clear and accurate. Outputs of QuGAN 2021 are distorted and blurred, especially for digit '3', whereas EQ-GAN has an unsatisfactory image quality while IQGAN yields stable and clear quantum images, including on the

real quantum hardware. It converges in 10 iterations and has a final fidelity score between generated quantum states and target quantum states equal to 0.966, thus being much better than other models. Furthermore, it is transformed using its trainable encoder, which speeds up learning and provides improved fidelity by 0.039. In addition, the trainable encoder outperforms its fixed counterpart by up to 4 percent for downstream classification tasks, especially if the input size is large and the model is complex.

To evaluate, the dataset is MNIST [36] and used PennyLane library [35] to implemented on the simulator and used 5-qubits on an IBM quantum computer [34]. It compares EQGAN [17] and IQGAN with ADAM optimizer [37], learning rate = 0.001, batch size = 32, and epochs = 30 will train also The learning rate is scheduled by CosineAnnealingLR with a T_{max} of 30.

The function of TE increases when the number of qubits increases, for example on MNIST-4 for input sizes of 2×2 and 4×4. The next improvement of TE increases with the model complexity, for instance, MNIST2 and MNIST-4 with both 16-qubit inputs. Both methods show consistent convergence after ten iterations, demonstrating that IQGAN achieves a high more than 0.92 state overlap with the target inputs. The suggested TE produced less fidelity (e.g., 0.85 vs. 0.90) in early rounds, but the final taught fidelity is ∼0.039 greater (0.966 vs. 0.927) than FE. Also, the acquired fidelity of TE rises quicker than that of FE, mostly due to the TE circuit's greater flexibility and adaptability and its richer expressive potential. The original pictures are downsampled to 1×2, 1×4, 1×6, and 1×8 vectors using the PCA technique. However, the quality of the target pictures improves monotonically with input size, and the fidelity of taught images to original inputs declines. A quantum system with more qubits and gates is more sensitive to noise. In this scenario, a more advanced IQGAN circuit raises the model's expressive capability, but quantum noise from excessive overhead diminishes overall fidelity.

Ultimately, IQGAN offers, on average, enhanced image quality relative to other sophisticated quantum GANs, alongside a faster convergence rate. It also demands reduced hardware resources and demonstrates improved noise tolerance for quantum generative modeling on NISQ devices. Possible further research areas include further application of IQGAN on bigger data sets such as CIFAR10 and ImageNet [38], as well as extensions of the generator network and the activation of superior techniques for error reduction.

3.4. Experimental Quantum Generative Adversarial Networks for Image Generation

He-Liang Huang et al. [19], in Experimental Quantum Generative Adversarial Networks for Image Generation (Abbreviation EXQGAN), provided two strategies based on the number of qubits and features. The goal of this method is the optimal use of quantum resources and the providing of high quality in image generation. These two methods are quantum patch GAN (QPGAN) which $N < \lceil \log M \rceil$ and quantum batch GAN (QBGAN) which $N > \lceil \log M \rceil$ where N is the qubits count and M is the feature dimension count. Specifically, the quantum patch GAN was made to use quantum resources better and generate high dimensional features, but quantum batch GAN is to train in parallel because there are enough quantum resources.

The quantum patch GAN where $N < \lceil \log M \rceil$ has the quantum generator and the classical discriminator such that because of lack of resources, the discriminator is classic. In this situation, the generator consists of several sub generators $\{G_t\}_{t=1}^{T}$ where T is $O\left(\frac{\lceil \log M \rceil}{N}\right)$. That is useful for a distributed quantum system to train in parallel or a single qubit system for linear training, while in classical GAN this method is equivalent mini batch [39]. Each qubit has the task to generate a part of the image that at the end of generating composes together and provides the final image. G_ts are parameterized with $U_{G_t}(\theta_t)$ that output is a state $|G_t(Z)\rangle$ ($|G_t(Z)\rangle = U_{G_t}(\theta_t)|Z\rangle$). Let $\widetilde{x}$ be the generated sample and x be real data, the loss function that has been used to optimize θ and γ is follow as:

$$\min_{\theta} \max_{\gamma} = E_{x \sim P_{data}(x)}[\log D_{\gamma}(x)] + E_{z \sim P(z)}(\log\{1 - D_{\gamma}\,[G_{\theta}\,(z)]\}) \tag{37}$$

where $P_{data}(x)$ is the distribution of real data and $P(z)$ is the probability of Gaussian distribution to enter the generator. The circuit of the quantum generator encodes data with two RY gates as input, then entangled qubits via CNOT gates and measurements for classical output. Figures 6 and 7 show quantum generator circuits and patch QGAN structure, respectively. To test the performance of the quantum patch GAN, it runs on a superconducting quantum processor to generate handwritten digit images for "0" and "1". The superconducting quantum processor has 12 qubits on a one-dimensional (1D) chain, with a maximum of six neighboring qubits selected during the experiment. About fidelities of single-qubit and controlled-Z gates are around 0.9994 and 0.985, respectively. The training 1000 examples are 8×8-pixel images with M = 64. In the experimental settings for quantum patch GAN, we set T = 4, N = 5.

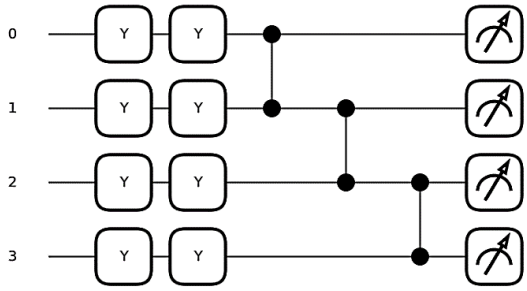

Figure 6. Scheme of quantum generator in quantum patch GAN.

