

Quantum Generative Diffusion Model: A Fully Quantum-Mechanical Model for Generating Quantum State Ensemble

Chuangtao Chen, Qinglin Zhao, *Senior Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, Zhimin He, Zhili Sun, *Senior Member, IEEE*, and Haozhen Situ

Abstract—Classical diffusion models have shown superior generative results. Exploring them in the quantum domain can advance the field of quantum generative learning. This work introduces Quantum Generative Diffusion Model (QGDM) as their simple and elegant quantum counterpart. Through a non-unitary forward process, any target quantum state can be transformed into a completely mixed state that has the highest entropy and maximum uncertainty about the system. A trainable backward process is used to recover the former from the latter. The design requirements for its backward process includes non-unitarity and small parameter count. We introduce partial trace operations to enforce non-unitarity and reduce the number of trainable parameters by using a parameter-sharing strategy and incorporating temporal information as an input in the backward process. We present QGDM's resource-efficient version to reduce auxiliary qubits while preserving generative capabilities. QGDM exhibits faster convergence than Quantum Generative Adversarial Network (QGAN) because its adopted convex-based optimization can result in better convergence. The results of comparing it with QGAN demonstrate its effectiveness in generating both pure and mixed quantum states. It can achieve 53.02% higher fidelity in mixed-state generation than QGAN. The results highlight its great potential to tackle challenging quantum generation tasks.

Index Terms—Variational quantum algorithms, Quantum machine learning, Quantum generative models, Denoising diffusion probabilistic models.

1 INTRODUCTION

GENERATIVE models have attracted widespread attention and achieved significant success in industrial fields [1]. Notably, in the past two years, large models such as ChatGPT and Stable Diffusion showed remarkable performance and creativity in human-machine dialogue and image generation [2], [3]. They have made technological breakthroughs and significantly impacted our daily life. Their essence is to learn the underlying probability distributions from a given dataset. Once trained, these models can generate new data that follow the distribution, demonstrating an understanding of the underlying patterns in the data.

Quantum generative models [4], as the quantum counterparts of classical generative models in quantum machine learning [5], [6], [7], have sparked great interest in the research community. They merge the advantages of quantum computing and classical generative models, offering new possibilities for data generation. There have been several

explorations of such models, including Quantum Generative Adversarial Network (QGAN) [8], [9], Quantum Circuit Born Machine (QCBM) [10], [11], Quantum Variational Autoencoder (QVAE) [12], and Quantum Boltzmann Machine (QBM) [13]. Among them, as the quantum counterpart of the classical Generative Adversarial Network (GAN) [14], [15], [16], [17], [18], QGAN has received considerable attention for its remarkable applications, *e.g.*, generating quantum states [9], [19], [20], [21], image generation [22], [23], [24], [25], and fitting classical distributions [26], [27], [28], [29], [30]. It provides a pathway for researchers to more effectively simulate and explore complex quantum systems, especially in generating and processing data related to quantum states.

1.1 Motivations

Despite their various applications, QGANs inherit challenges in convergence from GANs. QGANs may experience model collapse, leading to oscillations in the loss of the generator and discriminator without converging to a global optimum [21]. Furthermore, they face a unique challenge in the quantum domain as they cannot achieve convergence in generating mixed states [20]. Specifically, according to Helstrom's measurement theory, the analytic optimal output of the generator under a minimax optimization strategy is a pure state, which demonstrates its inability to produce a target mixed state. The aforementioned challenges make their training difficult and limit their broader application in critical quantum state generation tasks. These limitations highlight the need for developing more effective alternatives to perform the complex tasks of quantum state generation.

Corresponding author: Qinglin Zhao.

- Chuangtao Chen and Qinglin Zhao are with the Faculty of Innovation Engineering, Macau University of Science and Technology, Macao 999078, China. (e-mail: chuangtaochen@gmail.com; qlzhao@must.edu.mo)
- MengChu Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA. (e-mail: zhou@njit.edu)
- Zhimin He is with the School of Electronic and Information Engineering, Foshan University, Foshan 528000, China. (e-mail: zhimihe@gmail.com)
- Zhili Sun is with the Institute of Communication Systems, University of Surrey, GU2 7XH Guildford, U.K. (e-mail: Z.Sun@surrey.ac.uk)
- Haozhen Situ is with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China. (e-mail: situhaozhen@gmail.com)

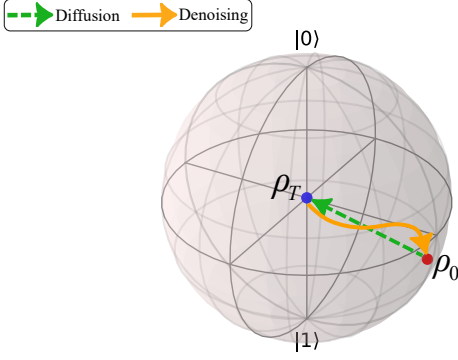


Fig. 1: The core idea of the proposed QGDM.

To do so, it is important to explore the quantum analogs of classical generative diffusion models [31], [32], [33], [34]. These models have been proven to have superior generative performance in many fields over GANs with better convergence [35], [36]. Their quantum analogs may offer similar benefits. For example, if we can set up a convex objective function that minimizes the distance between the target and generated quantum states, then using gradient descent to optimize the convex function can result in better convergence properties [37]. Therefore, the development of such models is expected to address the challenges faced by QGANs and make a breakthrough contribution to the field of quantum generative learning. These considerations have motivated us to undertake this work.

1.2 Contributions

Inspired by classical diffusion models, this paper introduces a novel quantum generative model known as QGDM for quantum state generation tasks. Fig.1 shows its core idea in a Bloch sphere: any target quantum state (ρ_0) can be transformed into a completely mixed state (ρ_T) through a forward (diffusion) process; subsequently, a trainable backward (denoising) process can be used to recover the former from the latter. Our objectives are to a) define a timestep-dependent forward process, b) design an effective quantum state backward process, and c) benchmark our model against other models in pure and mixed state generation tasks. We aim to make the following novel and significant contributions to the field of quantum generative learning:

- 1) We propose an elegant quantum version of diffusion model called QGDM. Any target quantum state can be diffused into a completely mixed state by using a forward process. Then, using a learnable backward process, the former can be recovered from the latter. The forward process employs a non-unitary depolarization channel model. To reverse the diffusion process, we introduce a partial trace operation during denoising, which traces out information from the subsystem and makes the backward process non-unitary. Additionally, to reduce the number of parameters, we use a parameter-sharing strategy and include temporal information as an input, allowing the backward processes at all timesteps to share the same parameters.

TABLE 1: Notations.

Notation	Description
ρ_0	Target Quantum State
ρ_t	System's Density Matrix at Timestep t
τ_t	Timestep Embedding Quantum State at Timestep t
α_t	Noise Parameter Dependent at Timestep t
$\bar{\alpha}_t$	Accumulated Noise Parameter $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$
$\mathcal{E}(\cdot, \cdot)$	Forward Process Function
$f_{\Theta}(\cdot, \cdot)$	Backward Process Function Parameterized by Θ
$\mathcal{T}(\omega, \cdot)$	Timestep Embedding Circuit Parameterized by ω
\mathcal{U}	Uniform Distribution
$U(\theta)$	Denoising Circuit Parameterized by θ
$\text{tr}(\cdot)$	Trace Operation
$\text{tr}_B(\cdot)$	Partial Trace Operation over Subsystem B
N	Number of Qubits in the Target State
N_τ	Number of Qubits in the Timestep Embedding Circuit
d	Dimension of Hilbert Space
λ	Hyperparameter in Loss Function
A^\dagger	Hermitian Conjugate of the Operator A
\mathcal{L}_t	Loss Function at Timestep t

- 2) We introduce a Resource-efficient version of QGDM (RQGDM) to minimize the number of auxiliary qubits required in the backward process. Its core idea is to utilize a parameterized quantum circuit to extract a low-dimensional representation of quantum data from its original high-dimensional Hilbert space. This compression is crucial for the backward process since the low-dimensional representation uses much fewer qubits than the input state. Consequently, RQGDM enhances the model's ability to tackle large-scale problems efficiently.
- 3) We conduct numerical simulation whose results demonstrate that the proposed models well outperform QGANs in generating both mixed states and pure states. Notably, in generating mixed states, QGDM achieves 22.17% higher fidelity than QGANs for tasks involving 1 to 4 qubits, while RQGDM achieves 64.66% higher one than QGANs for tasks involving 1 to 8 qubits.

The remainder of this article is structured as follows. Section 2 briefs the existing literature and the foundational work relevant to quantum generative models. Section 3 presents an in-depth exposition of QGDM. Section 4 introduces RQGDM. Section 5 analyzes the rationale behind the design of the backward process and presents simulation results. Section 6 gives the numerical simulation setup and comparative results. Finally, Section 7 concludes this article.

2 PRELIMINARY AND RELATED WORK

2.1 Preliminary

We first introduce the necessary quantum information background. For more details, please refer to Ref. [38]. The notations are summarized in Table 1.

2.1.1 Pure Quantum State

In quantum computing, a state of an N -qubit can be represented by a vector in a complex Hilbert space \mathbb{C}^d with unit length, denoted as $|\psi\rangle \in \mathbb{C}^d$, where the *ket* notation $|\rangle$ denotes a column vector and $d = 2^N$ is the dimension of Hilbert space. The *bra* notation $\langle\psi| = |\psi\rangle^\dagger$ represents a

row vector, with \dagger denoting the conjugate transpose. The evolution of a pure state can be represented by: $|\psi'\rangle = U|\psi\rangle$, where U is a unitary operator satisfying $UU^\dagger = \mathbb{I}$, and \mathbb{I} is the identity matrix.

2.1.2 Mixed Quantum State

A mixed state of a quantum system is a probabilistic mixture of pure states. It can be represented by a density matrix (or density operator): $\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|$, where the ensemble of pure states $\{p_i, |\psi_i\rangle\}$ indicates that the system is in state $|\psi_i\rangle$ with a probability of p_i . This matrix is particularly useful in contexts where the exact state of a system is not known or when considering the subsystems of entangled pairs. The evolution of a mixed state is described by $\rho' = U\rho U^\dagger$.

2.1.3 Tensor Product

The Hilbert space of a combined system, composed of two (or more) different quantum systems, is the tensor product of the Hilbert spaces of the subsystems. Specifically, if the states of the subsystems are $|\psi_A\rangle$ and $|\psi_B\rangle$ for pure states, or ρ_A and ρ_B for mixed states, the state of the composite system can be represented as $|\psi_A\rangle \otimes |\psi_B\rangle$ or $\rho_A \otimes \rho_B$, respectively. The tensor product operation allows us to describe the complete state of the composite system through the individual states of its components.

2.1.4 Partial Trace

Suppose that we have two quantum systems, A and B . Their composite system state is represented by the density matrix ρ_{AB} . The partial trace operation yields a reduced density matrix for subsystem A , calculated by $\rho_A = \text{tr}_B(\rho_{AB})$. Its definition is $\text{tr}_B(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|) = |a_1\rangle\langle a_2| \text{tr}(|b_1\rangle\langle b_2|)$, where $|a_1\rangle$ and $|a_2\rangle$ are any two vectors in A , while $|b_1\rangle$ and $|b_2\rangle$ are any two vectors in B .

