

Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

Abstract

We present high quality image synthesis results using diffusion probabilistic models, a class of latent variable models inspired by considerations from nonequilibrium thermodynamics. Our best results are obtained by training on a weighted variational bound designed according to a novel connection between diffusion probabilistic models and denoising score matching with Langevin dynamics, and our models naturally admit a progressive lossy decompression scheme that can be interpreted as a generalization of autoregressive decoding. On the unconditional CIFAR10 dataset, we obtain an Inception score of 9.46 and a state-of-the-art FID score of 3.17. On 256x256 LSUN, we obtain sample quality similar to ProgressiveGAN. Our implementation is available at <https://github.com/hojonathanho/diffusion>.

1 Introduction

Deep generative models of all kinds have recently exhibited high quality samples in a wide variety of data modalities. Generative adversarial networks (GANs), autoregressive models, flows, and variational autoencoders (VAEs) have synthesized striking image and audio samples [14, 27, 3, 58, 38, 25, 10, 32, 44, 57, 26, 33, 45], and there have been remarkable advances in energy-based modeling and score matching that have produced images comparable to those of GANs [11, 55].

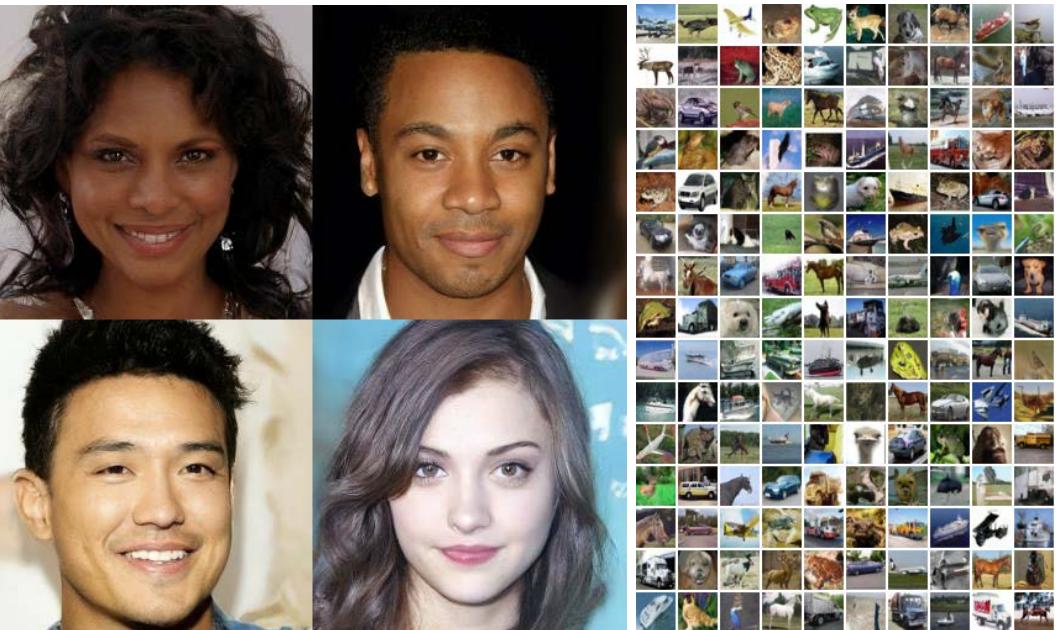


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

Mô hình xác suất khuếch tán khử nhiễu

Jonathan Ho UC
Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

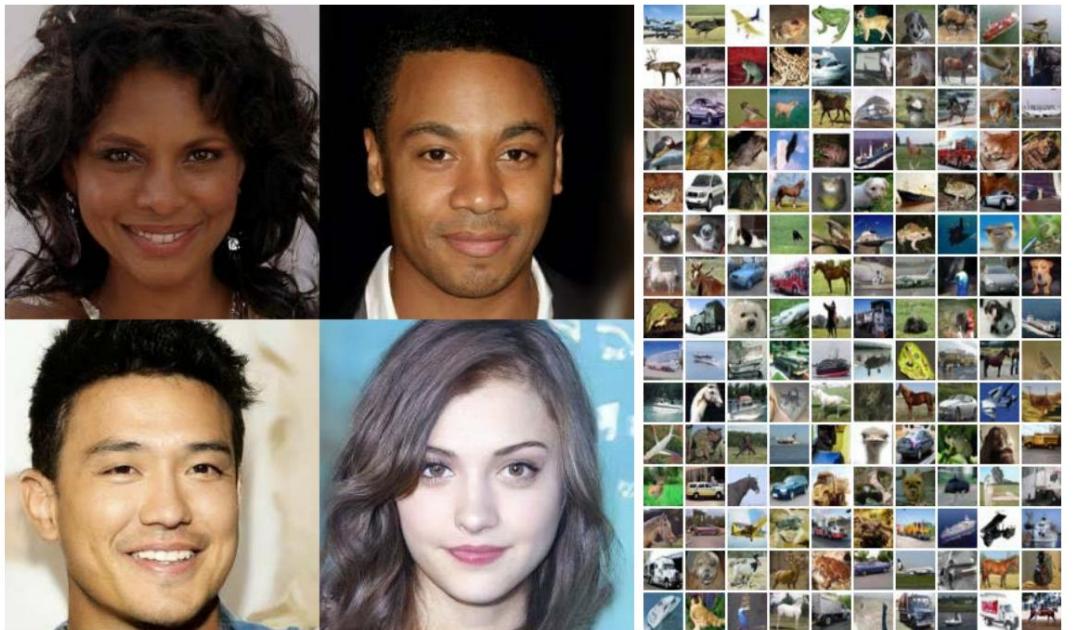
trùu tượng

Chúng tôi trình bày kết quả tổng hợp hình ảnh chất lượng cao bằng cách sử dụng mô hình xác suất khuếch tán, một loại mô hình biến tiệm ẩn lấy cảm hứng từ những cân nhắc từ nhiệt động lực học không cân bằng. Kết quả tốt nhất của chúng tôi thu được bằng cách huấn luyện trên một giới hạn biến thiên có trọng số được