
Quantum State Generation with Structure-Preserving Diffusion Model

Yuchen Zhu¹ Tianrong Chen¹ Evangelos A. Theodorou¹ Xie Chen² Molei Tao¹

Abstract

This article considers the generative modeling of the states of quantum systems, and an approach based on denoising diffusion model is proposed. The key contribution is an algorithmic innovation that respects the physical nature of quantum states. More precisely, the commonly used density matrix representation of mixed-state has to be complex-valued Hermitian, positive semi-definite, and trace one. Generic diffusion models, or other generative methods, may not be able to generate data that strictly satisfy these structural constraints, even if all training data do. To develop a machine learning algorithm that has physics hard-wired in, we leverage the recent development of Mirror Diffusion Model and design a previously-unconsidered mirror map, to enable strict structure-preserving generation. Both unconditional generation and conditional generation via classifier-free guidance are experimentally demonstrated efficacious, the latter even enabling the design of new quantum states when generated on unseen labels.

1. Introduction

Generative modeling, a specialized subfield of machine learning, is dedicated to creating models capable of generating new data points that closely match the distribution of a given dataset. These models adeptly capture and replicate the patterns, structures, and characteristics of the input data, allowing them to yield new instances that seem to originate from the same distribution. Among them, diffusion model (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021) has risen to prominence for its remarkable ability to produce high-quality data, especially notable in image generation (Rombach et al., 2022; Ho et al., 2022). The fundamental mechanism of diffusion model starts with adding noise to the data and progressively altering it into a tractable Gaussian prior with analytically available transition kernels. The model then effectively reverses this

noise addition, reconstructing the data from its noised state through the application of learned score functions (Anderson, 1982; Haussmann & Pardoux, 1986). This process not only results in easily obtainable conditional score functions but also simplifies regression objectives, significantly boosting their capability in managing complex, high-dimensional data distributions.

Diffusion model has showcased impressive capabilities in generating various types of dataset across different application area, such as video synthesis (Ge et al., 2023; Blattmann et al., 2023), language modeling (Austin et al., 2021; Lou et al., 2023) and point cloud generation (Zeng et al., 2022; Luo & Hu, 2021). With the recent rise of the AI4Science trend, the potential of diffusion model in tackling scientific problems also started being explored, including applications in biology (Jumper et al., 2021; Watson et al., 2022), chemistry (Duan et al., 2023; Hoogetboom et al., 2022), and climate science (Mardani et al., 2024). Other important notions relevant to this article include constrained generations (e.g., Liu et al., 2023; Fishman et al., 2023b;a; Lou & Ermon, 2023) and conditional generations (e.g., Dhariwal & Nichol, 2021; Ho & Salimans, 2022).

What is under-studied but important is the generative modeling of quantum data. This is not the same as quantum computing or quantum machine learning, because the goal is not to design quantum algorithms, but to develop algorithms that run on classical computers for handling quantum data. Similar to the already extremely successful application of the generation (and thus design) of molecular configurations (e.g., Watson et al., 2023; Duan et al., 2023), it would be useful to generate, for example, the states of quantum systems.

To be able to represent quantum states, which in general can be mixed states, one needs to be mindful of their structural constraints that originated from physics. For example, it is standard in quantum information and quantum computing to represent mixed states as complex-valued density matrices (Nielsen & Chuang, 2010), but such matrices have to be Hermitian, positive semidefinite, and trace one. Data-based generation of quantum density matrices therefore amounts to generative modeling in non-Euclidean spaces. Without strictly satisfying the structural constraints, the generated matrices will no longer correspond to quantum states. Un-

¹Georgia Institute of Technology ²California Institute of Technology. Correspondence to: Molei Tao <mtao@gatech.edu>.

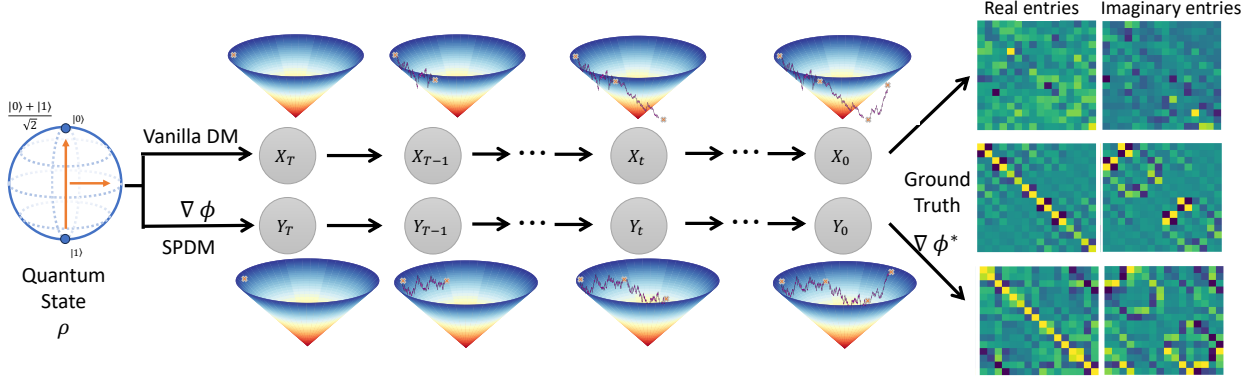


Figure 1. Structure-Preserving Diffusion Model (SPDM) is the first diffusion-based method for quantum state generation. It hard codes physical knowledge into the generative models to strictly satisfy the structural conditions of quantum states (lower path; dynamics inside the constraint cone). In contrast, vanilla Diffusion model (DM) falls short of generating reliable samples due to its failure to detect the sophisticated structures of quantum states in high dimensions (upper path; dynamics ignores the constraint cone, leading to unsatisfactory generation quality).

fortunately, generic generative modeling approaches will almost surely violate the constraints due to indispensable errors originated from finite data and compute. In fact, a seminal work (Carrasquilla et al., 2019) considered the unconditional generative modeling of quantum states using earlier machine learning tools (RBM and RNN), but errors may render density matrices reproduced from the generated ‘states’ no longer positive¹. This article proposes the first machine learning approach that generates density matrices similar to training data and strictly satisfies their structural constraints, based on hard-wiring these constraints into diffusion models via technique from convex optimization.

It should also be mentioned that machine learning has already demonstrated promising for quantum problems, and is thus rapidly becoming popular in the quantum domain. Some major applications accomplished so far included the classification and regression of quantum states (e.g., Huang et al., 2020; 2022), and quantum state tomography (e.g., Torlai et al., 2018), which, roughly speaking, corresponds to the process of estimating some latent quantum state through repeated, but necessarily noisy measurements of many copies of the same state. The generative modeling of quantum systems, on the other hand, is still in its infancy. In fact, one may ask, what is the point of generating more quantum states that are statistically like the quantum states already existing in a training data set?

