

Quantum generative adversarial networks based on a readout error mitigation method with fault tolerant mechanism

Run-Sheng Zhao(赵润盛), Hong-Yang Ma(马鸿洋), Tao Cheng(程涛), Shuang Wang(王爽), and Xing-Kui Fan(范兴奎)

Citation: Chin. Phys. B, 2024, 33 (**4**): 040304. DOI: 10.1088/1674-1056/ad02e7
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Simulation of optimal work extraction for quantum systems with work storage

Peng-Fei Song(宋鹏飞) and Dan-Bo Zhang(张旦波)
Chin. Phys. B, 2024, 33 (**2**): 020312. DOI: 10.1088/1674-1056/ad09d0

Quantum algorithm for minimum dominating set problem with circuit design

Haoying Zhang(张皓颖), Shaoxuan Wang(王绍轩), Xinjian Liu(刘新建), Yingtong Shen(沈颖童), and Yukun Wang(王玉坤)
Chin. Phys. B, 2024, 33 (**2**): 020310. DOI: 10.1088/1674-1056/ad02e5

Protected simultaneous quantum remote state preparation scheme by weak and reversal measurements in noisy environments

Mandal Manoj Kumar, Choudhury Binayak S., and Samanta Soumen
Chin. Phys. B, 2024, 33 (**2**): 020309. DOI: 10.1088/1674-1056/ace683

Proposal for sequential Stern-Gerlach experiment with programmable quantum processors

Meng-Jun Hu(胡孟军), Haixing Miao(缪海兴), and Yong-Sheng Zhang(张永生)
Chin. Phys. B, 2024, 33 (**2**): 020303. DOI: 10.1088/1674-1056/ad09ca

Observation of flat-band localized state in a one-dimensional diamond momentum lattice of ultracold atoms

Chao Zeng(曾超), Yue-Ran Shi(石悦然), Yi-Yi Mao(毛一屹), Fei-Fei Wu(武菲菲), Yan-Jun Xie(谢岩骏), Tao Yuan(苑涛), Han-Ning Dai(戴汉宁), and Yu-Ao Chen(陈宇翱)
Chin. Phys. B, 2024, 33 (**1**): 010303. DOI: 10.1088/1674-1056/ad0cd1

Quantum generative adversarial networks based on a readout error mitigation method with fault tolerant mechanism

Run-Sheng Zhao(赵润盛)[1], Hong-Yang Ma(马鸿洋)[1], Tao Cheng(程涛)[2], Shuang Wang(王爽)[2], and Xing-Kui Fan(范兴奎)[1,†]

[1] *School of Sciences, Qingdao University of Technology, Qingdao* 266033, *China*
[2] *School of Information and Control Engineering, Qingdao University of Technology, Qingdao* 266033, *China*

(Received 17 July 2023; revised manuscript received 12 October 2023; accepted manuscript online 13 October 2023)

Readout errors caused by measurement noise are a significant source of errors in quantum circuits, which severely affect the output results and are an urgent problem to be solved in noisy-intermediate scale quantum (NISQ) computing. In this paper, we use the bit-flip averaging (BFA) method to mitigate frequent readout errors in quantum generative adversarial networks (QGAN) for image generation, which simplifies the response matrix structure by averaging the qubits for each random bit-flip in advance, successfully solving problems with high cost of measurement for traditional error mitigation methods. Our experiments were simulated in Qiskit using the handwritten digit image recognition dataset under the BFA-based method, the Kullback–Leibler (KL) divergence of the generated images converges to 0.04, 0.05, and 0.1 for readout error probabilities of $p = 0.01$, $p = 0.05$, and $p = 0.1$, respectively. Additionally, by evaluating the fidelity of the quantum states representing the images, we observe average fidelity values of 0.97, 0.96, and 0.95 for the three readout error probabilities, respectively. These results demonstrate the robustness of the model in mitigating readout errors and provide a highly fault tolerant mechanism for image generation models.

Keywords: readout errors, quantum generative adversarial networks, bit-flip averaging method, fault tolerant mechanisms

PACS: 03.67.Ac, 03.67.Pp, 03.67.Lx **DOI:** 10.1088/1674-1056/ad02e7

1. Introduction

In the era of noisy-intermediate scale quantum (NISQ) computing, quantum machine learning has garnered significant attention and research accomplishment.[1,2] It has demonstrated remarkable contributions to various fields including high-energy physics, quantum chemistry, and artificial intelligence, holding great promise for practical problem-solving in quantum computing.[3,4] In particular, quantum generative adversarial networks (QGAN) have shown great potential in generative tasks, which are used to generate real-world instances, mainly in image generation, quantum state tomography, stock futures, etc.[5,6] Compared to classical GAN, QGAN efficiently handle high-dimensional data representation and generation, harnessing the benefits of superposition and parallel computation.[5,7,8] In 2019, Seth Lloyd *et al.* systematically investigated the theoretical framework of QGAN, marking a new era for this field.[7] Subsequently, Christa Zouf, Stefan Woerner, and others utilized QGAN models to generate discrete data, further advancing the capabilities of QGAN.[8–10] In 2021, Jian-wei Pan, He-liang Huang, and their team successfully demonstrated image generation on a superconducting device, paving the way for QGAN to be deployed in real-world scenarios and opening up new avenues for practical applications.[9] Leveraging their exponential computational speed advantage, QGAN have emerged as one of the most popular quantum neural network models in current research.

However, QGAN still encounters some challenges in quantum devices, especially readout noise caused by measurement instability. This problem can significantly affect the statistical results and the expressiveness of the overall circuit.[11–13] At the same time, it dominates and becomes the most important error type compared with other noises in quantum circuits.[14–16] Although some classical error correction methods like quantum error correction codes and zero-noise extrapolation techniques, etc. can mitigate certain types of noise, they are unable to effectively tackle the underlying problem of readout errors caused by inherent instability in the measurement gates.[17–19] Therefore, classical statistical based post-processing techniques are used to solve this problem.[15,20] This method involves measuring the conditional probability at each ground state and creating a response matrix for estimating the expected value of the true observables, which can mitigate readout errors in QGAN.[21] However, the process of measuring the bias for each ground state requires a significant consumption of measurement shots, making it impractical for scenarios with limited measurement budgets and medium to large-scale quantum circuits.[15,16,22]

To address the above problems, we use an efficient and economical method called bit-flip averaging (BFA).[23] BFA method simplifies the model of the response matrix by averaging the qubit flipping over the entire quantum circuit. By

†Corresponding author. E-mail: fanxingkui@126.com

© 2024 Chinese Physical Society and IOP Publishing Ltd

measuring only the elements of the first column, it becomes possible to estimate the entire response matrix. Therefore, BFA resolves the cost issue associated with full error mitigation and effectively mitigates readout errors using a finite number of measurements. This approach reduces the measurement cost from $O(2^{2n})$ to $O(2^n)$, providing a viable solution for mitigating readout errors in medium to large-scale quantum circuits.[24] In this paper, we design a scheme that can ingeniously integrate the BFA method and QGAN, and propose a new QGAN model that includes a fault-tolerant mechanism. We performed a simulation of the QGAN-based handwritten digit image recognition generation task on Qiskit. In this simulation, we do not consider readout errors caused by phase flips because they have no effect on the statistical results of the quantum circuit.[21,23] For the error mitigation process, we follow two main steps. Firstly, we analyze the influence of noise on the quantum circuit and compute the response matrix using the BFA model. Then we utilize the response matrix to estimate the true probability distribution after each training iteration on the QGAN-generated images. Our results show that new QGAN with BFA method can generate realistic images when the readout error probability is $p = 0.1$ or lower. Furthermore, within a constrained cost framework, full error mitigation only handles readout errors with probability as low as $p = 0.01$. To quantify the effectiveness of error mitigation, we employ metrics such as Kullback–Leibler (KL) divergence and state fidelity, both of which provide promising results. Our study shows that this new quantum neural network model can effectively handle readout errors at high error thresholds, thereby enhancing the overall robustness of the model. By transitioning QGAN from an idealized environment to a realistic and noisy domain, our study successfully bridges the gap between theory and reality and addresses the challenges posed by noisy environments. Thus, our study provides an efficient approach to the problem of image generation in practical, noisy quantum computing scenarios, providing valuable insights into noise-related challenges and their solutions.