thiết kế theo một kết nối mới giữa các mô hình xác suất khuếch tán và khớp điểm khử nhiễu với động lực học Langevin, và các mô hình của chúng tôi đương nhiên thừa nhận một sơ đồ giải nén tần hao lũy tiến có thể được hiểu là sự tổng quát hóa của giải mã tự hồi quy. Trên tập dữ liệu CIFAR10 vô điều kiện, chúng tôi nhận được điểm Khởi đầu là 9,46 và điểm FID hiện đại là 3,17. Trên LSUN 256x256, chúng tôi thu được chất lượng mẫu tương tự như ProgressiveGAN. Việc triển khai của chúng tôi có sẵn tại <https://github.com/hojonathanho/diffusion>.

1. Giới thiệu

Tất cả các loại mô hình sinh sâu gần đây đã thể hiện các mẫu chất lượng cao ở nhiều phương thức dữ liệu khác nhau. Mạng đối nghịch tổng hợp (GAN), mô hình tự hồi quy, luồng và bộ mã hóa tự động biến thiên (VAE) đã tổng hợp các mẫu âm thanh và hình ảnh nổi bật [14, 27, 3, 58, 38, 25, 10, 32, 44, 57, 26, 33, 45], và đã có những tiến bộ đáng chú ý trong mô hình hóa và so khớp điểm dựa trên năng lượng đã tạo ra hình ảnh có thể so sánh với hình ảnh của GAN [11, 55].

arXiv:2006.11239v2



Hình 1: Các mẫu được tạo trên CelebA-HQ 256 × 256 (trái) và CIFAR10 vô điều kiện (phải)

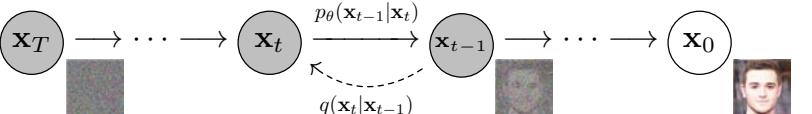


Figure 2: The directed graphical model considered in this work.

This paper presents progress in diffusion probabilistic models [53]. A diffusion probabilistic model (which we will call a “diffusion model” for brevity) is a parameterized Markov chain trained using variational inference to produce samples matching the data after finite time. Transitions of this chain are learned to reverse a diffusion process, which is a Markov chain that gradually adds noise to the data in the opposite direction of sampling until signal is destroyed. When the diffusion consists of small amounts of Gaussian noise, it is sufficient to set the sampling chain transitions to conditional Gaussians too, allowing for a particularly simple neural network parameterization.

