

# IQGAN: ROBUST QUANTUM GENERATIVE ADVERSARIAL NETWORK FOR IMAGE SYNTHESIS ON NISQ DEVICES

Cheng Chu   Grant Skipper   Martin Swamy   Fan Chen

Department of Intelligent Systems Engineering, Indiana University, Bloomington, IN, USA

## ABSTRACT

In this work, we propose IQGAN, a quantum Generative Adversarial Network (GAN) framework for multiqubit image synthesis that can be efficiently implemented on Noisy Intermediate Scale Quantum (NISQ) devices. We investigate the reasons for the inferior generative performance of current quantum GANs in our preliminary study and conclude that an adjustable input encoder is the key to ensuring high-quality data synthesis. We then propose the IQGAN architecture featuring a *trainable multiqubit quantum encoder* that effectively embeds classical data into quantum states. Furthermore, we propose a *compact quantum generator* that significantly reduces the design cost and circuit depth on NISQ devices. Experimental results on both IBM quantum processors and quantum simulators demonstrated that IQGAN outperforms state-of-the-art quantum GANs in qualitative and quantitative evaluation of the generated samples, model convergence, and quantum computing cost.

**Index Terms**— Quantum machine learning, generative adversarial networks, noisy intermediate-scale quantum

## 1. INTRODUCTION

**Related Work and Motivation.** The recent success of supervised Quantum Neural Networks (QNNs) [1, 2, 3, 4] has inspired several quantum Generative Adversarial Network (GAN) frameworks [5, 6, 7, 8, 9] that extend the application of QNNs to unsupervised generative learning. A GAN [10, 11] consists of a generator (G) that generates synthetic samples, and a discriminator (D) that tries to distinguish between true and fake data. Such an adversarial game converges to the point where G generates the same statistics as the true data and D is unable to discriminate between the true and generated data. Lloyd *et al.* [6] theoretically proved that quantum GANs exhibit potential exponential advantages over classical GANs in high-dimensional data synthesis. QuGAN18 [7] presented an implementation of quantum GAN, which suffers from convergence issues. QuGAN21 [8] and EQ-GAN [9] showed that a quantum fidelity based loss function stabilizes the training process, but both works generate data with low quality and require high implementation cost. In this work, we are motivated

**Table 1.** Comparisons between IQGAN and previous works (**Conv**: Convergence; **MultiQ**: multiple-qubit output).

Scheme	Loss	Conv	MultiQ	Quality	Cost
QGAN [6]	N/A	N/A	N/A	N/A	N/A
QuGAN18 [7]	Trace	✗	✗	Med.	High
QuGAN21 [8]	Fidelity	✓	✓	Low	High
EQ-GAN [9]	Fidelity	✓	✗	Low	Med.
<b>IQGAN</b>	Fidelity	✓	✓	High	Low

to advance previous work [8, 9] by proposing IQGAN, a robust multiqubit Quantum Generative Adversarial Network architecture for Image synthesis that can be efficiently implemented on today’s noisy intermediate-scale quantum (NISQ) devices. We summarize the comparisons between IQGAN and state-of-the-art (SOTA) quantum GANs in Table 1.

**Our Contributions.** This work makes the following contributions. (1) We study the reasons for the low generative performance in previous work [8, 9] and conclude that the standard quantum encoders limit the generative ability of a quantum GAN. We then propose a trainable multiqubit quantum encoder that achieves SOTA quality on the generated data (Section 3.1); (2) We present a compact generator circuit ansatz that reduces hardware cost and circuit depth compared with previous work [8, 9] (Section 3.2); (3) We demonstrate that IQGAN can be efficiently implemented on NISQ devices and provide the training procedure (Section 3.3); (4) We evaluate IQGAN on both IBM quantum processors and simulators and show that IQGAN outperforms the prior arts in qualitative and quantitative evaluations of the generated samples, model convergence, and quantum computing cost (Section 4).

