Denoising diffusion models for quantum state generation

Tiago de S. Farias, Alexandre C. Ricardo, Matheus da S. Fonsenca, Amanda G. Valério and Celso J. Villas-Boas

***Abstract*— We introduce a denoising diffusion probabilistic model (DDPM) for learning and sampling distributions of qubit pure states. By representing states as four-dimensional real vectors, we train a neural denoiser to invert a Gaussian corruption process. This approach provides a flexible, likelihood-based generative model that supports conditioning on physically meaningful quantities. We demonstrate its versatility across three distinct tasks: unconditionally generating states of maximum coherence, conditionally sampling states to match specified Bloch vector expectations, and learning the bimodal output distribution of a black-box quantum channel. Our results show that our model faithfully reproduces target distributions, yielding sharp modes for highly concentrated state ensembles.**

***Keywords*— Quantum State Generation, Diffusion Models, Conditional Generation, Qubits**

I. Introduction

Learning and sampling from distributions over quantum states is a central task in quantum information science, with critical applications in state preparation, quantum circuit compilation, quantum error mitigation, and device characterization [1], [2], [3], [4]. Classical generative models have recently shown significant promise for representing and generating quantum data, including quantum states and measurement outcomes [5], [6]. Among these, denoising diffusion probabilistic models have emerged as a robust and stable class of generative models [7], [8]. They are known for their training stability, high-quality sample generation, and a principled likelihood-based framework, which makes them an attractive candidate for quantum applications.

In this work, we adapt the classical DDPM framework for the task of generating qubit pure states and introduce conditioning mechanisms to produce states with specific, physically meaningful properties. Our primary contributions are: a minimal and effective DDPM formulation for single-qubit pure states using a four-dimensional real vector representation. This is complemented by canonicalization steps that enforce the unit norm and global phase constraints inherent to quantum states, a lightweight conditional network architecture that incorporates physical constraints by concatenating time and context embeddings at each processing block, enabling conditioning on properties such as Bloch vector expectation values or other scalar-based state descriptors. Also, we demonstrate the model's effectiveness across three distinct case studies: generating states of maximum coherence, conditioning on specific expectation values, and learning the output distribution of a non-unital "black box" quantum process, which produces characteristic bimodal patterns on the Bloch sphere.

Physics Department, Center for Exact Sciences and Technology, Federal University of São Carlos, 13565-905, São Carlos, SP, Brazil. E-mail: tiago.farias@ufscar.br.

II. Related Work

The use of classical generative models for quantum systems is an expanding field of research [9]. Early work explored the use of restricted Boltzmann machines and autoregressive models for representing wavefunctions [10]. More recently, other models like generative adversarial networks and normalizing flows have been applied to generate quantum states or measurement data. While powerful, these generative models can suffer from training instability and mode collapse [11].

Diffusion models and score-based methods offer a compelling alternative, providing strong mode coverage and stable training dynamics, making them well-suited for quantum-data applications [7]. Recent work has explored quantum-native diffusion models, which use quantum circuits for both the diffusion and denoising processes [12]. Other approaches have focused on hybrid quantum-classical models [13]. Our work aligns with a growing interest in using classical machine learning to solve problems in the quantum domain, where the diffusion process is classical, but the data being modeled is quantum in nature. This approach avoids the hardware limitations of near-term quantum devices while still leveraging the power of deep generative models [14].

III. Quantum state generation

Our method adapts the framework of denoising diffusion probabilistic models to learn distributions of pure single-qubit states. A DDPM consists of two processes: a fixed forward diffusion process that gradually adds Gaussian noise to the data, and a learned reverse denoising process that iteratively removes the noise to generate new samples.

A. Forward Diffusion Process

The forward process is a Markov chain that progressively corrupts an initial data sample x_0 over T timesteps. At each step t, a small amount of Gaussian noise is added according to a predefined variance schedule β_t. Given a data point x_{t-1}, the next point in the sequence is sampled from:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I). \tag{1}$$

A key property of this process is that we can sample x_t at any arbitrary timestep t directly from x_0. Letting $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, we have:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\alpha}_t x_0, (1-\bar{\alpha}_t)I). \tag{2}$$

As $t \to T$, $\bar{\alpha}_T \to 0$, and x_T approaches an isotropic Gaussian distribution $\mathcal{N}(0, I)$, from which it is easy to sample.