2. Related work

2.1. Quantum generative adversarial networks

The principle of QGAN is illustrated in Fig. 1, comprising a quantum generator and a classical discriminator. In QGAN, classical data can be encoded as quantum states, wherein a vector $\boldsymbol{v}_j$ in a normalized N-dimensional complex or real space can be represented as a $\log N$ quantum state $|v_j\rangle$. The covariance matrix of classical vectors can be expressed as the density matrix $\boldsymbol{\rho} = p_j \sum_j^N |v_j\rangle\langle v_j|$, where p_j represents the probability of each quantum state $|v_j\rangle$ in the N-dimensional vector. This allows classical information, such as images, audio, stocks, etc. to be encoded in a normalized quantum form, harnessing the exponential computational advantages offered by quantum systems.[25,26]

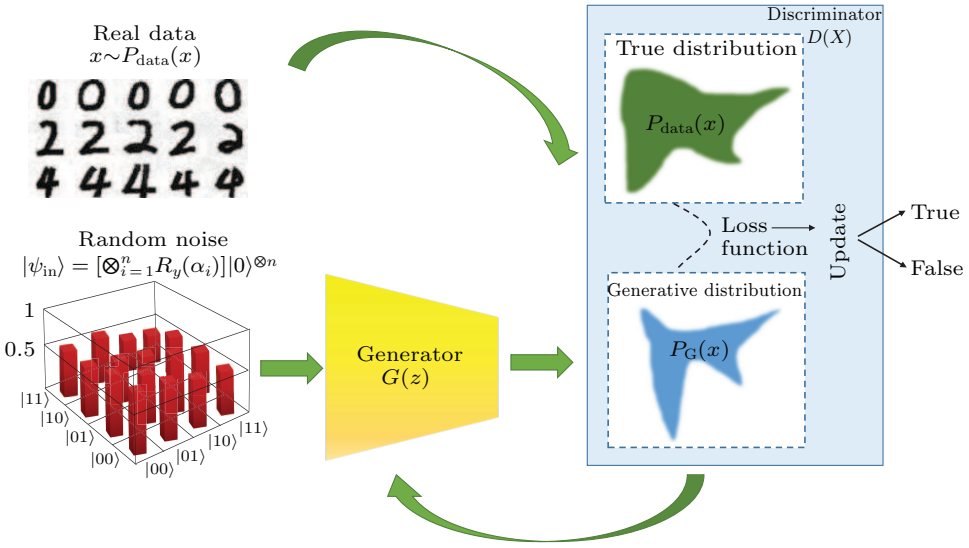


Fig. 1. Describe the working principle of the generated adversarial network. The yellow part is the quantum generator, and the dotted line on the far right is the classical discriminator.

Therefore, the essence of training a quantum generative adversarial network (QGAN) lies in generating a statistical distribution in density matrix form that closely resembles the real probability distribution. We represent the true probability distribution $\boldsymbol{P}_{\text{true}}(x)$ by sampling the classical training $X = x^0, x^1, \ldots, x^{s-1}$, where s is the number of data sets. In the context of image generation, these classical probability distributions are utilized to represent quantum image information. Specifically, the input state of the quantum generator is composed of random noise, which is employed to generate a fake probability distribution denoted as $\boldsymbol{P}_z(x)$ that approximates the true probability distribution $\boldsymbol{P}_{\text{true}}(x)$. The discrimi-

nator is trained on a classical computer and takes as input the probability distribution generated by the generator and the true probability distribution. Its output is a binary classification, where a value of "true" indicates that the input probability distribution aligns with the real probability distribution, while a value of "false" signifies a mismatch between the input probability distribution and the real one. The generator aims to fool the discriminator by generating realistic data, while the discriminator provides feedback to the generator by distinguishing generated data from real data. This is a training process involving adversarial games, and the training stops when the Nash equilibrium point is reached. At this time, the generated results are very close to the real probability distribution.

2.2. Full error mitigation

The traditional full error mitigation method is mainly divided into two parts:

(1) Calibration and measurement process, which is used to measure the readout error model M in the fixed device before designing the parametric variational unitary gate and training the QGAN. In n-qubit quantum circuits in noisy environments, we introduce the positive operator valued measure (POVM), which uses the n-bit string x to represent the measurement operator of 2^n elements as $\Pi_x = |x\rangle\langle x|$. In the measurement process, we assume that the observed measurement outcome is $|y\rangle$, and its real outcome is $|x\rangle$, so the new POVM element can be reduced to the diagonalized form

$$\Pi y = \sum_x \langle y|\boldsymbol{M}|x\rangle|x\rangle\langle x|, \quad x \in \{0,1\}^n, \tag{1}$$

where $\boldsymbol{M}$ is the response matrix, its approximate unbiased estimate can be obtained by normalizing the empirical average[20] after multiple measurements of different ground states. For simple understanding, we write the response matrix of a qubit as $\boldsymbol{M} = \begin{bmatrix} 0.6 & 0.3 \\ 0.4 & 0.7 \end{bmatrix}$. The probability of obtaining the measurement result $\langle 1|$ for the actual outcome $\langle 0|$ is 40%, while the probability of obtaining the measurement outcome $\langle 0|$ for the actual state $\langle 1|$ is 30%. We generalize it to n-qubit quantum system, then the response matrix can be written as

$$\boldsymbol{M} = \begin{bmatrix} p(\sigma'^1 \mid \sigma^1) & \cdots & p(\sigma'^1 \mid \sigma^n) \\ \vdots & \ddots & \vdots \\ p(\sigma'^n \mid \sigma^1) & \cdots & p(\sigma'^n \mid \sigma^n) \end{bmatrix}, \quad \sigma^i \in \{0,1\}^n. \tag{2}$$

The matrix in Eq. (2) captures the error probabilities associated with each quantum state across n qubits. Here $p(\sigma'^n|\sigma^n)$ represents an element of the $\boldsymbol{M}$ matrix, denoting a conditional probability that explains the likelihood of observing the state $|\sigma^n\rangle$ as $|\sigma'^n\rangle$. The response matrix $\boldsymbol{M}$ has a total of 2^{2n} parameters, requiring normalization through multiple measurements of these 2^{2n} parameters to obtain statistically significant results.

(2) The inverse of the response matrix $\boldsymbol{M}$ is used to estimate the true probability distribution in each training iteration for QGAN to reduce the readout error

$$\boldsymbol{P}_{\text{true}} = \boldsymbol{M}^{-1}\boldsymbol{P}_{\text{readout}}, \tag{3}$$

where $\boldsymbol{P}_{\text{readout}}$ and $\boldsymbol{P}_{\text{true}}$ represent the probability distribution in the presence of readout errors and the true probability distribution in the absence of readout errors, respectively.

The readout noise is correlated with the device in which it occurs, resulting in the identical readout noise for any quantum circuit within a given device. Consequently, by performing a calibration measurement once, we can utilize error mitigation techniques to alleviate errors in other quantum circuits,[27] However, this full error mitigation approach requires multiple measurements for each computational basis, resulting in a measurement cost of $O(2^{2n})$. In the next chapter, we design an executable algorithm on the QGAN network in conjunction with an efficient and cost-saving bit-flip averaging approach to error mitigation.

3. Algorithm analysis and design

3.1. Model design

In order to measure quantum circuits of a specific structure in calibration ease, we need to assume the corresponding structure of QGAN in advance, mainly a quantum generator and a classical discriminator. A quantum generator is used to generate fake samples, consisting of a parameterized quantum circuit (PQC), as shown in Fig. 2.[28–30] The PQC is composed of multiple layers of parametric variational unitary gate, such as $R_y(\theta)$, $R_z(\theta)$, etc. These gates represent parametric angles of rotation around the Y or Z axis, where θ represents the angle of rotation. Additionally, there are global parameter gates, which are untrained quantum gates responsible for connecting different qubits, such as CNOT and CZ gates, representing controlled NOT and controlled Z gates respectively.[31] The input to the quantum generator is a random angular noise vector rotated around the Y axis

$$\begin{aligned} |z\rangle &= |\psi_{\text{in}}\rangle = \otimes_{i=1}^n R_y(\alpha_i)\langle 0|_1\langle 0|_2\cdots\langle 0|_n \\ &= [\otimes_{i=1}^n R_y(\alpha_i)]\langle 0|^{\otimes n}, \end{aligned} \tag{4}$$

where the gate $R_y(\alpha_i)$ on the first layer corresponds to the Y-rotation gate with a random parameter α_i, $\alpha_i \in [0,\pi]$, the mathematics of $R_y(\theta)$ and $R_z(\theta)$ are expressed as

$$\begin{aligned} R_y(\theta) &= \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, \\ R_z(\theta) &= \begin{bmatrix} e^{-i(\frac{\theta}{2})} & 0 \\ 0 & e^{i(\frac{\theta}{2})} \end{bmatrix}. \end{aligned} \tag{5}$$