Diffusion models are straightforward to define and efficient to train, but to the best of our knowledge, there has been no demonstration that they are capable of generating high quality samples. We show that diffusion models actually are capable of generating high quality samples, sometimes better than the published results on other types of generative models (Section 4). In addition, we show that a certain parameterization of diffusion models reveals an equivalence with denoising score matching over multiple noise levels during training and with annealed Langevin dynamics during sampling (Section 3.2) [55, 61]. We obtained our best sample quality results using this parameterization (Section 4.2), so we consider this equivalence to be one of our primary contributions.

Despite their sample quality, our models do not have competitive log likelihoods compared to other likelihood-based models (our models do, however, have log likelihoods better than the large estimates annealed importance sampling has been reported to produce for energy based models and score matching [11, 55]). We find that the majority of our models’ lossless codelengths are consumed to describe imperceptible image details (Section 4.3). We present a more refined analysis of this phenomenon in the language of lossy compression, and we show that the sampling procedure of diffusion models is a type of progressive decoding that resembles autoregressive decoding along a bit ordering that vastly generalizes what is normally possible with autoregressive models.

2 Background

Diffusion models [53] are latent variable models of the form $p_\theta(\mathbf{x}_0) := \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$, where $\mathbf{x}_1, \dots, \mathbf{x}_T$ are latents of the same dimensionality as the data $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. The joint distribution $p_\theta(\mathbf{x}_{0:T})$ is called the *reverse process*, and it is defined as a Markov chain with learned Gaussian transitions starting at $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \quad (1)$$

What distinguishes diffusion models from other types of latent variable models is that the approximate posterior $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$, called the *forward process* or *diffusion process*, is fixed to a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule β_1, \dots, β_T :

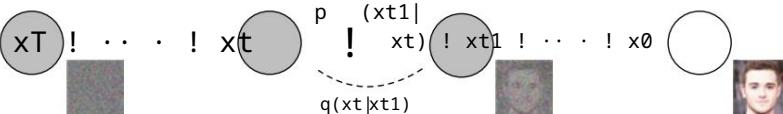
$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad (2)$$

Training is performed by optimizing the usual variational bound on negative log likelihood:

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] = \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] =: L \quad (3)$$

The forward process variances β_t can be learned by reparameterization [33] or held constant as hyperparameters, and expressiveness of the reverse process is ensured in part by the choice of Gaussian conditionals in $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$, because both processes have the same functional form when β_t are small [53]. A notable property of the forward process is that it admits sampling \mathbf{x}_t at an arbitrary timestep t in closed form: using the notation $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$, we have

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}) \quad (4)$$



Hình 2: Mô hình đồ họa có hướng được xem xét trong nghiên cứu này.

Bài viết này trình bày sự tiến bộ trong các mô hình xác suất khuếch tán [53]. Mô hình xác suất khuếch tán (mà chúng tôi sẽ gọi cho ngắn gọn là “mô hình khuếch tán”) là một chuỗi Markov được tham số hóa được đào tạo bằng cách sử dụng suy luận biến phân để tạo ra các mẫu khớp với dữ liệu sau thời gian hữu hạn. Các chuyển đổi của chuỗi này được học để đảo ngược quá trình khuếch tán, đó là chuỗi Markov dần dần thêm nhiều vào dữ liệu theo hướng ngược lại với việc lấy mẫu cho đến khi tín hiệu bị phá hủy. Khi sự khuếch tán bao gồm một lượng nhỏ nhiều Gaussian, việc thiết lập các chuyển đổi chuỗi lấy mẫu thành Gaussian có điều kiện cũng là đủ, cho phép tham số hóa mạng thần kinh đặc biệt đơn giản.

