

# Quantum Generative Neural Networks for Imaging Applications

Sunil Vittal  
*Department of Mathematics*  
*Princeton University*  
Princeton, NJ, USA  
sv2954@princeton.edu

David Ramirez  
*SenSIP Center*  
*Arizona State University*  
Tempe, AZ, USA  
dframire@asu.edu

Glen Uehara  
*SenSIP Center*  
*Arizona State University*  
Tempe, AZ, USA  
guehara@asu.edu

Gennaro De Luca  
*School of Computing and Artificial Intelligence*  
*Arizona State University*  
Tempe, AZ, USA  
gennaro.deluca@asu.edu

Andreas Spanias  
*SenSIP Center*  
*Arizona State University*  
Tempe, AZ, USA  
spanias@asu.edu

**Abstract**—Quantum Machine Learning has risen in popularity due its potential to provide an exponential speed-up in training neural networks and other machine learning models. While simple examples include the Variational Quantum Classifier and Quantum Kernel methods, we focus on generative approaches within Quantum Machine Learning. In particular, we build Quantum-Classical Hybrid Generative Adversarial Networks (qGANs) to generate small-scale gray-scale images of the digit zero from the MNIST dataset. We find that our proposed hybrid model achieves a higher average PSNR than a classical GAN, a lower FID score of 222.949, but a lower Inception score of 1.160.

**Index Terms**—Quantum Machine Learning, Generative Models, GANs, Patch qGAN, MNIST

## I. INTRODUCTION

Due to Quantum Machine Learning’s potential to provide advantages over classical machine learning, people are now looking towards quantum generative models for image generation. For instance, quantum computing has applications to optimization problems, thus quantum computing can be utilized to speed up the optimization process while training machine learning models. [1], [2] Another key application is data storage, where by relying on smaller scale computations via qubits and quantum circuits, one can map large amounts of classical data to a relatively small number of qubits. [3] Finally, there is also talk of Quantum Supremacy, which states that quantum computers can solve problems that classical computers cannot solve. [4], [5]

Nonetheless, while the potential of quantum computing is high, the field is in the Noisy Intermediate Scale Quantum (NISQ) era, meaning current systems are affected by noisy errors, forcing us to utilize few qubits. The aforementioned applications motivate the construction of fault-tolerant quantum computers which will allow us to fully observe the advantages of Quantum machine learning. [1]–[5]

Many QML methods involve a hybrid classical-quantum algorithm containing a parametrized quantum circuit with trainable parameters. These circuits are analogous to classical neural networks where the qubits are the neurons and the parameters are trainable weights. For instance, one might map their classical data to a quantum state vector, run a parametrized quantum circuit, acquire the probabilities of each outcome in this circuit, put this into a classical optimizer, and restart this process from the parametrized circuit, effectively “training” the model. [3]

Current QML Generative Models focus on small scale images and are quantum adaptations of well known classical models. Quantum approaches are based off of the Quantum Circuit Born Machine (QCBM) due to the abilities of these models to capture classical probability distributions with quantum states, thus leveraging the probabilistic nature of quantum states. [6] Moreover, such quantum architectures showcase the ability for quantum computers to capture features and symmetries that classical computer struggle to do.

An example of such a quantum circuit is the Instantaneous Quantum Polynomial (IQP) circuit which prepares a state with measurement probabilities from the distribution  $H^{\otimes n} U H^{\otimes n} |0\rangle^{\otimes n}$ , where  $H$  is the Hadamard Gate and  $U$  is a product of diagonal single-qubit or two-qubit gates. In fact, simulating this probability distribution on a classical computer is a  $\#P$  problem. [7] Due to the effectiveness of these circuits on quantum architectures, we will utilize IQPs in our work.

As for the baseline generative model, a popular generative approach to QML involves the Quantum Generative Adversarial Network (qGAN) [8], a quantum version of the classical GAN [9]. qGANs often have a quantum generator and classical discriminator, but the quantum generator may also have a classical component.

While QML theoretically gives an exponential speedup, the classical simulation of quantum computation is exponentially slower as the number of qubits increases, both motivating

efficient quantum algorithms within these models and larger quantum computers (CITE SOMETHING HERE?). An example of a resource efficient qGAN is the quantum patch GAN [10] which utilizes multiple sub-generators in the form of quantum circuits, but these were utilized on  $8 \times 8$  MNIST [11] images.