The seminal paper by Carrasquilla et al. (2019) provided an answer, namely, it can facilitate state tomography. The idea is, that physical measurements are expensive, and generative modeling could help obtain more virtual ‘measurements’

¹This was noted, e.g., by Huang et al. (2020) for an analogous reconstruction procedure considered in their work; their discussion was not commenting on (Carrasquilla et al., 2019) though.

from physical ones, hence improving the accuracy of the estimated latent state. However, the power of generative modeling can go much beyond this. For example, it can provide an estimation of the statistical distribution of states in the data, hence providing further quantitative insights for constraining the latent state and probing the physical process, if the goal is state tomography. In fact, some consideration of this aspect was already included in (Carrasquilla et al., 2019) too. However, diffusion model as a more recent generative approach is posited to be particularly effective for estimating density near the data manifold, which means such an estimation is efficacious even if each datum is high-dimensional while there aren’t (comparatively) many data points, as long as there is some low dimensional structure in the data. This is oftentimes the case for image and video generations, and their empirical successes were actually the root of this belief. We anticipate quantum state data to be similar; in fact, they are high-dimensional since dimension grows exponentially with the number of qubits, but scarce due to the cost of measurements, and this is a case where diffusion generative model could be advantageous over traditional density estimation approaches. Meanwhile, note density estimation via diffusion model is standard and not our innovation. Therefore, to make a new contribution, this article will instead focus on another benefit of (diffusion) generative modeling, namely conditional generation, which can even lead to the creation of new quantum states.

More precisely, given training data that carry different labels (e.g., traditionally labels could be ‘astronaut’, ‘horse’, etc., but in our case they can be ‘insulating’, ‘magnetic’, ‘topological’, etc.), we can not only generate under a label that already has training data (e.g., ‘astronaut’, or ‘magnetic’), but also generate under a label that has no training data (e.g., the famous ‘horse-riding astronaut’ (Heaven, 2022),

or, this time, a motivating perspective of creating ‘topological magnetic’ states). This article humbly starts a first step toward this goal of scientific innovation, essentially via the interpolation/extrapolation of quantum data. The inter-/extra-polation, however, is in the sense of probability distributions but not sample points, because when one tries to design new states, it may not make sense to take, e.g., a specific ‘topological’ state and a specific ‘magnetic’ state and force a new ‘topological magnetic’ state based on them two alone. Instead, the entire probability distributions are needed, and that is how our conditional generative modeling approach becomes useful.

2. Background

2.1. Unconditional Generation via Diffusion Model

Denoising diffusion generative models in Euclidean space admit many descriptions, with focuses on different perspectives, and here we adopt the stochastic differential equations (SDE) description (Song et al., 2021). Given samples of \mathbb{R}^d -valued random variable x_0 that follows the data distribution p_0 , denoising diffusion adopts a forward noising process followed by a backward denoising generation process to generate more samples of p_0 .

The forward process transports the (unknown latent) data distribution to a known, easy-to-sample distribution by evolving the initial condition via an SDE,

$$dx_t = f(x_t, t)dt + g(t)dw_t. \quad (1)$$

The evolution of the distribution’s density p_t , given by $x_t \sim p_t(\cdot)$, can be characterized by a partial differential equation (PDE) known as the Fokker-Planck Equation (FPE),

$$\frac{\partial}{\partial t}p_t(x) = -\text{div}(p_t(x)f(x, t)) + \frac{g(t)^2}{2}\Delta p_t(x).$$

Upon certain choices of the drift f and diffusion coefficient g , the solution to the FPE will approach some limiting distribution. For example, $f(x, t) = -x$, $g(t) = \sqrt{2}$ corresponds to the well-known Variance-Preserving (VP) scheme, also known as Ornstein–Uhlenbeck process. In this case, p_∞ will be a standard Gaussian $\mathcal{N}(0, I)$.

The backward process then utilizes the time-reversal of the SDE (1) (Anderson, 1982). More precisely, if one considers

$$dy_\tau = (-f(y_\tau, T - \tau) + g(T - \tau)^2 s(y_\tau, T - \tau))d\tau + g(T - \tau)dw_\tau, \quad (2)$$

with $y_0 \sim p_T$, where s , known as the score function, is given by

$$s(x, t) = \nabla \log p_t(x),$$

then we have $y_\tau \sim p_{T-\tau}$, i.e. $y_\tau = x_{T-\tau}$ in distribution. In particular, the T -time evolution of (2), y_T , will follow the data distribution p_0 .

In practice, one considers evolving the forward dynamics for finite but large time T , so that $p_T \approx \mathcal{N}(0, \sigma_T^2 I)$, and then initialize the backward dynamics using $y_0 \sim \mathcal{N}(0, \sigma_T^2 I)$ and simulate it numerically till $\tau = T$ to obtain approximate samples of the data distribution. Critically, the score function s needs to be estimated in the forward process.

To do so, the score $s(x, t) = \nabla \log p_t(x)$ is often approximated using a neural network $s_\theta(x, t)$. For linear forward SDE, it is typically trained by minimizing an objective based on denoising score matching (Vincent, 2011), namely

$$\mathbb{E}_t \mathbb{E}_{x_0 \sim p_0} \mathbb{E}_{x_t \sim p_t(\cdot|x_0)} \|s_\theta(x_t, t) - \nabla \log p_t(x_t|x_0)\|^2$$

where $\nabla \log p_t(x_t|x_0)$ is the conditional score derived from the solution of (1) with a given initial condition.

2.2. Diffusion Guidance for Conditional Generation

For the task of conditional generation, we have an additional input y which is often a class label (e.g., a text sequence), and the goal is to sample from the conditional distribution $p(x|y)$ given training data $\{x_i, y_i\}_i$. For diffusion model, this means that we need the conditional score function $\nabla \log p(\cdot|y)$. After applying Bayes’ rule, it’s clear that we can decompose the conditional score into two parts,

$$\nabla \log p(x|y) = \nabla \log p(x) + \nabla \log p(y|x),$$

where $\nabla \log p(x)$ is the usual score function of the data distribution, and $\nabla \log p(y|x)$ is the gradient of the conditional probability of the addition input being y . $\nabla \log p(x)$ can be learned by training a diffusion model on unconditional data, and we need an estimator of $\nabla \log p(y|x)$ to generate samples from the conditional distribution $p(\cdot|y)$. Noticed that $p(y|x)$ can be approximated by training a discriminative model such as a classifier based on the data-label pair (x, y) . In practice, practitioners often use pre-trained classifier models to estimate $\nabla \log p(y|x)$.

Dhariwal & Nichol (2021) exploited this fact and proposed the technique of classifier-based diffusion guidance, originally for boosting the sample quality generated by diffusion models. Instead of $p(x|y)$, they proposed to sample from a condition-enhanced distribution,

$$p_\gamma(x|y) \propto p(x)p(y|x)^\gamma.$$

The score of this condition-enhanced distribution can be computed by

$$\nabla \log p_\gamma(x|y) = \nabla \log p(x) + \gamma \nabla \log p(y|x).$$

γ is called the strength of the guidance, which amplifies the influence of the conditioning when setting to a scale larger than 1. When $\gamma > 1$, the distribution is sharpened and focused onto its mode that corresponds to the condition

y (Dieleman, 2022). By tuning γ , classifier guidance allows us to capture the influence of the condition signal.

However, classifier-based diffusion guidance can become impractical due to the expensive cost of training a separate classifier model. Such a classifier often requires to be noise-robust to handle the noise-corrupted input x during the sampling process. Ho & Salimans (2022) proposed classifier-free diffusion guidance to sample from the condition-enhanced distribution $p_\gamma(x|y)$ without the need for an extra discriminative model. Their insight came from another writing of Bayes' rule: $\nabla \log p_\gamma(x|y)$ can also be decomposed as,

$$\nabla \log p_\gamma(x|y) = (1 - \gamma)\nabla \log p(x) + \gamma\nabla \log p(x|y).$$

Therefore, classifier-free diffusion guidance motivates the training of a model that functions both as the conditional score $\nabla \log p(x|y)$ and the unconditional score $\nabla \log p(x)$, depending on whether an additional condition input y is given. We will use classifier-free guidance due to improved performance and computational cost.