2.2 Quantum Generative Models

Quantum generative models, a pivotal sector in quantum machine learning, encompass various models including QCBM [10], QBM [13], QVAE [12], and QGAN [8], [9].

QCBM uses quantum circuits to generate a classical bit string from specific probability distributions [10]. This model has substantially advanced [11], [39], [40], [41], finding its applications in various fields, *e.g.*, Monte Carlo simulations [42], financial data generation [43], and joint distribution learning [43], [44]. QBM functioning as a quantum counterpart of the classical Boltzmann Machine [45], [46], uses a quantum device to prepare Boltzmann distributions, by replacing the units in the Boltzmann Machine with qubits and substituting the energy function with a parameterized Hamiltonian. QBM was first demonstrated in [13], has its enhancements [47], [48]. Khoshaman et al. [12] introduced QVAE, using QBM as a prior in Discrete VAE [49], with developments including annealer-based [50] and gate-based QVAE [51]. In the realm of QGANs, Lloyd et al. [8] developed a comprehensive taxonomy, classifying them by the quantum or classical nature of their components. This was empirically supported by Dallaire-Demers and Killoran [9], who demonstrated the efficacy of QGAN in generating the desired quantum data. Since then, the field has continuously explored applications in quantum state generation [20], [21],

[52], image generation [22], [23], [24], and discrete data generation [27], [28], [30]. For an in-depth review of these quantum generative models, please refer to [4].

During our research, several studies [53], [54], [55] have proposed quantum analogs to the classical diffusion model. Parigi et al. [53] introduce several strategies and tools that could be applied to develop quantum versions of diffusion models. However, they fail to offer specific implementation examples and experimental validation. In contrast, our work introduces a detailed concept and delves into technical details, including comparisons with other quantum generative models. Zhang et al. [54] propose quantum denoising diffusion probabilistic model (QuDDPM) to generate individual pure states from a target distribution. Instead, our work aims to generate the average state of the distribution, which is also an objective considered in other research on quantum generative models [9], [20], [21]. Additionally, the backward process in QuDDPM uses a different set of parameters at each timestep. In contrast, we use a parameter-sharing strategy, ensuring that the same set of parameters is used for the backward process at every timestep. This approach keeps the total number of parameters constant, regardless of the number of timesteps, thereby avoiding an increase in parameters as the number of timesteps grows. Cacioppo et al. [55] use a classical approach for their diffusion process, while the denoising process is performed via a trainable quantum circuit for the denoising task. However, our method is fully quantum-mechanical, encompassing both diffusion and denoising processes.

3 QUANTUM GENERATIVE DIFFUSION MODEL (QGDM)

3.1 General Framework

Fig. 2 shows a framework of QGDM. It includes a forward process (or diffusion one) $\mathcal{E}(\rho_{t-1}, t)$ and a backward process (or denoising one) $f_{\Theta}(\rho_t, t)$. $\mathcal{E}(\rho_{t-1}, t)$ takes an N -qubit quantum state ρ_{t-1} and a timestep scalar t as input, producing a quantum state ρ_t with added noise. Similar to classical diffusion models, in QGDM, a “timestep” refers to a discrete step in the model’s processes where noise is sequentially added to (forward process) or subtracted from a quantum state (backward process). In this paper, $\mathcal{E}(\rho_{t-1}, t)$ is modeled as a depolarizing channel. When $t = 0$, ρ_0 represents the target quantum state. In contrast, at $t = T$, ρ_T is a completely mixed state. The denoising process employs a model $f_{\Theta}(\rho_t, t)$ with trainable parameters Θ , which denoises the noisy input quantum state ρ_t , producing a cleaner quantum state ρ_{t-1} . After applying the function $f_{\Theta}(\cdot, \cdot)$ for T times to ρ_T , the target quantum state ρ_0 is recovered.

Remark 1: Let us discuss the difference between the notations used in this paper and those used in classical diffusion model papers [32], [35], [36]. In quantum information, the quantum operation formalism $\rho' = \mathcal{E}(\rho)$ is a powerful tool for describing the evolution of quantum states [38], where \mathcal{E} represents a quantum operation, and ρ and ρ' are the initial and final states, respectively. This formalism is especially useful for describing open quantum systems or the stochastic evolution of quantum states. In this paper, we utilize the quantum operation formalism, denoting forward and backward processes as $\mathcal{E}(\rho_{t-1}, t)$

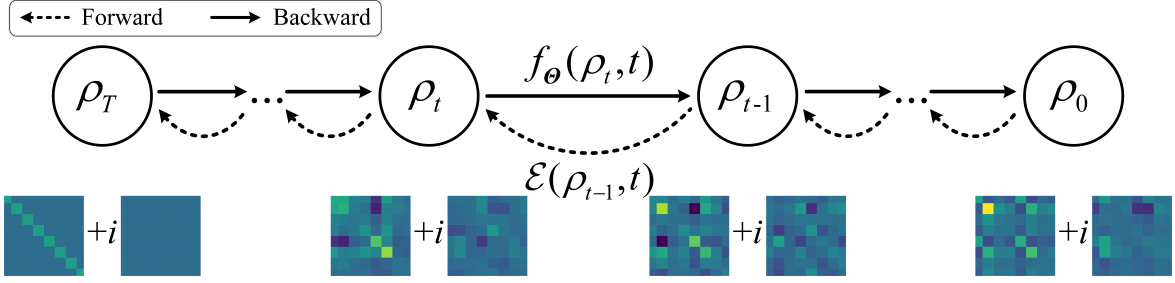


Fig. 2: The framework of the proposed Quantum Generative Diffusion Model (QGDM).

and $f_{\Theta}(\rho_t, t)$, respectively. These notations do not explicitly include the output term. In contrast, in classical diffusion models [31], [32], forward and reverse distributions are defined as $q(x_t|x_{t-1})$ and $p_{\Theta}(x_{t-1}|x_t)$. These notations explicitly include the output term (*i.e.*, x_t in $q(x_t|x_{t-1})$ and x_{t-1} in $p_{\Theta}(x_{t-1}|x_t)$). The difference between the notations in our paper and those in classical diffusion model papers can be summarized as follows. In the classical case, the evolution process of a sample (x_{t-1} or x_t) is probabilistic. Therefore, the forward or backward process uses conditional probability notation. In the quantum case, the system's state is described by a density matrix. The evolution of the density matrix is deterministic, but the density matrix itself represents a probabilistic quantum system. Thus, we use the quantum operation formalism to describe the forward and backward processes. This distinction between quantum and classical systems aids readers who may not be familiar with quantum computation in understanding our notations.

3.2 Forward Process

In the natural world, diffusion is governed by non-equilibrium thermodynamics, driving systems toward equilibrium [56]. As a system undergoes evolution, its entropy increases until it reaches the maximum at equilibrium, indicating that there is no further evolution. The classical diffusion model draws inspiration from this diffusion mechanism [31]. In quantum information theory, the depolarizing channel is considered as a “worst-case scenario” [57]. Under its influence, a quantum state gradually loses its original information, eventually transforming into a completely mixed state. This state is characterized by the maximum entropy, representing the pinnacle of randomness in quantum systems. At this point, our uncertainty about the system has also reached its maximum.

An N -qubit completely mixed state is mathematically represented as $\mathbb{I}/2^N = \sum_i \frac{1}{2^N} |i\rangle \langle i|$. This denotes that each computational basis state $|i\rangle$ occurs with an equal probability of $\frac{1}{2^N}$. This uniform probability distribution across all computational bases shows the state to be maximally random, similar to the uniform distribution in classical data. At timestep t , the diffusion process of QGDM adds depolarizing noise to an N -qubit quantum state ρ_{t-1} :

$$\begin{aligned} \rho_t &= \mathcal{E}(\rho_{t-1}, t) \\ &= (1 - \alpha_t)\mathbb{I}/d + \alpha_t\rho_{t-1}, \end{aligned} \quad (1)$$

where $d = 2^N$ denotes the dimension of Hilbert space and $\alpha_t \in [0, 1]$ is a timestep-dependent scalar. Eq. (1) indicates

that the input quantum state ρ_{t-1} remains unchanged with a probability of α_t , while with a probability of $1 - \alpha_t$, it is replaced by the completely mixed state \mathbb{I}/d . This indicates the loss of original information in the quantum system. Therefore, the depolarizing channel is non-unitary because information is lost when quantum states pass through it, making the process irreversible. By setting an appropriate sequence of $\{\alpha_t\}$, a series of depolarizing channel sequences acting on the quantum state ρ_0 can eventually yield the output $\rho_T \approx \mathbb{I}/d$.

3.2.1 Efficient Quantum State Transition to Arbitrary Timestep

Eq. (1) mathematically describes a single-step diffusion process. It is inefficient to repeatedly apply \mathcal{E} to obtain a specific timestep's quantum state ρ_t . To enhance the transition efficiency from the initial state ρ_0 to a specific state ρ_t , we introduce Theorem 1.

Theorem 1: For any $t \in \{1, \dots, T\}$, the relationship between ρ_0 and ρ_t can be described by:

$$\rho_t = (1 - \bar{\alpha}_t)\mathbb{I}/d + \bar{\alpha}_t\rho_0, \quad (2)$$

where $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. Thus, $\bar{\alpha}_t \leq \bar{\alpha}_{t-1}$ since $\alpha_i \in [0, 1], \forall i$.

Proof:

$$\begin{aligned} \rho_t &= (1 - \alpha_t)\mathbb{I}/d + \alpha_t\rho_{t-1} \\ &= (1 - \alpha_t)\mathbb{I}/d + \alpha_t[(1 - \alpha_{t-1})\mathbb{I}/d + \alpha_{t-1}\rho_{t-2}] \\ &= (1 - \alpha_t\alpha_{t-1})\mathbb{I}/d + \alpha_t\alpha_{t-1}\rho_{t-2} \\ &= (1 - \alpha_t\alpha_{t-1})\mathbb{I}/d \\ &\quad + \alpha_t\alpha_{t-1}[(1 - \alpha_{t-2})\mathbb{I}/d + \alpha_{t-2}\rho_{t-3}] \\ &= (1 - \alpha_t\alpha_{t-1}\alpha_{t-2})\mathbb{I}/d + \alpha_t\alpha_{t-1}\alpha_{t-2}\rho_{t-3} \\ &= \dots \\ &= \left(1 - \prod_{i=1}^t \alpha_i\right)\mathbb{I}/d + \prod_{i=1}^t \alpha_i\rho_0 \\ &= (1 - \bar{\alpha}_t)\mathbb{I}/d + \bar{\alpha}_t\rho_0. \quad \square \end{aligned} \quad (3)$$

Thus, Eq. (2) enables the direct transition from ρ_0 to a quantum state ρ_t in any chosen timestep.

3.2.2 Noise Schedule

The choice of α_t must ensure that $\bar{\alpha}_T \rightarrow 0$ and $\alpha_t \in [0, 1]$ for all t . We use the cosine noise schedule from the classical diffusion model [34] to select α_t :

$$\alpha_t = \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, \quad (4)$$

where

$$\bar{\alpha}_t = \frac{g(t)}{g(0)}, \quad g(t) = \cos\left(\frac{t/T + s}{1 + s} \cdot \frac{\pi}{2}\right)^2, \quad (5)$$

with s being a small offset hyperparameter. We note that there are other noise schedules, such as the linear noise schedule [32] and learned one [58]. Exploring the impacts of other noise schedules on QGDM's performance, or designing specific noise schedules for QGDM, is an interesting research direction, which we leave for future work.