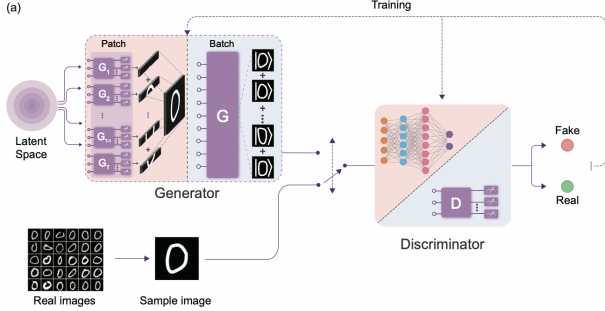


Fig. 1. patch qGAN architecture from [10]

Other works (i.e. [12]) utilized this patch qGAN approach, generating  $28 \times 28$  MNIST digits and Fashion-MNIST [13] dataset images. Other approaches involve feeding the generator output into a classical layer (i.e [14]), which provides stronger optimization since it is difficult to zero out pixels with a purely quantum generator. These kinds of models, however, remove the quantum benefit of fewer trainable parameters.

We combine both approaches into a quantum patch GAN that features a classical layer within each patch, to combat the issues with a purely quantum generator mentioned in [12]. Moreover, we present a novel circuit architecture and form comparisons between our model, a patch qGAN with a purely quantum generator, and a classical GAN with respect to Peak Signal-to-Noise Ratio (PSNR) during training, Frechet Inception Distance (FID) [15] scores, and Inception scores [16]. We find that purely quantum method start with a higher baseline PSNR score that peaks quicker but lower than a patch qGAN with a classical component.

## II. PRELIMINARIES

### A. Generative Adversarial Networks

GANs feature two neural networks, a generator and a discriminator. These two networks play a minimax game where the generator attempts to generate a fake image and the discriminator must determine whether it is real or fake. In particular, given a distribution of values  $p_G$ , we sample a noise vector  $\mathbf{z} \sim p_G$  which is an input to our generator  $G$ .  $G$  desires to learn this distribution of values and transform it into the distribution that matches the image dataset. This minimax adversarial game can be described via the following objective function:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{target}}} [\log(D(x))] - \mathbb{E}_{z \sim p_G} [\log(1 - D(G(z)))] \quad (1)$$

### B. Quantum Computing

For more information on the information presented in this section, we refer the readers to [17]. The foundations of quantum computing lie in *qubits* which are the building blocks of information in quantum computers. Qubits can be mathematically represented as a point on the *Bloch Sphere* where a qubit  $|\psi\rangle = \cos(\frac{\theta}{2})|0\rangle + e^{i\varphi}\sin(\frac{\theta}{2})|1\rangle$  is a general state on the bloch sphere. Just how classical circuits undergo transformations via logic gates, qubits undergo changes via Quantum Logic Gates which are represented by Unitary matrices. These Unitary gates can be thought of as rotations on the bloch sphere. The general form of Unitary Quantum Logic Gate is

$$U(\theta, \varphi, \lambda) = \begin{pmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda}\sin(\frac{\theta}{2}) \\ e^{i\varphi}\sin(\frac{\theta}{2}) & e^{i(\varphi+\lambda)}\cos(\frac{\theta}{2}) \end{pmatrix} \quad (2)$$

Additionally, the general form a rotation logic gate is

$$R(\theta, \varphi, \omega) = \begin{pmatrix} e^{-i(\varphi+\omega)/2}\cos(\frac{\theta}{2}) & -e^{i(\varphi-\omega)/2}\sin(\frac{\theta}{2}) \\ e^{-i(\varphi-\omega)/2}\sin(\frac{\theta}{2}) & e^{i(\varphi+\omega)/2}\cos(\frac{\theta}{2}) \end{pmatrix} \quad (3)$$

With these gates defined, we can define more common quantum gates, such as the Pauli Matrices

$$X = U(\pi, 0, 0) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (4)$$

$$Y = U(\pi, \frac{\pi}{2}, \frac{\pi}{2}) = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad (5)$$

$$Z = U(0, -\pi, 0) = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad (6)$$

Since quantum computing can be described through linear algebra, one often refers to the state of the circuit as the state vector, comprised of the coefficients on the *computational basis*. The computational basis for  $n$  qubits is a basis of  $2^n$  vectors, where we often represent with with respect to the classical binary representation of their index. Moreover, this basis has an additional special property, for a state vector

$$|\psi\rangle = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_{2^n} \end{pmatrix} \quad (7)$$

$\sum_{i=1}^{2^n} |p_i|^2 = 1$ , so the state vector also encodes the probabilities of getting each computational basis state after *measurement*.