B. Reverse Denoising Process

The reverse process aims to learn the distribution $q(x_{t-1}|x_t)$ to reverse the noising process, starting from a sample $x_T \sim \mathcal{N}(0, I)$ and iteratively denoising it to produce a clean sample x_0. This reverse transition is intractable, so we approximate it with a neural network, $p_\theta(x_{t-1}|x_t)$, parameterized by θ.

The training objective is to minimize the difference between the true and learned reverse processes. It has been shown that this can be simplified to training a neural network $\epsilon_\theta(x_t, t)$ to predict the noise that was added to x_0 to obtain x_t. The loss function is formulated as the mean squared error between the actual and predicted noise:

$$L(\theta) = \mathbb{E}_{t,x_0,\epsilon}[||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)||^2], \quad (3)$$

where $\epsilon \sim \mathcal{N}(0, I)$ is the sampled noise and t is sampled uniformly from $\{1, \ldots, T\}$.

For conditional generation, where we want to generate states with a specific property c, the model is modified to take this context vector as an additional input, $\epsilon_\theta(x_t, t, c)$. The context is injected into the model at each processing block, guiding the denoising process toward generating samples that exhibit the desired property.

C. Sampling from the Learned Model

Once the model ϵ_θ is trained, we can generate new quantum states by starting with random noise $x_T \sim \mathcal{N}(0, I)$ and iteratively applying the denoising step for $t = T, T-1, ...1$:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z, \quad (4)$$

where $z \sim \mathcal{N}(0, I)$ is new noise added at each step, and σ_t^2 is the variance of the reverse process, typically set to β_t. When conditioning on a property c, we simply use the conditional model $\epsilon_\theta(x_t, t, c)$ in the update step. Finally, the output vector x_0 from the model is renormalized to ensure it represents a valid physical quantum state, enforcing unit norm and a consistent global phase.

IV. Results

In this section, we present the results of our diffusion-based approach to quantum state generation. We demonstrate the flexibility of the model across three distinct tasks. First, we show that the model can learn a specific, non-trivial distribution of quantum states by generating maximum-coherence states concentrated around the equator of the Bloch sphere. Second, we illustrate the power of conditional generation by tasking the model to produce states that match target expectation values, showing control over the properties of the generated state. Finally, we showcase the ability of the model to learn the output of a "black-box" quantum process, successfully capturing the bimodal distribution resulting from a simulated non-unital quantum channel. We begin by detailing the model's implementation and training configuration.

A. Implementation details

We represent a single-qubit pure state $|\psi\rangle = (\alpha + i\beta)|0\rangle + (\gamma + i\delta)|1\rangle$ as a four-dimensional real vector $[\alpha, \beta, \gamma, \delta]$. Our denoising network ϵ_θ is an MLP-UNet, a Multi-Layer Perceptron Autoencoder with skip connections [15]. Each block in the network concatenates the current hidden vector with a learned linear time embedding and an optional context embedding derived from user-provided targets.

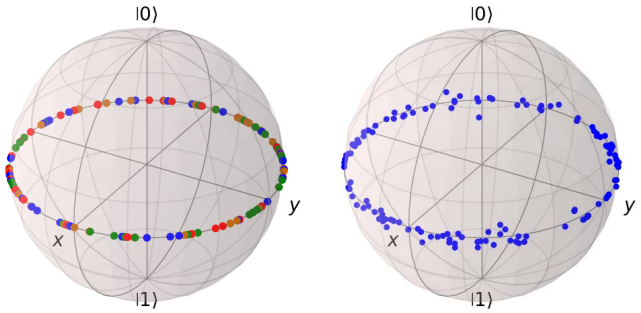
The training loss is the mean-squared error between the true and predicted Gaussian noise. We use a linear noise schedule with $\beta_{\text{start}} = 10^{-4}$ and $\beta_{\text{end}} = 10^{-2}$ over $T = 1000$ diffusion steps, unless specified otherwise. The model is trained using the Adam optimizer [16] with a learning rate of 10^{-4} and a batch size of 128. Fidelity between the generated and target distributions is assessed qualitatively by visualizing the states on the Bloch sphere. For a quantitative metric, we compute the Kullback-Leibler (KL) divergence between the distributions of pairwise fidelities for the training and generated datasets [17]. This procedure is as follows: first, for each set of states (training and generated), we compute the quantum fidelity $F(\rho_i, \rho_k)$ for all unique pairs of states within that set. The resulting two collections of fidelity values are then binned to create normalized probability distributions. The KL divergence is then calculated between these two distributions. A low KL divergence value signifies that the internal geometric structure of the generated ensemble of states closely reproduces that of the target distribution.