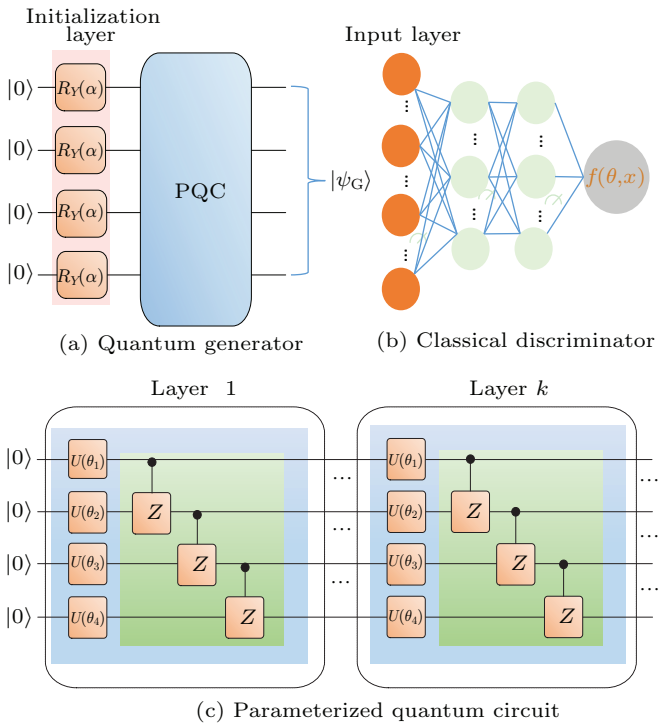


Fig. 2. (a) The structure of the quantum generator. (b) The structure of a classical discriminator. The discriminator consists of four layers of neurons, with the first layer as the input layer, the second and third layers as hidden layers, and the last layer applying a sigmoid activation function to the output. (c) A parameterized quantum circuit, which is the basic structure of a quantum generator. It consists of a global unitary gate and a trainable parameterized unitary gate. The parameters are optimized for each training and the results are obtained through measurement.

The essence of the PQC is a variational quantum algorithm that transforms the initialized quantum state through unitary gates with variational parameters. The output formula of the variational quantum algorithm is as follows:

$$\begin{aligned} |\psi_z\rangle &= (\otimes_{i=1}^{n} \boldsymbol{U}(\theta_i^l)\boldsymbol{U}(\theta_i^{l-1})\cdots\boldsymbol{U}(\theta_i^1)) \\ &\quad \times \left(\prod_{j=1}^{l} \otimes_{k=1}^{n-1} CZ_{k,k+1}\right)|\psi_{\text{in}}\rangle \\ &= \sum_{i=0}^{2^n-1} \sqrt{p_i}|i\rangle, p_i \in [0,1], \end{aligned} \tag{6}$$

where $|\psi_z\rangle$ represents the generated quantum state, and the coefficient p_i represents the probability of the measurement result corresponding to the ground state $|i\rangle$. $\boldsymbol{U}(\theta_i^l)$ represents a unitary gate on the ith qubit in the kth layer, which includes gates such as $R_y(\theta)$ and $R_z(\theta)$ as described earlier. CZ denotes the controlled-Z gate that operates on two qubits. All unitary gates within the same layer are represented by a tensor product. The total number of layers in the quantum circuit is denoted as 'l'. Finally, the result of the density matrix is obtained by projection measurement, and the measurement operator is the tensor product of all qubit projection operators.

In the image generation study conducted in this paper, we employ a quantum circuit consisting of four sub-generators, each with a specific structure as depicted in Fig. 3. These subgenerators are responsible for generating different parts of the image, which are eventually combined to form a complete image. Here, each subgenerator follows the same architecture and consists of four qubits. The first layer acts as a random initialization layer, and the PQC is represented by Eq. (6). The probability distribution generated by the sub-generator is expressed as $G(Z) = [G^{(1)}(z), \ldots, G^{(i)}(z), G^{(4)}(z)]$, where $G^{(i)}(z)$ represents the projection measurement result on the ith subgenerator.

As shown in Fig. 2(b), the structure of the discriminator adopts a classical full-connected neural network architecture designed for binary classification. The output of the discriminator is recorded as $D(x)$ and $D(G_{\theta_g}(z))$, which represent the evaluation results of the discriminator on the real sample and the generated sample, respectively, expressed as "true" or "false". Among the neurons in different layers, the initial input layer contains 64 neurons, which are subsequently fed to the first hidden layer using the ReLU activation function, and the hidden layer maps 64 neurons into 16. Finally, these 16 neurons are mapped to a final single sigmoid activation function for nonlinear output, producing values from 0 to 1.

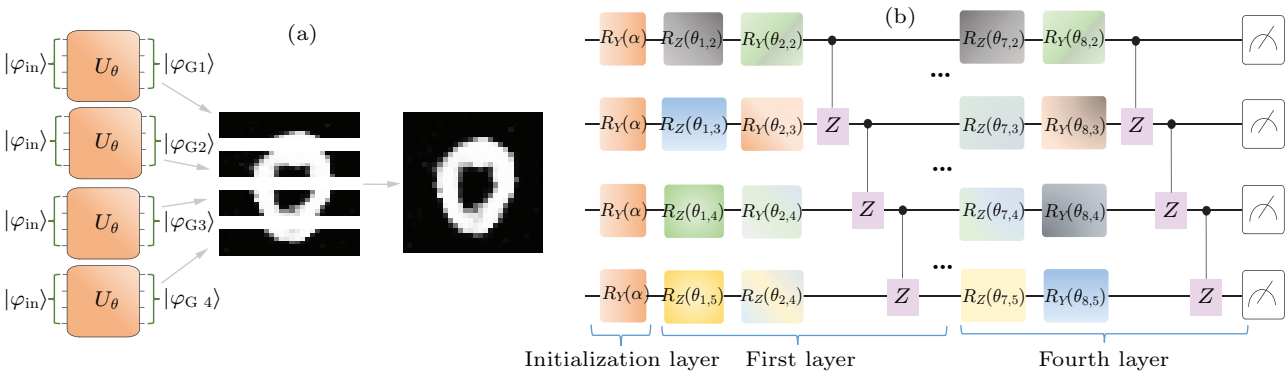


Fig. 3. (a) The combined image generated by the 4 quantum generators. (b) The specific structure of each quantum generator used in the experiment. The first layer is randomly rotated R_y gates. There are four layers in the middle layer, and each layer consists of a layer of R_z gates, a layer of R_y gates and a layer of CZ gates. The last layer is the measuring gate.

3.2. Bit-flip averaging

Unlike full error mitigation,[32,33] bit-flip averaging (BFA) simplifies the structure of the response matrix by averaging different bit-flips, as shown in Fig. 4. The bit-flip operations have minimal cost and high fidelity, especially when integrated with PQC. Furthermore, classical X-gates used for post-processing are executed on classical computer without incurring the overhead of quantum resources, so we do not need to consider the negative impact of bit-flips.[23]

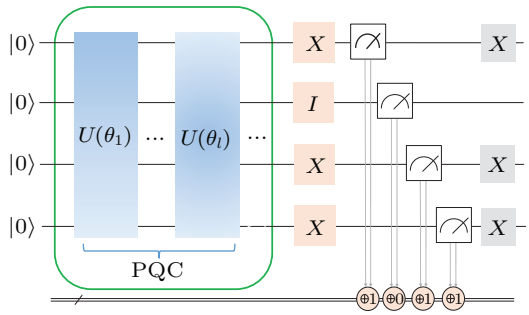


Fig. 4. The principle of bit-flip averaging. Bit-flips are represented by random X-gates while ensuring that the probability of being flipped to each ground state is the same. In the calibration phase there is no PQC structure (shown in the green border) and the response matrix is computed by measuring the initial state $|0000\rangle$ result. In error mitigation, the quantum generator containing the PQC structure is used to measure the readout results and estimate the true results based on the response matrix, where a bit-flip averaging operation of the same principle is performed before the measurement. The corresponding classical X-gate operation is used in classical post-processing to flip the corresponding flipped qubits again to ensure consistent results.