Note that since  $p_i \in \mathbb{C}$  for all  $1 \leq i \leq 2^n$ , distinct state vectors can have the same probabilities for certain outcomes, which motivates a different interpretation of the 'state' of our system. *Density matrices* encode all information of a state and is a quantum analog for storing all statistical information related to a system just like a covariance matrix in statistics. In particular, a density matrix  $\rho$  encodes the probabilities of each outcome on the diagonal of the matrix while encoding the correlations between each computational basis state in the rest of the entries.

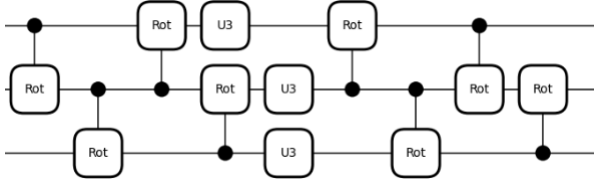


Fig. 2. Ansatz for the encoder. This is applied  $L$  times, where  $L$  is the number of layers.

### III. OUR PROPOSED QGAN ARCHITECTURE

We propose two qGANs, one with a fully quantum generator, and another with a quantum-classical hybrid generator. Call these qGAN and qCGAN, respectively. Both qGANs have 32 patches, one for each row in the  $32 \times 32$  image output, similar to how [12] set up their model. Each patch for both the qGAN and qCGAN consists of a complex circuit ansatz with two parts, an encoder and the patch sub-generator. In the qCGAN, however, each patch output is fed into a small classical fully-connected layer to improve the output accuracy.

#### A. The Encoder

The encoder in the quantum generator takes in an input vector  $\mathbf{z} \sim \pi \cdot \mathcal{N}(0, 1)$  which are parameters for an arbitrary noise matrix  $S \in SU(2^n)$ . This  $S$  is decomposes as follows

$$S = \sum_{i=1}^{2^n-1} z_i P_i \quad (8)$$

for  $P_i \in \{I, X, Y, Z\}^{\otimes n} \setminus \{I^{\otimes n}\}$ . After applying the following ansatz: Many circuits in past literature utilize more specified gates, but this limits the variability and expression of quantum circuits. Particularly, one of the main differences between classical and quantum computing is the 3D world qubits live in. These general rotation gates allow us to capture all three dimensions in the optimization process. In Figure 2, we see the circuit for a single layer in the Encoder.

#### B. The Patch Sub-Generator

We take the output from the encoder,  $\rho_E$  and perform some manipulations on it. In particular, we'd like to take this output data and use an IQP embedding on it, but IQP embeddings are a linear embedding.  $\rho_E$  is a  $8 \times 8$  matrix, but 8 qubits is too expensive to classically simulate, so we'd like to be able to use  $\rho_E$  on 5 qubits.

We utilize the partial trace and properties of density matrices to accomplish this. Given  $\rho_E$  on three qubits, the partial trace gives us a subsystem  $\rho'$  on  $m < 3$  qubits. As a result, for  $\rho_E$  on  $n$  qubits, we perform  $\binom{n}{m}$  partial traces, giving us a collection  $P = \{\rho_i\}_{i=1}^{\binom{n}{m}}$  of  $n$  subsystems. Note that these matrices have dimension  $2^{n-1} \times 2^{n-1}$ , allowing a more equitable simulation cost. To capture all information acquired from these partial traces, we take a convex combination of all  $\rho \in P$ . In particular, the input  $\rho_I$  into the patch sub-generator is:

$$\rho_I = \sum_{\rho_i \in P} x_i \rho_i \quad (9)$$

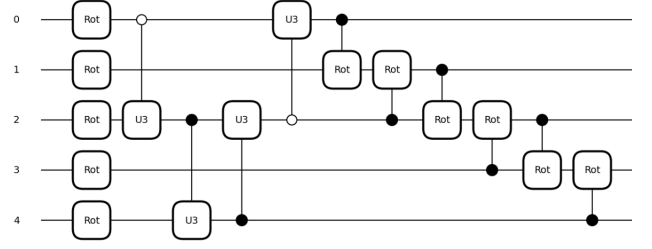


Fig. 3. Ansatz for the Patch Sub-Generator on 6 qubits. This is repeated  $L'$  times.