**Limitations and Scope.** The potential exponential advantage of quantum GANs over classical GANs is proved in [6]. In this work, we focus on the situation where the data is classical while the generator and discriminator are quantum, same as [8]. We implement a small-scale IQGAN and evaluate its performance against prior arts on MNIST dataset – a small dataset considered by classical machine learning but SOTA dataset on QNN evaluation [8, 12, 13, 14]. The limitations on the scale of the implementation and the dataset are due to the NISQ technology capabilities. The implementation scale of IQGAN and its corresponding generative performance will improve with the rapidly improving technology. IQGAN demonstrates how NISQ computers can be used to perform nontrivial generative learning tasks with a reasonable performance by overcoming the current quantum obstacles.

THIS WORK WAS SUPPORTED IN PART BY NSF CAREER AWARD CNS-2143120.

## 2. BACKGROUND AND PRELIMINARY ANALYSIS

### 2.1. Quantum GANs Background

Figure 1 shows the mainstream quantum GAN (QGAN) architecture [8, 9], which consists of (1) a data encoder  $S(\mathbf{x})$  that embeds a real classical input  $\mathbf{x}$  to a quantum state  $|\psi_\sigma\rangle$ ; (2) a Variational Quantum Circuit (VQC) based generator  $G(\theta_g)$  that generates synthetic data represented as a quantum state  $|\psi_\rho(\theta_g)\rangle$ ; and (3) a SWAP test based discriminator  $D(|\psi_\sigma\rangle, |\psi_\rho\rangle)$  that measures the fidelity between the real data  $|\psi_\sigma\rangle$  and the fake data  $|\psi_\rho\rangle$ . Due to the limited number of qubits in NISQ devices, high-dimensional inputs are down-sampled using Principal Component Analysis (PCA) [8], while the generated data is obtained by first measuring the qubits of the generator, i.e.,  $q_3, q_4$  in Figure 1, on the probability of the  $|0\rangle$  state, and then transforming the measured vector into a high-dimensional data using inverse PCA. Note that we consider two-qubit inputs and a  $G(\theta_g)$  with a single circuit block as an example. The input size and number of blocks in  $G(\theta_g)$  can be adjusted to fit the problem of interest.

**Classical-to-Quantum Encoder,  $S(\mathbf{x})$ .** Angle encoding is the most widely used method in QNNs [8, 13, 14, 15] due to its noise immunity and simplicity of implementation. Angle encoding encodes an  $N$ -dimensional data as the radians of rotation gates acting on  $N$  qubits, i.e.,  $|\psi_{\mathbf{x}}\rangle = \bigotimes_{i=0}^{N-1} R(x_i)|0\rangle$ , where  $R$  can be one of or a combination of the rotate gates  $\{RX, RY, RZ\}$  and  $\otimes$  represent the tensor product operation.

**VQC Circuit Ansatz,  $G(\theta_g)$ .** Mainstream QNNs [13, 16, 17] adopt a VQC ansatz constructed by parameterized single-qubit rotation gates followed by nearest-neighbor coupling of qubits using fixed two-qubit CNOT gates, as illustrated in Figure 1. Such a circuit ansatz has demonstrated superior expressive capability in various applications. The logic behind such designs is that single-qubit rotations provide a way to parameterize circuits, while two-qubit CNOT gates provide maximal entanglement between the two target qubits.

**SWAP Test Circuit,  $D(|\psi_\sigma\rangle, |\psi_\rho\rangle)$ .** A SWAP test circuit is a standard quantum module used for quantum fidelity measurement. As shown in Figure 1, it consists of one ancillary qubit  $q_0$ , two Hadamard gates (i.e., H gate), and several controlled SWAP gates (i.e., CSWAP gate) that interchange the two quantum states under test only if  $q_0$  is in state  $|1\rangle$ . The ancillary qubit  $q_0$  is finally measured on the z-basis and the probability it yields a measured output  $|0\rangle$  is  $P_0 = \frac{1 + \langle\psi_\sigma|\psi_\rho\rangle^2}{2}$ .  $P_0$  is defined as quantum fidelity as it provides a good estimate of how close two states are, i.e.,  $P_0 = \frac{1}{2}$  for two orthogonal states and  $P_0 = 1$  for two exactly same states.