B. Maximum Coherent States

In this first case study, we task the model with learning the distribution of maximum-coherence states. In the computational basis, these are pure states whose Bloch vectors lie on the equator of the Bloch sphere, corresponding to an equal superposition of the $|0\rangle$ and $|1\rangle$ states. To create the training dataset, we generate states that lie exactly on the equator. This is achieved by starting with the $|0\rangle$ state and applying a sequence of rotations: a random rotation around the x-axis followed by a fixed $\pi/2$ rotation around the y-axis. This procedure produces a dataset of states that are uniformly distributed in azimuth ϕ with a fixed polar angle $\theta = \pi/2$, as depicted in the figure 1a.

The diffusion model is then trained on this dataset in an unconditional manner, without explicit context provided to the network during training. Upon sampling from the trained model, we observe that the generated states are sharply concentrated around the Bloch sphere's equator. This is confirmed visually by plotting the Bloch vectors of the generated states and quantitatively by calculating the L_1-norm of coherence for each state, which is consistently close to its maximum value of 1. This result demonstrates the model's ability to learn and faithfully reproduce a highly specific, non-trivial distribution of quantum states without direct supervision.

Figure 5b shows the training results. the training outcomes demonstrate an average coherence of 0.9992, accompanied by a standard deviation of 0.0000022 among the samples, indicateing the network's substantial capacity to generate coherent states. A comparison of the fidelities was conducted using



(a) (b)

Fig. 1: Bloch sphere representation of the quantum coherent states used as training data (a) and generated by the trained diffusion model (b).

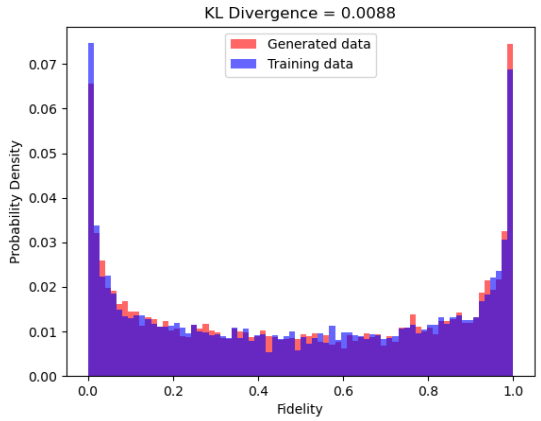


Fig. 2: Distribution of pairwise fidelities for training (blue) and generated (red) datasets. KL divergence = 0.0088.

the KL divergence metric, as depicted in Figure 2. The result obtained was a low divergence of 0.0088. This value highlights the proximity of the training data to the data generated by the network, thereby demonstrating effective task learning.

C. Generating States with Specified Expectation Values

We next demonstrate the capacity of the model for conditional generation. In this scenario, the model is tasked with generating a pure state $|\psi\rangle$ whose Pauli expectation values match a specified target Bloch vector $v = (\langle\sigma_x\rangle, \langle\sigma_y\rangle, \langle\sigma_z\rangle)$. This requires the model to learn the relationship between the physical properties of the state and its representation.

To achieve this, we modify the training process to include conditioning. The training dataset is composed of pairs: a pure state vector (sampled uniformly from the surface of the Bloch sphere) and its corresponding Bloch vector, which serves as a context vector c. During training, the denoising network $\epsilon_\theta(x_t, t, c)$ receives this context vector as an additional input at every diffusion step. This guides the network to learn how to reverse the noise process in a way that is dependent on the target expectation values.

At the time of sampling, any normalized 3D vector as the target condition can be provided to the model. The model then performs the reverse diffusion process, starting from random Gaussian noise and conditioning each denoising step on this target vector. The model successfully generates states whose calculated Bloch vectors are tightly clustered around the provided targets. We verify this by directly comparing the expectation values of the final generated states against the input target vectors.