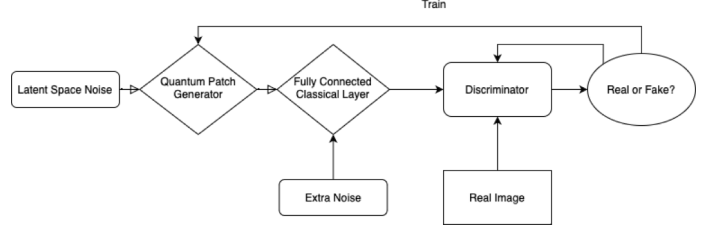


Fig. 4. High level description of the proposed qCGAN architecture.

where the  $x_i$  are classical trainable parameters to train which part of the initial encoding is 'most useful'. Passing  $\rho_I$  into IBM Qiskit's [18] IQP embedding, we acquire an embedding on  $2^{n-1}$  qubits. In our use case of  $32 \times 32$  images,  $n = 3$  and we utilize 5 or 6 qubits by performing a Multi-Controlled Toffoli gate from four embedded qubits onto a fifth and sixth qubit. In Figure 3, we see the ansatz that is applied  $L'$  times after this IQP circuit.

#### C. Measurement and Classical Layers

Just as in [10], [12], we perform a partial measurement and trace out an ancilla qubit from our quantum output. Mathematically, our output state is  $|\psi\rangle = U_C(z)|0\rangle^{\otimes n}$  where  $U_C$  is the unitary representing our circuit operation. Since image generation requires non-linear transformations, we take a projective measurement  $\Pi = |0\rangle\langle 0|^{\otimes n_A}$  and acquire a new state:

$$|\phi\rangle = \frac{\text{Tr}_A(\Pi \otimes \mathbb{I} |\psi\rangle\langle\psi|)}{\langle\psi|\Pi \otimes \mathbb{I}|\psi\rangle} \quad (10)$$

where  $n_A$  is the number of ancilla qubits and  $\mathbb{I}$  is the identity matrix on the rest of the qubits. This essentially means that we trace out the ancillary qubit and re-normalize the remaining state. We further post-process by dividing the remaining state vector by its max. In the qCGAN, we pass this output into a fully connected classical layer with extra sampled noise to complete the patch.

### IV. METHODS AND RESULTS (TODO)

#### A. Hardware Specs and Training Process

Experiments conducted and described below were completed on an Apple Macbook Pro with an Apple M2 CPU, 16 GB RAM, and macOS Ventura 13.0. Moreover, circuits and quantum neural networks were developed using the PennyLane

Neural Network	Optimizer	Learning Rate
Hybrid Generator	Adam	0.002
Quantum Generator	Adam	0.001
Classical Generator	Adam	0.0002
Discriminator 1	SGD	0.01
Discriminator 2	SGD	0.02

Fig. 5. Optimizers and Learning Rates for the training process of each neural network. Discriminator 1 is used for the Hybrid and Quantum Generators. Discriminator 2 is used for the Classical Generator. Both Discriminators have the same model architecture and are fully-connected neural networks, but have different learning rates due to the different generators.

[19] API. We compare and contrast three models: the Hybrid qCGAN, the qGAN, and a classical GAN. Each model completed a training process on five mini-batches of 100  $32 \times 32$  zero digit images from the MNIST dataset.

These models are compared qualitatively via their produced images and quantitatively via their Peak Signal-to-Noise Ratio (PSNR) between the target and produced images during the training process and Frechet Inception Distance (FID) from an unseen mini-batch of 200 zero digit images. Moreover, we also compute the Inception score of each model.

Note that there is a large difference in the number of parameters in each generator. In the qCGAN, the generator contains 40896 trainable parameters, the qGAN generator has 39840 trainable parameters, but the classical GAN's generator has 682752 trainable parameters. Moreover, the typical method of training a QNN's weights is the parameter-shift method [20], but this is computationally expensive as the number of circuit parameters increases. As a result, all models were trained using classical backpropagation.

### B. Qualitative Evaluation of Generated Images

In the qGAN, we notice that there is quite a bit of noise and hazy pixel distribution. Moreover, there are random white pixels away from the digits in question. This showcases some difficulties in fully quantum generators as image size increases as this behavior isn't noticeable in [10]. Furthermore, in Figure 6, we observe a mode collapse as the generator has low image variation, but there is still some variation as we see the pixel intensities are different and the shape is not exactly the same but very similar.