**Training Objective.** The parameter matrix  $\theta_g$  can be equivalently viewed as the weights in classical neural networks and is trained iteratively to minimize a fidelity based loss shown in Equation 1.

$$\min_{\theta_g} L(\theta_g) = \min_{\theta_g} [1 - \langle\psi_\sigma|\psi_\rho(\theta_g)\rangle^2] \quad (1)$$

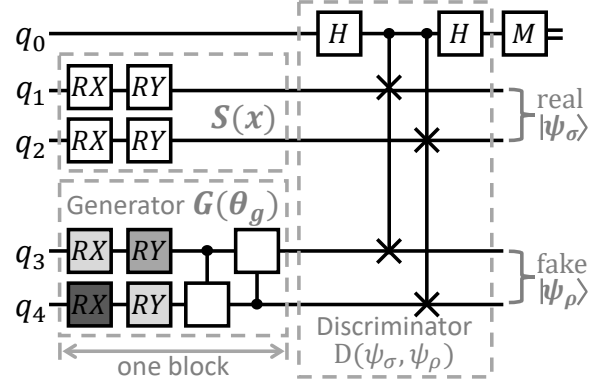


Fig. 1. A standard QGAN (shaded gates are trainable).

### 2.2. Preliminary Analysis

**Low-quality Generated Images.** Although prior work [8, 9] achieved training convergence with the fidelity based loss, their generated data is of low quality. We show the fidelity between the learned quantum state  $|\psi_\rho\rangle$  and the target quantum state  $|\psi_\sigma\rangle$  using angle encoding in Equation 2~3. For the sake of simplicity, we highlight a special case of Equation 3 in Equation 4 by adding  $2n\pi$  to the learned angle  $\theta_\rho$ . It is clear that the same fidelity and hence the same loss can be achieved with different parameter sets due to the periodic repeatability of the sine and cosine functions, resulting in distorted generated images (See Figure 2). Although input data can be mapped to the range  $[0, \pi]$  before applying the angle encoding [8, 16], such unfaithful normalization prevents a QNN from learning an accurate model and producing high-quality output. *We conclude that the de facto angle encoding (even with normalization) fails to ensure high-quality output in a fidelity based GAN framework. We investigate if a new quantum encoding method with learnability can improve performance for a quantum GAN.*

$$\langle\psi_\rho|\psi_\sigma\rangle^2 = \left( \left[ \cos \frac{\theta_\rho}{2} \quad \sin \frac{\theta_\rho}{2} \right] \begin{bmatrix} \cos \frac{\theta_\sigma}{2} \\ \sin \frac{\theta_\sigma}{2} \end{bmatrix} \right)^2 \quad (2)$$

$$= \left( \cos \frac{\theta_\rho}{2} \cos \frac{\theta_\sigma}{2} + \sin \frac{\theta_\rho}{2} \sin \frac{\theta_\sigma}{2} \right)^2 \quad (3)$$

$$= \left( \cos \frac{\theta_\rho + 2n\pi}{2} \cos \frac{\theta_\sigma}{2} + \sin \frac{\theta_\rho + 2n\pi}{2} \sin \frac{\theta_\sigma}{2} \right)^2 \quad (4)$$

**Complex and High-Cost Generator.** The standard circuit ansatz for generator shown in Figure 1 requires each pair of nearest-neighbors to be maximally entangled using expensive two-qubit CNOT gates. We hypothesize that (1) a flexible amount of entanglement rather than fixed maximal entanglement may perform better for a QNN algorithm; (2) a generator without entanglement gates may be good enough to provide acceptable generative capability since the encoding circuit  $S(\mathbf{x})$  normally embeds target data without any entanglement gates. Therefore, *we proposed to explore the replacement of fixed CNOT gates with other two-qubit gates (e.g., fixed or parameterized), or even completely remove two-qubit gates in a generator circuit. We investigate the effect of two-qubit gates on the performance of a generative ansatz and set out to reduce the circuit complexity of a generator.*