3.3 Backward Process

The objective of a backward process in QGDM is to restore ρ_0 from ρ_T . We should first address two considerations.

Firstly, since the denoising process is essentially the inverse of a non-unitary diffusion process, it must also be non-unitary to effectively denoise. Secondly, in principle, we need to train T distinct denoising processes to deal with T different denoising demands if we model the denoising process in a form of $f_{\Theta_t}(\rho_t)$. To reduce the number of trainable parameters in QGDM, we make these processes share the same parameters by using the timestep t as an input. This integration is crucial for shaping $f_{\Theta}(\rho_t, t)$, as it allows the model to effectively utilize temporal information to accurately estimate ρ_{t-1} while maintaining a small number of parameters.

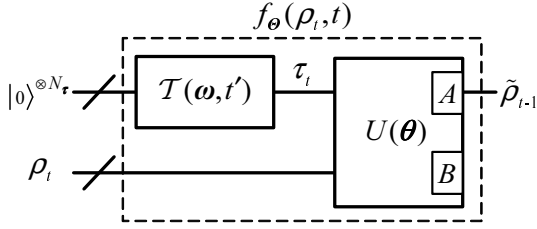


Fig. 3: The denoising process framework, $f_{\Theta}(\rho_t, t)$, is composed of a timestep embedding circuit $\mathcal{T}(\omega, t')$ and a denoising circuit $U(\theta)$, where $\Theta = \{\omega, \theta\}$.

To endow the denoising process with the aforementioned two characteristics, we achieve non-unitary properties by tracing out a part of the subsystem. Additionally, we use a parameterized circuit to embed temporal information into the quantum system. Our design for the denoising process $f_{\Theta}(\rho_t, t)$ is illustrated in Fig. 3. $f_{\Theta}(\rho_t, t)$ comprises two modules: a timestep embedding circuit $\mathcal{T}(\omega, t')$ parameterized by ω and $t' = t\pi/T$, and a denoising circuit $U(\theta)$ parameterized by θ , thus trainable parameters $\Theta = \{\omega, \theta\}$. A timestep embedding circuit $\mathcal{T}(\omega, t')$, operating on an auxiliary register with N_{τ} qubits, is utilized to produce state τ_t , which carries temporal information. Subsequently, the composite system $\tau_t \otimes \rho_t$ is fed into the denoising circuit $U(\theta)$. Afterward, all qubits are regrouped into two registers: A with N qubits and B with N_{τ} qubits. We trace out B to get the predicted denoised quantum state $\tilde{\rho}_{t-1}$.

3.3.1 Timestep Embedding Circuit

A critical consideration when designing the denoising process is how to encode the temporal information into a

quantum state. This is a classical-to-quantum encoding challenge, where such methods as qubit encoding [59], [60] or amplitude encoding [61] could be used to embed t into a quantum system. However, the qubit encoding method only maps the timestep sequence uniformly onto a ring on the Bloch sphere, which lacks expressiveness and flexibility. The amplitude encoding method requires a number of gates that is exponential in the number of qubits [62]. To enhance the expressiveness and flexibility of timestep embedding while maintaining an appropriate gate complexity, this work utilizes the quantum embedding method [63] to obtain a timestep embedding state τ_t with N_{τ} qubits, *i.e.*,

$$\tau_t = \mathcal{T}(\omega, t') (|0\rangle \langle 0|)^{\otimes N_{\tau}} \mathcal{T}^{\dagger}(\omega, t'), \quad (6)$$

where scalar $t' = t\pi/T$ is the mapping of timestep t into the range $[0, \pi]$, and ω denotes the trainable parameters in the timestep embedding circuit.

The architecture of the timestep embedding circuit is shown in the supplementary file.

3.3.2 Denoising Circuit

The denoising circuit $U(\theta)$ acts on the composite system $\tau_t \otimes \rho_t$. Subsequently, all qubits are regrouped into subsystems A and B , where they possess N and N_{τ} qubits, respectively. By tracing out B , we obtain the predicted output:

$$\begin{aligned} \tilde{\rho}_{t-1} &= f_{\Theta}(\rho_t, t) \\ &= \text{tr}_B \left(U(\theta) (\tau_t \otimes \rho_t) U^{\dagger}(\theta) \right). \end{aligned} \quad (7)$$

Using subsystem B to output $\tilde{\rho}_{t-1}$ seems intuitive. However, such a design would result in the model failure. This failure is caused by the slight difference between ρ_t and ground truth ρ_{t-1} , which poses a training challenge. Rather than producing the intended output, the optimizer often merely copies its input. The next Section provides the detailed examination of this issue, including numerical simulations to demonstrate the concept.

For details on the architecture of the denoising circuit, please refer to the supplementary file.

3.4 Training and Generation

The training objective of QGDM is to maximize the quantum fidelity between the predicted denoised state $\tilde{\rho}_{t-1}$ and the ground truth ρ_{t-1} . The loss function is defined as follows:

$$\min_{\Theta} \mathcal{L} = \mathcal{L}_0 + \lambda \mathbb{E}_{t \in \mathcal{U}(2, \dots, T)} [\mathcal{L}_{t-1}], \quad (8)$$

where λ is a hyperparameter used to balance the losses \mathcal{L}_0 and $\mathbb{E}_{t \in \mathcal{U}(2, \dots, T)} [\mathcal{L}_{t-1}]$, and we find that setting a small λ improves the generative effect of QGDM. $\mathcal{U}(2, \dots, T)$ represents a uniform distribution that gives values from 2 to T . We have:

$$\mathcal{L}_{t-1} = 1 - F(\rho_{t-1}, \tilde{\rho}_{t-1}), \quad (9)$$

where $F(\cdot, \cdot)$ is the quantum fidelity function [38], [57], used to measure the distance between two quantum states ρ and σ :

$$F(\rho, \sigma) = \left(\text{tr} \sqrt{\sqrt{\rho} \sigma \sqrt{\rho}} \right)^2. \quad (10)$$

QGDM training is realized via in Algorithm 1. In each iteration, we compute \mathcal{L}_0 and the expectation $\mathbb{E}_{t \in \mathcal{U}(2, \dots, T)}[\mathcal{L}_{t-1}]$ to update the trainable parameters until the training loss converges or the maximum number of iterations is reached. The expected value is estimated by using the Monte Carlo approach.

Once the parameters are trained, we can iteratively obtain the target state ρ_0 from the completely mixed state ρ_T by using ω and θ . ρ_0 is then generated via Algorithm 2. The algorithm starts with a completely mixed state and uses the trained denoising process to generate the target state via T steps. Therefore, its complexity is $\mathcal{O}(T)$.

Algorithm 1 QGDM training.

- 1: **Input:** Noise schedule parameters $\{\bar{\alpha}_t\}_{t=1}^T$, batch size BS, weight parameter λ , number of qubits of the target state N , number of qubits of the timestep embedding state N_τ , total timestep of the diffusion process T , maximum iteration number, target quantum state ρ_0 , learning rate η , completely mixed state \mathbb{I}/d .
 - 2: $\theta \sim \mathcal{U}(0, \pi)$, $\omega \sim \mathcal{U}(0, \pi)$
 - 3: **repeat**
 - 4: $\rho_1 \leftarrow (1 - \bar{\alpha}_1) \mathbb{I}/d + \bar{\alpha}_1 \rho_0$
 - 5: $\tau_1 \leftarrow \mathcal{T}(\omega, \pi/T) (|0\rangle\langle 0|)^{\otimes N_\tau} \mathcal{T}^\dagger(\omega, \pi/T)$
 - 6: $\tilde{\rho}_0 \leftarrow \text{tr}_B(U(\theta) (\tau_1 \otimes \rho_1) U^\dagger(\theta))$
 - 7: $\mathcal{L}_0 \leftarrow 1 - F(\rho_0, \tilde{\rho}_0)$
 - 8: $\mathcal{L} \leftarrow 0$
 - 9: **for** i from 1 to BS **do**
 - 10: Sample a unique timestep t from the uniform distribution $\mathcal{U}(2, \dots, T)$
 - 11: $\rho_t \leftarrow (1 - \bar{\alpha}_t) \mathbb{I}/d + \bar{\alpha}_t \rho_0$
 - 12: $\tau_t \leftarrow \mathcal{T}(\omega, t\pi/T) (|0\rangle\langle 0|)^{\otimes N_\tau} \mathcal{T}^\dagger(\omega, t\pi/T)$
 - 13: $\tilde{\rho}_{t-1} \leftarrow \text{tr}_B(U(\theta) (\tau_t \otimes \rho_t) U^\dagger(\theta))$
 - 14: $\rho_{t-1} \leftarrow (1 - \bar{\alpha}_{t-1}) \mathbb{I}/d + \bar{\alpha}_{t-1} \rho_0$
 - 15: $\mathcal{L} \leftarrow \mathcal{L} + 1 - F(\rho_{t-1}, \tilde{\rho}_{t-1})$
 - 16: **end for**
 - 17: $\mathcal{L} \leftarrow \mathcal{L}_0 + \lambda \frac{1}{\text{BS}} \mathcal{L}$
 - 18: $\omega \leftarrow \omega - \eta \nabla_\omega \mathcal{L}$; $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
 - 19: **until** \mathcal{L} converges or the number of iterations reaches the maximum
 - 20: **Output:** Optimal parameters ω^*, θ^*
-

Algorithm 2 QGDM generation.

- 1: **Input:** Number of qubits of the timestep embedding state N_τ , total timestep of the diffusion process T , the optimal parameters ω^*, θ^* , completely mixed state $\rho_T = \mathbb{I}/d$.
 - 2: Initialize $\tilde{\rho}$ as completely mixed state: $\tilde{\rho} \leftarrow \rho_T$
 - 3: **for** t from T to 1 **do**
 - 4: $t' \leftarrow t\pi/T$
 - 5: $\tau_t \leftarrow \mathcal{T}(\omega^*, t') (|0\rangle\langle 0|)^{\otimes N_\tau} \mathcal{T}^\dagger(\omega^*, t')$
 - 6: Update the generation state: $\tilde{\rho} \leftarrow \text{tr}_B(U(\theta^*) (\tau_t \otimes \tilde{\rho}) U^\dagger(\theta^*))$ using the previous value of $\tilde{\rho}$.
 - 7: **end for**
 - 8: **Output:** The final generation state $\tilde{\rho}_0 = \tilde{\rho}$
-

4 RESOURCE-EFFICIENT QUANTUM GENERATIVE DIFFUSION MODEL (RQGDM)

The number of qubits required for the QGDM denoising process is $N_\tau + N$. To reduce the number of auxiliary qubits, we propose a new design for the denoising process $f_\Theta(\rho_t, t)$. We call this prototype as RQGDM.

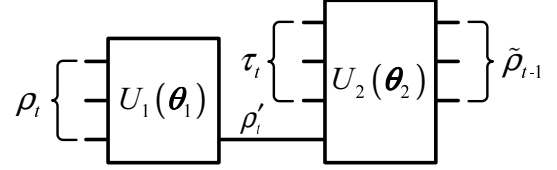


Fig. 4: The backward process design in Resource-efficient Quantum Generative Diffusion Models (RQGDM).