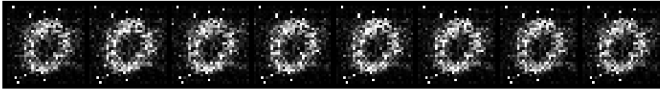


Fig. 6. Gray scale zero digit images generated by the patch qGAN after five mini-batches of 100 zero digit images.

In the classical model images in Figure 7, we note that classical GANs are capable of generating each MNIST digit from a probability distribution, showing an advantage above quantum generators, meaning that feeding one type of image

leads to quick learning and thus the same mode collapse we observed in the qGAN. Nonetheless, the generated images are much less noisy.



Fig. 7. Gray scale zero digit images generated by the Classical GAN after five mini-batches of 100 zero digit images.

In the qCGAN images in Figure 8, we are able to correct the issues present in the qGAN, with a less noisy image, less random white pixels, and more varied images with regards to both shape and pixel intensity.



Fig. 8. Gray scale zero digit images generated by the patch qCGAN after five mini-batches of 100 zero digit images.

### C. Quantitative Evaluation of Generated Images

From the generated images, we expect the patch qGAN to have the worst quantitative performance amongst the three models, but the results between the other two models are interesting. If we look at the PSNRs relative to the image seen from the mini-batch in Figures 9 and 10, we notice that the patch-qGAN starts at a higher PSNR and peaks much lower than the Classical GAN and the qCGAN. Moreover, there is much less variance. While the Classical GAN reaches a high

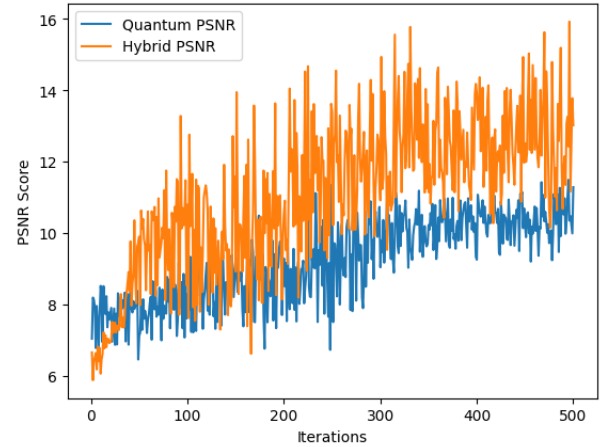


Fig. 9. PSNR relative to the current image in the training process in the qGAN and qCGAN

PSNR quicker than the qCGAN, it plateaus quicker, while the qCGAN keeps a growth trend, resulting in a higher average PSNR of 11.102 with the classical GAN having an average PSNR of 10.680. The qCGAN had a negligibly higher variance compared to the classical GAN, but both models' PSNRs had around three times the variance of the qGAN.

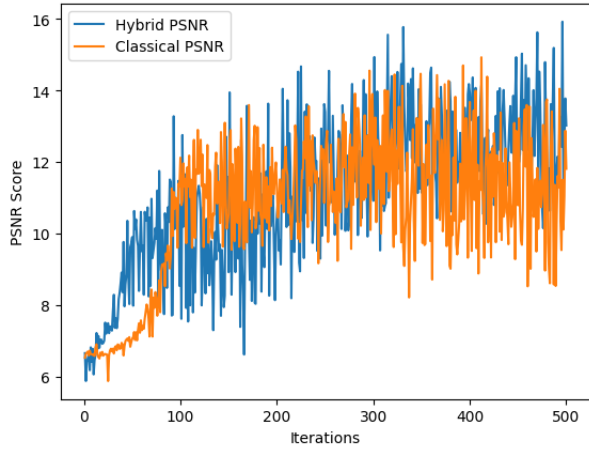


Fig. 10. PSNR relative to the current image in the training process in the qCGAN and the Classical GAN

Model	FID Score	Inception Score
qCGAN	222.949	1.160
Classical GAN	278.875	1.264
qGAN	353.866	1.113

Fig. 11. FID and Inception scores for each model.

For FID scores, the qCGAN had an FID score of 222.949, the Classical GAN had an FID score of 278.875, and the qGAN had an FID score of 353.866. Note that these scores are rather inflated since we are using gray-scale images. According to these FID scores, the qCGAN generated images that were closest to the target image distribution, which is likely due to the lack of mode collapse.