Các mô hình khuếch tán rất dễ xác định và đào tạo hiệu quả, nhưng theo hiểu biết tốt nhất của chúng tôi, chưa có bằng chứng nào cho thấy chúng có khả năng tạo ra các mẫu chất lượng cao. Chúng tôi cho thấy các mô hình khuếch tán thực sự có khả năng tạo ra các mẫu chất lượng cao, đôi khi tốt hơn kết quả được công bố trên các loại mô hình tổng hợp khác (Phần 4). Ngoài ra, chúng tôi cho thấy rằng việc tham số hóa nhất định của các mô hình khuếch tán cho thấy sự tương đương với điểm khử nhiễu phù hợp với nhiều mức tiếng ôn trong quá trình huấn luyện và với động lực học Langevin đã được ủ trong quá trình lấy mẫu (Phần 3.2) [55, 61]. Chúng tôi đã thu được kết quả chất lượng mẫu tốt nhất bằng cách sử dụng tham số hóa này (Phần 4.2), vì vậy chúng tôi coi sự tương đương này là một trong những đóng góp chính của chúng tôi.

Bất chấp chất lượng mẫu của chúng, các mô hình của chúng tôi không có khả năng ghi nhận ký cạnh tranh so với các mô hình dựa trên khả năng khác (tuy nhiên, các mô hình của chúng tôi có khả năng ghi nhận ký tốt hơn so với việc lấy mẫu tầm quan trọng đã được ước tính lớn đã được báo cáo để tạo ra cho các mô hình dựa trên năng lượng và khớp điểm [11, 55]). Chúng tôi thấy rằng phần lớn độ dài mã không mất dữ liệu của mô hình của chúng tôi được sử dụng để mô tả các chi tiết hình ảnh không thể nhận thấy (Phần 4.3). Chúng tôi trình bày một phân tích tinh tế hơn về hiện tượng này bằng ngôn ngữ nén tồn tại và chúng tôi chỉ ra rằng quy trình lấy mẫu của các mô hình khuếch tán là một loại giải mã lũy tiến giống như giải mã tự hồi quy theo thứ tự bit khai quát hóa rất nhiều những gì thường có thể xảy ra với các mô hình tự hồi quy.

2 Bối cảnh

Các mô hình khuếch tán [53] là các mô hình biến tiệm ẩn có dạng $p_\theta(\mathbf{x}_0) := p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$, trong đó \mathbf{x}_T là các mô hình biến tiệm ẩn có nghĩa là cùng chiều với dữ liệu $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. Sự phân phối chung $\mathbf{x}_1, \dots, \mathbf{x}_T$ được gọi là quá trình ngược lại, và nó được định chuỗi Markov với các chuyển đổi Gaussian đã học bắt đầu từ $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \boldsymbol{\mu}_\theta(\mathbf{x}_{t-1}, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_{t-1}, t)) \quad (1)$$

Điều phân biệt mô hình khuếch tán với các loại mô hình biến tiệm ẩn khác là $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$ sau gần đúng, được gọi là quá trình chuyển tiếp hoặc quá trình khuếch tán, được cố định vào chuỗi Markov để dần dần thêm nhiều Gaussian vào dữ liệu theo một biểu đồ phương sai β_1, \dots, β_T :

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \mathbf{x}_{t-1} + \beta_t \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad (2)$$

Việc đào tạo được thực hiện bằng cách tối ưu hóa giới hạn biến thiên thông thường đối với khả năng ghi nhận ký âm:

$$\mathbb{E}[-\log p_\theta(\mathbf{x}_0)] \leq \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] = \text{Phương trình } \log p(\mathbf{x}_T) - \log \frac{p_\theta(\mathbf{x}_T|\mathbf{x}_{t-1})}{q(\mathbf{x}_T|\mathbf{x}_{t-1})} =: L \quad (3)$$

Phương sai của quy trình thuận β_t có thể được học bằng cách tham số hóa lại [33] hoặc được giữ không đổi dưới dạng siêu tham số và tính biểu thị của quy trình ngược lại được đảm bảo một phần bằng cách chọn các điều kiện Gaussian trong $p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})$, bởi vì cả hai quy trình đều có dạng hàm tương tự khi β_t nhỏ [53]. Một đặc tính đáng chú ý của quy trình chuyển tiếp là nó cho phép lấy mẫu \mathbf{x}_t tại dấu thời gian tùy ý t ở dạng đóng: sử dụng ký hiệu $\alpha_t := 1 - \beta_t$ và $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$, ta có

$$\mathbf{x}_t := q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I})$$

Efficient training is therefore possible by optimizing random terms of L with stochastic gradient descent. Further improvements come from variance reduction by rewriting L (3) as:

$$\mathbb{E}_q \left[\underbrace{D_{KL}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t))}_{L_{t-1}} - \log p_\theta(\mathbf{x}_0 | \mathbf{x}_1) \right] \quad (5)$$