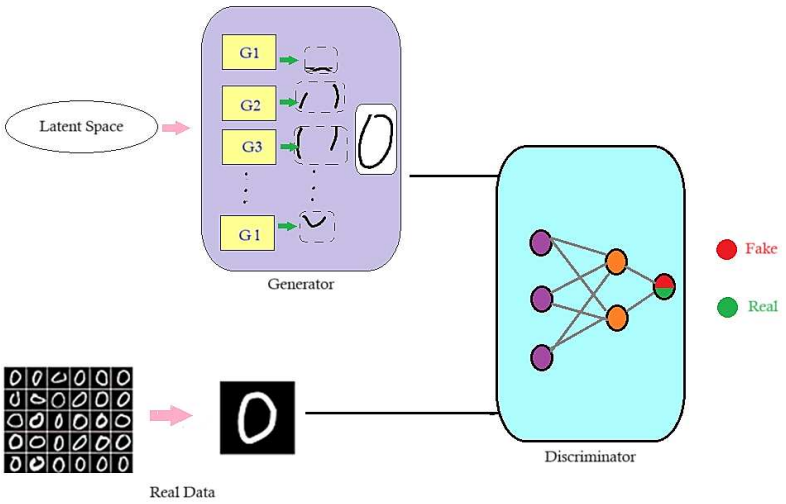

Figure 7. Scheme of quantum patch GAN.

In the quantum batch GAN where $N > \lceil \log M \rceil$ both the generator and the discriminator are in quantum mode because enough resources are available. The loss function is the same as (35), but the method work is different. First, the N qubits available are divided into two parts: the feature register (R_F) and the index register (R_I). The R_F will encode with N_F qubits that consist of the feature information, and the R_I records with are encoded with $\frac{1}{\sqrt{N_e}} \sum_i |i\rangle_I |x_i\rangle_F$ by the use of the amplitude encoding method. Having an index register helps us work on N_e to gain the gradient more efficiently. Since one can train N_e samples through superposition simultaneously, the training process is more efficient because of the existence of quantum batch GAN. On the other hand, quantum batch GAN uses the quantum discriminator for binary classification, measuring one qubit should be sufficient for determining between real and fake images. As a result, quantum batch GAN requires very few observations. To validate its generative capabilities, Quantum Batch uses GAN to generate gray-scale bar images. Experimental parameter settings T = 1, N = 3, $|B_k|$ = 1 (or $N_I = 0$), and the quantum generator has a total of nine trainable parameters [40].

Quantum discrimination has a total of 12 trainable parameters. Quantum batch GAN can achieve the same FD (the Fréchet distance) [41] score as quantum patch GAN, indicating that it can be used to solve image processing challenges. This indicates a significantly lower performance of quantum batch GAN compared to quantum patch GAN, due to its quantum selectivity major. Due to the restricted number of training parameters used in the medium, we can see the decreased performance of the quantum batch GAN against the quantum patch GAN. In summary, this work is a first step towards connecting quantum and traditional generative learning. It shows that near-term quantum devices could be used for important applications, like generating real-world digital images. Additionally, these findings suggest that quantum GANs might offer advantages, such as reducing the number of training parameters and improving computational efficiency in the NISQ environment [42]. Finally, the outcomes revealed that the proposed QGAN carried out small-scale grayscale images with high realism. Finally, the performance of the QGAN was evaluated by the fidelity score of 0.912 of the fake quantum states with the real ones when the number of training iterations was about 15. The convergence of the three parameters did not significantly change through training, apart from slight fluctuations caused by increased hardware noise. Paired with the evaluation of the generated samples based on comparisons, the performance of the QGAN was shown to improve across the iterations to be a near-likeness of the structure pattern in the original data. In the aspect of hardware efficiency, fewer qubits and fewer numbers of circuits were implemented into the model and hence were fine-tuned to meet NISQ device constraints. On this account, the QGAN outcomes were found to be adequately favorable with the standard GAN models while using the limited data set considered in the research. However, they found that general scaling to larger, analytically intensive datasets is still not possible due to some constraints in quantum hardware that exist today.

3.5. Hybrid Quantum–Classical Generative Adversarial Network for High-Resolution Image Generation

In their research, Shu Lok Tsang et al. [20] propose the continuation of the quantum patch GAN, introducing a novel aspect by merging it with the Wasserstein GAN gradient penalty (WGAN-GP). This integration is intended to facilitate the generation of high-resolution grayscale images measuring 28×28 pixels. Quantum sub-generators, based on quantum computation, are utilized on image patches to reduce the number of trainable parameters yet achieve approximately the same image quality as classical WGAN-GP models. It has better convergence inherited from WGAN, and the other is improved in training and not using PCA for dimension reduction. In this method, the generator receives an N-dimensional vector $\boldsymbol{z}$ from a specific probability distribution such as uniform or Gaussian (P_z) then a layer R_Y rotation encodes $\mathbf{z}$ that will have $|z\rangle$:

$$|z\rangle = R_Y^1(z_1)R_Y^2(z_2)\dots R_Y^N(z_N)\,|0\rangle^{\otimes N} \tag{38}$$

where z_i is the rotation angle and $|0\rangle^{\otimes N}$ is the initial state. After going through layers that parameterize each qubit with $R(\phi,\theta,\omega) = R_Z(\omega)R_Y(\theta)R_Z(\phi)$ [43] then each qubit will be entangled with an adjacent qubit by the CNOT gate. This order (ZYZ) is suitable when you aim to implement them on real quantum computers. The parametric layers are unitary matrices like $U(\boldsymbol{\phi}_i, \boldsymbol{\theta}_i, \boldsymbol{\omega}_i)$ that apply on state $|z\rangle$ that generates a part of the image by sub-generator [20].