The initial response matrix in full error mitigation is shown in Eq. (1), where each element of the matrix is a conditional probability that the measured ground state is σ' when the true ground state is σ, denoted by the equation $M_{\sigma\to\sigma'} = p(\sigma'|\sigma)$.

In the context of BFA, the initial conditional probability of different true ground state after a random bit-flip becomes $p(\hat{X}^j\sigma'|\hat{X}^j\sigma)$, where we use the X-Pauli operator $\hat{X}$ to denote the bit-flip operation, and j denotes randomly flipped qubits expressed in binary integer form. For example, $j = 1011$ implies that the X-Pauli operator $\hat{X}^{1011}$ performs the 1st, 3rd and 4th qubits flips on the target state, which for an n-qubit ground state would result in a flip of 2^n. This random bit-flipping ensures that any ground state can be randomly flipped into any computational basis in the same Hilbert space. For any conditional probability $p(\sigma'|\sigma)$ in the full error mitigation matrix, the formula for averaging over all possible random flips on the ground state is as follows:

$$\tilde{M}_{\sigma\to\sigma'} = \tilde{p}(\sigma'|\sigma) = \frac{1}{2^n}\sum_j (\hat{X}^j\sigma'|\hat{X}^j\sigma), \tag{7}$$

where $\tilde{p}(\sigma'|\sigma)$ is a new conditional probability after bit-flip averaging. From this equation we have $\tilde{p}(\sigma'|\sigma) = \tilde{p}(\hat{X}^j\sigma'|\hat{X}^j\sigma)$, therefore, the elements of this response matrix are symmetrically equal on the diagonal and the antidiagonal. This symmetry-equal structure can be obtained by bit-flip averaging, where the result of measuring only part of the ground state can represent the whole matrix. Therefore, the first column element of the matrix can be obtained based on the measurements under bit-flip averaging when the input ground state is $|0\cdots0\rangle_n$, to represent the entire BFA matrix

$$\tilde{p}(\sigma'|\sigma) = \tilde{p}(\hat{X}^{\sigma'}\sigma||0\cdots0\rangle_n), \tag{8}$$

where the ground state σ' is represented as binary integers, i.e., if σ and σ' are $|01\rangle$ and $|10\rangle$, respectively, then it can be expressed as $\hat{X}^{(10)}|01\rangle = |11\rangle$. The formula says that for any element $\tilde{p}(\sigma'|\sigma)$ of the matrix, it can be represented by a different conditional probability in the first column, and for clarity and visibility we can write $\tilde{p}(\hat{X}^{\sigma'}\sigma||0\cdots0\rangle_n)$ as $\tilde{p}(\hat{X}^{\sigma'}\sigma)$. In addition, this formula has another meaning, which expresses the probability of an error at different qubits. Suppose that in a bit-flip averaging of two-qubit system, the probability that the ground state $|01\rangle$ undergoes bit-flip averaging to obtain the result $|10\rangle$ can be expressed as $\bar{p}(\hat{X}^{(10)}01) \equiv \bar{p}(11)$, which corresponds to the probability that the first and the second quantum bits are in error (the elements of the first column of the matrix represent the probability of error for different qubits). Based on this symmetry and antisymmetry, this new matrix can be expressed as a succinct sparse representation for ease of computation: $\tilde{M} = \sum_j p(j)\hat{X}^j$, where $p(j)$ is the probability of error for different qubits corresponding to different j. To ensure the symmetry and antisymmetry of such matrices, we need to use the same flipping principle for bit-flip averaging of the output of the parametrized quantum circuits, then use the response matrix $\tilde{M}$ for error mitigation according to Eq. (3).

3.3. Loss function

The loss function of the quantum generative adversarial network is defined with reference to the classical model as

$$\begin{aligned} L(\theta_g, \theta_d) = {} & E_{x\sim P_{\text{data}}(x)}[D_{\theta_d}(x)] \\ & + E_{x\sim P(z)}[(1 - D_{\theta_d}(G_{\theta_g}(z)))], \end{aligned} \tag{9}$$

where $D_{\theta_d}(x)$ and $D_{\theta_d}(G_{\theta_g}(z))$ represent the outputs of the discriminator when given real and generated samples, respectively. $G_{\theta_g}(z)$ represents the measurement results generated by the generator, and x is the real sample. The generator aims to minimize the loss function in order to generate probability distributions that closely resemble real data. On the other hand, the discriminator aims to maximize the loss function by optimizing its parameters. This optimization ensures that $D(x)$ assigns high probabilities to real samples classified as real, while $D(\theta_g)$ assigns low probabilities to generated samples labeled as real.[34]

However, classical GAN often encounter the problem of vanishing gradients in the generator during discriminator feedback. To alleviate this problem, a possible solution is to train

the generator and discriminator independently using a non-saturated loss function. By adopting this approach, we can alleviate the problem of vanishing gradients and facilitate more efficient learning of the generator and discriminator.[35] The loss function is rewritten as

$$L_{\mathrm{G}}(\theta_{\mathrm{g}},\theta_{\mathrm{d}}) = -E_{x\sim p(z)}[D_{\theta_{\mathrm{d}}}(G_{\theta_{\mathrm{g}}}(z))], \tag{10}$$

$$L_{\mathrm{D}}(\theta_{\mathrm{g}},\theta_{\mathrm{d}}) = -E_{x\sim p_{\mathrm{data}}(x)}[D_{\theta_{\mathrm{d}}}(x)] - E_{x\sim p(z)}[\log(1-D_{\theta_{\mathrm{d}}}(G_{\theta_{\mathrm{g}}}(z)))]. \tag{11}$$

Equations (10) and (11) denote the loss functions of the generator and discriminator, respectively. We fix the generator, calculate the derivative of the loss function, and derive the optimal value of the discriminator $D\left(G_{\theta_{\mathrm{g}}}(z)\right)$ as follows:

$$D_{\mathrm{opt}}\left(G_{\theta_{\mathrm{g}}}(z)\right) = \frac{\boldsymbol{P}_{\mathrm{data}(x)}}{\boldsymbol{P}_{\mathrm{data}(x)}+\boldsymbol{P}_{G(x)}}. \tag{12}$$

Equation (12) represents the output value of the discriminator when the loss function reaches the extreme point, where $\boldsymbol{P}_{\mathrm{data}(x)}$ and $\boldsymbol{P}_{G(x)}$ represent the probability distribution of real data and generated data, respectively.

Next fix the discriminator and substitute $D_{\mathrm{opt}}\left(G_{\theta_{\mathrm{g}}}(z)\right)$ into the loss function. When $\boldsymbol{P}_{\mathrm{data}(x)} = \boldsymbol{P}_{G(x)}$, the loss function of the generator is minimized. In this case, the probability distribution generated by the generator approaches the real probability distribution, and the discriminator can no longer distinguish the real data from the generated data, indicating that the Nash equilibrium point has been reached.

3.4. Model training

3.4.1. Data preprocessing

We use Pytorch[36,37] for classical training and IBM Qiskit[38,39] to simulate the associated quantum circuits and readout errors, in addition, we use "Mitiq" package to implement the "full error mitigation" approach. The dataset used in this paper is images of handwritten digits with a resolution of 8×8 pixels, as cited in Ref. [40]. Since the color and position of each image pixel are represented by classical numbers, we can represent the pixel position and color information in image using quantum states and their coefficients according to the quantum image (FRQI) encoding method.[41–43] We set the number of iterations to 350 and use the Adam optimizer, and set the initial learning rates of the generator and discriminator to $\alpha_{\mathrm{G}} = 0.05$ and $\alpha_{\mathrm{D}} = 0.001$ respectively. Subsequently, we trained the QGAN following the training procedure outlined in Table 1.