**Table 2.** Comparison on quantum GANs with different two-qubit gate (2QGate).

2QGate	Block #	Normalized Cost	Fidelity	$\theta_0$	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$	$\theta_7$
CNOT	2	$2.8\times$	0.813	N/A							
ISWAP	2	$3.7\times$	0.954	N/A							
CRX( $\theta$ )	2	$6\times$	0.969	-0.007 0.003	-0.023 0.036	-0.228 1.799	0.034 0.018	-0.093 -0.118	-0.154 0.013	-0.015 0.033	3.348 3.167
CROT( $\phi, \theta, \omega$ )	2	$6\times$	0.969	0.911 0.079	-0.057 -0.086	2.049 -0.068	0.059 -0.971	-1.883 -0.043	0.013 0.098	2.827 0.262	0.090 -0.062
w/o 2QGate	N/A	$1\times$	0.969	N/A							

### 3. IQGAN: DESIGN AND IMPLEMENTATION

In this section, we present the design and implementation of the proposed IQGAN framework. The IQGAN architecture has two innovative components: (1) a variational encoder  $S(\arcsin(\mathbf{x} \cdot \theta_s))$  that adaptively embeds classical training data into quantum states; and (2) a compact quantum generator  $G(\theta_g)$  without costly two-qubit entanglement gates.

#### 3.1. Trainable Multiqubit Quantum Encoder

Instead of using a fixed encoder (FE) circuit  $S(\mathbf{x})$ , we investigate trainable encoder (TE) using PennyLane [18] and show that a properly designed variational classical-to-quantum encoder results in improved performance for a QNN model (See results in Section 4). We introduce adaptivity into the encoder circuit by defining a variational encoder function  $S(\arcsin(\mathbf{x} \cdot \theta_s))$  with trainable parameters  $\theta_s$ . To obtain an angle encoding, we utilize the  $\arcsin$  function, which is common practice in QNN encoding [16]. The parameter set  $\theta_s$  is pre-trained to produce faithful quantum presentations in which data from different clusters are separated. Specifically, the encoder is trained with a pre-train dataset (CIFAR10 [19] in this work)  $\mathcal{T}=\{(\mathbf{x}_i, y_i) | 0 \leq i \leq N-1\}$ , where  $\mathbf{x}_i$  is an  $n$ -dimensional vector and  $y_i$  is the label. We prepare a quantum ensemble  $\sigma_{\mathbf{y}_k} = \frac{1}{N_k} \sum_{j=0}^{N_k-1} |\psi_\sigma(x_j)\rangle \langle \psi_\sigma(x_j)|$  by uniformly sampling  $N_k$  inputs from class  $y_k$ , i.e.,  $\mathcal{T}_k = \{(\mathbf{x}_j, y_k) | 0 \leq j \leq N_k-1\}$ , and feeding them into the encoder. We then train the encoder to obtain a set of optimal parameters  $\theta_s^*$  that maximize the distance between  $\sigma_{\mathbf{y}_k}$  and  $\sigma_{\mathbf{y}_m}$  when  $k \neq m$ . IQGAN uses the pre-trained values of  $S(\arcsin(\mathbf{x} \cdot \theta_s^*))$  as the initial parameters of the encoder.

#### 3.2. Compact Quantum Generator

To investigate the impact of two-qubit entanglement gates on the performance of a generator circuit ansatz, we replace the default CNOT gates in Figure 1 with widely used two-qubit gates, including fixed ISWAP gates, and parameterized CRX( $\theta$ ) and CROT( $\phi, \theta, \omega$ ) gates. We refer interested readers to [20] for detailed explanations of each candidate gate. We provide a full comparison between different generator circuit ansatzes in Table 2. The default CNOT gates do not perform very well and actually perform worse than other types of two-qubit entanglement gates. Parameterized gates, i.e., CRX and CROT, perform better than fixed ISWAP gates (i.e., 0.954 vs. 0.969), which is attributed to the trainable amount

**Table 3.** Hardware cost of different quantum GAN schemes ( $n$ : input size;  $b$ : repeated VQC block number; 1QG#/2QG#: one-/two-qubit gates number; Param.#: parameter number).

Scheme	Qubit#	1QG#	2QG#	Param.#
QuGAN21 [8]	$2n+1$	$nb+1$	$4nb$	$5nb$
EQ-GAN [9]	$2n+1$	$2nb+n+2$	$(b+1)n$	$2nb$
<b>IQGAN</b>	$2n+1$	$2nb+n+2$	$n$	$2nb$

of entanglement. However, the hardware overhead of CRX and CROT gates is significantly higher than fixed two-qubit gates. In particular, we find that a generator without two-qubit gates, denoted as *w/o 2QGate* in Table 2 achieves the same performance as CRX and CROT with  $6\times$  reduced hardware cost. Therefore, we propose to implement the generator in IQGAN with no two-qubit gates.