$$\left|\psi_{G_i}\right\rangle = U(\boldsymbol{\phi}_i, \boldsymbol{\theta}_i, \boldsymbol{\omega}_i)|z\rangle \tag{39}$$

As we know, the nonlinearity for activation function in neural network classic is a strong benefit. Therefore, we apply a partial measurement like M on ancilla qubits whereas after measurement we have the situation of qubits as follows:

$$|\psi_D\rangle = Tr_A\left(\frac{M \otimes I|\psi_{G_i}\rangle\langle\psi_{G_i}|}{\langle\psi_{G_i}|M \otimes I|\psi_{G_i}\rangle}\right) \tag{40}$$

If the M is replaced with $|0\rangle\langle 0|^{\otimes A}$ then we have:

$$\rho_D = Tr_A\left(\frac{\left(|0\rangle\langle 0|^{\otimes A}\right) \otimes I|\psi_{G_i}\rangle\langle\psi_{G_i}|}{\langle\psi_{G_i}|\left(|0\rangle\langle 0|^{\otimes A}\right) \otimes I|\psi_{G_i}\rangle}\right) \tag{41}$$

The existence $|\psi_{G_i}\rangle$ in the denominator and numerator causes nonlinearity. After measurement of each qubit as a sub-generator we have:

$$G_i(z) = \left[\rho(0),\ \rho(1), \ldots, \rho\left(2^{D-1}\right)\right] \tag{42}$$

We want each part of the generator's output to fall between 0 and 1, so they can be considered pixel values. While we could technically use the probabilities as pixel values, doing so would cause issues due to the normalization constraint, preventing us from obtaining the desired pixel values. Therefore, we use the following formula:

$$G_i'(z) = \frac{G_i(z)}{\max(G_i(z))} \tag{43}$$

Because the quantum circuit's outputs are a power of 2 s, we preserve the initial $\frac{H \times W}{P}$ pixels to construct the necessary patch dimensions. Finally, the output from all the sub-generators is combined to generate a picture of dimension $H \times W$.

$$G(z) = \left[G_1'(z),\ \ldots, G_p'(z)\right]^T \tag{44}$$

The whole of this process is collected in Algorithm 2.

Algorithm 2. To generate an Image from the Patch Quantum Generator [20]

Input: Image dimensions $H \times W$, number of ancilla qubits A, number of data qubits D, number of sub-generator layers L, number of patches P, generator parameters $\theta = [\theta_1, \ldots,\ \theta_P]$, latent variable $\boldsymbol{z}$.
for $i = 1, \ldots,\ P$ **do**
$|\psi_i\rangle \longleftarrow U_{i,\ L,\ \theta_i}|z\rangle$
$\rho_D \leftarrow Tr_A\left(\frac{\left(|0\rangle\langle 0|^{\otimes A}\right)\otimes I|\psi_{G_i}\rangle\langle\psi_{G_i}|}{\langle\psi_{G_i}|\left(|0\rangle\langle 0|^{\otimes A}\right)\otimes I|\psi_{G_i}\rangle}\right)$
Measure ρ_D in computational basis to obtain
$G_i(z) \leftarrow \left[\rho(0),\ \rho(1), \ldots, \rho\left(2^{D-1}\right)\right]$
$G_i'(z) \leftarrow \frac{G_i(z)}{\max(G_i(z))}$
Discard excess pixel values to obtain
$G_i''(z) \leftarrow G_i'(z)\left[: \frac{HW}{P}\right]$
end for
$G(z) = \left[G_1''(z),\ \ldots, G_p''(z)\right]^T$
Return $G(z)$

In WGAN-GP, the critic's role is to take an image and calculate the Wasserstein distance. There are challenges for the quantum critic. First, loading high-dimensional data into a quantum circuit is difficult because it demands a lot of quantum resources. Second, the learning process for quantum circuits is not as well understood as it is for classical neural networks. The training algorithm for PQWGAN is similar to the WGAN-GP training

algorithm but uses quantum generators (see Algorithm 3). With the generator divided into sub-generators, we need to update the parameters of each sub-generator based on the loss calculated from the entire image by gradient [44]. The loss function $L(w, \theta)$ depends on both the critic and generator parameters. If we have N_G sub generators, each with n parameters, the generator's parameters can be represented as a vector:

$$\theta = [\theta_1, \ldots, \theta_{N_G}] = [\theta_{1,1}, \ldots, \theta_{1.n}, \ldots, \theta_{N_G,1}, \ldots, \theta_{N_G,n}] \tag{45}$$

The gradient of the j-th parameter of the i-th sub-generator is:

$$\frac{\partial \langle L(\omega,\theta)\rangle}{\partial \theta_{i,j}} = \frac{1}{2}\left(\left\langle L\left(\omega, \theta_{1,1}, \ldots, \theta_{i,j} + \frac{\pi}{2}, \ldots, \theta_{N_G,n}\right)\right\rangle - \left\langle L\left(\omega, \theta_{1,1}, \ldots, \theta_{i,j} - \frac{\pi}{2}, \ldots, \theta_{N_G,n}\right)\right\rangle\right) \tag{46}$$

Algorithm 3. PQWGAN training algorithm [20]