3.4.2. Training process

We use the "QuantumCircuit" class in Qiskit's QasmSimulator simulator to build quantum circuits and define methods to represent various unitary gates.[44] To obtain the subgenerator results, we generate a quantum state for projection measurement, yielding a new set of measurement probabilities

$$G^{(i)} = [P^{(i)}(0),P^{(i)}(1),\ldots,P^{(i)}(j),\ldots,P^{(i)}(2^N-1)], \quad N=4, \tag{13}$$

where $P^{(i)}(j)$ represents the measurement probability of the jth quantum state in the ith sub-generator, where $j\in[0,2^N]$ and N is the number of qubits. In the context of an image, it represents information related to the jth pixel. $G^{(i)}$ represents a part of handwritten digital image content generated by a sub-generator. Therefore, the entire image can be represented as follows:

$$\tilde{x} = [\tilde{x}^{(1)},\ldots,\tilde{x}^{(N_{\mathrm{G}})}], \quad N_{\mathrm{G}}=4, \tag{14}$$

Algorithm 1 Training process

1. Select a handwritten digit recognition image dataset and set a random number seed.
2. Build a quantum circuit and measure the error model using the BFA method.
3. Construct four quantum sub-generators and a classical fully connected neural network, setting the number of circuits per subgenerator, $n=4$. Initialize different learning rates $\alpha_{\mathrm{G}}=0.3$ and $\alpha_{\mathrm{D}}=0.01$

   Wight: *Generator* θ = *Rand num between* $[0,\pi]$

   Discriminator random parameter γ

   Epochs: $K=350$

1: **while** θ, γ less than K **do**

2: Train the discriminator on real and generated data:

3: Computer the loss function according to the updated parameter γ

4: Computer $Grad_{\mathrm{D}} = \mathrm{d}L/\mathrm{d}\gamma$

5: Update $\gamma = \gamma - \alpha_{\mathrm{D}} Grad_{\mathrm{D}}$

6: Train the generator on generated data:

7: Perform projection measurements on the resulting quantum states and use the measured error model $\tilde{M}$ (response matrix) for error mitigation.

8: Computer the loss function according to the updated parameters θ

9: Computer $Grad_{\mathrm{G}}$

10: Update $\theta = \theta - \alpha_{\mathrm{G}} Grad_{\mathrm{G}}$

11: Representation as an image based on projection measurements

12: **return** result

where $N_{\rm G}$ represents the number of sub-generators. In our case, we utilize 4 parameterized quantum circuits (PQC) to represent the synthesis of images consisting of 2^4 pixel points generated by each sub-generator. During training, we update the gradient of the loss function by backpropagation in a classical computer. For the classical discriminator, its gradient can be directly calculated as

$$\nabla_\gamma L_{\rm D}\left(\theta_{\rm g},\theta_{\rm d}\right)=\frac{\partial L(\theta,\gamma)}{\partial\gamma},\quad \gamma=\theta_{\rm d}. \tag{15}$$

The gradient $\nabla_\gamma L_{\rm D}\left(\theta_{\rm g},\theta_{\rm d}\right)$ represents the derivative calculation of the discriminator loss function $L_{\rm D}\left(\theta_{\rm g},\theta_{\rm d}\right)$ with respect to the weight parameters γ of the discriminator's fully connected neural network.

In the quantum generator, we use the unbiased estimation method,[45] which is obtained by computing the original expectation twice. Its gradient formula is as follows:

$$\begin{aligned}\nabla_{\theta_j}L_{\rm G}(x)=&\frac{1}{2}\big[L_{\rm G}(\theta_1,\theta_2,\ldots,\theta_l^{+},\ldots,\theta_n,\gamma)\\&-L_{\rm G}(\theta_1,\theta_2,\ldots,\theta_l^{-},\ldots,\theta_n,\gamma)\big].\end{aligned} \tag{16}$$

This formula represents the partial derivative of the loss function of the sub-generator with respect to the lth parameter θ_l. Here $(\theta_l)^{\pm}=\theta_j\pm\pi/2$ denotes the rotation of the gate parameter θ_l by $\pi/2$ and $-\pi/2$, respectively.

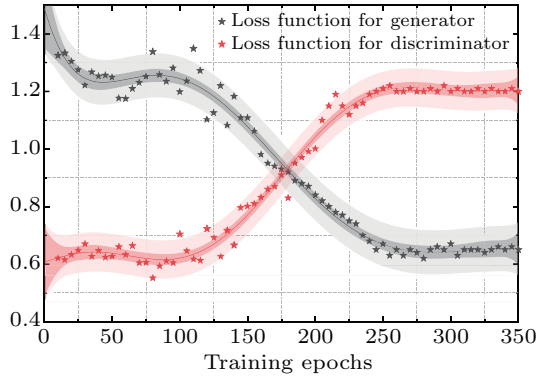


Fig. 5. The loss function for training images in a noise-free environment. The gray dots are the loss function of the quantum generator, and the red dots are the loss function of the classical discriminator. The figure shows the loss function for every five iterations, fitting an approximate curve and plotting the confidence space.

Our training process achieves the optimization of the loss function by continuously updating the gradients of the generator and the discriminator, and the formula is as follows:

$$\theta^k=\theta^{k-1}-\alpha_{\rm G}\,\nabla_\theta L_{\rm G}\left(\theta_{\rm g},\theta_{\rm d}\right), \tag{17}$$

$$\gamma^k=\gamma^{k-1}-\alpha_{\rm D}\,\nabla_\gamma L_{\rm D}(\theta_{\rm g},\theta_{\rm d}). \tag{18}$$

The index "k" in the equation represents the kth iteration during the training process. $\alpha_{\rm G}$ and $\alpha_{\rm D}$ represent the learning rates of the generator and discriminator, respectively. θ^k and γ^k represent the parameters of the generator and discriminator updated during the kth iteration. Figure 5 shows the loss function we trained in the noise-free environment. When the final loss function no longer changes, the Nash equilibrium point is reached,[46] indicating that our image synthesis has converged to the optimal training expectation.

3.4.3. Error mitigation

In order to simulate readout noise in the same training environment, we utilized the "ReadoutError" class in Qiskit and set the probabilities of three different readout error probabilities (bit error flip rate) as $p=0.01$, $p=0.05$, and $p=0.1$.[47] Following the principle outlined in the "Bit-flip averaging" section, we constructed an error model by measuring the readout probability distribution. To incorporate random bit flipping, we designed a loop structure after constructing the quantum circuits. Within this loop, we introduced random numbers while traversing the circuit. Specifically, when generating random numbers, if the value was less than 0.5, we applied an X-gate flip to the corresponding qubit. This randomized process ensured that each qubit underwent a probabilistic flip operation. Through these steps, we were able to mitigate the effects of coarse measurements, thus creating a more robust and accurate error model. In both the full error mitigation method and the bit-flip averaging (BFA) method, we performed the calibration process by measuring it 120 times. For the BFA method this involved looping through the process 120 times to acquire the necessary measurement data. It is worth noting that while classical computers can simulate a large number of circuit measurements, real quantum computers have limitations in terms of the number of measurements they can perform at a manageable cost.

To ensure that the experimental setup closely reflects the constraints of real quantum devices and increase the realism of the experiment, we intentionally refrained from simulating a larger number of measurements. This approach aligns with the limitations imposed by current quantum hardware and allows us to better understand and address the challenges associated with practical implementation on real quantum devices.[48] After measuring the error model, denoted as $\tilde{\boldsymbol{M}}$, our approach aligns with the error mitigation technique described in the "Full error mitigation" section.[49–51] This involves estimating the true density matrix after each iterative training and ultimately encoding authentic density matrices into visually compelling digital images. In our setup, since the discriminator runs on a classical computer, our focus for error mitigation is the quantum generator in QGAN. To estimate the true probability distribution $\boldsymbol{P}_{\rm true}$, we need to compute its inverse from the measured response matrix

$$\boldsymbol{P}_{\rm true}=\tilde{\boldsymbol{M}}^{-1}\boldsymbol{P}_{\rm readout}, \tag{19}$$

where $\tilde{\boldsymbol{M}}^{-1}$ represents the inverse of the response matrix $\tilde{\boldsymbol{M}}$. $\boldsymbol{P}_{\rm true}$ and $\boldsymbol{P}_{\rm readout}$ denote the true quantum state distribution in

the absence of noise and the measured quantum state distribution in the presence of readout errors respectively. In order to calculate the inverse matrix quickly and conveniently, we use the method in Ref. [23]. The first step of this method is to diagonalize the response matrix $\tilde{\boldsymbol{M}}$. Next, we supplement and analyze the solution process of the inverse matrix to better understand the response matrix $\tilde{\boldsymbol{M}}$. Its diagonalization can be realized by using the Hadamard matrix. The transformation equation is as follows:

$$\boldsymbol{H}^{\otimes n}\tilde{\boldsymbol{M}}\boldsymbol{H}^{\otimes n}=\sum_j \widetilde{p}_j \boldsymbol{Z}^j. \tag{20}$$