Inspired by quantum data compression methods [64], [65], the core idea of RQGDM is to first use a parameterized quantum circuit to learn a low-dimensional representation of quantum data from its original high-dimensional Hilbert space for the denoising process. This low-dimensional representation uses fewer qubits than the input state (e.g., one qubit in our approach), thereby reducing the total number of qubits required for the denoising process. In RQGDM, only $N + 1$ qubits are needed to generate an N -qubit quantum state, which enhances its efficiency in terms of resource utilization. Fig. 4 illustrates the denoising process circuit for a 3-qubit target state using RQGDM. The polluted state ρ_t is compressed by a parameterized circuit $U_1(\theta_1)$, with θ_1 as its trainable parameters, condensing its information into the final qubit. This qubit, containing concentrated information, is then combined with τ_t to form a composite system. Another circuit, $U_2(\theta_2)$, processes this expanded state across $N + 1$ qubits. After applying $U_2(\theta_2)$, the system's last qubit is discarded, leaving the reduced density matrix of the remaining N qubits as the predicted output state $\tilde{\rho}_{t-1}$:

$$\tilde{\rho}_{t-1} = \text{tr}_4 \left[U_2(\theta_2) \left(\tau_t \otimes \rho'_t \right) U_2^\dagger(\theta_2) \right], \quad (11)$$

where τ_t is given by (6) and ρ'_t is calculated by tracing out the first two qubits of the output from the U_1 circuit:

$$\rho'_t = \text{tr}_{12} \left(U_1(\theta_1) \rho_t U_1^\dagger(\theta_1) \right). \quad (12)$$

Therefore, the trainable parameters for the denoising process $f_\Theta(\rho_t, t)$ of RQGDM are $\Theta = \{\omega, \theta_1, \theta_2\}$. Both QGDM and RQGDM share the same diffusion process, training, and generation pipeline, but they differ in the denoising process.

5 CRITICAL EVALUATION OF DENOISING IN QGDM

In this section, we provide a detailed analysis of QGDM's denoising process and present numerical simulation results to support our analysis results.

A straightforward design for the denoising process $f_\Theta(\rho_t, t)$ involves using a trainable circuit $U(\theta)$ that acts on the extended system $\tau_t \otimes \rho_t$ and then traces out τ_t -subsystem (i.e., all the qubits of τ_t), which can be described as:

$$\text{tr}_{\tau_t\text{-subsystem}} \left(U(\theta) (\tau_t \otimes \rho_t) U^\dagger(\theta) \right) = \tilde{\rho}_{t-1}. \quad (13)$$

However, the noise added to the quantum state in a diffusion step is negligible, resulting in only a trivial difference between ρ_t and ρ_{t-1} . In the case of tracing out the register of τ_t and aiming for $\tilde{\rho}_{t-1} \approx \rho_t$, it can mislead the optimizer, which tends to adjust the circuit $U(\theta)$ to produce an output that closely approximates its input, thus minimizing the training loss. However, this adjustment deviates from the intended role of the denoising circuit.

The numerical simulation results illustrating the limitations of this design approach are shown in Fig. 5. Fig. 5(a) displays the decreasing trend of the training loss over epochs. Fig. 5(b) displays the Hilbert-Schmidt (HS) distance [38], [57] between the input and output states of the denoising circuit $U(\theta)$ during training. As a measure used in quantum information to quantify the distance between two quantum states, HS distance is defined as $D_{\text{HS}}(\rho, \sigma) = \sqrt{\text{tr}[(\rho - \sigma)^2]}$. Fig. 5(c) displays the trend of generation fidelity of QGDM over timestep t . The orange solid line represents the fidelity of the generated states with the target state, while the blue dashed line represents the theoretical fidelity curve that the model should achieve. This result shows the model fails to change the input state during the generation process.

Figs. 5(a)-(b) indicate that with increasing epochs, both training loss and HS distance rapidly decrease. A reduction in HS distance shows that the difference between the input state $\tau_t \otimes \rho_t$ and the output state $U(\theta)(\tau_t \otimes \rho_t)U^\dagger(\theta)$ becomes minimal, ultimately converging to zero. At this point, tracing over the subsystem containing τ_t would yield a result close to ρ_t . Fig. 5(c) shows that QGDM cannot alter the input state during the generation process. The denoising process merely outputs the quantum state fed into it. Hence the fidelity between the generated quantum state and target remains unchanged.

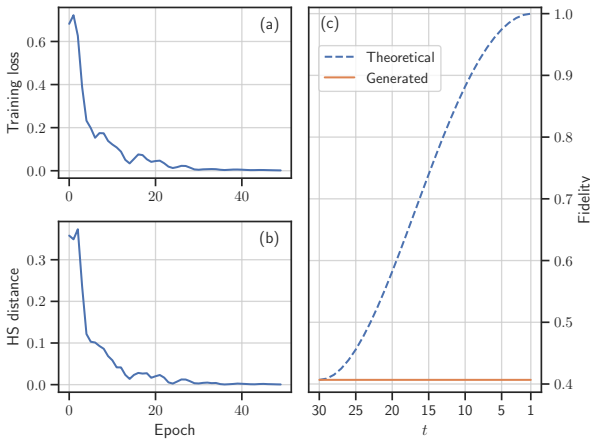


Fig. 5: (a) Training loss with respect to the iterations of training. (b) Hilbert-Schmidt (HS) distance between the input and output state of denoising circuit $U(\theta)$. (c) After training, the generation fidelity of QGDM with respect to the timestep.

The comparison between the final generated quantum state $\tilde{\rho}_0$ (shown in the first row) and ρ_0 (displayed in the second row) is illustrated in Fig. 6. Given the complex nature of the density matrix, the figure separately draws the real and imaginary parts, with the former on the left and the

latter on the right. This visualization clearly shows that $\tilde{\rho}_0$ remains equal to the completely mixed state ρ_T , indicating that $U(\theta)$ essentially reproduces the input state. This highlights the critical need for a well-designed denoising process in QGDM that effectively integrates information between the noisy input state ρ_t and the timestep embedding state τ_t , especially considering the minor differences in quantum states across adjacent timesteps. The results and discussions in our study intend to provide valuable insights for the future development of similar quantum machine learning models.

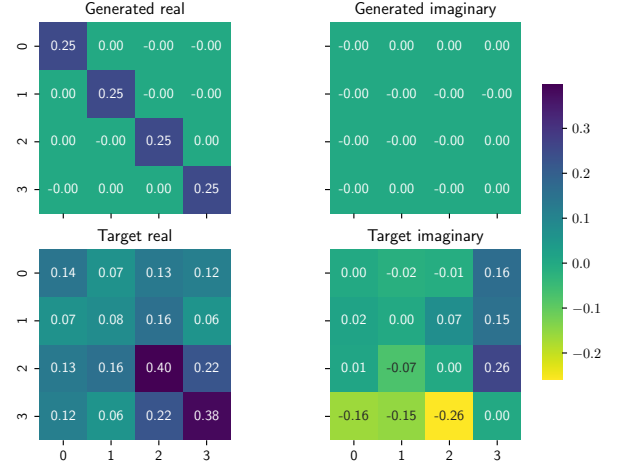


Fig. 6: Visualization of the generated density matrix and target density matrix.

Finally, we observe that $U(\theta)$ is not necessarily optimized to identity matrix \mathbb{I} every time, although it seems straightforward to use \mathbb{I} as a simple way to keep input and output states the same. Learning algorithms do not always choose to do so. This is because different ensembles of quantum states can be transformed into each other by $U(\theta)$ while maintaining the same density matrix. There may be more than one transform in the hypothesis space that can satisfy this condition. After training, which one of them is chosen depends on the position of $U(\theta)$ in the hypothesis space when the parameters θ are initialized.

6 NUMERICAL SIMULATIONS

Our evaluation of the proposed models involves generating a variety of random quantum states, encompassing both pure and mixed states. We have conducted a comparative analysis of our models, QuGAN [9], and EQ-GAN [21]. QuGAN implementation follows the original framework in PennyLane [66], whereas EQ-GAN is built by using the source code in [67].

For more details on the numerical simulations, including training hyperparameters, target state preparation, and the training and generation metrics of our models, please refer to the supplementary file.

6.1 Setup

We use the Tensorcircuit framework [68] to simulate quantum circuits and the Tensorflow framework [69] for parameter optimization, with the optimizer Adam [70]. During

TABLE 2: Comparison of pure state generation between QuGAN, EQ-GAN, and our proposed models.

Models	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$	$N = 8$
QuGAN [9]	$0.999 \pm 5e-5$	$0.996 \pm 4e-3$	$0.995 \pm 4e-3$	$0.993 \pm 3e-3$	$0.990 \pm 4e-3$	$0.986 \pm 3e-3$	$0.978 \pm 6e-3$	$0.971 \pm 1e-1$
EQ-GAN [21]	1.00 ± 0	$0.978 \pm 5e-2$	$0.947 \pm 4e-2$	$0.896 \pm 8e-2$	$0.898 \pm 9e-2$	$0.872 \pm 9e-2$	$0.833 \pm 1e-1$	$0.824 \pm 8e-2$
QGDM (Ours)	$0.999 \pm 3e-5$	$0.999 \pm 2e-4$	$0.999 \pm 2e-3$	$0.996 \pm 8e-3$	/	/	/	/
RQGDM (Ours)	/	$0.994 \pm 8e-3$	$0.999 \pm 6e-4$	$0.999 \pm 4e-4$	$0.997 \pm 4e-3$	$0.995 \pm 1e-2$	$0.994 \pm 4e-3$	$0.990 \pm 1e-2$

training, a larger learning rate may make it difficult for the model to converge, while setting the learning rate too small makes the model converge too slowly. To better optimize the model, we use a cosine decay method [71] to dynamically adjust the learning rate.

Our numerical simulations cover a number of qubits ranging from 1 to 8. QGDM requires $2N$ qubits to generate an N -qubit target state. Therefore, in our simulations, QGDM achieves results for $N \leq 4$. EQ-GAN can generate mixed states for $N \leq 7$, as simulating the $N = 8$ task demands an unattainable amount of computer memory. At $N = 1$, RQGDM degenerates into QGDM. Therefore, in the next two subsections, the results for QGDM at $N > 4$, RQGDM at $N = 1$, and EQ-GAN at $N = 8$ in mixed-state generation are excluded. All simulations are repeated 10 times with different random seeds to obtain the statistical characteristics of our results.

6.2 Pure State Generation

6.2.1 Impact of N_τ

We first investigate the impact of N_τ in QGDM for generating pure states. The results are shown in Fig. 7. Generation fidelity improves with N_τ when $N_\tau \leq N$, achieving the maximum when $N_\tau = N$ for all N .