Input: Gradient penalty coefficient λ, Critic iterations per generator iteration n_C, number of epochs n_{epochs}, batch size m, Adam hyperparameters $\eta_1,\ \eta_2,\ \beta_1,\ \beta_2$. Initialize critic parameters ω, sub-generator parameters θ.
for $epoch = 1, \ldots,\ n_{epochs}$ **do**
for $t = 1, \ldots,\ n_C$ **do**
for $i = 1, \ldots, m$ **do**
Sample real data $x \sim P_{data}$, latent variable $z \sim p_z$ random number $\epsilon \sim U[0,1]$
$x' \leftarrow quantum - generator\ (\theta, z)$
$\hat{x} \leftarrow \epsilon x + (1-\epsilon)x'$
$L_D^{(i)} \leftarrow D(x') - D(x) + \lambda(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2$
end for
$\omega \leftarrow Adam\left(\nabla_\omega \frac{1}{m} \sum_{i=1}^{m} L_D^{(i)}, \omega,\ \eta_1,\ \beta_1,\ \beta_2\right)$
end for
for $i = 1, \ldots, m$ **do**
Sample real data $x \sim P_{data}$, latent variable $z \sim p_z$
$x' \leftarrow quantum - generator\ (\theta, z)$
$L_G^{(i)} \leftarrow -D(x')$
end for
$\theta \leftarrow Adam\left(\nabla_\omega \frac{1}{m} \sum_{i=1}^{m} L_G^{(i)}, \theta,\ \eta_2,\ \beta_1,\ \beta_2\right)$
end for

To evaluate, the datasets are the MINST and the FMINST [45]; the first one is handwritten digits, and the second one is clothes, shoes, and accessories. All images are 28×28 pixels and 1000 samples of each class are used as a train set. The implementation was used with Python 3, PyTorch, and PennyLane. The classical discriminator in PQWGAN with the role of a critic is fully connected with two hidden layers with 512 and 256 neurons. Also, the activation function is the ReLU, and the output layer has a neuron without an activation function. While, the WGAN-GP has three hidden layers with 256, 512, and 1024 neurons, and the activation function is the ReLU. Also, the output layer has tanh as the activation. About hyperparameters, $\lambda = 10$, $n_C = 5$, the optimizer is Adam [37], $\beta_1 = 0$, $\beta_2 = 0.9$, and learning rates are 0.1 and 0.0002. As an experience, the uniform distribution is better than Gaussian, so the uniform distribution with [0, 1] is used for implementation. The batch size is 25 and the iterations are 600. For the MNIST dataset, digits 0 and 1 will be generated, and for the FMNIST dataset, T-shirts and trousers.

The results show its efficacy in generating high-resolution (28×28) grayscale images with a reduced number of trainable parameters than classical models. It shows that PQWGAN was able to generate high-quality images similar to classical WGAN-GP models for the MNIST dataset, such as digit generation "0" and "1" with the trainable parameters. The model also does well on more complex datasets, the Fashion MNIST (FMNIST) and

tri-class MNIST (digits '0', '1', and '3'), while more complexity causes lower performance in terms of keenness, although this required more layers and qubits to improve it. The results show the state of patches, qubits, layers, and total parameters: With increasing the number of patches, the number of qubits and layers reduces, and total parameters are nearly equal. For example, when there is one patch with 10 qubits and 153 layers, the number of parameters is 5049. Whereas, with 28 patches with 5 qubits and 10 layers, the total parameters are still 5040. Increasing layers with the same number of patches, such as 28 patches and 15 layers, the parameters will be 7560. Therefore, increasing the number of qubits, like 10 qubits, needs more layers (153) to maintain a consistent parameter count (5049). This balance indicates a swap between qubits and layers while ensuring parameter stability across configurations. There is an indirect relation between the number of layers and parameters; it means more parameters with fewer layers for NISQ devices. In particular, this makes scalability and resource-efficient quantum generative modeling optimized for particular tasks and with hardware limitations.

For example, PQWGAN is shown to have encoded the underlying data distribution in the latent space successfully. On the MNIST dataset, more complexity needs further challenges, such as tri-class MNIST and FMNIST datasets, but PQWGAN still has good performance despite the need for more time.

4. Discussion and Comparative Analysis of the Papers

These papers focus on the same topics: improvement of QGAN architectures for noisy intermediate-scale quantum (NISQ) devices and optimizing them under limitations such as the number of qubits and noisy environments. In all studies, image generation is a main task, from simple 2×2 grayscale images to more complex 28×28 grayscale MNIST and Fashion MNIST (FMNIST).

All of these frameworks focus on computational efficiency by moving from classical GANs in reducing the number of trainable parameters and maintaining better performance. For compatibility with existing hardware, they use quantum circuit optimization techniques for minimum depth in the circuits and the minimum number of gates. Robustness to quantum noise also appears in the form of decoherence and gate errors that each paper tries to provide solutions for. There is a summary of the papers in Table 1.

Collectively, the papers look into different quantum and hybrid quantum (classical) generative adversarial networks (QGANs) for image synthesis and optimization of architectures for noisy intermediate-scale quantum (NISQ) devices. While common issues such as scalability, noise resilience, and efficient resources are shown, each paper states different advancements. While they differ in detail, all of them use quantum properties like superposition and entanglement to learn better.

EQ-GAN introduces a fidelity-based loss function for convergence and stability during training. It proposes quantum-specific fidelity metrics to generate data that are stable but have limitations in scalability because it focuses on synthetic datasets and QRAM-based image generation.

QuGANs use the loss metric in their model, which is quantum state fidelity, and it generates 2×2 grayscale images with relatively low parameter counts (~5040). Although QuGAN obtains strong fidelity scores and robustness in quantum noise, it is limited by the small datasets and scalability.

IQGAN expands these methods by including a trainable quantum encoder and a compact generator that lowers hardware costs without reduction in performance. This structure generates 8×8 grayscale images well. It is both scalable and efficient while still using moderate resources. Moreover, its robustness to noise makes it a good candidate for NISQ devices.