Meanwhile, for Inception scores, the qCGAN got a score of 1.160, the Classical GAN got a score of 1.264, and the qGAN got a score of 1.113, so even though the qCGAN generated images closer to the target images, the Classical GAN generated 'higher quality' images.

## V. CONCLUSION

Although in the early stages of image generation, we can see the possible performance benefits from quantum-classical hybrid models. Our hybrid model, which was primarily quantum, was able to achieve a higher average PSNR score and lower FID score than its classical counterpart. It's important to highlight that the classical component of the qCGAN was an extremely small Dense layer with 32 input features and 32 output features, yet it caused an extremely large improvement to the quantum component of the qCGAN.

While minimizing the classical portion of the qCGAN was intentional, it's also important to note that the above results were with respect to one type of image from the MNIST dataset. Standard classical GANs are able to generate all digits from random noise, but current quantum models lack the expressibility to capture multiple kinds of images

from a distribution, likely due to their small-scale nature. This motivates the construction of a Class-Conditional qGAN, where we can trigger which class of image to generate from a dataset.

Thinking along other avenues, we also note quantum generative models is still on small gray-scale images. How can we get to generating color images efficiently? If we can generate color images and develop an efficient Class-Conditional GAN, we will be much closer to realizing the Quantum Advantage promised by QML.

Promising avenues involve some of the methods above, but some aspects we did not consider are different IQP circuits, richer embeddings with IQP circuits, measuring in different bases. Creatively applying these methods and more could contribute to solving the aforementioned problems.

## VI. ACKNOWLEDGEMENTS

This project is funded by the National Science Foundation (NSF) Award 1953745. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the NSF.

## REFERENCES

- [1] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, "Power of data in quantum machine learning," *Nature Communications*, vol. 12, Art. no. 2631, 2021.
- [2] Cerezo M, Verdon G, Huang H Y, et al, "Challenges and opportunities in quantum machine learning," *Nature Computational Science*, vol. 2, pp. 567-576, 2022.
- [3] G. De Luca, "A Survey of NISQ Era Hybrid Quantum-Classical Machine Learning Research," *Journal of Artificial Intelligence and Technology*, vol. 2, no. 1, pp. 9–15, 2021.
- [4] J. Preskill, "Quantum Computing and the Entanglement Frontier," *Bulletin of the American Physical Society*, vol. 58, 2013.
- [5] X.-Y. Liu and Z. Zhang, "Classical Simulation of Quantum Circuits Using Reinforcement Learning: Parallel Environments and Benchmark," *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023.
- [6] M. Benedetti, D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, "A generative modeling approach for benchmarking and training shallow quantum circuits," *npj Quantum Information*, vol. 5, no. 1, p. 45, 2019.
- [7] M. J. Bremner, A. Montanaro, D. J. Shepherd, Average-case complexity versus approximate simulation of commuting quantum computations, *Phys. Rev. Lett.* 117, 080501 (2016).
- [8] P.-L. Dallaire-Demers and N. Killoran, "Quantum generative adversarial networks," arXiv:1804.08641 [quant-ph], 2018.
- [9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.
- [10] H.-L. Huang, Y. Du, M. Gong, Y. Zhao, et. al, "Experimental Quantum Generative Adversarial Networks for Image Generation," arXiv:2010.06201 [quant-ph], 2020.
- [11] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141-142, 2012.
- [12] S. L. Tsang, M. T. West, S. M. Erfani, and M. Usman, "Hybrid Quantum-Classical Generative Adversarial Network for High Resolution Image Generation," arXiv:2212.11614 [cs.LG], 2022.
- [13] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms," 2017.
- [14] R. Shu, X. Xu, M.-H. Yung, and W. Cui, "Variational Quantum Circuits Enhanced Generative Adversarial Network," arXiv:2402.01791v1 [cs.LG], 2024.

- [15] M. Heusel, H. Ramsauer, T. Unterthiner, and B. Nessler, "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- [16] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved Techniques for Training GANs," *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [17] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed. Cambridge: Cambridge University Press, 2010.
- [18] A. Javadi-Abhari et al, "Quantum Computing with Qiskit," arXiv:2405.08810 [quant-ph], 2024.
- [19] V. Bergholm et al, "PennyLane: Automatic differentiation of hybrid quantum-classical computations," 2018. arXiv:1811.04968.
- [20] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, "General Parameter-Shift Rules for Quantum Gradients," arXiv:2107.12390 [quant-ph], 2021.