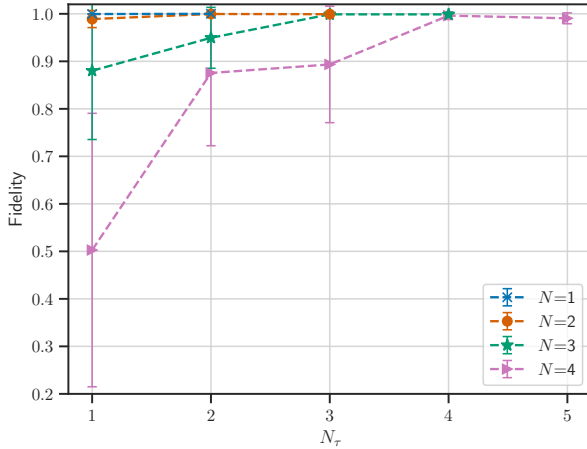


Fig. 7: The generation fidelity of QGDM in generating target pure states for different qubits with respect to N_τ .

However, when $N_\tau > N$, except for the case $N = 1$, fidelity decreases slightly in other cases. The reason for this may be that there is a redundant qubit that complicates the denoising process. The denoising circuit aims to predict the denoising state $\tilde{\rho}_t$ by merging information from τ_t and ρ_t , and then projecting its prediction onto the first N qubits of the τ_t state. These qubits are denoted as register A in Fig. 3. In cases where $N_\tau > N$, there is an extra qubit in the state

τ_t compared to ρ_t . This additional qubit may be redundant, complicating the process of merging and output prediction.

When $N_\tau < N$, register A contains a part of the qubits of ρ_t . However, the presence of overlapping qubits between register A and the qubits of ρ_t may diminish the efficacy of $U(\theta)$ in such scenarios since a part of the information from ρ_t is already stored in A . In Section 5, we have investigated a well-intentioned but faulty denoising design. This design involves transferring all qubits from ρ_t to the output register, maximizing the overlap between A and the qubits of ρ_t . This extreme case strengthens our hypothesis, illustrating the potential implications.

Based on the results in Fig. 7, subsequent simulations default to $N_\tau = N$ as this setting yields the best generative results with our proposed method.

6.2.2 Pure States Generation Results

In Fig. 8, we compare QGDM and RQGDM with QuGAN [9] and EQ-GAN [21] to assess their fidelity in generating the desired pure state. The results indicate that as the number of qubits in the target state increases, the fidelity of all models decreases.

QuGAN maintains high fidelity at lower qubit counts. However, its median fidelity decreases as the number of qubits increases, along with an increase in variance. For instances with $N \geq 7$, QuGAN's fidelity falls below 0.99 for all 10 experiments, indicating its limitations in handling more complex quantum states. EQ-GAN achieves a perfect fidelity of 1.0 when the system has only one qubit. However, its fidelity decreases as the number of qubits grows, displaying the largest variance among the four models. From 3 qubits onward, EQ-GAN's fidelity consistently stays below 0.99, highlighting its challenges in generating pure states. The fidelity of QGDM and RQGDM remains mostly above 0.99 across all qubit numbers, with more concentrated data distributions, indicating their better scalability than QuGAN and EQ-GAN. In most scenarios, RQGDM surpasses the other three models in terms of median fidelity and demonstrates greater stability at higher qubit counts than its peers. In summary, QGDM excels in smaller quantum systems, while RQGDM offers a more scalable solution across a broader range of qubit configurations.

To better evaluate the performance differences among models, we present the generative fidelity values for each model in Table. 2. The fidelity of QuGAN diminishes as the number of qubits increases, starting from $0.999 \pm 5e-5$ at $N = 1$ to $0.971 \pm 1e-1$ at $N = 8$. It is worth noting that its fidelity drops below 0.99 for $N \geq 6$, indicating a decline in performance as the quantum system size increases. The fidelity of QuGAN from $N = 1$ to 5 remains above 0.99, showing a downward trend for larger systems, although these values are consistently lower than those achieved by our proposed method. EQ-GAN achieves the best initial

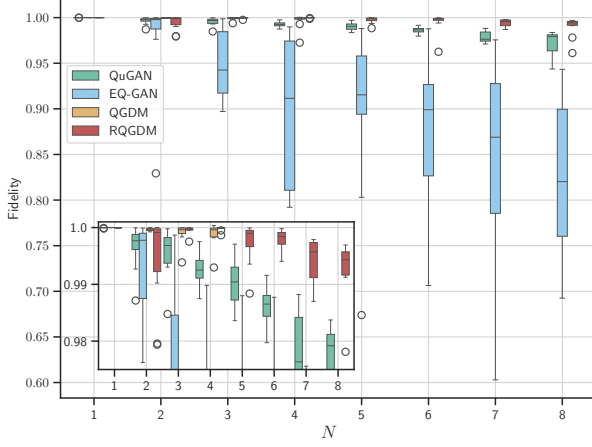


Fig. 8: Pure state generation results. The circles represent outliers, which are data points that significantly deviate from the rest of the data.

result, producing fidelities of 1 at $N = 1$. However, as the number of qubits increases, EQ-GAN shows the lowest average fidelity among the four models, along with the highest variance. QGDM exhibits stable fidelity, with an average of 0.999 at $N = 1, 2$, and 3, and $0.996 \pm 8e-3$ at $N = 4$. Importantly, EQ-GAN consistently demonstrates high fidelity across all numbers of qubits, even achieving 0.99 at $N = 8$. This underscores the superior robustness and accuracy of RQGDM in pure state generation, highlighting the effectiveness and scalability of our proposed models.

6.2.3 Timestep Embedding State Visualization

Fig. 9 shows selected examples visualizing a single-qubit timestep embedding state sequence $\{\tau_t\}_{t=1}^T$ after training, with $T = 30$. Different colors are used to denote the states in each timestep t . Each example demonstrates that τ_t shifts its position on the Bloch sphere surface coherently as t progresses. The density of the states varies as well. Figs. 9(a)-(b) show a more dispersed distribution on the Bloch sphere surface. Figs. 9(c)-(d) illustrate cases where states are more clustered. Figs. 9(e)-(f) display an intermediate situation, with the state sequence moderately and coherently distributed over the surface of the Bloch sphere.

Compared to qubit encoding methods [59], [60], our approach to timestep embedding, as inspired by [63], is more expressive and flexible. The qubit encoding method generates states that are equidistantly arranged in a ring on the Bloch sphere surface. This restricts subsequent learning models to only learning from such fixed patterns. In contrast, our approach allows the timestep embedding states to be adaptable, enhancing their multifunctionality. This could potentially decrease the difficulty of learning for subsequent models and improve the overall model’s performance.

6.3 Mixed State Generation

Fig. 10 shows the fidelity of four quantum generative models for generating mixed states. The results indicate that the fidelity of QuGAN and EQ-GAN decreases as the number of qubits increases, while QGDM and RQGDM maintain high generation fidelity in large-scale tasks. Specifically, QuGAN

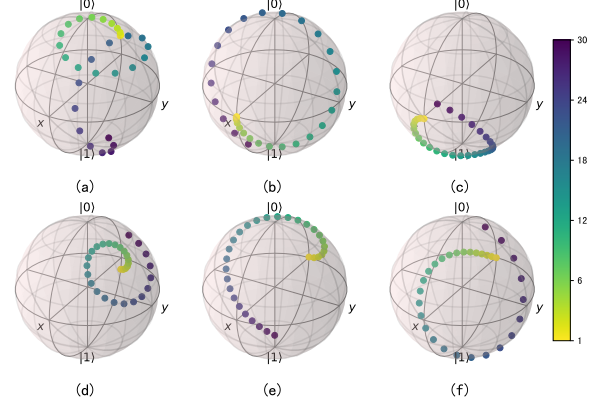


Fig. 9: Selected examples visualizing single-qubit timestep embedding states. Our method allows the timestep embedding state to adaptively distribute itself on the surface of the Bloch sphere while maintaining temporal coherence. This demonstrates that our approach to encoding temporal information is expressive and flexible.

exhibits a continuous decrease in fidelity with an increase in N , sharply falling from above 0.9 at $N = 1$ to approximately 0.3 at $N = 8$ for most fidelity points and the median. EQ-GAN also shows a significant decrease in fidelity but maintains a better median and data point distribution than QuGAN. These findings suggest that EQ-GAN outperforms QuGAN in generating mixed states, but both are difficult to handle the task of generating mixed states with multiple qubits. In contrast, QGDM and RQGDM consistently show a median fidelity above 0.99 across all scenarios. QGDM surpasses both QuGAN and EQ-GAN at $N = 1$, with most fidelity points above 0.99. For $N = 2, 3$, and 4, RQGDM has a slight advantage over QGDM, with RQGDM having more concentrated data points. Notably, at $N \geq 5$, RQGDM is significantly better than QuGAN and EQ-GAN, maintaining all fidelity scores above 0.9.

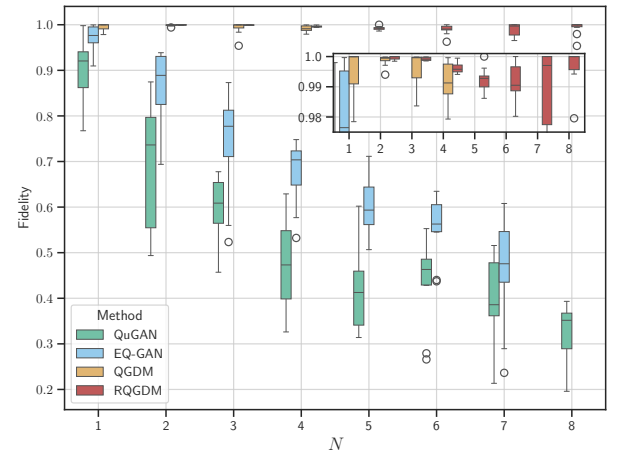


Fig. 10: Mixed state generation results. The circles represent outliers, which are data points that significantly deviate from the rest of the data.

In Table 3, we present a detailed comparison of the fidelity among these four methods. QuGAN starts at a

TABLE 3: Comparison of mixed state generation performance between QuGAN, EQ-GAN, and our proposed models.

Models	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$	$N = 8$
QuGAN [9]	$0.902 \pm 6e-2$	$0.694 \pm 1e-1$	$0.598 \pm 6e-2$	$0.477 \pm 9e-2$	$0.414 \pm 8e-2$	$0.439 \pm 9e-2$	$0.396 \pm 8e-2$	$0.324 \pm 6e-2$
EQ-GAN [21]	$0.972 \pm 3e-2$	$0.867 \pm 8e-2$	$0.741 \pm 1e-1$	$0.676 \pm 7e-2$	$0.602 \pm 6e-2$	$0.555 \pm 6e-2$	$0.463 \pm 1e-1$	/
QGDM (Ours)	$0.995 \pm 7e-3$	$0.999 \pm 1e-3$	$0.992 \pm 1e-2$	$0.992 \pm 7e-3$	/	/	/	/
RQGDM (Ours)	/	$0.999 \pm 6e-4$	$0.999 \pm 5e-4$	$0.996 \pm 2e-3$	$0.993 \pm 4e-3$	$0.990 \pm 1e-2$	$0.992 \pm 2e-2$	$0.993 \pm 2e-2$

fidelity of $0.902 \pm 6e-2$ at $N = 1$, declining to $0.324 \pm 6e-2$ at $N = 8$. This trend highlights QuGAN’s limitations regarding scalability. EQ-GAN has a lower standard deviation in most cases (except for the case $N = 3$), while generally displaying higher average fidelity than QuGAN. In contrast, QGDM and RQGDM maintain a fidelity of greater than 0.99 across all cases, accompanied by a low standard deviation, indicating their superior scalability. Notably, our methods achieve fidelity of up to 0.999 in some cases, demonstrating their excellent generative capabilities. In particular, RQGDM achieves $0.992 \pm 1e-2$ for $N = 7$ and maintains $0.993 \pm 1e-2$ for $N = 8$, demonstrating the ability of our models to maintain high fidelity generation even on larger scale problems.