2.3. Density Matrix Representation of Quantum State

Since we will consider the unconditional and conditional generations of quantum states, let's review some of their basics. A quantum state describes the physical status of a quantum system. For generality and with some motivations being quantum computing and quantum information in mind, we consider mixed states, which can be represented by density matrices (in theory, they are operators, but once we fix the bases of the Hilbert space, they become finite dimensional objects).

Precisely, a (quantum) density matrix is a matrix with each element being a **complex** number, which satisfies the following structural constraints (e.g., Nielsen & Chuang, 2010):

It is Hermitian, with trace 1, and positive semi-definite.

Notations: Here we denote the set of complex Hermitian matrices of dimension n as \mathbb{C}_n ,

$$\mathbb{C}_n = \{X \in \mathbb{C}^{n \times n}, X = X^\dagger\}, \quad (3)$$

where X^\dagger is the conjugate transpose of X , defined through $(X^\dagger)_{ij} = \overline{X_{ji}}$. We also denoted the set of positive definite complex Hermitian matrices of dimension n as \mathbb{C}_n^+

$$\mathbb{C}_n^+ = \{X \in \mathbb{C}_n, X \succ 0\}. \quad (4)$$

Definition 2.1. A n -qubit quantum state ρ is an element of $\mathbb{C}_{2^n}^+$ with trace 1.

Because of the structural constraints, the generative modeling of quantum states is nontrivial but an interesting machine-learning problem. Ideally speaking, if there were

an infinite amount of data and infinite computational resources (so that generative modeling can be exact), the training data already implicitly carry information about the structural constraints, and then the generated data will also respect the constraints, thus yielding true quantum states. Nevertheless, in practice the generation is plagued with various sources of error (e.g., statistical error of finite samples, (score) function approximation error, finite-time approximation of $T \rightarrow \infty$, numerical integration of the backward process, etc.) and generic generative modeling approaches that are not built to respect those constraints will likely generate samples that violate them, and the samples will no longer possess physical meaning and are hence useless. Therefore, in the following, we hardwire these important physical knowledge into our generative model, so that the constraints will be **exactly** satisfied.

3. Method

Our task is to generate new samples of quantum mixed states, from a distribution characterized by the training data. Out of the three structure constraints of density matrix representation of quantum states, Hermitianity is easy to handle as one can just work with the upper triangular part of the matrix. Trace one condition will only be minimally violated due to generation error, and we simply normalize the generated result by dividing its trace to strictly enforce this constraint. The most nontrivial constraint is the positive semi-definiteness. Since this is a convex constraint, we can leverage the recent approach of Mirror Diffusion Model (MDM) (Liu et al., 2023), which is a class of diffusion models that generate data on convex-constrained sets, to enable exact generation on \mathbb{C}_n^+ . To achieve this goal, we work out a complex and multivariate extension of the results in MDM.

3.1. Mirror Map with Real Numbers

More precisely, the approach of MDM relies on a tool from convex optimization known as mirror map. It can create a nonlinear bijection that maps the data from the constrained primal space to an unconstrained dual space, which is Euclidean. This bijection serves as a coordinate transformation, i.e. a pair of exact encoder and decoder. A mirror map $\nabla\phi$ is defined as follows:

Definition 3.1. Given a convex constrained set, $\mathcal{M} \subseteq \mathbb{R}^d$, $\nabla\phi$ is called mirror map if $\phi : \mathcal{M} \rightarrow \mathbb{R}$ is a twice differentiable function that is strictly convex and satisfying:

$$\nabla\phi(\mathcal{M}) = \mathbb{R}^d \quad \text{and} \quad \lim_{x \rightarrow \partial\mathcal{M}} \|\nabla\phi\| \rightarrow \infty.$$

$\nabla\phi : \mathcal{M} \rightarrow \mathbb{R}^d$ is an invertible map from the constrained primal space \mathcal{M} to an unconstrained dual space $\nabla\phi(\mathcal{M})$, which is also called the mirror space.

Its inverse enjoys a pleasant characterization from convex analysis (Rockafellar, 2015). Namely, by defining the dual

function of ϕ as $\phi^*(y) = \sup_{x \in \mathcal{M}} \langle x, y \rangle - \phi(x)$, the inverse (as a mapping) of $\nabla \phi$ is the same as the gradient of ϕ^* , i.e.

$$\begin{aligned}\nabla \phi^*(\nabla \phi(x)) &= x, \quad \forall x \in \mathcal{M}, \\ \nabla \phi(\nabla \phi^*(y)) &= y, \quad \forall y \in \mathbb{R}^d.\end{aligned}$$

Therefore, $(\nabla \phi, \nabla \phi^*)$ acts as a pair of encoder-decoder that transform each datum between the primal constrained space \mathcal{M} and the unconstrained mirror space $\nabla \phi(\mathcal{M})$.

3.2. Mirror Map for Complex-Valued Density Matrices

A complication is that our data consists of complex-valued matrices, different from real vectors and matrices typically considered in convex optimization. Therefore, what remains to be described is the construction of a strictly convex potential ϕ for the class of positive definite Hermitian matrices. It turns out that the negative von Neumann entropy satisfies the need.

The von Neumann entropy (e.g., Bengtsson & Życzkowski, 2017) is an extension of the concept of Shannon entropy to quantum states. For $X \in \mathbb{C}_n^+$, the negative von Neumann entropy is defined as,

$$\phi(X) = \text{Tr}(X \log X). \quad (5)$$

The mirror map and its inverse are computed analytically as

$$\nabla \phi(X) = I + \log X, \quad \nabla \phi^*(Y) = \exp(Y - I), \quad (6)$$

where \log and \exp are matrix logarithm and matrix exponential. Specifically, we define the matrix logarithm $\log X$ to be $\log X = \sum_i (\log \lambda_i) q_i q_i^\dagger$ if X admits spectral decomposition $X = \sum_i \lambda_i q_i q_i^\dagger$, where $\lambda_i > 0$ are positive real numbers, $\{q_i\}$ are a set of complex-valued vectors that are also orthonormal basis of \mathbb{C}^n .

Proposition 3.2. $(\nabla \phi, \nabla \phi^*)$ given by (6) is a pair of bijective mappings between the set of positive definite complex Hermitian matrices \mathbb{C}_n^+ and the set of complex Hermitian matrices \mathbb{C}_n . Moreover, $\nabla \phi^*$ is the inverse of $\nabla \phi$.

Proof. Please see Appendix.A.2.