Batch Size		X	Y	Z
128	Min. val.	0.00028	0.00079	0.00025
	Max. Val.	0.23918	0.15103	0.20432
1024	Min. val.	0.00005	0.00016	0.00038
	Max. Val.	0.05895	0.05826	0.06071

TABLE I: Smallest and largest absolute difference between the target and obtained expected value for 100 states generated from backward diffusion process. Targeting expectation values sampled from a normal distribution (with Bloch vector normalized after sampled). For the forward and backward process we use $T = 1000$. For the training, the number of epochs was 10^5 and we used two numbers for the batch size: 128 and 1024.

Table I presents the maximum and minimum differences between the target and obtained expectation values for σ_x, σ_y, and σ_z. The target values were sampled from a normal distribution, with the Bloch vector normalized before sampling. We compare two training batch sizes: 128 and 1024, with $T = 10^3$ and 10^5 epochs, respectively. Figure 3 displays the data as violin plots.

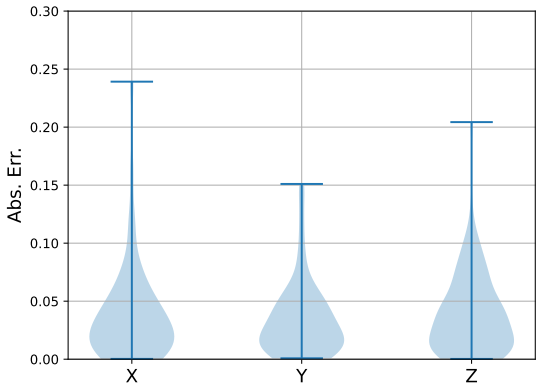
Our results clarify that increasing the batch size yields more accurate generated states. The 1024 batch size configuration achieved superior performance, with the largest difference being $6.1 \cdot 10^{-3}$ and the smallest only $5.0 \cdot 10^{-5}$. While the 128 batch size case showed poorer worst-case performance (maximum difference of 2.4×10^{-2}), its best result achieved a small difference of $2.5 \cdot 10^{-4}$. The violin plots in Figure 3 further reveal that high-accuracy states occur more frequently than low-accuracy ones.

The results show that careful adjustment of the number of training epochs and steps is crucial, as both insufficient and excessive values for any of them can reduce accuracy. Furthermore, we verify that increasing the batch size improves solution quality. Finally, these results demonstrate effective and precise control over the generative process, enabling the on-demand creation of quantum states with desired physical properties.

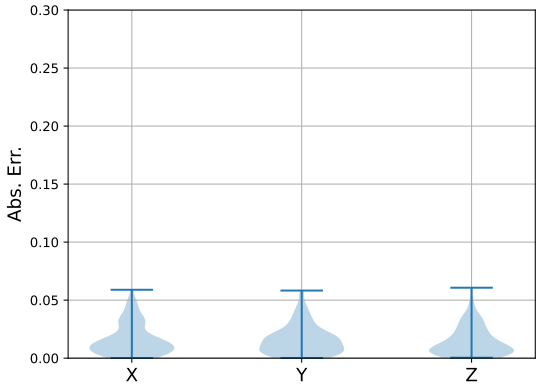
D. Black-box

To test the ability of the model to learn the statistical signature of an unknown quantum process, we simulate a "black box" designed to produce a complex, bimodal distribution of quantum states. This scenario is analogous to the practical task of quantum process tomography, where one aims to characterize the dynamics of a device by observing its effect on a set of input states.

Our simulated black box mimics a non-unital quantum channel that concentrates initial states toward two antipodal points on the Bloch sphere. The process is constructed as follows: for a batch of initial states sampled uniformly from



(a)



(b)

Fig. 3: Violin plot for the absolute difference between the target and obtained expected value for 100 states generated from backward diffusion process with training batch size of 128(a) and 1024(b) and number of epochs equal to 10^5. Targeting expectation values sampled from a normal distribution (with Bloch vector normalized after sampled). For the forward and backward process we use $T = 1000$.

the Bloch sphere, we first probabilistically divide them into two groups. Each group is assigned a target pole (the north pole $|0\rangle$ and the south pole $|1\rangle$). The states in each group are then evolved by a process that first rotates the Bloch sphere to align the target pole with the $+\hat{z}$ axis, then applies a stochastic unraveling of an amplitude damping channel. The amplitude damping channel is described by the Kraus operators:

$$K_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \qquad K_1 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}, \tag{5}$$

with γ representing the amplitude damping strength. Instead of computing the average mixed-state output, we simulate a single quantum trajectory for each input state. A "no-jump" event (application of a normalized K_0) or a "jump" event (application of a normalized K_1) is stochastically chosen based on its respective probability. This ensures that a pure input state evolves into a pure output state. After the damping event,