We can quantify the performance improvement of our models relative to the QGAN-based models by calculating the relative change [72] across different model groups (*i.e.*, our models and QGAN-based models). By extracting the best fidelity values for each N from different model groups in Table 3 and calculating the average fidelities, we find that our models demonstrates a 53.02% relative performance improvement over QGAN-based models. For detailed calculations, please refer to the supplementary file.

Our method has demonstrated significant advantages in comparison with QuGAN and EQ-GAN. The results indicate that QuGAN and EQ-GAN perform poorly in the tasks of generating mixed states. The fidelity decreases as the number of qubits increases, revealing the limitations of these models in handling multi-qubit tasks. This limitation might be due to QGANs’ optimization strategies, which rely on the min-max saddle point game, which is generally less effective than using gradient descent for convex functions. It is well known that training classical GAN models is prone to issues such as model collapse, instability, and non-convergence. With an exponential increase in the dimensions of the density matrix, there may be higher demands on the training stage, such as the need for a more suitable learning rate. Given these conditions, the QGANs’ unstable training performance may significantly impact the final generation results. In contrast, the training process of our models, which minimizes a convex function using gradient descent, shows more user-friendly and stable training performance. This might explain why our model well outperforms QuGAN and EQ-GAN.

7 CONCLUSION

Inspired by classical diffusion models, this paper has introduced a Quantum Generative Diffusion Model (QGDM), as a novel quantum generative approach for generating a quantum state ensemble. Its fundamental concept is that any target quantum state can be converted into a completely mixed state through a non-unitary forward process. Following this, a trainable backward process is employed to reconstruct the target state from the completely mixed

state. In the forward process, a timestep-dependent depolarizing channel incrementally converts the input state into a maximally random state, known as a completely mixed state. The denoising process involves a parameterized function composed of a timestep embedding circuit and a denoising circuit, employing the concept of parameter sharing. Specifically, the timestep embedding circuit embeds temporal information into a quantum state, while the denoising circuit processes the polluted quantum state along with the timestep embedding state to produce a denoised state. To minimize auxiliary qubits in QGDM, we introduce a novel resource-efficient version of QGDM called RQGDM. Numerical simulations, conducted up to 8 qubits, compare our models with QuGAN and EQ-GAN. The results indicate that both QGDM and RQGDM surpass QuGAN and EQ-GAN in generating both pure and mixed quantum states. Notably, our models achieve 53.02% higher fidelity in mixed-state generation tasks than QGANs. These results highlight their excellent performance and scalability.

Our work introduces new models to the field of quantum generative learning. These models overcome the convergence issues faced by QGANs when generating mixed states, providing powerful new tools for the challenging task of quantum state generation. Many of the designs and findings in our paper are pioneering and can serve as prototypes and valuable inspirations for future research on quantum versions of diffusion models.

Future work should explore the impact of different noise schedules on generation performance. Additionally, designing more efficient structures for a denoising process to reduce resource requirements, including the number of auxiliary qubits and the total number of quantum gates, is a worthwhile direction to pursue. Investigating the relationship between problem properties and the number of different time steps can provide important insights for further understanding QGDM. Theoretically analyzing and proving that QGDM converges faster than QGAN is an important and open issue. This could provide theoretical support for quantum versions of classic diffusion models and deepen our understanding of the advantages of these models.

ACKNOWLEDGMENTS

This work is supported by the Science and Technology Development Fund, Macau SAR (0093/2022/A2, 0076/2022/A2, and 0008/2022/AGJ), the National Key Research and Development Program of China (2023YFB2703800) and Guangdong Basic and Applied Basic Research Foundation (No. 2022A1515140116). The authors are thankful to Prof. Wenmin Wang for his valuable discussions.

REFERENCES

- [1] S. Bond-Taylor, et al., "Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [2] OpenAI, "Gpt-4 technical report," <https://cdn.openai.com/papers/gpt-4.pdf>, 2023.
- [3] R. Rombach, et al., "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [4] J. Tian, et al., "Recent advances for quantum neural networks in generative learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [5] J. Biamonte, et al., "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.
- [6] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Physical Review Letters*, vol. 122, no. 4, p. 040504, 2019.
- [7] J. Shi, et al., "Parameterized hamiltonian learning with quantum circuit," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 6086–6095, 2022.
- [8] S. Lloyd and C. Weedbrook, "Quantum generative adversarial learning," *Physical review letters*, vol. 121, no. 4, p. 040502, 2018.
- [9] P.-L. Dallaire-Demers and N. Killoran, "Quantum generative adversarial networks," *Physical Review A*, vol. 98, no. 1, p. 012324, 2018.
- [10] M. Benedetti, et al., "A generative modeling approach for benchmarking and training shallow quantum circuits," *npj Quantum Information*, vol. 5, no. 1, p. 45, 2019.
- [11] J.-G. Liu and L. Wang, "Differentiable learning of quantum circuit born machines," *Physical Review A*, vol. 98, no. 6, p. 062324, 2018.
- [12] A. Khoshman, et al., "Quantum variational autoencoder," *Quantum Science and Technology*, vol. 4, no. 1, p. 014001, 2018.
- [13] M. H. Amin, E. Andriyash, et al., "Quantum boltzmann machine," *Physical Review X*, vol. 8, no. 2, p. 021050, 2018.
- [14] H. Lin, et al., "How generative adversarial networks promote the development of intelligent transportation systems: A survey," *IEEE/CAA Journal of Automatica Sinica*, 2023.
- [15] G. Cai, et al., "Unsupervised domain adaptation with adversarial residual transform networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 3073–3086, 2019.
- [16] J. Chen, et al., "Mogan: Morphologic-structure-aware generative learning from a single image," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.
- [17] Z. Chen, et al., "Supervised anomaly detection via conditional generative adversarial network and ensemble active learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7781–7798, 2022.
- [18] J. Zhu, et al., "Label-guided generative adversarial network for realistic image synthesis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3311–3328, 2022.
- [19] S. Chakrabarti, et al., "Quantum wasserstein generative adversarial networks," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [20] P. Braccia, et al., "How to enhance quantum generative adversarial learning of noisy information," *New Journal of Physics*, vol. 23, no. 5, p. 053024, 2021.
- [21] M. Y. Niu, et al., "Entangling quantum generative adversarial networks," *Physical Review Letters*, vol. 128, no. 22, p. 220505, 2022.
- [22] H.-L. Huang, et al., "Experimental quantum generative adversarial networks for image generation," *Physical Review Applied*, vol. 16, no. 2, p. 024051, 2021.
- [23] D. Silver, et al., "Mosaic: Quantum generative adversarial networks for image generation on nisy computers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7030–7039.
- [24] S. L. Tsang, et al., "Hybrid quantum-classical generative adversarial network for high resolution image generation," *IEEE Transactions on Quantum Engineering*, 2023.
- [25] N.-R. Zhou, et al., "Hybrid quantum-classical generative adversarial networks for image generation via learning discrete distribution," *Signal Processing: Image Communication*, vol. 110, p. 116891, 2023.
- [26] C. Zoufal, et al., "Quantum generative adversarial networks for learning and loading random distributions," *npj Quantum Information*, vol. 5, no. 1, p. 103, 2019.
- [27] J. Zeng, et al., "Learning and inference on generative adversarial quantum circuits," *Physical Review A*, vol. 99, no. 5, p. 052306, 2019.
- [28] H. Situ, et al., "Quantum generative adversarial network for generating discrete distribution," *Information Sciences*, vol. 538, pp. 193–208, 2020.
- [29] J. Romero and A. Aspuru-Guzik, "Variational quantum generators: Generative adversarial quantum machine learning for continuous distributions," *Advanced Quantum Technologies*, vol. 4, no. 1, p. 2000003, 2021.
- [30] S. Chaudhary, et al., "Towards a scalable discrete quantum generative adversarial neural network," *Quantum Science and Technology*, vol. 8, no. 3, p. 035002, 2023.
- [31] J. Sohl-Dickstein, et al., "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*. PMLR, 2015, pp. 2256–2265.
- [32] J. Ho, et al., "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.
- [33] J. Song, et al., "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2020.
- [34] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8162–8171.
- [35] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.
- [36] C. Luo, "Understanding diffusion models: A unified perspective," *arxiv:2208.11970*, 2022.
- [37] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [38] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.
- [39] Y. Du, M.-H. Hsieh, et al., "Expressive power of parametrized quantum circuits," *Physical Review Research*, vol. 2, no. 3, p. 033125, 2020.
- [40] D. Zhu, et al., "Training of quantum circuits on a hybrid quantum computer," *Science advances*, vol. 5, no. 10, p. eaaw9918, 2019.
- [41] B. Coyle, et al., "The born supremacy: quantum advantage and training of an ising born machine," *npj Quantum Information*, vol. 6, no. 1, p. 60, 2020.
- [42] O. Kiss, et al., "Conditional born machine for monte carlo event generation," *Physical Review A*, vol. 106, no. 2, p. 022612, 2022.
- [43] B. Coyle, et al., "Quantum versus classical generative modelling in finance," *Quantum Science and Technology*, vol. 6, no. 2, p. 024013, 2021.
- [44] E. Y. Zhu, et al., "Generative quantum learning of joint probability distribution functions," *Physical Review Research*, vol. 4, no. 4, p. 043092, 2022.
- [45] S. E. Fahlman, et al., "Massively parallel architectures for al: Netl, thistle, and boltzmann machines," in *National Conference on Artificial Intelligence, AAAI*, 1983.
- [46] D. H. Ackley, et al., "A learning algorithm for boltzmann machines," *Cognitive Science*, vol. 9, no. 1, pp. 147–169, 1985.
- [47] M. Kieferová and N. Wiebe, "Tomography and generative training with quantum boltzmann machines," *Physical Review A*, vol. 96, no. 6, p. 062327, 2017.
- [48] C. Zoufal, et al., "Variational quantum boltzmann machines," *Quantum Machine Intelligence*, vol. 3, pp. 1–15, 2021.
- [49] J. T. Rolfe, "Discrete variational autoencoders," in *International Conference on Learning Representations*, 2016.
- [50] N. Gao, et al., "High-dimensional similarity search with quantum-assisted variational autoencoder," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 956–964.
- [51] J. Li and S. Ghosh, "Scalable variational quantum circuits for autoencoder-based drug discovery," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 340–345.
- [52] M. Benedetti, et al., "Adversarial quantum circuit learning for pure state approximation," *New Journal of Physics*, vol. 21, no. 4, p. 043023, 2019.
- [53] M. Parigi, et al., "Quantum-noise-driven generative diffusion models," *arxiv:2308.12013*, 2023.
- [54] B. Zhang, et al., "Generative quantum machine learning via denoising diffusion probabilistic models," *Physical Review Letters*, vol. 132, no. 10, p. 100602, 2024.