3.3. Generative Modeling in Primal Space Based on Diffusion in Dual Space

The gradient map of negative von Neumann entropy, $\nabla \phi$, allows us to transform the problem of learning and sampling the data distribution $p_{data}(X)$ over \mathbb{C}_n^+ , to that for a new distribution $\tilde{p}_{data} = ([\nabla \phi_\#]p_{data})$ over the mirror space \mathbb{C}_n , where $\#$ is the push-forward operation. Moreover, although $\mathbb{C}_n \subseteq \mathbb{C}^{n \times n}$ is still a convex-constrained set due to each element being a Hermitian matrix, \mathbb{C}_n enjoys a much simpler structure due to the fact that $\mathbb{C}_n \cong \mathbb{R}^{n^2}$, i.e. it is isomorphic to an unconstrained real Euclidean space of dimension n^2 . This is because each complex Hermitian matrix

can be represented using only its upper or lower triangular entries. Since complex Hermitian has real diagonal values, the diagonal contributes n dimension. There are $n(n-1)/2$ strict upper triangular entries, each having a complex value. Each complex entry can be represented with a point in \mathbb{R}^2 . Therefore, the total dimension is n^2 and linearity guarantees the isomorphism.

Since diffusion models leverage a neural network parameterization of the score function, and most neural networks have not been designed for complex numbers, we utilize the isomorphism mentioned above and further transform $\nabla \phi(X)$ into a vector in \mathbb{R}^{n^2} . This enables us to represent the score function based on simple Euclidean space again.

3.4. Summary of Methodology

Our approach can be summarized in the following steps. We take quantum state data in \mathbb{C}_n^+ and apply the mirror map to transform them into \mathbb{C}_n . We extract the upper triangular entries and create their real representation as a vector in the dual space \mathbb{R}^{n^2} . On the dual space, we build a diffusion model that transforms the distribution $\tilde{p}_{data} = ([\nabla \phi_\#]p_{data})$ to a Gaussian measure. We train the score neural network s_θ with a score-matching loss on the dual space.

For quantum state generation, we simulate the reverse-time SDE or marginally equivalent probability flow ODE with the learned score network s_θ to generate new samples $Y_{new} \sim \tilde{p}_{data}$, which is on \mathbb{R}^{n^2} . We compute its representation \tilde{Y}_{new} as a complex Hermitian matrices through $\tilde{Y}_{new} = \text{Isomorphism}_{\mathbb{R}^{n^2} \rightarrow \mathbb{C}_n}(Y_{new})$, where $\text{Isomorphism}_{\mathbb{R}^{n^2} \rightarrow \mathbb{C}_n}$ stands for the isomorphic transformation between the two spaces. We generate a new quantum state by applying the inverse of mirror map, $X_{new} = \nabla \phi^*(\tilde{Y}_{new})$, which approximately satisfies the desired data distribution $p_{data}(X)$.

4. Experiments

As this work is the first on the generative modeling of quantum data, there is no existing approach to compare to, although both statistical and physical means for assessing the success still exist. Moreover, no public dataset has been curated either, and we will therefore prove the concept using physical but synthetic data.

More precisely, we will consider three labeled classes of data, corresponding to quantum mixed states of a 4-qubit system with three different levels of entanglements. We will first conduct unconditional generation, from the distribution of the union of all classes. Then we will perform conditional generation, from the same union distribution but conditioned on the class label; results similar to that of individually generating from each class would be seen. Finally, we

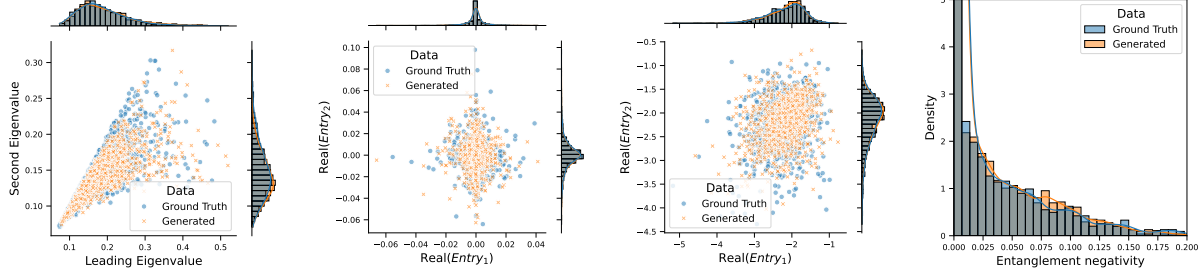


Figure 2. Unconditional generation of the mixture of three classes of quantum states. Each figure shows an observable for comparison between unconditionally generated samples and its corresponding ground truth samples. (1) Leading eigenvalue versus second eigenvalue. (2) Real value of matrix entries in the primal space (\mathbb{C}_{16}^+). (3) Real value of matrix entries in the dual space (\mathbb{C}_{16}). (4) Distribution of entanglement negativity measured between qubit 1 and the rest of the system.

will investigate conditional generation beyond the known class labels, i.e., the ability to interpolate/extrapolate across existing classes in the training data.

4.1. Problem Setup

To demonstrate the efficacy of our method, we consider an example of generating quantum states based on training data consisting of quantum states with three different levels of entanglement between qubits.

Product state. Let $\rho_1, \rho_2, \rho_3, \rho_4 \in \mathbb{C}_1^+$ be quantum states of one qubit. The product state with no entanglement for a 4-qubit system is given by

$$\rho_{\text{prod}} = \rho_1 \otimes \rho_2 \otimes \rho_3 \otimes \rho_4. \quad (7)$$

To simplify the setting, we consider $\rho_{i=1,2,3,4}$ to be sampled i.i.d. from the same distribution p_{bit} over quantum states of one qubit. Product state is a quantum state of the system that corresponds to the situation where there is no quantum entanglement between any of the two qubits, i.e., each qubit has a quantum state that can be independently described regardless of others.

Pairwisely entangled state. We now entangle qubits to create entangled state $\rho_{\text{entg},2}$ from product state ρ_{prod} . Let $U_{12}, U_{34} \in U(16)$ be complex-valued unitary matrices of dimension 16, where U_{12} corresponds to the quantum entanglement between qubit 1 and 2, U_{34} corresponds to the quantum entanglement between qubit 3 and 4. A pairwisely entangled state for the 4-qubit system is given by

$$\rho_{\text{entg},2} = (U_{12}U_{34})\rho_{\text{prod}}(U_{12}U_{34})^\dagger. \quad (8)$$

Such states correspond to a situation where there is simultaneous quantum entanglement between qubits 1 and 2, as well as qubits 3 and 4, but no entanglement between any other qubit pair. Since U_{12} and U_{34} refer to entanglement between only two qubits, the two unitary matrices are created with complex unitary matrices of dimension 4.

In this experiment, we let $M_{ij} \in \mathbb{C}^{4 \times 4}$, $1 \leq i < j \leq 4$ be unitary matrices of dimension 4, sampled from the Haar

measure of $U(4)$. Haar measure generalizes the concept of uniform distribution to compact Lie groups. U_{12}, U_{34} are then created from M_{12}, M_{34} in the following way:

$$U_{12} = M_{12} \otimes \mathbb{I}_{4 \times 4}, \quad U_{34} = \mathbb{I}_{4 \times 4} \otimes M_{34}$$

where $\mathbb{I}_{n \times n}$ is the identity matrix.