EXQGAN is built with the concept of patching, which divides the generator into sub-generators that also help optimize resources. This method shows good scalability for generating 8×8 grayscale images of handwritten characters and bar patterns, and with distributed quantum resources, tries to save resources.

Table 1. A summary of the papers.

Aspect	EQ-GAN	QuGAN	IQGAN	EXQGAN	PQWGAN
Main Contribution	Introduced fidelity-based loss for convergence.	Quantum fidelity as the loss metric.	Trainable encoder and compact generator design.	Two strategies: Quantum Patch GAN and Batch GAN.	Hybrid quantum-classical approach for high-resolution images.
Datasets	QRAM state generation, synthetic samples.	2×2 grayscale images (MNIST).	8×8 grayscale images (“0” and “1”).	Handwritten digits (“0” and “1”) and bar images.	MNIST and Fashion MNIST (28×28 grayscale).
Scalability	Limited to QRAM and synthetic datasets.	2×2 image size, limited scalability.	Efficient for 8×8; challenges with higher complexity.	Scalable through patching; challenges in high parameter efficiency.	Best scalability, up to 28×28 high-resolution images.
Number of Parameters	~5000	~5040	~5000–7600	~5000–7600	~5000–5500 (5376)
Generator Design	Fidelity-based quantum generator.	Parameterized quantum circuits.	Compact generator with reduced entanglement gates.	Divided generator into sub-generators.	Hybrid quantum-classical patches.
Discriminator Design	Quantum fidelity measurements.	Quantum discriminator.	Classical or quantum discriminator.	Classical for Patch GAN, quantum for Batch GAN.	Classical discriminator in patches.
Performance Metrics	Fidelity, convergence stability.	Fidelity, Hellinger distance.	Image quality, hardware efficiency.	Fréchet distance (FD), efficiency.	High-resolution quality, latent space interpolation.

PQWGAN shows the improvement of scalability, as hybrid quantum-classical methods are used to generate high-resolution, 28×28 grayscale images from MNIST and Fashion MNIST datasets. PQWGAN combines quantum sub-generators with classical patches using a combination of Wasserstein GAN with the Gradient Penalty (WGAN-GP) and then generates high-quality images with fewer parameters than their classical counterparts.

Though PQWGAN succeeds when it works on larger datasets, it needs more hardware for more complexity. Their key commons are their focus on improvement to NISQ limitations, circuit design optimization, and using the lower trainable parameters while increasing performance. The topic of noise resilience and finding ways to minimize the impact of decoherence and gate error are followed in all papers. However, scalability varies significantly.

EQ-GAN and QuGAN work on small datasets, while IQGAN, EXQGAN, and PQWGAN talk significantly higher resolution and scale to higher scales; also, PQWGAN is making the most significant improvements in scalability. IQGAN and EXQGAN affect the balance between parameter counts and image qualities successfully based on compact designs and patch-based strategies that are suitable for limited hardware.

Hybrid methods focus on higher-resolution tasks that are more complex and can be used by PQWGAN. Performance metrics like fidelity scores, Fréchet distances, and latent space interpolation show different strengths. PQWGAN achieves the best results in high-resolution generation, while IQGAN and EXQGAN balance efficiency and quality for moderate resolutions. Table 2 shows the noise resilience and stability of complex data based on performance metrics.

Table 2. Comparison of Performance Metrics, Noise Resilience, and Scalability Across QGAN Models.

Paper	Performance Metric	Noise Resilience	Scalability to Complex Data
EQ-GAN	Fidelity, convergence	Moderate	Low
QuGAN	Fidelity, Hellinger dist.	Moderate	Low
IQGAN	Image quality, FD score	High	Moderate
EXQGAN	FD score	High	Moderate
PQWGAN	Resolution, interpolation	High	High

5. Challenges and Limitations

The PQWGAN model identifies challenges related to data complexity and convergence, suggesting that future research should focus on optimizing quantum circuits and scaling up quantum models as quantum technology advances. Both EQ-GAN and QuGAN emphasize the need to address noise-related issues and refine the training process to enhance convergence and generative performance, especially in more complex scenarios.

QPGAN and QBGAN underscore the potential for optimizing quantum GAN architectures, particularly in managing larger datasets and mitigating issues such as "barren plateaus" in the parameter space. Challenges arising affect QGANs in a great way due to the following reasons: leakage of information in quantum systems and complexity arising since information may leak in the intermediate process, resulting in various challenges to QGANs. Present-day quantum computers have problems such as limitations in the coherence times of qubits; current gates have errors; all these factors affect the performance and reliability of the systems. The QGANs are affected in generating sophisticated and high-resolution images because of the limited underlying qubits of current quantum computers. Moreover, it could be said that quantum noise influences computational processes, which means that the quality of the generated images could be compromised. Training stability is another major challenge that organizations face concerning big data analytics. Because of the adversarial structure in GANs, the training is unstable per se, and the QGANs suffer from quantum noise and errors further deteriorating the stability. Data representation also presents challenges in this regard. There are issues with efficiently loading classical image data into registers of a quantum system while preserving information. It is also seen that high-dimensional image data needs to be processed carefully to avoid exponential resource consumption in quantum form. Demanding requirements are necessary to create QGANs, which makes demand rather high.

To be precise, high-q image synthesis requires many qubits and some quantum gates, which are not available now at present. The assessment of QGANs may be problematic and the benchmarking much more so. Since there are no standards for comparing QGANs to classical types of GANs, it is necessary to create them in order to understand the strengths and weaknesses of the method proposed. Algorithm design and selection shall be considered with great attention. This result is obtained because effective QGANs might require hybrid quantum-classical approaches, in which some calculations are performed using classical computers.