Equation (20) represents the diagonalization matrix obtained through the Hadamard matrix transformation. Here $\boldsymbol{H}^{\otimes n}$ denotes the Hadamard matrix for n qubits, while $\boldsymbol{Z}^j$ represents the phase-flip operator, which operates on the same principles as $\boldsymbol{X}^j$. The diagonal elements of the Pauli matrix $\boldsymbol{Z}^j$ correspond to the phase-flip operation

$$Z^j_{ii}=\Pi_k(-1)^{i_k j_k}=(-1)^{i\cdot j}. \tag{21}$$

Equation (21) represents the ith diagonal element in Pauli's matrix $\boldsymbol{Z}$, $i\cdot j$ is the dot product. It can be expressed in the form of a Hadamard matrix

$$H_{ij}=\frac{(-1)^{ij}}{\sqrt{2}}\rightarrow Z^j_{ii}=\sqrt{2^n}H^{\otimes n}_{ij}, \tag{22}$$

where H_{ij} is the corresponding element in the Hadamard matrix. Thus the eigenvalues of $\tilde{\boldsymbol{M}}$ can be given by diagonal elements of $\sum_j \widetilde{p}_j \boldsymbol{Z}^j$

$$\begin{aligned}\lambda_i &= \sum_j \widetilde{p}_j Z^j_{ii}=\sum_j \widetilde{p}_j(-1)^{j\cdot i}=\sqrt{2^n}\sum_j H^{\otimes n}_{ij}\widetilde{p}_j\\ &= \sqrt{2^n}(H^{\otimes n}_{ij}\widetilde{p})_i\rightarrow \lambda=\sqrt{2^n}\boldsymbol{H}^{\otimes n}\widetilde{p}.\end{aligned} \tag{23}$$

The Hadamard matrix transformation can also be represented by λ_i, which is the ith eigenvalue and λ represents the set of eigenvalues for all elements. In Eq. (23), $\widetilde{p}$ is a probability vector composed of $\widetilde{p}_j$ values, corresponding to the values in the first column of $\tilde{\boldsymbol{M}}$. Therefore, the inverse transformation of diagonalization can be expressed as

$$\boldsymbol{H}^{\otimes n}\tilde{\boldsymbol{M}}^{-1}\boldsymbol{H}^{\otimes n}=\boldsymbol{\Lambda}^{-1}=\mathrm{diag}(1/\lambda_0,\ldots,1/\lambda_{2n-1}). \tag{24}$$

Equation (24) is the projection of $\boldsymbol{H}^{\otimes n}\tilde{\boldsymbol{M}}^{-1}\boldsymbol{H}^{\otimes n}$ on different Pauli matrices $\boldsymbol{Z}^j$. Finally we de-diagonalize the matrix with the formula

$$\begin{aligned}\widetilde{q}(j) &= \frac{1}{2^n}\mathrm{Tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^j)\\ &= \frac{1}{\sqrt{2^n}}(\boldsymbol{H}^{\otimes n}\lambda^{-1})_j\rightarrow\tilde{\boldsymbol{M}}^{-1}=\sum_j \widetilde{q}(j)\boldsymbol{X}^j.\end{aligned} \tag{25}$$

Based on the equation $\mathrm{Tr}(\boldsymbol{Z}(s)\boldsymbol{Z}(t))=2^n\delta_{st}$, the coefficient $\widetilde{q}(j)$ is obtained by multiplying $\boldsymbol{\Lambda}^{-1}$ with different $\boldsymbol{Z}^j$ matrices. It serves as a coefficient for $\tilde{\boldsymbol{M}}^{-1}$ and is multiplied by different $\boldsymbol{X}^j$ matrices to obtain the inverse matrix after diagonalization cancellation.

We find that the inverse matrix is a simple symmetric structure. Combining Eqs. (19) and (25) allows us to perform error mitigation on the estimation of the quantum circuit. The quantum generator estimates the true density matrix at the output of each iteration, and each training result restores an approximate true probability distribution.

4. Simulation analysis

We performed image training in three different environments: a noise-free quantum circuit, the BFA method, and full error mitigation with readout noise. Simulation experiments were conducted to demonstrate the advantages of the BFA method in mitigating errors during image generation. Figure 6 displays the results of image generation, illustrating that training the model with readout errors causes difficulties in generating realistic samples. However, our observations revealed that using the BFA method allows us to generate images that closely resemble those generated in a noise-free environment. This empirical evidence highlights the effectiveness of the BFA method in reducing the impact of readout errors and improving the quality of generated images. By incorporating the BFA method into the training process, we achieve significant improvements in image generation performance.

In order to quantify the performance of image generation, we evaluate its metrics using the Kullback–Leibler (KL) divergence,[52] which measures the similarity between the probability distributions of generated samples and target samples

$$D_{\mathrm{KL}}(P\parallel Q)=\sum_{x=0}^{S-1}P(x)\log\frac{P(x)}{Q(x)}, \tag{26}$$

where $Q(x)$ represents the probability distribution of the generated data, while $P(x)$ represents the target probability distribution and S is the pixel distribution of the image composition. The formula represents the discrepancy between the probability distributions formed by the values of each pixel.[53,54] A lower value of KL divergence indicates a closer resemblance of the generated probability distribution to the true probability distribution.

During training, the KL divergence between the generated image and the real image is computed in post-processing on a classical computer after every ten iterations of image generation, the results are shown in Fig. 7. During the training process, the KL divergence decreases as the number of training times increases and the value of the KL divergence tends to minimize after the system reaches the Nash equilibrium point. The convergence of the KL divergence is very stable in an environment without readout errors, when the value stays floating around 0.03 after more than 300 training sessions. This shows that our model is able to generate highly realistic sample distributions and in the absence of errors, demonstrating that our QGAN structure can effectively solve the pattern collapse problem.

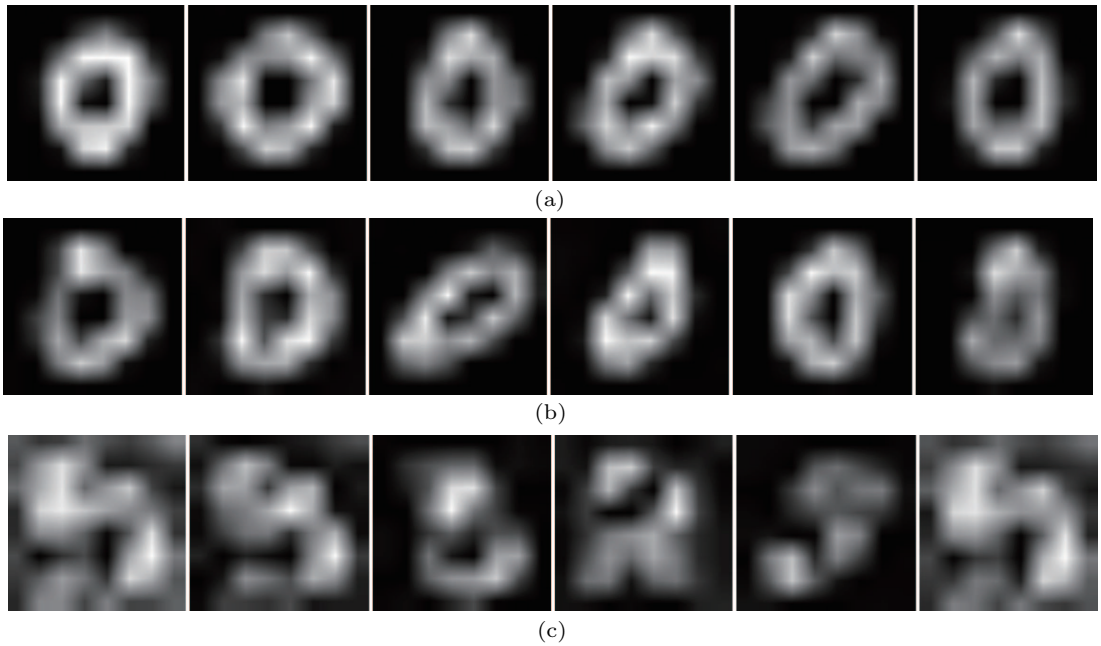


Fig. 6. The training results of 8×8 pixel handwritten digit images. (a) The images generated in the noise-free environment, (b) the images generated using the BFA method in the readout error simulation, and (c) the generated result under readout error without using any error mitigation method.