Fully entangled state. We now entangle more qubits to create further entangled state $\rho_{\text{entg},\text{all}}$ from product state ρ_{prod} . For $1 \leq i < j \leq 4$, let $U_{ij} \in U(16)$ be complex unitary matrices of dimension 16, where U_{ij} corresponds to the quantum entanglement between qubit i and j . Consider a fully entangled state for the 4-qubit system given by

$$\rho_{\text{entg},\text{all}} = \left(\prod_{i < j} U_{ij} \right) \rho_{\text{prod}} \left(\prod_{i < j} U_{ij} \right)^\dagger. \quad (9)$$

Fully entangled state is a quantum state of the system that corresponds to a situation where there is simultaneous entanglement between any pair of qubits. Each matrix U_{ij} corresponds to the entanglement between only two qubits, thus the matrix is created using complex unitary matrices of dimension 4.

U_{ij} are created from M_{ij} in the following way:

$$\begin{aligned} U_{13} &= \text{Perm}_{23}(M_{13} \otimes \mathbb{I}_{4 \times 4}), & U_{12} &= M_{12} \otimes \mathbb{I}_{4 \times 4}, \\ U_{14} &= \text{Perm}_{13}(\mathbb{I}_{4 \times 4} \otimes M_{14}), & U_{23} &= \mathbb{I}_{2 \times 2} \otimes M_{23} \otimes \mathbb{I}_{2 \times 2}, \\ U_{24} &= \text{Perm}_{23}(\mathbb{I}_{4 \times 4} \otimes M_{24}), & U_{34} &= \mathbb{I}_{4 \times 4} \otimes M_{34}, \end{aligned}$$

where Perm_{ij} permutes the rows and columns that correspond to qubit i and j , $\mathbb{I}_{n \times n}$ is the identity matrix.

4.2. Data Preparation

This section describes how we generate the three classes of quantum states, which altogether constitute our training set.

Single qubit quantum state distribution. In the generation process of product state, we consider the scenario where each qubit has a quantum state sampled i.i.d. from the same distribution p_{bit} , which is defined by the following procedure. Let $Y = Q\Lambda Q^\dagger$, where $Q \sim \text{Haar}(U(2))$, $\Lambda =$

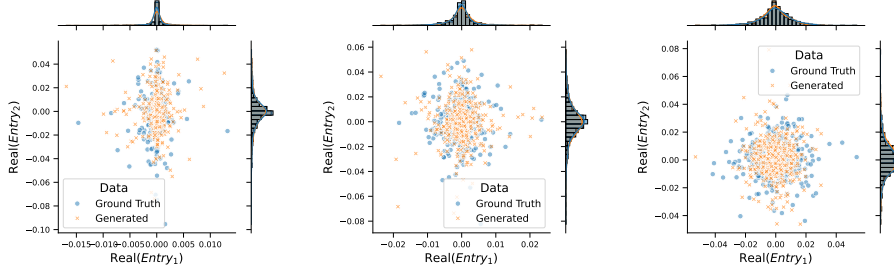


Figure 3. Conditional generation with known labels. Each figure shows a comparison between conditionally generated samples of one class and its corresponding ground truth samples in the primal space \mathbb{C}_{16}^+ . (1) Real value of matrix entries for product states. (2) Real value of matrix entries for pairwise entangled states. (3) Real value of matrix entries for fully entangled states.

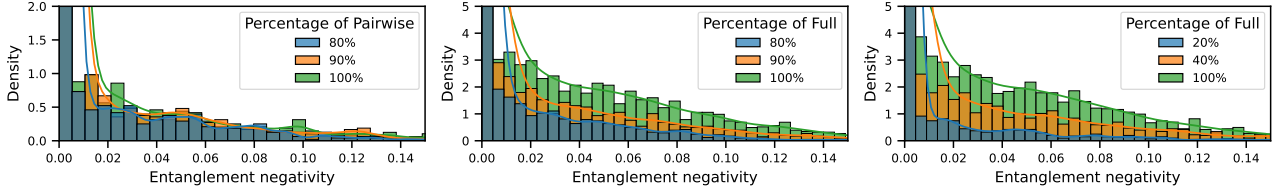


Figure 4. Conditional generation with unseen labels. Each figure represents the distribution of the entanglement negativity of samples generated with unseen labels that are convex combinations of one-hot encoding of seen labels. Entanglement negativity is measured between qubit 1 and the rest of the system. (Left) Interpolation between product state and pairwise entangled state. (Middle) Interpolation between pairwise entangled states and fully entangled states. (Right) Interpolation between product states and fully entangled states.

$\text{diag}([\lambda_1, \lambda_2])$ with $\lambda_1, \lambda_2 \sim \text{Uniform}[\lambda_{\min}, \lambda_{\max}]$. $X \in \mathbb{C}_1^+$ is computed as

$$X = \frac{1}{\text{Tr}(Y)} Y. \quad (10)$$

We consider p_{bit} to be the distribution of X generated via eq.(10). By construction, X is a positive definite Hermitian matrix with trace 1, thus a valid quantum state.

Haar measure of $U(n)$. To generate the training data for pairwise entangled and fully entangled states, we chose to sample from the Haar measure on $U(4)$ to create unbiased quantum entanglement matrices U_{ij} between qubit i and j . $U(n)$ is the set of matrices defined as,

$$U(n) = \{M \in \mathbb{C}^{n \times n}, M^\dagger M = M M^\dagger = \mathbb{I}_n\}.$$

Sampling a probability distribution on $U(n)$, such as the Haar measure, is also an interesting task. One may view it as a sampling problem on a constrained set, but note $U(n)$ is actually a non-convex constrained set, and thus common constrained sampling tools (e.g., projected Langevin (Bubeck et al., 2015), sophisticated MCMC walkers (Lovász & Simonovits, 1993; Lovász, 1999; Kannan & Narayanan, 2009; Chen et al., 2018), mirror Langevin (Zhang et al., 2020; Li et al., 2022)) may not suit well. Fortunately, due to the Lie group structure of $U(n)$, the problem can be solved by considering the dynamics,

$$\begin{cases} \dot{g} = g\xi, \\ d\xi = -\gamma\xi dt + \sqrt{2\gamma}dW_g, \end{cases} \quad (11)$$

where $g \in U(n)$, $\xi \in \mathfrak{g} = \mathfrak{u}(n)$ is an element in the Lie algebra, W_g is the Brownian motion over the vector space $\mathfrak{u}(n)$. The Lie algebra and Brownian motion is defined as,

$$\mathfrak{u}(n) = \{M \in \mathbb{C}^{n \times n}, M^\dagger = -M\}, dW_g = \sum_{i=1}^{n^2} dW_i \cdot e_i$$

where $\{e_1, \dots, e_{n^2}\}$ spans $\mathfrak{u}(n)$, dW_1, \dots, dW_{n^2} are independent one dimensional Brownian motion. One notable property of the diffusion process described in (11) is that it converges to a unique invariant distribution and the g -marginal of its invariant distribution is the Haar measure of $U(n)$ (Kong & Tao, 2024). Therefore, by numerically integrating (11), we can approximately sample uniformly from the unitary group $U(4)$ to generate the quantum entanglement matrices. This numerical integration can be done computational-efficiently with g exactly remaining on $U(4)$ via the operator splitting approach (Tao & Ohsawa, 2020).

4.3. Evaluation

In this section, we discuss the metrics and observables (since the data distribution is high dimensional) used for evaluating the sample quality of the generated quantum states. Both statistical and physical quantities will be used.