The greatest challenge in designing such combinations of these algorithms is to identify which will be more efficient. Similar to classical GANs, QGANs are also prone to overfitting the training data and tend to develop poor generalization when tested on new data. It is always a question of how to make the QGANs generalizable. Moreover, the existing quantum systems can only process small data sets, which in a way hinders the learning of complicated image distributions in QGANs.

When it comes to applying the ideas to solve global issues, there may be some measures paid to ensure trainability and avoid reaching a barren plateau [46]. The optimization of a minimax loss in Equation (17) encounters barren plateaus particularly, what effects various loss functions and approaches to optimization have on quantum GANs. Understanding this topic to the greatest extent enables the development of more efficient quantum GANs. An approach to overcome barren plateaus may be the proposal of a barren plateaus-

immune ansatz [47,48], instead of a hardware-efficient ansatz for the implementation of the corresponding quantum generator or discriminator. Moreover, the idea of the suggested quantum patch GAN shows that there is a way to get rid of barren plateaus and noise.

Quantum patch GAN training ability can be justified with the help of the decomposition of large-size issues into ones of a smaller size. Because of these reasons, one perhaps exciting research area to explore is to carry out experimental studies of quantum GANs on larger data sets. It should be noted that even though quantum GANs are capable of partial optimization of quantum system imperfections, there is a main guideline on how to enhance the performance of quantum GANs which is to contribute to the constant enhancement of quantum processors, for instance, the extent of qubits, the level of connectivity, the degree of system noise, and the period of decoherence time but in term of time it takes time. As a part of the study to understand the usefulness of quantum GANs, it can implement the structure on quantum computers of continually growing capabilities to solve various complicated real-world generation problems.

6. Conclusions

The discussed papers demonstrate several architectures, properties, and developments in quantum GANs (QGANs) for generating images and point to helpful paths for improving their performance in quantum machine learning. Each method has strengths and weaknesses, which suggest ways to improve scalability, noise resilience, and parameter efficiency and provide for superior and more robust quantum generative modeling.

In terms of accuracy and convergence, EQ-GAN and QuGAN focus on convergence and accuracy. An improvement of 53% accuracy to 69% on sampling on variational QRAM is obtained by EQ-GAN. Additionally, it demonstrates high convergence capability, even globally optimal Nash equilibrium in noisy environments, and is higher than other QGAN models. The QuGAN is also trying to minimize Hellinger distance and achieves a 48.33% reduction of 25 epochs that better test performance on IBM-Q "Melbourne". It also shows that the Quantum Patch GAN (QPGAN) and Quantum Batch GAN (QBGAN) have high fidelity and efficiency under limitations on current quantum resources. While QPGAN is good at structured image generation, QBGAN is efficient with training using quantum parallelism.

In terms of parameter efficiency, it is one of the most important developments in QGANs. It shows that PQWGAN has high-quality image generation with 5376 trainable parameters (three orders of magnitude fewer than 1.46 M for classical WGAN-GP) while maintaining similar performance. Also, QBGAN has good fidelity with only nine parameters of the generator and twelve parameters of the discriminator, showing its efficiency. However, EQ-GAN and QuGAN focus on stability and robustness under noise conditions instead of parameter efficiency. The differences in the QGAN provide the different priorities that these methods may prioritize based on the applications and hardware limitations they are designed for.

In terms of training and noise resilience, robustness is important for QGAN structures, and reduction in noise is an important factor as well. EQ-GAN and QuGAN show considerable robustness to noise, ensuring their compatibility with NISQ devices. For instance, QuGAN efficiency increased when evaluating experiments on noisy platforms such as IBM-Q "Melbourne," such that QuGAN stays convergence in the long run. In addition, it is indicated that QPGAN and QBGAN are noise-resistant, where QPGAN has used features of parallel training while QBGAN has used quantum superposition for efficient training.

Finally, these papers organize how QGAN research is built from basic models of EQ-GAN and QuGAN that pay attention to stability and robustness and then go forward with the latest frameworks of PQWGAN that update hybrid quantum-classical approaches to scalability and parameter reduction. However, there are still problems such as noise influence and quantum hardware, but these papers describe what further investigations should be based on. Table 3 shows the summary of contributions and insights from QGAN research papers.

Table 3. Summary of key contributions and insights from QGAN research papers.

Aspect	Key Insights
Accuracy and Convergence	EQ-GAN achieves 69% accuracy on variational QRAM, QuGAN reduces Hellinger Distance by 48.33%. QPGAN excels in structured image generation, QBGAN uses parallelism.
Parameter Efficiency	PQWGAN achieves high quality with 5376 parameters, QBGAN uses only 9 parameters in the generator. EQ-GAN and QuGAN focus on robustness over parameter efficiency.
Image Generation Quality	PQWGAN and QPGAN generate high-quality images but PQWGAN struggles with sharpness on complex images like trousers. EQ-GAN and QuGAN focus on distribution improvement.
Training and Noise Resilience	EQ-GAN and QuGAN are resilient to noise and perform well on noisy quantum hardware. QPGAN benefits from parallel training, QBGAN uses quantum superposition for stability.
Overall Contribution	From foundational designs (EQ-GAN, QuGAN) to advanced hybrid architectures (PQWGAN), these papers address scalability, efficiency, and noise resilience for real-world QGAN applications.

7. Future Studies and Applications

There will be a need for further research on the quantum generative adversarial networks (QGANs) by incorporating other sophisticated quantum algorithms such as quantum neural networks (QNNs) and quantum reinforcement learning (QRL) for better learning of complex data distributions and better adaptability and robustness. Dynamic feedback loops of quantum-classical could also be able to optimize the training. Decoherence and gate errors are significantly important for scaling QGANs, and for this reason, error management and fault tolerance are very essential. Moreover, other distributed quantum computing strategies and existing architectures suitable for larger datasets, such as CIFAR10 and ImageNet, will enhance scalability, therefore increasing efficiency in terms of parameters.