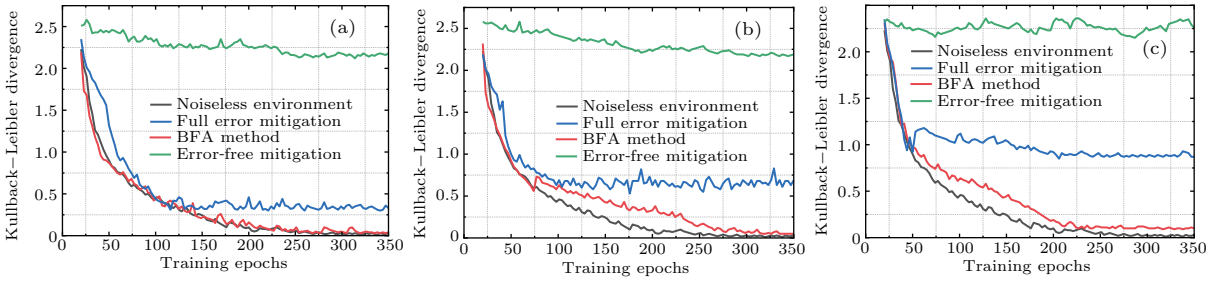


Fig. 7. The three plots depict the KL divergence as a line graph, with different colors representing different mitigation methods. The green line represents error-free mitigation, the red line represents the BFA method, the blue line represents full error mitigation, and the gray line represents a noiseless environment. Each corresponding scatter plot is fitted with a line to illustrate the trend of the KL divergence. (a) KL divergence at a readout error probability of $p = 0.01$, (b) KL divergence at a readout error probability of $p = 0.05$, and (c) KL divergence at a readout error probability of $p = 0.1$.

We conducted simulation experiments on readout errors, and by observing the experimental results, we found that the KL divergence of the model output showed extremely high fluctuations and oscillations at the beginning of the training period without any error mitigation measures. Despite improvement during the continuous optimization process of machine learning, the KL divergence value never converges to a minimum value due to the readout noise present in the measurement circuit itself. This experimental result clearly demonstrates that the continuous optimization training model is unable to resolve errors caused by readout noise, even when the probability of readout error is very low (e.g., $p = 0.01$).

Through the application of the BFA method, we observed significant improvements. First, during the calibration phase, we performed 120 projection measurements for each quantum sub-generator. Considering that the sub-generator is a four-qubit quantum circuit model (shown in Fig. 3), the time complexity of the measurement is satisfying $O(2^n)$. From the experimental results in Fig. 7, we can clearly see the effectiveness of the BFA method in error mitigation. When the readout error probabilities are $p = 0.01$ and $p = 0.05$, the KL divergence values converge to 0.04 and 0.05, respectively, which are almost equal to the convergence values of the KL divergence in the noise-free environment. Even with an readout error probability of $p = 0.1$, the KL divergence converges to a satisfactory level close to 0.1. While it may not achieve the same level of performance as in the absence of readout errors, it is still capable of generating reasonably realistic images, as illustrated in Fig. 6. In the context of full error mitigation, the response matrix cannot faithfully represent the error model during the calibration phase due to limited measurements and the necessity to measure every ground state. Additionally, the Kullback–Leibler (KL) divergence for the three readout error probabilities reaches a convergence turning point quite early and then remains at relatively high values thereafter.

To evaluate the error mitigation performance of the BFA method in terms of flexibility and generalization, we employ state fidelity[55,56] as a measure. However, since the discrimi-

nator is a classical model, real images are represented directly as classical pixel values without quantum encoding. Therefore, it is not possible to quantify the quantum state fidelity of real images. As an alternative, we compute the fidelity between the quantum states generated by different error mitigation methods in the training batch and the corresponding quantum states generated in the noise-free environment with the same training parameters. This experimental condition guarantees that there are only readout error-induced gaps between the two resulting quantum states. The formula for quantum state fidelity is as follows:

$$D_{\sigma}^{\text{fidelity}} = [\text{Tr}\sqrt{\sigma^{1/2}\rho\ \sigma^{1/2}}], \tag{27}$$

where σ denotes the quantum state generated in a noise-free environment, ρ is the quantum state corrected by the BFA method, the full error mitigation method and the error-free mitigation environment for different noise levels. Similarly, every ten training iterations, we ignore the image generation link, only calculate its density matrix, and average the fidelity of multiple training to get the result shown in Fig. 8.

Figure 8 illustrates the fidelity under different error mitigation methods for various readout error probabilities. We found that using the BFA method could reach a higher fidelity level in each training epoch. With all three different probabilities of readout noise, the BFA method can keep the average fidelity of the density matrix after error mitigation around 0.96, generating our desired target quantum state through the higher level of fidelity.

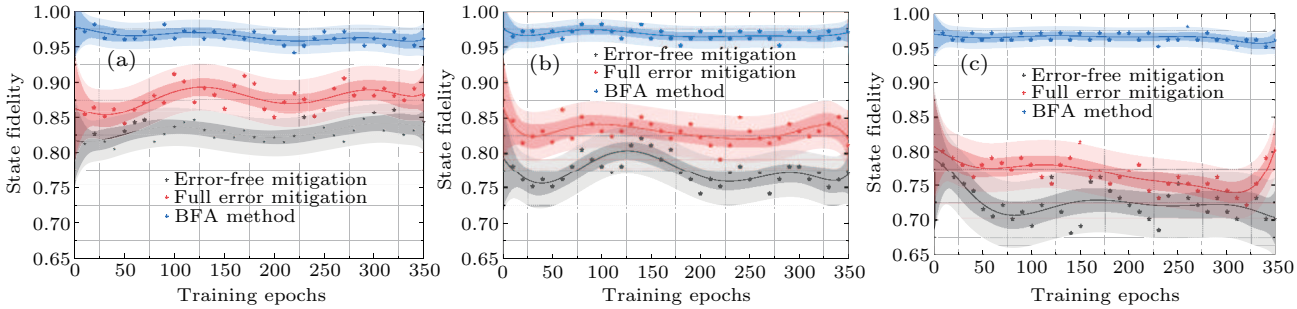


Fig. 8. As shown in the above figure, the gray dots represent the case without error mitigation, the blue dots represent the BFA method, and the red dots represent full error mitigation. We can plot the corresponding curves and confidence intervals based on the data points. (a)–(c) The fidelity of states in different scenarios corresponding to readout error probability of $p = 0.01$, $p = 0.05$, and $p = 0.1$, respectively.

In contrast to the BFA method, the full error mitigation approach did not achieve the expected results. Even at a lower error probability ($p = 0.01$), the fidelity value of 0.87 is less than ideal. Furthermore, at $p = 0.05$ and $p = 0.1$, the average fidelity of the density matrix sharply drops to around 0.5. This signifies that under the full error mitigation approach, the QGAN struggles to generate high-fidelity target quantum states. In the absence of error mitigation methods, we observe that it is impossible to generate high-fidelity samples regardless of the training parameters. Even in the best-case scenario ($p = 0.01$), the maximum fidelity is only around 0.85, which does not allow us to achieve the desired results. Through fidelity experiments, we once again demonstrate the effectiveness of the BFA method for the QGAN model, especially when dealing with readout error probabilities below 0.1.

5. Conclusion

We combine QGAN with the BFA method to form the training mechanism of the BFA-QGAN model. This model can obtain an accurate error model in a limited measurement environment and can achieve error mitigation with high fault tolerance performance.[57–59] Simple classical post-processing methods prove that variational quantum algorithms under high-cost measurement shots can be presented in the form of high-fidelity output density matrices. Through our model, the problems caused by readout noise to QGAN are solved.

Relative to the full error mitigation method, the BFA-QGAN model can effectively handle measurement noise up to a threshold of 0.1, which strongly verifies the robustness of QGAN in this paper. Furthermore, fidelity experiments confirm that our study is not limited to image generation, demonstrating the potential to be applicable to various generation tasks, thereby enhancing the generalization ability of the fault-tolerant mechanism.[60]

Moving forward, our future research endeavors will focus on exploring fault tolerance mechanisms in large-scale quantum circuits with an increased number of qubits. Additionally, we will investigate methods for enabling QGANs to achieve superior generalization capabilities in noisy environments. The theoretical framework and practical implementation presented in this study serve as a valuable reference for the advancement of quantum machine learning in noisy environments. Furthermore, we offer valuable insights for exploring quantum neural network models incorporating more robust fault tolerance mechanisms.[61,62]

Acknowledgements

Project supported by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2021MF049) and Joint Fund of Natural Science Foundation of Shandong

Province (Grant Nos. ZR2022LLZ012 and ZR2021LLZ001).