Entanglement negativity. Entanglement negativity is a non-negative measure of the amount of quantum entanglement present in a quantum state. Entanglement negativity is an entanglement monotone whose value does not increase un-

Table 1. Unconditional and conditional generation of quantum states. We report metrics computed using 3000 generated samples and ground truth samples. {Sliced Wasserstein Distance (SWD), Maximum Sliced Wasserstein Distance (MSWD), Energy Maximum Mean Discrepancy (Energy-MMD), 1-Wasserstein distance (W_1)} are computed using density matrices of generated quantum state. Negativity (W_1) is computed by measuring the W_1 distance between the distribution of entanglement negativity. More details on evaluation metrics can be found in Appendix A.4. The last row is just providing a reference for interpreting how small values mean close distributions, as it compares the training set against randomly generated density matrices (16×16 Hermitian with eigenvalues uniformly sampled from $[0, 1]$ and eigenspace matrix sampled from Haar on $U(16)$), for which metrics values should be large.

| | Class | SWD ↓ | MSWD ↓ | W_1 ↓ | Energy-MMD ↓ | Negativity(W_1) ↓ |
|----------------------------|------------|---------|---------|---------|--------------|-----------------------|
| Unconditional | Mixture | 0.00054 | 0.00094 | 0.020 | 0.000076 | 0.0000110 |
| | Product | 0.00052 | 0.00079 | 0.011 | 0.000070 | 0.0000025 |
| Conditional | Pairwisely | 0.00064 | 0.00100 | 0.018 | 0.000082 | 0.0000061 |
| | Fully | 0.00056 | 0.00092 | 0.026 | 0.000100 | 0.0000062 |
| Magnitude reference | | – | 0.0854 | 0.1554 | 1.7484 | 1.1236 |

der local operations and classical communication. Consider a composite quantum system consisting of two subsystems A and B , described by a density matrix ρ in a Hilbert space that is the tensor product of the spaces of A and B . We denote the transpose operator on the space of subsystem A as T_A . We denote the partial transpose of ρ with respect to subsystem A as ρ^{Γ_A} , which can be computed as $\rho^{\Gamma_A} = (T_A \otimes I)\rho$. Then, the entanglement negativity $\mathcal{N}(\rho)$ is defined as,

$$\mathcal{N}(\rho) = \left| \sum_{\lambda_i < 0} \lambda_i \right| = \sum_i \frac{|\lambda_i| - \lambda_i}{2}$$

where λ_i are eigenvalues of ρ^{Γ_A} . A high value of $\mathcal{N}(\rho)$ represents a high level of quantum entanglement.

Distributional-wise metric. To evaluate the similarity between high-dim probability distributions, we mainly use Sliced Wasserstein Distance (SWD) (Kolouri et al., 2019), Maximum Sliced Wasserstein Distance (MSWD) (Deshpande et al., 2019), and Maximum Mean Discrepancies (MMD) (Gretton et al., 2012), in addition to the standard Wasserstein distance, as they suffer less from the curse of dimensionality. Within the family of MMDs, we consider Energy-MMD, which is computed with kernel $k(x, y) = -\|x - y\|_2$. We also compute the 1-Wasserstein distance based on values of the physically meaningful low-dim observable of entanglement negativity.

4.4. Training

We train on a dataset of 900,000 samples, distributed equally across all three classes of quantum states. The entire dataset is first transformed to the dual space by applying the mirror map. Then a DDPM is trained on the transformed dataset. We train for 1,000,000 iterations with AdamW optimizer, using an initial learning rate of 10^{-3} and decay rate of 0.995 every 1000 AdamW steps. We summarize our neural network architecture in Appendix A.3. We adopt the same data post-processing and metrics evaluation used in (Chen et al., 2023), and we elaborate them in Appendix A.4.

4.5. Results

SPDM accurately learns the distribution of quantum states. Table 1 summarizes our quantitative results for both conditional and unconditional generation of three classes of quantum states. Figure 3.3 and Figure 4 depict the generated samples compared against ground truth. Our method manages to learn accurately the correct distribution of eigenvalues, which is extremely sensitive to all the entries in the matrix. We note that product states, pairwisely entangled states, and fully entangled states considered in this work share an identical distribution of eigenvalues since the spectrum of matrices is invariant under unitary conjugations. However, they have distinct marginal distributions for each entry due to the quantum entanglement. Our method successfully predicts the correct joint distribution of all the entries in the density matrix, which is also confirmed by small values in all distribution-based metrics listed in Table 1.

SPDM generates the correct level of quantum entanglement between qubits. Other than eigenvalues and distribution-based metrics, another assessment of generation quality is whether the generated samples demonstrate the same level of entanglement as the training data. We use entanglement negativity to capture the degree of quantum entanglement between two subsystems. As is evident from Table 1 and Figure 3.3, our method accurately generates samples with the same amount of quantum entanglement as the training data. This implies that under our framework, diffusion models manage to learn the sophisticated relationships between entangled entries and reproduce such entanglements in the generation process.

SPDM enables the design of physically-meaningful new quantum states. We generate new labels by taking a convex combination of the one-hot encodings of seen classes. Then we conditionally generate samples that correspond to these new labels with our model. Figure 4 demonstrated the quantum entanglement level of the resulting data. When

generating under labels that are mixtures of two classes with different entanglements, the model produces data with an entanglement level in the middle ground of the two classes, which is in accordance with the interpolation ratio we used in generating new labels. This implies that our method manages to learn a meaningful embedding of class labels that also reflects the interpolation of entanglement level among quantum data. The model allows an extrapolation to unseen quantum states. This means that our model is capable of designing quantum states with specific (new) entanglement level controlled by a manually created label.

5. Conclusion

We developed the first diffusion-based method for the generative modeling of quantum states. Our method hard codes physical knowledge into the generative models and generates samples that strictly satisfy the structural constraints of quantum states. We demonstrate that the model accurately learns and reproduces the distribution of quantum states represented by high dimensional density matrix, and can even create physically-meaningful quantum states of unseen level of entanglements, via conditional generation. In the future, it will be an interesting task to design neural network architectures for parametrizing the score function that specialize in quantum state generation for even better generation quality.

6. Impact Statement

This paper presents work whose goal is to advance the field of machine learning and quantum sciences. There could be potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