#### 3.3. Implementation of IQGAN

**Hardware Implementation Cost.** Multiqubit gates are decomposed in practice into basic single-qubit and two-qubit gates supported by specific gate libraries. We assume an ideal quantum circuit library that supports all types of single-qubit and two-qubit gates and provide a quantitative comparison of design cost between IQGAN and prior arts in Table 3. IQGAN requires the minimum hardware resource and reduces hardware cost by at least  $6\times$  in the simplest case when the generator uses only one VQC block, as detailed in Table 2.

**Training Procedure.** IQGAN requires a two-step training process. At the beginning of the algorithm, the encoder  $S(\arcsin(\mathbf{x} \cdot \theta_s))$  is initialized by arbitrary parameters  $\theta_s^0$  and trained on a pre-train dataset to obtain the optimized parameter set of  $\theta_s^*$  as explained in Section 3.1. We then utilize  $S(\arcsin(\mathbf{x} \cdot \theta_s^*))$  in an IQGAN framework with uniformly initialized  $\theta_g^0$ , and train the whole model to minimize the objective function in Equation 1. After the model reaches its convergence, we record the optimal parameter set of  $\theta_g^*$ .

## 4. EXPERIMENTS AND RESULTS

### 4.1. Experimental Setup

**Schemes and Benchmarks.** We compare IQGAN with two SOTA quantum GANs [8, 9] using the MNIST [21] dataset. Note that we exclude QuGAN2018 [7] because it suffers from convergence oscillation and is difficult to train in practice [9]. We demonstrate the proof-of-principle of IQGAN using a 5-qubit IBM quantum processor, *ibmqquito* [22]. We confirm that IQGAN is reasonably accurate in generating synthetic images on real NISQ devices.

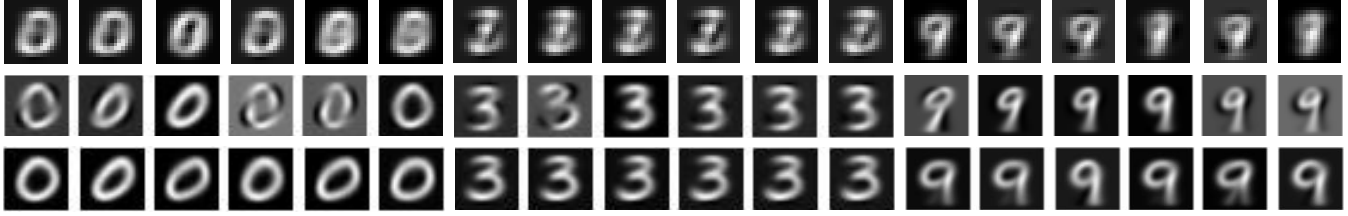


Fig. 2. MNIST images generated by QuGAN2021 [8] (1st row), EQ-GAN [9] (2nd row), and IQGAN (bottom row).

Table 4. Accuracy comparison of Fixed Encoder (FE) and Trainable Encoder (TE) on different subsets of MNIST.

Task	Input Size	Qubit #	Accuracy (%)	
			FE	TE
MNIST-2	4*4	16	89.5	<b>90.9</b>
MNIST-4	2*2	4	43.2	<b>45.6</b>
MNIST-4	4*4	16	45.9	<b>49.4</b>
MNIST-8	4*4	16	23.25	<b>24.3</b>

**Simulation.** All quantum GANs are implemented using PennyLane [18], a quantum computing software library. We train QuGAN2021 [8] using their open-source code by following their training configuration. We implement EQ-GAN [9] by following their design strategy but extending their original design to support multiqubit data. EQ-GAN [9] and IQGAN are trained with the ADAM optimizer, while the learning rate, batch size, and the number of epochs are set respectively as 0.001, 32, and 30. The learning rate is scheduled by CosineAnnealingLR with a  $T_{max}$  of 30.