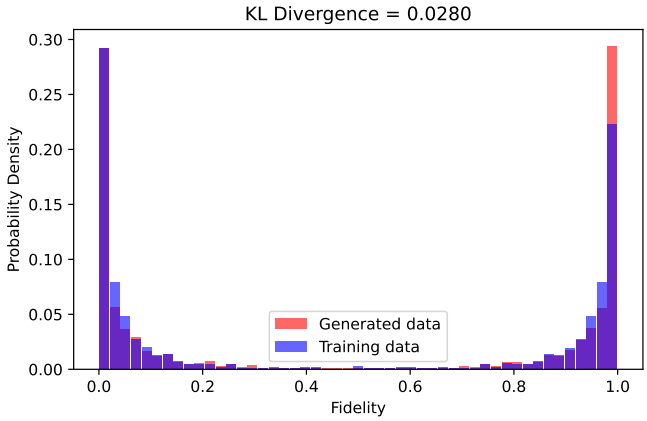


Fig. 4: Distribution of pairwise fidelities for training (blue) and generated (red) datasets. KL divergence = 0.0280.

a final rotation is applied to return to the original frame. This entire process results in a dataset of pure states heavily concentrated in two antipodal caps on the Bloch sphere.

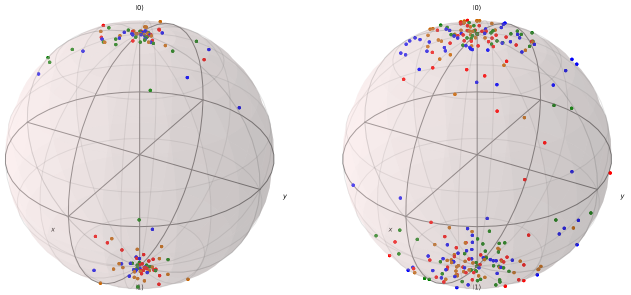
The diffusion model was trained unconditionally on this bimodal dataset, with no information provided about the underlying black-box dynamics. Upon sampling, the model successfully reproduces the two-lobed distribution, generating states clustered around the two poles. This demonstrates the capacity of the model to learn complex and physically-motivated distributions arising from unknown quantum dynamics, highlighting its potential as a tool for characterizing and simulating quantum processes.

To quantitatively evaluate the fidelity of the generated quantum state distribution in the black-box task, we computed the KL divergence between the distributions of pairwise fidelities in the training set and the samples generated by the trained diffusion model. As shown in Fig. 4, the KL divergence between these two distributions is 0.0280, indicating a close match. This low divergence value suggests that the relationship among the generated states closely replicates those of the target distribution, providing strong evidence that the diffusion model has learned the underlying structure of the black-box output.

Figure 5a illustrates the Bloch sphere distribution of the training data obtained from the simulated stochastic quantum process, and the generated distribution (Figure 5b), which exhibits a similar bimodal structure, with states clustering prominently around the same two poles. Importantly, these generated states are novel samples produced by the trained diffusion model from isotropic Gaussian noise and were not seen during training. This confirms that the model does not merely memorize training examples but successfully generalizes to new latent inputs, capturing the complex, multimodal structure of the black-box process's output. The close alignment of both the geometric features and pairwise fidelities between the training and generated datasets reinforces the model's capacity to learn a distribution that faithfully reproduces the stochastic output statistics of an unknown quantum transformation.

We emphasize that the current approach models the

marginal output distribution of the black-box process, without conditioning on specific input states. While this does not constitute full quantum process tomography, it is appropriate in settings where the process is either not directly controllable or where only output sampling is available. In such scenarios, accurately capturing the statistical structure of the output ensemble is valuable in its own right, for instance, for simulating device behavior or validating physical models from empirical data.



(a) (b)

Fig. 5: Bloch sphere representation of the quantum states used as training data (a) and generated by the trained diffusion model (b).

V. Conclusions

We have demonstrated that denoising diffusion probabilistic models are a simple and effective tool for learning and sampling distributions of single-qubit pure states. By representing qubit states as four-dimensional real vectors and applying a post-sampling canonicalization step, we can train stable denoisers capable of generating high-fidelity quantum state distributions.