- Anderson, B. D. Reverse-time diffusion equation models. Stochastic Processes and their Applications, 12(3):313–326, 1982.
- Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and Van Den Berg, R. Structured denoising diffusion models in discrete state-spaces. Advances in Neural Information Processing Systems, 34:17981–17993, 2021.
- Bengtsson, I. and Życzkowski, K. Geometry of quantum states: an introduction to quantum entanglement. Cambridge university press, 2017.
- Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. Align your latents: High-resolution video synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22563–22575, 2023.
- Bubeck, S., Eldan, R., and Lehec, J. Finite-time analysis of projected langevin monte carlo. Advances in Neural Information Processing Systems, 28, 2015.
- Carrasquilla, J., Torlai, G., Melko, R. G., and Aolita, L. Reconstructing quantum states with generative models. Nature Machine Intelligence, 1(3):155–161, 2019.
- Chen, T., Liu, G.-H., Tao, M., and Theodorou, E. A. Deep momentum multi-marginal schrödinger bridge. NeurIPS, 2023.
- Chen, Y., Dwivedi, R., Wainwright, M. J., and Yu, B. Fast mcmc sampling algorithms on polytopes. The Journal of Machine Learning Research, 19(1):2146–2231, 2018.
- Deshpande, I., Hu, Y.-T., Sun, R., Pyrros, A., Siddiqui, N., Koyejo, S., Zhao, Z., Forsyth, D., and Schwing, A. G. Max-sliced wasserstein distance and its use for gans. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10648–10656, 2019.
- Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. Advances in neural information processing systems, 34:8780–8794, 2021.
- Dieleman, S. Guidance: a cheat code for diffusion models, 2022. URL <https://benanne.github.io/2022/05/26/guidance.html>.
- Duan, C., Du, Y., Jia, H., and Kulik, H. J. Accurate transition state generation with an object-aware equivariant elementary reaction diffusion model. Nature Computational Science, 2023.
- Fishman, N., Klarner, L., De Bortoli, V., Mathieu, E., and Hutchinson, M. Diffusion models for constrained domains. TMLR, 2023a.
- Fishman, N., Klarner, L., Mathieu, E., Hutchinson, M., and De Bortoli, V. Metropolis sampling for constrained diffusion models. NeurIPS, 2023b.
- Ge, S., Nah, S., Liu, G., Poon, T., Tao, A., Catanzaro, B., Jacobs, D., Huang, J.-B., Liu, M.-Y., and Balaji, Y. Preserve your own correlation: A noise prior for video diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 22930–22941, 2023.
- Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. The Journal of Machine Learning Research, 13(1):723–773, 2012.
- Hausmann, U. G. and Pardoux, E. Time reversal of diffusions. The Annals of Probability, pp. 1188–1205, 1986.

- Heaven, W. D. Horse riding astronaut, 2022. URL <https://www.technologyreview.com/2022/04/06/1049061/dalle-openai-gpt3-ai-agi-multimodal-image-generation/>.
- Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.
- Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research*, 23(1):2249–2281, 2022.
- Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.
- Huang, H.-Y., Kueng, R., and Preskill, J. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.
- Huang, H.-Y., Kueng, R., Torlai, G., Albert, V. V., and Preskill, J. Provably efficient machine learning for quantum many-body problems. *Science*, 377(6613):eabk3333, 2022.
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnoy, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
- Kannan, R. and Narayanan, H. Random walks on polytopes and an affine interior point method for linear programming. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pp. 561–570, 2009.
- Kolouri, S., Nadjahi, K., Simsekli, U., Badeau, R., and Rohde, G. Generalized sliced wasserstein distances. *Advances in neural information processing systems*, 32, 2019.
- Kong, L. and Tao, M. Convergence of kinetic langevin monte carlo on lie groups. *arXiv:2403.12012*, 2024.
- Li, R., Tao, M., Vempala, S. S., and Wibisono, A. The mirror langevin algorithm converges with vanishing bias. In *International Conference on Algorithmic Learning Theory*, pp. 718–742. PMLR, 2022.
- Liu, G.-H., Chen, T., Theodorou, E. A., and Tao, M. Mirror diffusion models for constrained and watermarked generation. *NeurIPS*, 2023.
- Lou, A. and Ermon, S. Reflected diffusion models. *ICML*, 2023.
- Lou, A., Meng, C., and Ermon, S. Discrete diffusion language modeling by estimating the ratios of the data distribution. *NeurIPS*, 2023.
- Lovász, L. Hit-and-run mixes fast. *Mathematical programming*, 86:443–461, 1999.
- Lovász, L. and Simonovits, M. Random walks in a convex body and an improved volume algorithm. *Random structures & algorithms*, 4(4):359–412, 1993.
- Luo, S. and Hu, W. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845, 2021.
- Mardani, M., Brenowitz, N., Cohen, Y., Pathak, J., Chen, C.-Y., Liu, C.-C., Vahdat, A., Kashinath, K., Kautz, J., and Pritchard, M. Residual diffusion modeling for km-scale atmospheric downscaling. 2024.
- Nielsen, M. A. and Chuang, I. L. *Quantum computation and quantum information*. Cambridge university press, 2010.
- Rockafellar, R. *Convex analysis*. Princeton landmarks in mathematics and physics. Princeton University Press, 2015.
- Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *ICML*, 2015.
- Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021.
- Tao, M. and Ohsawa, T. Variational optimization on lie groups, with examples of leading (generalized) eigenvalue problems. In *International Conference on Artificial Intelligence and Statistics*, pp. 4269–4280. PMLR, 2020.
- Torlai, G., Mazzola, G., Carrasquilla, J., Troyer, M., Melko, R., and Carleo, G. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, 2018.
- Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. Broadly applicable and accurate protein design by integrating structure prediction networks and diffusion generative models. BioRxiv, pp. 2022–12, 2022.

Watson, J. L. et al. De novo design of protein structure and function with rfdiffusion. Nature, 2023.

Zeng, X., Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., and Kreis, K. Lion: Latent point diffusion models for 3d shape generation. arXiv preprint arXiv:2210.06978, 2022.

Zhang, K. S., Peyré, G., Fadili, J., and Pereyra, M. Wasserstein control of mirror langevin monte carlo. In Conference on Learning Theory, pp. 3814–3841. PMLR, 2020.

A. Appendix

A.1. Vanilla SGM on Quantum State Dataset

In our research, we demonstrate that conventional score-based generative models (SGMs) fall short in reliably generating high-dimensional quantum datasets. Figure 5 exemplifies that, although these SGMs can precisely mimic the distribution of a 1-qubit quantum state (4 dimensions), they face significant challenges due to the curse of dimensionality when the dimensionality is expanded to a 4-qubit state (256 dimensions), as highlighted in Figure 6. This limitation likely stems from the current neural network parameterization strategies. Identifying an efficient network architecture capable of effectively capturing the essence of quantum states remains a pivotal, yet unresolved, question, which we aim to address in future work.

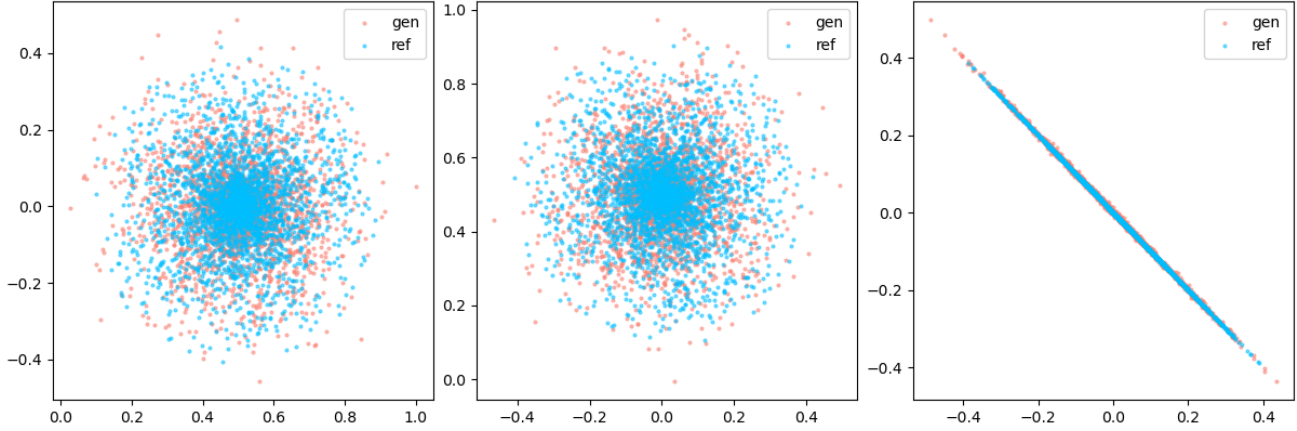


Figure 5. SGM can easily recover the distribution of 1 qubit quantum state.