- [55] A. Cacioppo, *et al.*, "Quantum diffusion models," *arxiv:2311.15444*, 2023.
- [56] S. R. De Groot and P. Mazur, *Non-equilibrium thermodynamics*. Courier Corporation, 2013.
- [57] M. M. Wilde, *Quantum information theory*. Cambridge university press, 2013.
- [58] D. Kingma, *et al.*, "Variational diffusion models," *Advances in Neural Information Processing Systems*, vol. 34, pp. 21 696–21 707, 2021.
- [59] E. Stoudenmire and D. J. Schwab, "Supervised learning with tensor networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [60] E. Grant, *et al.*, "Hierarchical quantum classifiers," *npj Quantum Information*, vol. 4, no. 1, p. 65, 2018.
- [61] P. Rebentrost, *et al.*, "Quantum support vector machine for big data classification," *Physical Review Letters*, vol. 113, no. 13, p. 130503, 2014.
- [62] R. LaRose and B. Coyle, "Robust data encodings for quantum classifiers," *Physical Review A*, vol. 102, no. 3, p. 032420, 2020.
- [63] S. Lloyd, *et al.*, "Quantum embeddings for machine learning," *arxiv:2001.03622*, 2020.
- [64] K. H. Wan, *et al.*, "Quantum generalisation of feedforward neural networks," *npj Quantum Information*, vol. 3, no. 1, p. 36, 2017.
- [65] J. Romero, *et al.*, "Quantum autoencoders for efficient compression of quantum data," *Quantum Science and Technology*, vol. 2, no. 4, p. 045001, 2017.
- [66] V. Bergholm, *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv:1811.04968*, 2018.
- [67] [Online]. Available: https://github.com/tensorflow/quantum/tree/research/eq_gan
- [68] S.-X. Zhang, *et al.*, "Tensorcircuit: a quantum software framework for the nisq era," *Quantum*, vol. 7, p. 912, 2023.
- [69] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, and G. S. C. *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [70] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arxiv:1412.6980*, 2014.
- [71] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2016.
- [72] J. Bennett and W. Briggs, *Using & understanding mathematics: a quantitative reasoning approach*. Pearson, 2019.

Supplementary File for “Quantum Generative Diffusion Model: A Fully Quantum-Mechanical Model for Generating Quantum State Ensemble”

Chuangtao Chen, Qinglin Zhao, *Senior Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, Zhimin He, Zhili Sun, *Senior Member, IEEE* and Haozhen Situ

This supplementary file provides detailed additional information of our proposed models. Section A introduces the hyperparameters used in QGDM and RQGDM models. Section B details the calculation of relative performance improvement in mixed state generation. Section C presents the architecture details of the timestep embedding circuit and the denoising circuit. Section D presents the details of the target state preparation in the numerical simulations. Section E presents the training and generation curves for QGDM and RQGDM. Finally, Section F derives the fidelity boundary between a quantum state and the completely mixed state.

APPENDIX A

THE HYPERPARAMETERS OF THE PROPOSED MODELS

TABLE I: The hyperparameters of QGDM.

Parameters	Pure state / Mixed state			
	$N = 1$	$N = 2$	$N = 3$	$N = 4$
$L_{\mathcal{T}}$	5			
λ	0.02			
s	0.008			
Batch size	16			
T	30			
Epoch	200			
Initial learning rate	0.3			
Final learning rate	0.01			
Learning rate decay steps	200			
L	1		1 / 2	2 / 3

TABLE II: The hyperparameters of RQGDM.

Parameters	Pure state / Mixed state						
	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$	$N = 7$	$N = 8$
$L_{\mathcal{T}}$	5						
λ	0.02						
s	0.008						
Batch size	16						
T	30						30 / 90
Epoch	200			200 / 500			
Initial learning rate	0.3			0.5			
Final learning rate	0.01			0.07			
Learning rate decay steps	200			200 / 500			
L	1	1 / 2		1 / 3		1 / 4	

APPENDIX B

CALCULATION OF RELATIVE PERFORMANCE IMPROVEMENT IN MIXED STATE GENERATION

We use the relative change [1] to quantify the improvement of our models over QGAN-based models in the mixed state generation task. First, we calculate the average of the best fidelity values for our models across different N :

$$\bar{F}_{\text{Ours}}^* = \frac{0.995 + 0.999 + 0.999 + 0.996 + 0.993 + 0.990 + 0.992 + 0.993}{8} = 0.994625. \quad (\text{A1})$$

Then, we calculate the average of the best fidelity values for QGAN-based models across different N :

$$\bar{F}_{\text{QGANs}}^* = \frac{0.972 + 0.867 + 0.741 + 0.676 + 0.602 + 0.555 + 0.463 + 0.324}{8} = 0.65. \quad (\text{A2})$$

The relative performance improvement can be calculated as follows:

$$\frac{\bar{F}_{\text{Ours}}^* - \bar{F}_{\text{QGANs}}^*}{\bar{F}_{\text{QGANs}}^*} \times 100\% = \frac{0.994625 - 0.65}{0.65} \times 100\% = 53.0192\% \approx 53.02\%. \quad (\text{A3})$$

APPENDIX C

CIRCUIT ARCHITECTURE

C.1 Timestep Embedding Circuit

The architecture of an timestep embedding circuit with $N_\tau = 4$ qubits is shown in Fig. 1. The circuit includes single-qubit gates R_x and R_y , and the two-qubit gate $ZZ(\phi) = \exp(-i\phi(Z \otimes Z)/2)$. Our proposed method maps temporal information to appropriate positions on the Bloch sphere (to be demonstrated by our numerical simulations) while maintaining linear complexity in the number of quantum gates, which is sufficient to meet our requirements.

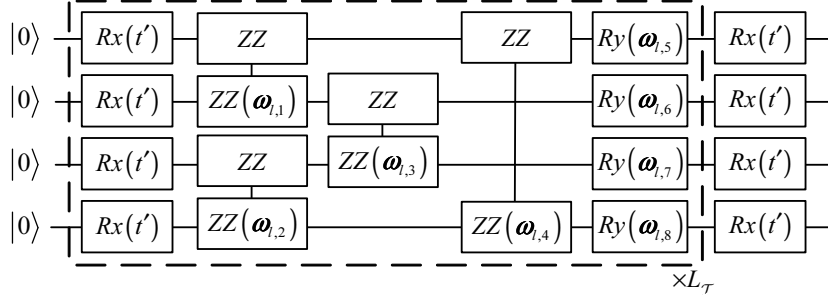


Fig. 1: The conventional ansatz [2] is used for our timestep embedding with trainable parameter ω , and given parameter $t' = t\pi/T$. The circuit in the dashed lines is repeated for L_τ times. Where $l \in \{1, \dots, L_\tau\}$.

C.2 Denoising Circuit

The architecture of a 4-qubit denoising circuit $U(\theta)$ is shown in Fig. 2. The circuit includes single-qubit gates R_z and R_x , and the two-qubit gate $XX(\phi) = \exp(-i\phi(X \otimes X)/2)$. This circuit architecture using $\mathcal{O}(N^2)$ two-qubit gates may affect the efficiency of our method. However, our focus is to evaluate the effectiveness of our method. To ensure that this evaluation is not affected by the circuit design, we choose a more complex architecture, even though it increases the gate complexity. This decision allows our results to accurately represent the potential of our method. In addition, it may not be necessary for us to maintain a consistent circuit architecture across all timesteps. Adaptable circuits that change architecture at different timesteps can reduce the total number of gates in QGDM. We leave the task of designing efficient circuit architectures for QGDM for future work. An effective approach is to use quantum architecture search algorithms [3], [4], [5], [6].

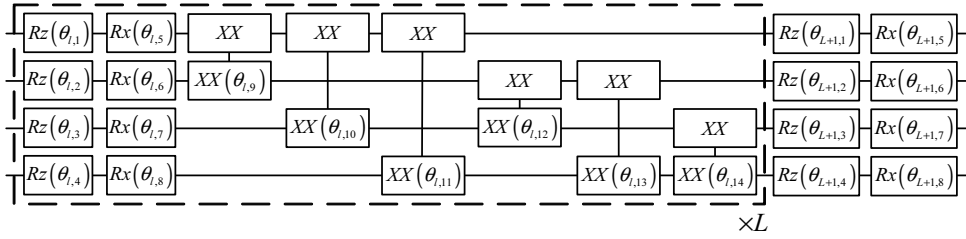


Fig. 2: The architecture of a 4-qubit denoising circuit $U(\theta)$, where θ denotes trainable parameters. To enhance its learning capability, the dashed-boxed part is repeated L times. $l \in \{1, \dots, L\}$.

APPENDIX D

TARGET STATE PREPARATION

D.1 Pure State

Fig. 3 illustrates the circuit to obtain a target pure state $|\psi\rangle$. All parameters in the target state generation circuit are uniformly sampled from the range $[0, \pi]$.

D.2 Mixed State

Quantum mixed states are generated through $\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i|$, where $\{|\psi_i\rangle\}$ is randomly produced by the circuit shown in Fig. 3. The probability vector $\{p_i\}$ is obtained by uniform sampling from $(0, 1]$ and normalized through a softmax operation to ensure $\sum_i p_i = 1$ and $p_i > 0$ for any index i . In this paper, we generate a target mixed state by probabilistically mixing two random pure states.

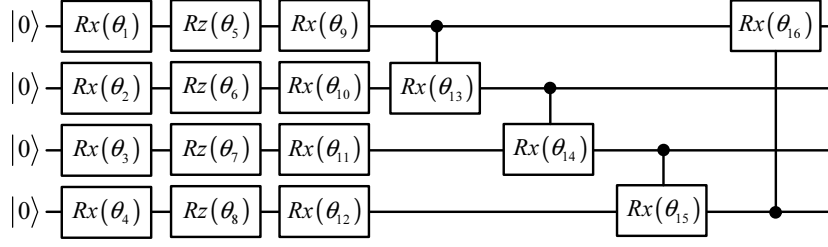


Fig. 3: The architecture of a 4-qubit circuit to prepare a target pure state. All the parameters θ are initialized in $[0, \pi]$.

APPENDIX E

MODEL TRAINING AND GENERATION CURVES

Fig. 4 displays the metrics during the training and generation of QGDM, which include the loss values and fidelity of the generated quantum states. In contrast, Figs. 5-6 show the metrics of RQGDM.

Each scenario (number of qubits, pure or mixed state) includes two subfigures: the left subfigure illustrates the trend of training loss function value as the number of epochs increases, while the right subfigure shows how fidelity changes over timesteps. The red dashed line indicates the fidelity changes between the noisy quantum state and target one during the diffusion process, as time steps increase (indicated by the red arrow from right to left). The solid blue line represents the trend of fidelity between the generated quantum state and target one during the generation process (indicated by the blue arrow, from left to right). The shaded area in the figure represents the range of standard deviation.

The loss curves indicate that the loss functions of both QGDM and RQGDM converge to near zero as training epochs increase, and the number of epochs required to converge to zero increases with N , regardless of whether the target state is pure or mixed. Moreover, generating mixed states requires more iterations than generating pure ones, especially as N increases. This suggests that generating mixed states is more challenging than generating pure ones.