The ability to condition the generation process on physical properties, such as Bloch expectation values, is a powerful feature that enables targeted state generation. Furthermore, our results show that these models can effectively learn the output statistics of black-box quantum processes, suggesting their potential use in quantum process characterization and verification.

This framework can be extended to multi-qubit systems by modeling the joint probability distribution of tensor-product states or by developing more sophisticated representations that capture entanglement properties. Future work could explore more scalable network architectures, the generation of mixed quantum states, and the integration of these models into experimental workflows for quantum device characterization and control.

Data availability

The code for reproducing the results presented in this work is available at `github.com/tiago939/CDDM_QSG`.

Acknowledgments

This work was supported by the São Paulo Research Foundation (FAPESP), under Grants No. 2023/15739-3, No. 2023/18240-0, No. 2022/00209-6, No. 2023/14831-3, No. 2023/15739-3 and No. 2025/03962-5, the Coordination for the Improvement of Higher Education Personnel (CAPES) - Finance Code 001, and the National Council for Scientific and Technological Development (CNPq) with Grants No. 311612/2021-0 and No. 140467/2022-0.

References

[1] Jun Gao, Lu-Feng Qiao, Zhi-Qiang Jiao, Yue-Chi Ma, Cheng-Qiu Hu, Ruo-Jing Ren, Ai-Lin Yang, Hao Tang, Man-Hong Yung, and Xian-Min Jin. Experimental machine learning of quantum states. *Phys. Rev. Lett.*, 120:240501, Jun 2018.

[2] Hailan Ma, Daoyi Dong, Ian R. Petersen, Chang-Jiang Huang, and Guo-Yong Xiang. Neural networks for quantum state tomography with constrained measurements. *Quantum Information Processing*, 23(9):317, September 2024.

[3] Élie Genois, Jonathan A. Gross, Agustin Di Paolo, Noah J. Stevenson, Gerwin Koolstra, Akel Hashim, Irfan Siddiqi, and Alexandre Blais. Quantum-tailored machine-learning characterization of a superconducting qubit. *PRX Quantum*, 2:040355, Dec 2021.

[4] Hao-Kai Zhang, Chenghong Zhu, Mingrui Jing, and Xin Wang. Statistical analysis of quantum state learning process in quantum neural networks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 33133–33160. Curran Associates, Inc., 2023.

[5] Juan Carrasquilla, Giacomo Torlai, Roger G. Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, March 2019.

[6] Xuegang Li, Wenjie Jiang, Ziyue Hua, Weiting Wang, Xiaoxuan Pan, Weizhou Cai, Zhide Lu, Jiaxiu Han, Rebing Wu, Chang-Ling Zou, Dong-Ling Deng, and Luyan Sun. Experimental demonstration of reconstructing quantum states with generative models. *Science Bulletin*, 70(10):1572–1575, May 2025.

[7] Bingzhi Zhang, Peng Xu, Xiaohui Chen, and Quntao Zhuang. Generative quantum machine learning via denoising diffusion probabilistic models. *Phys. Rev. Lett.*, 132:100602, Mar 2024.

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020.

[9] Hiroyuki Tezuka, Shumpei Uno, and Naoki Yamamoto. Generative model for learning quantum ensemble with optimal transport loss. *Quantum Machine Intelligence*, 6(1):6, January 2024.

[10] Christa Zoufal. Generative quantum machine learning. *arXiv:2111.12738*, 2021.

[11] Manuel S. Rudolph, Sacha Lerch, Supanut Thanasilp, Oriel Kiss, Oxana Shaya, Sofia Vallecorsa, Michele Grossi, and Zoë Holmes. Trainability barriers and opportunities in quantum generative modeling. *npj Quantum Information*, 10(1):1–18, November 2024.

[12] Gino Kwun, Bingzhi Zhang, and Quntao Zhuang. Mixed-state quantum denoising diffusion probabilistic model. *Phys. Rev. A*, 111:032610, Mar 2025.

[13] Glauco Lima, Ernestas Filatovas, Marco Marcozzi, and Remigijus Paulavičius. A review of quantum-based diffusion models in generative ai. *Vilnius University Open Series*, page 109–120, May 2025.

[14] Yehui Tang, Mabiao Long, and Junchi Yan. Quadim: A conditional diffusion model for quantum state property estimation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[15] Weihao Weng and Xin Zhu. Inet: Convolutional networks for biomedical image segmentation. *IEEE Access*, 9:16591–16603, 2021.

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2017.

[17] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.