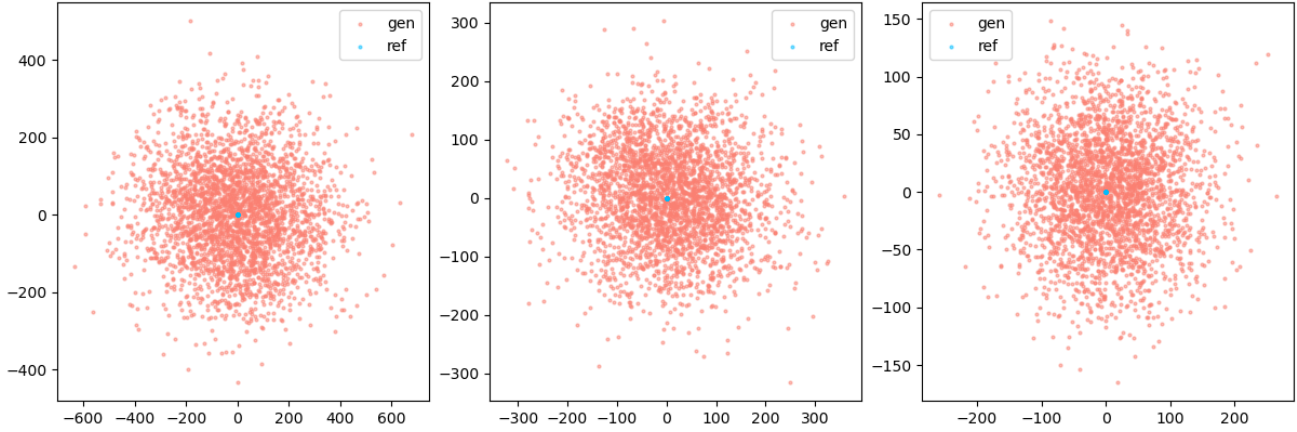


Figure 6. When the dimension increases, SGM cannot easily recover the distribution of the 4 qubit quantum state due to the complex data structure.

A.2. Proof of Proposition.3.2

Proof. We first show that $\nabla\phi$ is a mapping from \mathbb{C}_n^+ to \mathbb{C}_n and $\nabla\phi^*$ is a mapping from \mathbb{C}_n to \mathbb{C}_n^+ . For any $X \in \mathbb{C}_n^+$, $Y \in \mathbb{C}_n$, we denote their spectral decomposition as $X = \sum_i \lambda_i q_i q_i^\dagger$, $Y = \sum_i w_i p_i p_i^\dagger$ with complex-valued eigenvectors

$\{q_i\}$ and $\{p_i\}$ form orthonormal basis of \mathbb{C}^n , and eigenvalues $\lambda_i > 0$, $w_i \in \mathbb{R}$,

$$\begin{aligned}\nabla\phi(X) &= I + \log X = I + \sum_i (\log \lambda_i) q_i q_i^\dagger \\ (\nabla\phi(X))^\dagger &= I^\dagger + \sum_i \overline{\log \lambda_i} (q_i q_i^\dagger)^\dagger = I + \sum_i (\log \lambda_i) q_i q_i^\dagger \\ &= \nabla\phi(X)\end{aligned}$$

This implies that $\nabla\phi(X) \in \mathbb{C}_n$ is a complex Hermitian matrix for any $X \in \mathbb{C}_n^+$. Similarly,

$$\begin{aligned}\nabla\phi^*(Y) &= \exp(Y - I) = \sum_i \exp(w_i - 1) p_i p_i^\dagger \\ (\nabla\phi^*(Y))^\dagger &= \sum_i \overline{\exp(w_i - 1)} (p_i p_i^\dagger)^\dagger = \nabla\phi^*(Y)\end{aligned}$$

Since $\nabla\phi^*(Y)$ has positive eigenvalues $\exp(w_i - 1)$, $\nabla\phi^*(Y) \in \mathbb{C}_n^+$ for any $Y \in \mathbb{C}_n$.

Now we need to show that $\nabla\phi$ and $\nabla\phi^*$ are inverse of each other.

$$\begin{aligned}\nabla\phi^*(\nabla\phi(X)) &= \exp(I + \sum_i \log \lambda_i q_i q_i^\dagger - I) \\ &= \sum_i \lambda_i q_i q_i^\dagger = X\end{aligned}$$

This implies that $\nabla\phi^* \circ \nabla\phi$ is the identity map on \mathbb{C}_n^+ .

$$\begin{aligned}\nabla\phi(\nabla\phi^*(Y)) &= \nabla\phi(\exp(\sum_i (w_i - 1) p_i p_i^\dagger)) \\ &= I + \log(\sum_i \exp(w_i - 1) p_i p_i^\dagger) \\ &= I + \sum_i (w_i - 1) p_i p_i^\dagger = Y\end{aligned}$$

Similarly, $\nabla\phi \circ \nabla\phi^*$ is the identity map on \mathbb{C}_n . Therefore, we conclude that $(\nabla\phi, \nabla\phi^*)$ is a pair of bijective mappings that are inverse of each other. \square

A.3. Training and neural network

Our neural network takes an input pair (x, t, c) , where x is the spatial input, t is the time, and c is the condition. Time t is passed through a module to generate a standard sinusoidal embedding and then fed to 2 fully connected layers with Sigmoid Linear Unit (SiLU) and generate an output t_{out} . Spatial input x is passed through an MLP with 8 residual blocks, each containing 4 linear layers with hidden dimension 512 and SiLU activation. This generates an output y_{out} . The condition c , considered to be a one-hot encoding in our case, is fed to 3 fully connected layers with SiLU and generates an output c_{out} . Our final output out is computed through,

$$\text{out} = \text{Outmod}(\text{GroupNorm}(y_{\text{out}} + t_{\text{out}} + c_{\text{out}}))$$

where Outmod is an out module that consists of 4 fully connected layers with hidden dimension 512 and SiLU activation, GroupNorm stands for group normalization.

A.4. Evaluation

The 1-Wasserstein Distance suffers from the curse of dimensionality seriously which is pointed out in Appendix.C in (Chen et al., 2023). Therefore, we use various metrics to comprehensively evaluate the distance between high dimensional distributions. For the scalar distribution of negativity, we stick to using 1-Wasserstein distance. In the main results, when evaluating the sample quality of generated quantum states (which is a density of 512 dimensions), we use the Sliced-Wasserstein Distance (SWD) type and Maximum Mean Discrepancies (MMD) type of metrics as our major criterion. These

metrics include Energy Maximum Mean Discrepancy (Energy-MMD), Max-sliced Wasserstein distance (MSWD), and Sliced-Wasserstein Distance (SWD). Here, Energy-MMD is the MMD distance computed with kernel $k(x, y) = -\|x - y\|_2$. Our implementation is adapted from [Geoloss](#) (W_1 , Energy-MMD) and [POT](#) (SWD, MSWD).