## 4.2. Results and Analysis

**Effectiveness of the Trainable Encoder.** To compare the faithful data embedding capability between the *de facto* fixed encoder (FE) and the proposed trainable encoder (TE), we apply both FE and TE to downstream classification tasks and report their accuracy in Table 4. TE boosts the model accuracy under different input sizes. We also make two observations that can be extended for general quantum machine learning algorithm: (1) The effectiveness of TE increases as the input qubit size increases, e.g., the accuracy improvement comparison on MNIST-4 for input sizes of  $2 \times 2$  and  $4 \times 4$ ; (2) The effectiveness of TE increases with the model complexity, e.g., the accuracy improvement comparison between MNIST-2 and MNIST-4 with both 16-qubit inputs.

**Comparison on Image Quality.** As visualized in Figure 2, QuGAN2021 [8] shows limited generative capability. For images of 0 and 9, a blurred outline can be identified. However, for images of 3 with more complex visual structures, QuGAN2021 [8] fails to reproduce the images. The results of EQ-GAN [9] demonstrate an unstable image quality with a large deviation. For instance, the 4th, 5th, 8th, 13th, 17th, and 18th images are highly distorted. Note that the results for QuGAN2021 [8] and EQ-GAN [9] are simulated data, while the results on IQGAN are collected on `ibmq-quito` [22]. Compared with previous quantum GANs, we see IQGAN achieves a stable and consistent high-quality output even on NISQ devices.

**Convergence of IQGAN.** In Figure 3, we present the

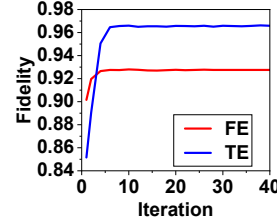


Fig. 3. IQGAN Convergence.

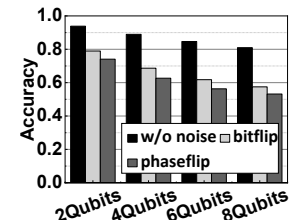


Fig. 4. Impact of noises.

measured fidelity of the discriminator in IQGAN. We reported results with both fixed encoder (FE) and trainable encoder (TE). A stable convergence within ten iterations can be observed for both configurations, confirming that IQGAN converges to a high (i.e.,  $>0.92$ ) state overlap with the target inputs. Compared with a standard FE, the proposed TE achieved a lower (e.g., 0.85 vs. 0.90) fidelity in the first several iterations, however, the final learned fidelity of TE is  $\sim 0.039$  higher (i.e., 0.966 vs. 0.927) than FE. Furthermore, the learned fidelity of TE increases faster than that of FE, which is mainly attributed to the higher flexibility and adaptability of the TE circuit and the richer expressive power.

**Impact of Quantum Errors.** We also explore the impact of quantum errors on IQGAN performance. We consider bit-flip and phase-flip [23, 24], which are the two most significant error types on NISQ devices. We down-sample the original images to  $1 \times 2$ ,  $1 \times 4$ ,  $1 \times 6$ , and  $1 \times 8$  vectors using PCA algorithm, and report the fidelity of generated images in Figure 4. Although the quality of the target images increases monotonically with the input size, the fidelity between learned images and original inputs decreases. This phenomenon is counterintuitive but can be explained in quantum GAN when quantum errors are considered. First, the increased input sizes require an increased number of qubits. Second, as the dimension of the data increases, the required generative capability of the generator also increases. Therefore, a more deep generator circuit with an increased number of blocks is required to fit the data. A quantum system with more qubits and gates is more susceptible to noise. In this case, while a more complex IQGAN circuit increases the expressive power of the model, quantum noise from excessive overhead reduces the overall fidelity.

## 5. CONCLUSION

We propose IQGAN, a quantum generative adversarial network for image synthesis that can be implemented on NISQ devices. We compared IQGAN with previous work and demonstrate that IQGAN outperforms the state-of-the-arts in image quality and quantum hardware implementation cost.