References

[1] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195

[2] Bartolucci S, Birchall P, Bombín H, Cable H, Dawson C, Segovia M G, Johnston E, Kieling K, Nickerson N, Pant M, Pastawski F, Rudolph T and Sparrow C 2023 *Nat. Commun.* **14** 912

[3] Stajic J 2023 *Science* **339** 1163

[4] Schuld M and Killoran N 2019 *Phys. Rev. Lett.* **122** 040504

[5] Zhou N R, Zhang T F, Xie X W and Wu J Y 2023 *Signal Processing: Image Communication* **110** 116891

[6] Niu M Y, Zlokapa A, Broughton M, Boixo S, Mohseni M, Smelyanskyi V and Neven H 2022 *Phys. Rev. Lett.* **128** 220505

[7] Lloyd S and Weedbrook C 2018 *Phys. Rev. Lett.* **121** 040502

[8] Stein S A, Baheri B, Chen D, Mao Y, Guan Q, Li A, Fang B and Xu S 2021 *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)* pp. 71–81

[9] Huang H L, Du Y, Gong M, *et al*. 2021 *Phys. Rev. Appl.* **16** 024051

[10] Zoufal C, Lucchi A and Woerner S 2019 *npj Quantum Information* **5** 103

[11] Preskill J 2018 *Quantum* **2** 79

[12] Saki A A, Alam M and Ghosh S 2021 *2021 22nd International Symposium on Quality Electronic Design (ISQED)* pp. 186-191

[13] Harrow A W and Montanaro A 2017 *Nature* **549** 203

[14] Maciejewski F B, Baccari F, Zimborás Z and Oszmaniec M 2021 *Quantum* **5** 464

[15] Nachman B, Urbanek M, Jong W A D and Bauer C W 2020 *npj Quantum Information* **6** 84

[16] Berg E V D, Minev Z K and Temme K 2022 *Phys. Rev. A* **105** 032620

[17] Livingston W P, Blok M S, Flurin E, Dressel J, Jordan A N and Siddiqi I 2022 *Nat. Commun.* **13** 2307

[18] He A, Nachman B, Jong W A D and Bauer C W 2020 *Phys. Rev. A* **102** 012426

[19] Pascuzzi V R, He A, Bauer C W, Jong W A D and Nachman B 2022 *Phys. Rev. A* **105** 042406

[20] Bravyi S, Sheldon S, Kandala A, Mckay D C and Gambetta J M 2021 *Phys. Rev. A* **103** 042605

[21] Borras K, Chang S Y, Funcke L, Grossi M, Hartung T, Jansen K, Kruecker D, Kühn S, Rehm F, Tüysüz C and Vallecorsa S 2023 *J. Phys.: Conf. Series* **2438** p. 012093

[22] Geller M R and Sun M 2021 *Quantum* **6** 025009

[23] Smith A W R, Khosla K E, Self C N and Kim MS 2021 *Science Advances* **7** eabi8009

[24] Wang C, Li X, Xu H, *et al*. 2022 *npj Quantum Information* **8** 3

[25] Daley A J, Bloch I, Kokail C, Flannigan S, Pearson N, Troyer M and Zoller P 2022 *Nature* **607** 667

[26] Gyongyosi L and Imre S 2019 *Computer Science Review* **31** 51-71

[27] Temme K, Bravyi S and Gambetta J M 2017 *Phys. Rev. Lett.* **119** 180509

[28] Ostaszewski M, Grant E and Benedetti M 2021 *Quantum* **5** 391

[29] Chu J, Li D, Yang X P, Song, S Q, Han Z K, Yang Z, Dong Y Q, Zheng W, Wang Z M, Yu X M, Lan D, Tan X S and Yu Y 2020 *Phys. Rev. Appl.* **13** 064012

[30] Houssein E H, Abohashima Z, Elhoseny M and Mohamed W M 2022 *Expert Systems with Applications* **2022** 116512

[31] Postler L, Heuβen S, Pogorelov I, Rispler M, Feldker T, Meth M, Marciniak C D, Stricker R, Ringbauer M, Blatt R, Philipp S, Markus M and Thomas M 2022 *Nature* **605** 675

[32] Funcke L, Hartung T, Jansen K, Kühn S, Stornati P and Wang X Y 2022 *Phys. Rev. A* **105** 062404

[33] Fisher M P A, Khemani V, Nahum A and Vijay S 2018 *Annual Review of Condensed Matter Physics* **14** 335

[34] Goodfellow I, Abadie P J, Mirza M, Xu B, Farley W D, Ozair S, Courville A and Bengio Y 2020 *Communications of the ACM* **63** 139-144

[35] Zeng J F, Wu Y F, Liu J G, Wang L and Hu J P 2019 *Phys. Rev. A* **99** 052306

[36] Paszke A, Gross S, Massa F, *et al* 2019 *Advances in neural information processing systems* **32**

[37] Imambi S, Prakash K B and Kanagachidambaresan G R 2021 *Programming with TensorFlow: Solution for Edge Computing Applications* **87** 104

[38] Alexander T, Kanazawa N, Egger D J, Capelluto L, Wood C J, Abhari A J and McKay D C 2020 *Quantum Science and Technology* **5** 044006

[39] Shaik H E, Rangaswamy N 2020 *2020 5th International conference on computing, communication and security (ICCCS)* pp. 1–6

[40] Dheeru D and Casey G 2017 *UCI machine learning repository*

[41] Yu Y M, Gao J, Mu X Y and Wang S M 2023 *Quantum Information Processing* **22** 180

[42] Wang Y N, Song Z Y, Ma Y L, Hua N and Ma H Y 2021 *Acta Phys. Sin.* **70** 230302 (in Chinese)

[43] Jiang Y Y, Chu P C, Zhang W B and Ma H Y 2022 *Chin. Phys. B* **31** 040307

[44] Chiribella G, D'Ariano G M and Perinotti P 2008 *Phys. Rev. Lett.* **101** 060401

[45] Liu J G and Wang L 2018 *Phys. Rev. A* **98** 062324

[46] Facchinei F and Kanzow C 2007 *4OR* **5** 173

[47] Johnstun S, Van H and Jean-François 2021 *American Journal of Physics* **89** 935

[48] Chang S Y, Agnew E, Combarro E, Grossi M, Herbert S and Vallecorsa S 2023 *J. Phys.: Conf. Series* **2438** 012062

[49] McArdle S, Yuan X and Benjamin S 2019 *Phys. Rev. Lett.* **122** 180501

[50] Cattaneo M, Rossi M A and García-Pérez G 2023 *PRX quantum* **4** 010324

[51] LaRose R, Mari A, Kaiser S, Karalekas P J, Alves A A, Czarnik P, Mandouh M E, Gordon M H, Hindy Y, Robertson A, Thakre P, Wahl M, Samuel D, Mistri R, Tremblay M, Gardner N, Stemen N T, Shammah N and Zeng W J 2022 *Quantum* **6** 774

[52] Bu Y, Zou S, Liang Y and Veeravalli V V 2018 *IEEE Transactions on Information Theory* **64** 2648

[53] Lu T A, Qiu Z H, Zhang Z B and Zhong J G 2020 *Optics and Lasers in Engineering* **134** 106301

[54] Duro D C, Franklin S E and Dub M G 2012 *Remote sensing of environment* **118** 259

[55] Cozzini M, Ionicioiu R and Zanardi P 2007 *Phys. Rev. B* **76** 104420

[56] Rahman A U, Haddadi S, Pourkarimi M R and Ghominejad M 2022 *Laser Physics Letters* **19** 035204

[57] Huang H Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H and McClean J R 2021 *Bulletin of the American Physical Society* **12** 2631

[58] An Z and Zhou DL 2019 *Europhys. Lett.* **126** 60002

[59] Fawzi O, Grospellier A and Leverrier A 2020 *Communications of the ACM* **64** 106

[60] Liang Y C, Yeh Y H, Mendonça P E, Teh R Y, Reid M D and Drummond P D 2019 *Reports on Progress in Physics* **82** 076001

[61] Liu Y C, Arunachalam S and Temme K 2021 *Nat. Phys.* **17** 1013

[62] Ajagekar A and You F Q 2021 *Applied Energy* **303** 117628