Conditional Image Generation or Time Series models could be developed as task-specific hybrid frameworks to expand the use of QGANs, which can be complemented by latent space analysis for improving interpolation quality and understanding the impacts of quantum operations on the generative capabilities of QGANs. Public benchmarking protocols and common datasets should be set to compare QGANs with CGANs and other frameworks to accommodate scalable, robust, and comparably efficient metrics. The practical uses of QGAN are numerous and progressive.

In data synthesis and augmentation, qGANs can create new samples of data for gross classes such as healthcare, genomics, and cybersecurity and mitigate the scarcity of such datasets. In drug design, it can perform molecular modeling much faster and may help to build new compounds when coupled with quantum chemistry algorithms. Real-world applications of QGANs are expected to lie in environmental and climate modeling by mimicking patterns, the promotion of renewable energy systems by enhancement, and the prediction of natural disasters. In finance, QGANs can produce fake financial time-series data, find the best portfolio allocation methods, and emulate the market's behavior to be usable for fraud identification and risk analysis. QGANs also have future scope in super-resolution image synthesis for scenarios like medical imagery, remote sensors, autonomous vehicles, quantum radar, precise direction communication signal processing, etc.

In education, the interactive tools powered by QGAN can be used to teach teachers and other professionals what quantum computing is and how to design prototypes of algorithms within virtual spaces. Furthermore, QGANs could be useful for other combinatorial objectives such as Max-Cut and Traveling Salesman through initializing quantum approximate optimization algorithms (QAOAs) or classical optimization solvers efficiently. Specifically in the creative industries, it could produce content for music, visual

arts, and even the narratives behind a story, all of which would make the application for quantum-assisted creativity richer. Integration with gaming and virtual reality could create compelling first-person experiences. That is why, overcoming current limitations and building on distinctive strengths, QGANs can become an indispensable foundation for extending the collaboration between quantum computing and AI, machine learning, and their applications in various branches of science to technology and practical life.

Author Contributions: Conceptualization, M.P. and V.S.S.; methodology, M.P.; validation, V.S.S. and R.K.; formal analysis, R.K.; writing—original draft preparation, M.P.; writing—review and editing, R.K.; visualization, M.P.; supervision, V.S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to thank the anonymous reviewers and editors for their valuable comments and guidance to make our work more comprehensive.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *arXiv* **2014**, arXiv:1406.2661. [CrossRef]
2. Lloyd, S.; Weedbrook, C. Quantum Generative Adversarial Learning. *Phys. Rev. Lett.* **2018**, *121*, 040502. [CrossRef] [PubMed]
3. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.
4. Goldsmith, D.; Mahmud, M.M.H. Machine Learning for Quantum Computing Specialists. *arXiv* **2024**, arXiv:2404.18555.
5. Dejen, A.; Ridwan, M. A Review of Quantum Computing. *Int. J. Math. Sci. Comput.* **2022**, *8*, 49–59. [CrossRef]
6. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information, 10th Anniversary Edition*; Cambridge University Press: Cambridge, UK, 2010; ISBN 978-1-139-49548-6.
7. Steane, A. Quantum Computing. *Rep. Prog. Phys.* **1998**, *61*, 117. [CrossRef]
8. Bausch, J. Recurrent Quantum Neural Networks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1368–1379.
9. Nema, P.; Nene, M.J. Pauli Matrix Based Quantum Communication Protocol. In Proceedings of the 2020 IEEE International Conference on Advent Trends in Multidisciplinary Research and Innovation (ICATMRI), Buldhana, India, 30 December 2020; pp.·1–6.
10. Ngo, T.A.; Nguyen, T.; Thang, T.C. A Survey of Recent Advances in Quantum Generative Adversarial Networks. *Electronics* **2023**, *12*, 856. [CrossRef]
11. Zhang, Z.; Luo, C.; Yu, J. Towards the Gradient Vanishing, Divergence Mismatching and Mode Collapse of Generative Adversarial Nets. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; Association for Computing Machinery: New York, NY, USA, 2019; pp. 2377–2380.
12. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A. Improved Training of Wasserstein GANs. *arXiv* **2017**, arXiv:1704.00028.
13. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. *arXiv* **2016**, arXiv:1606.03498.
14. Arjovsky, M.; Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. *arXiv* **2017**, arXiv:1701.04862.
15. Dallaire-Demers, P.-L.; Killoran, N. Quantum Generative Adversarial Networks. *Phys. Rev. A* **2018**, *98*, 012324. [CrossRef]
16. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum Principal Component Analysis. *Nat. Phys.* **2014**, *10*, 631–633. [CrossRef]
17. Niu, M.Y.; Zlokapa, A.; Broughton, M.; Boixo, S.; Mohseni, M.; Smelyanskyi, V.; Neven, H. Entangling Quantum Generative Adversarial Networks. *arXiv* **2021**, arXiv:2105.00080. [CrossRef]
18. Stein, S.A.; Baheri, B.; Chen, D.; Mao, Y.; Guan, Q.; Li, A.; Fang, B.; Xu, S. QuGAN: A Quantum State Fidelity Based Generative Adversarial Network. In Proceedings of the 2021 IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, USA, 17–22 October 2021; IEEE: Broomfield, CO, USA, 2021; pp. 71–81.
19. Huang, H.-L.; Du, Y.; Gong, M.; Zhao, Y.; Wu, Y.; Wang, C.; Li, S.; Liang, F.; Lin, J.; Xu, Y.; et al. Experimental Quantum Generative Adversarial Networks for Image Generation. *Phys. Rev. Appl.* **2021**, *16*, 024051. [CrossRef]
20. Tsang, S.L.; West, M.T.; Erfani, S.M.; Usman, M. Hybrid Quantum–Classical Generative Adversarial Network for High-Resolution Image Generation. *IEEE Trans. Quantum Eng.* **2023**, *4*, 1–19. [CrossRef]
21. Garcia-Escartin, J.C.; Chamorro-Posada, P. Swap Test and Hong-Ou-Mandel Effect Are Equivalent. *Phys. Rev. A* **2013**, *87*, 052330. [CrossRef]

22. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Supplementary Information for "Quantum Supremacy Using a Programmable Superconducting Processor". *Nature* **2019**, *574*, 505–510. [CrossRef]
23. Brock, A.; Donahue, J.; Simonyan, K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv* **2019**, arXiv:1809.11096. [CrossRef]
24. Variational Quantum Circuits for Deep Reinforcement Learning | IEEE Journals & Magazine | IEEE Xplore. Available online: https://ieeexplore.ieee.org/abstract/document/9144562 (accessed on 10 July 2024).
25. DiVincenzo, D.P. Quantum Gates and Circuits. *Proc. R. Soc. Lond. Ser. Math. Phys. Eng. Sci.* **1998**, *454*, 261–276. [CrossRef]
26. Crooks, G.E. Gradients of Parameterized Quantum Gates Using the Parameter-Shift Rule and Gate Decomposition. *arXiv* **2019**, arXiv:1905.13311. [CrossRef]
27. Cong, I.; Choi, S.; Lukin, M.D. Quantum Convolutional Neural Networks. *Nat. Phys.* **2019**, *15*, 1273–1278. [CrossRef]
28. Huang, H.-Y.; Broughton, M.; Cotler, J.; Chen, S.; Li, J.; Mohseni, M.; Neven, H.; Babbush, R.; Kueng, R.; Preskill, J.; et al. Quantum Advantage in Learning from Experiments. *Science* **2022**, *376*, 1182–1186. [CrossRef] [PubMed]
29. Tacchino, F.; Barkoutsos, P.; Macchiavello, C.; Tavernelli, I.; Gerace, D.; Bajoni, D. Quantum Implementation of an Artificial Feed-Forward Neural Network. *Quantum Sci. Technol.* **2020**, *5*, 044010. [CrossRef]
30. Chu, C.; Skipper, G.; Swany, M.; Chen, F. IQGAN: Robust Quantum Generative Adversarial Network for Image Synthesis on NISQ Devices. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Rhodes Island, Greece, 4–10 June 2023.
31. Yu, C.-H.; Gao, F.; Lin, S.; Wang, J. Quantum Data Compression by Principal Component Analysis. *Quantum Inf. Process.* **2019**, *18*, 249. [CrossRef]
32. Yen-Chi Chen, S.; Yoo, S.; Fang, Y.-L.L. Quantum Long Short-Term Memory. *arXiv* **2020**, arXiv:2009.01783. [CrossRef]
33. cifar10 | TensorFlow Datasets. Available online: https://www.tensorflow.org/datasets/catalog/cifar10 (accessed on 23 November 2024).
34. IBM Quantum Computing. Available online: https://www.ibm.com/quantum/ (accessed on 11 July 2024).
35. PennyLane. Available online: https://pennylane.ai/ (accessed on 11 July 2024).
36. Mnist | TensorFlow Datasets. Available online: https://www.tensorflow.org/datasets/catalog/mnist (accessed on 23 November 2024).
37. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
38. Imagenet2012 | TensorFlow Datasets. Available online: https://www.tensorflow.org/datasets/catalog/imagenet2012 (accessed on 23 November 2024).
39. Li, M.; Zhang, T.; Chen, Y.; Smola, A.J. Efficient Mini-Batch Training for Stochastic Optimization. In Proceedings of the Proceedings of the 20th ACM SIGKDD international conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 661–670.
40. Schuld, M.; Killoran, N. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.* **2019**, *122*, 040504. [CrossRef]
41. Dowson, D.C.; Landau, B.V. The Fréchet Distance between Multivariate Normal Distributions. *J. Multivar. Anal.* **1982**, *12*, 450–455. [CrossRef]
42. Preskill, J. Quantum Computing in the NISQ Era and Beyond. *arXiv* **2018**, arXiv:1801.00862. [CrossRef]
43. Bocharov, A.; Roetteler, M.; Svore, K.M. Efficient Synthesis of Universal Repeat-Until-Success Circuits. *arXiv* **2014**, arXiv:1404.5320. [CrossRef] [PubMed]
44. Schuld, M.; Bergholm, V.; Gogolin, C.; Izaac, J.; Killoran, N. Evaluating Analytic Gradients on Quantum Hardware. *Phys. Rev. A* **2019**, *99*, 032331. [CrossRef]
45. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv* **2017**, arXiv:1708.07747.
46. McCoy, D.E.; Feo, T.; Harvey, T.A.; Prum, R.O. Structural Absorption by Barbule Microstructures of Super Black Bird of Paradise Feathers. *Nat. Commun.* **2018**, *9*, 1–8. [CrossRef]
47. Cerezo, M.; Sone, A.; Volkoff, T.; Cincio, L.; Coles, P.J. Cost Function Dependent Barren Plateaus in Shallow Parametrized Quantum Circuits. *Nat. Commun.* **2021**, *12*, 1791. [CrossRef]
48. Zhang, K.; Hsieh, M.H.; Liu, L.; Tao, D. Toward Trainability of Quantum Neural Networks. *arXiv* **2020**, arXiv:2011.06258. [CrossRef]