As shown in Section F, the fidelity between any quantum state ρ and the completely mixed state is $\frac{1}{d} \leq F(\rho, \mathbb{I}/d) \leq 1$. For pure states, the fidelity between the target state and the completely mixed state is $1/d$. Therefore, the standard deviation of all diffusion fidelity curves (red dashed lines) for pure states is zero. For mixed states, the lower bound of the fidelity is $\frac{1}{d}$. As N increases, the random quantum states appear more similar [7], causing the variance of the fidelity to decrease. Therefore, at $t = T$ (i.e., the far left of the horizontal axis), both the fidelity values and the standard deviation decrease with N .

In all scenarios, the fidelity curve during generation (blue solid line) increases as the timestep t decreases, eventually approaching 1 with the smallest standard deviation. The overlap between the generation fidelity curve and the diffusion one decreases as the number of qubits increases. As the number of qubits increases, the generation-fidelity curve diverges more from the diffusion-fidelity curve. Additionally, the generation fidelity curve (blue solid line) becomes steeper as N increases.

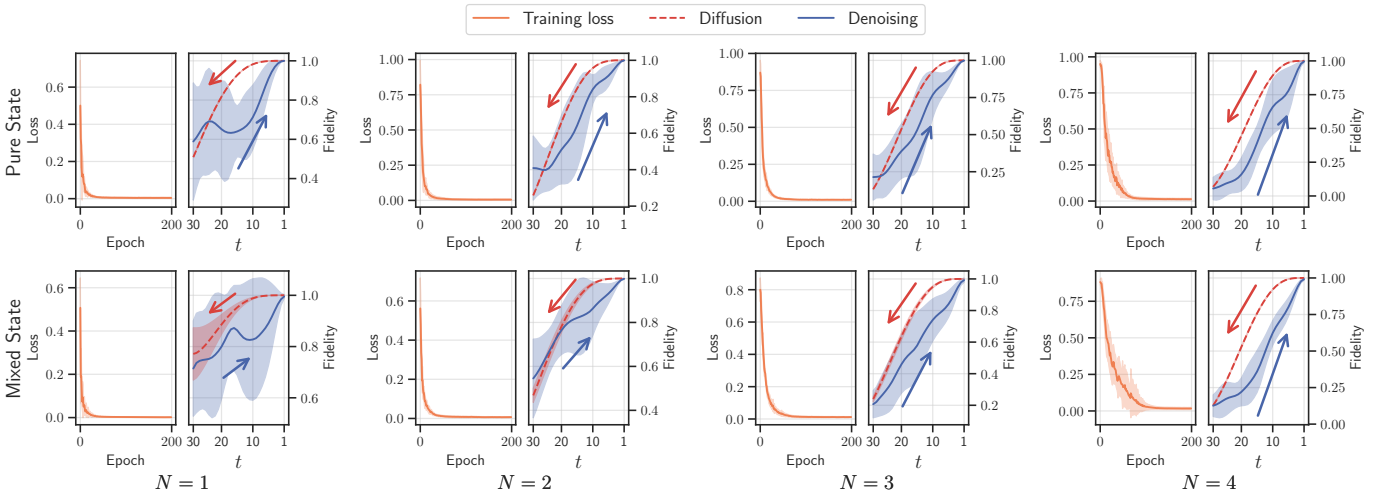


Fig. 4: Metrics of training and generation process of QGDM.

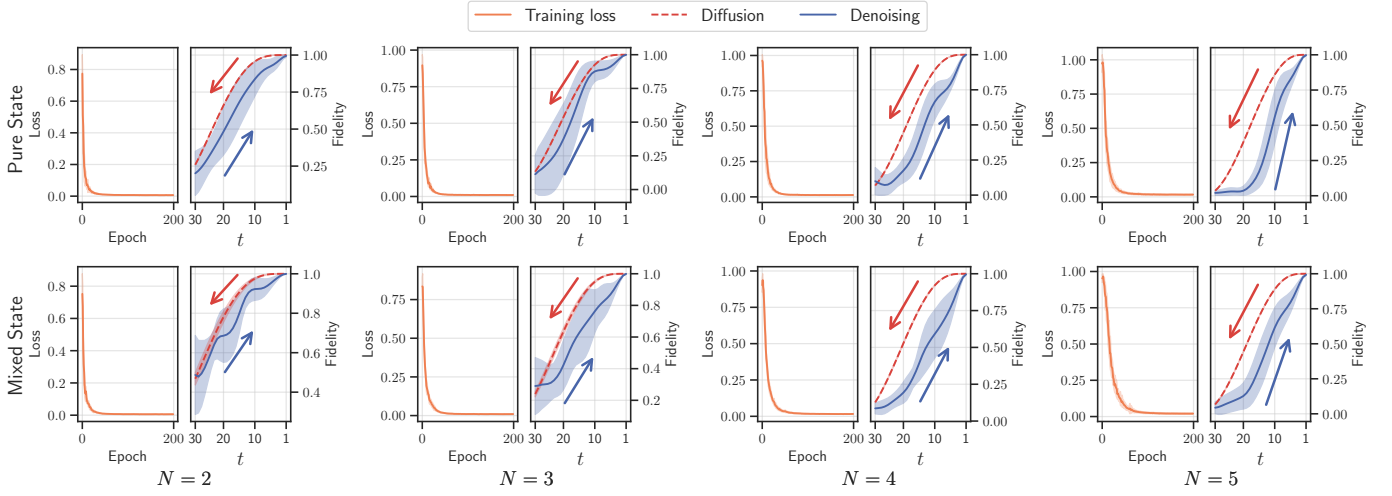


Fig. 5: Metrics of training and generation process of RQGDM when $N = 2, 3, 4$ and 5 .

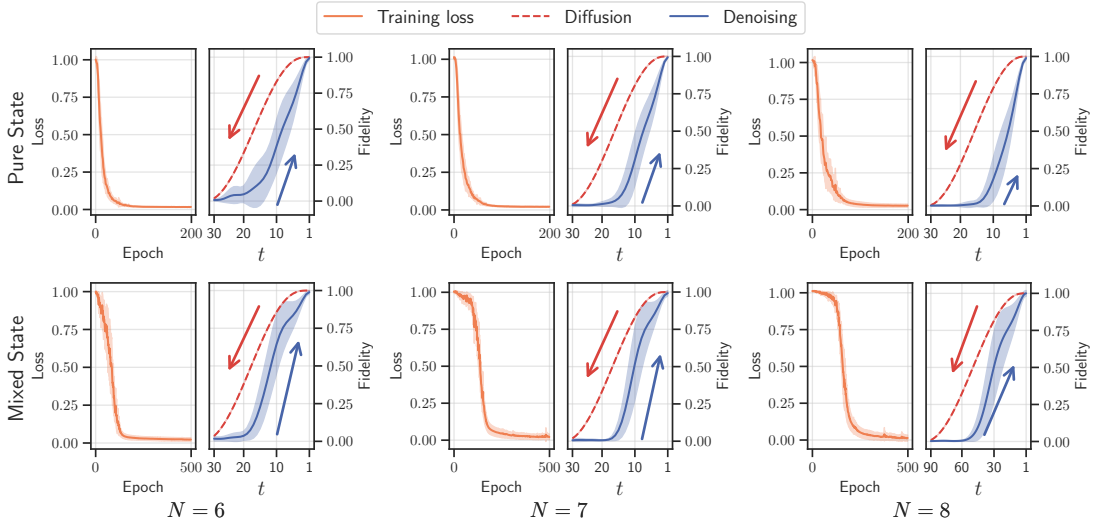


Fig. 6: Metrics of training and generation process of RQGDM when $N = 6, 7$ and 8 .

APPENDIX F

DERIVATION OF FIDELITY BOUNDARY BETWEEN A QUANTUM STATE AND COMPLETELY MIXED STATE

We start by considering the definition of quantum fidelity F between a quantum state ρ and the completely mixed state $\frac{\mathbb{I}}{d}$, where \mathbb{I} is the identity matrix and d is the dimension of the Hilbert space. The fidelity is given by:

$$\begin{aligned}
 F\left(\rho, \frac{\mathbb{I}}{d}\right) &= \left(\text{tr} \sqrt{\sqrt{\rho} \frac{\mathbb{I}}{d} \sqrt{\rho}}\right)^2 \\
 &= \frac{1}{d} (\text{tr} \sqrt{\rho})^2 \\
 &= \frac{1}{d} \left(\text{tr} \sum_k \sqrt{\lambda_k} |k\rangle\langle k|\right)^2 \\
 &= \frac{1}{d} \left(\sum_{i,k} \sqrt{\lambda_k} \langle i|k\rangle\langle k|i\rangle\right)^2 \\
 &= \frac{1}{d} \left(\sum_k \sqrt{\lambda_k}\right)^2.
 \end{aligned} \tag{A4}$$

Three distinct cases need to be considered.

1. If ρ is the completely mixed state, then fidelity reaches its maximum:

$$F\left(\rho, \frac{\mathbb{I}}{d}\right) = 1. \quad (\text{A5})$$

2. If ρ is a mixed state that is not the completely mixed state, the eigenvalues $\{\lambda_k\}$ are not necessarily equal but are non-negative and sum to one. This results in:

$$\begin{aligned} F\left(\rho, \frac{\mathbb{I}}{d}\right) &= \frac{1}{d} \left(\sum_k \sqrt{\lambda_k} \right)^2 \\ &> \frac{1}{d} \sum_k \left(\sqrt{\lambda_k} \right)^2 \\ &= \frac{1}{d}. \end{aligned} \quad (\text{A6})$$

Thus, if ρ is a mixed state that is not the completely mixed state, the fidelity holds $F\left(\rho, \frac{\mathbb{I}}{d}\right) > \frac{1}{d}$.

3. If ρ is a pure state ($\rho = |\psi\rangle\langle\psi|$), where ρ has a unique eigenvalue 1 and its corresponding eigenvector is $|\psi\rangle$. Then

$$F\left(\rho, \frac{\mathbb{I}}{d}\right) = \frac{1}{d}. \quad (\text{A7})$$

In summary, the fidelity boundary between any quantum state ρ and the completely mixed state $\frac{\mathbb{I}}{d}$ is

$$\frac{1}{d} \leq F\left(\rho, \frac{\mathbb{I}}{d}\right) \leq 1. \quad (\text{A8})$$

REFERENCES

- [1] J. Bennett and W. Briggs, *Using & understanding mathematics: a quantitative reasoning approach*. Pearson, 2019.
- [2] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, "Quantum embeddings for machine learning," *arXiv:2001.03622*, 2020.
- [3] Z. He, L. Li, S. Zheng, Y. Li, and H. Situ, "Variational quantum compiling with double Q-learning," *New Journal of Physics*, vol. 23, no. 3, p. 033002, 2021, IOP Publishing.
- [4] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, "Differentiable quantum architecture search," *Quantum Science and Technology*, vol. 7, no. 4, p. 045023, 2022, IOP Publishing.
- [5] Y. Du, T. Huang, S. You, M.-H. Hsieh, and D. Tao, "Quantum circuit architecture search for variational quantum algorithms," *npj Quantum Information*, vol. 8, no. 1, p. 62, 2022, Nature Publishing Group UK London.
- [6] Z. He, M. Deng, S. Zheng, L. Li, and H. Situ, "Training-free quantum architecture search," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.
- [7] P. Hayden, D. W. Leung, and A. Winter, "Aspects of generic entanglement," *Communications in Mathematical Physics*, vol. 265, pp. 95–117, 2006, Springer.