## 6. REFERENCES

- [1] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta, “Supervised Learning with Quantum-Enhanced Feature Spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.
- [2] Hsin-Yuan Huang et al., “Quantum advantage in learning from experiments,” *Science*, vol. 376, no. 6598, pp. 1182–1186, 2022.
- [3] Francesco Tacchino, Panagiotis Barkoutsos, Chiara Macchiavello, Ivano Tavernelli, Dario Gerace, and Daniele Bajoni, “Quantum Implementation of An Artificial Feed-forward Neural Network,” *Quantum Science and Technology*, vol. 5, no. 4, pp. 044010, 2020.
- [4] Iris Cong, Soonwon Choi, and Mikhail D Lukin, “Quantum Convolutional Neural Networks,” *Nature Physics*, vol. 15, no. 12, pp. 1273–1278, 2019.
- [5] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost, “Quantum Algorithms for Supervised and Unsupervised Machine Learning,” *arXiv e-prints*, p. arXiv:1307.0411, July 2013.
- [6] Seth Lloyd and Christian Weedbrook, “Quantum Generative Adversarial Learning,” *Phys. Rev. Lett.*, vol. 121, pp. 040502, Jul 2018.
- [7] Pierre-Luc Dallaire-Demers and Nathan Killoran, “Quantum Generative Adversarial Networks,” *Phys. Rev. A*, vol. 98, pp. 012324, Jul 2018.
- [8] Samuel A. Stein, Betis Baheri, Daniel Chen, Ying Mao, Qiang Guan, Ang Li, Bo Fang, and Shuai Xu, “QuGAN: A Quantum State Fidelity based Generative Adversarial Network,” in *IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2021, pp. 71–81, IEEE.
- [9] Murphy Yuezhen Niu, Alexander Zlokapa, Michael Broughton, Sergio Boixo, Masoud Mohseni, Vadim Smelyanskiy, and Hartmut Neven, “Entangling Quantum Generative Adversarial Networks,” *arXiv preprint arXiv:2105.00080*, 2021.
- [10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672–2680.
- [11] Alec Radford, Luke Metz, and Soumith Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015.
- [12] Johannes Bausch, “Recurrent quantum neural networks,” *Advances in neural information processing systems (NeurIPS)*, vol. 33, pp. 1368–1379, 2020.
- [13] Hanrui Wang, Yongshan Ding, Jiaqi Gu, Yujun Lin, David Z Pan, Frederic T Chong, and Song Han, “Quantumnas: Noise-adaptive search for robust quantum circuits,” in *IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2022, pp. 692–708.
- [14] Edward Farhi and Hartmut Neven, “Classification with quantum neural networks on near term processors,” *arXiv preprint arXiv:1802.06002*, 2018.
- [15] Ryan LaRose and Brian Coyle, “Robust data encodings for quantum classifiers,” *Physical Review A*, vol. 102, no. 3, pp. 032420, 2020.
- [16] Samuel Yen-Chi Chen, Shinjae Yoo, and Yao-Lung L Fang, “Quantum long short-term memory,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8622–8626.
- [17] Jun Qi and Javier Tejedor, “Classical-to-quantum transfer learning for spoken command recognition based on quantum neural networks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8627–8631.
- [18] PennyLane, “PennyLane,” <https://pennylane.ai/>.
- [19] Alex Krizhevsky, Geoffrey Hinton, et al., “Learning multiple layers of features from tiny images,” *Master’s thesis, Department of Computer Science, University of Toronto*, 2009.
- [20] pennylane, “Quantum operators,” <https://docs.pennylane.ai/en/stable/introduction/operations.html>.
- [21] Yann LeCun, “The mnist database of handwritten digits,” <http://yann.lecun.com/exdb/mnist/>, 1998.
- [22] IBM, “IBM Quantum Computing,” <https://quantum-computing.ibm.com/>.
- [23] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles, “Noise-Induced Barren Plateaus in Variational Quantum Algorithms,” *Nature communications*, vol. 12, no. 1, pp. 1–11, 2021.
- [24] Lena Funcke, Tobias Hartung, Karl Jansen, Stefan Kühn, Paolo Stornati, and Xiaoyang Wang, “Measurement Error Mitigation in Quantum Computers through Classical Bit-Flip Correction,” *arXiv preprint arXiv:2007.03663*, 2020.