(See Appendix A for details. The labels on the terms are used in Section 3.) Equation (5) uses KL divergence to directly compare $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ against forward process posteriors, which are tractable when conditioned on \mathbf{x}_0 :

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}), \quad (6)$$

$$\text{where } \tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} \mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} \mathbf{x}_t \quad \text{and} \quad \tilde{\beta}_t := \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t \quad (7)$$

Consequently, all KL divergences in Eq. (5) are comparisons between Gaussians, so they can be calculated in a Rao-Blackwellized fashion with closed form expressions instead of high variance Monte Carlo estimates.

3 Diffusion models and denoising autoencoders

Diffusion models might appear to be a restricted class of latent variable models, but they allow a large number of degrees of freedom in implementation. One must choose the variances β_t of the forward process and the model architecture and Gaussian distribution parameterization of the reverse process. To guide our choices, we establish a new explicit connection between diffusion models and denoising score matching (Section 3.2) that leads to a simplified, weighted variational bound objective for diffusion models (Section 3.4). Ultimately, our model design is justified by simplicity and empirical results (Section 4). Our discussion is categorized by the terms of Eq. (5).

3.1 Forward process and L_T

We ignore the fact that the forward process variances β_t are learnable by reparameterization and instead fix them to constants (see Section 4 for details). Thus, in our implementation, the approximate posterior q has no learnable parameters, so L_T is a constant during training and can be ignored.

3.2 Reverse process and $L_{1:T-1}$

Now we discuss our choices in $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$ for $1 < t \leq T$. First, we set $\Sigma_\theta(\mathbf{x}_t, t) = \sigma_t^2 \mathbf{I}$ to untrained time dependent constants. Experimentally, both $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t$ had similar results. The first choice is optimal for $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the second is optimal for \mathbf{x}_0 deterministically set to one point. These are the two extreme choices corresponding to upper and lower bounds on reverse process entropy for data with coordinatewise unit variance [53].

Second, to represent the mean $\mu_\theta(\mathbf{x}_t, t)$, we propose a specific parameterization motivated by the following analysis of L_t . With $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$, we can write:

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)\|^2 \right] + C \quad (8)$$

where C is a constant that does not depend on θ . So, we see that the most straightforward parameterization of μ_θ is a model that predicts $\tilde{\mu}_t$, the forward process posterior mean. However, we can expand Eq. (8) further by reparameterizing Eq. (4) as $\mathbf{x}_t(\mathbf{x}_0, \epsilon) = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$ for $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and applying the forward process posterior formula (7):

$$L_{t-1} - C = \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[\frac{1}{2\sigma_t^2} \left\| \tilde{\mu}_t \left(\mathbf{x}_t(\mathbf{x}_0, \epsilon), \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t(\mathbf{x}_0, \epsilon) - \sqrt{1 - \bar{\alpha}_t} \epsilon) \right) - \mu_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t) \right\|^2 \right] \quad (9)$$

$$= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t(\mathbf{x}_0, \epsilon) - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon \right) - \mu_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t) \right\|^2 \right] \quad (10)$$

Do đó, có thể đào tạo hiệu quả bằng cách tối ưu hóa các số hạng ngẫu nhiên của L với độ dốc giảm dần ngẫu nhiên. Những cải tiến hơn nữa đến từ việc giảm sai bằng cách viết lại L (3) dưới dạng:

$$\text{Phương trình DKL}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T)) + \sum_{t>1} \frac{DKL(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)}{L_0} \quad (5)$$

(Xem Phụ lục A để biết chi tiết. Các nhãn trên các thuật ngữ được sử dụng trong Phần 3.) Phương trình (5) sử dụng phân kỳ KL để so sánh trực tiếp $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ với các phần sau của quá trình thuận, có thể xử lý được khi được điều hòa trên \mathbf{x}_0 : $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) = N(\mathbf{x}_{t-1}; \mu^\top \mathbf{x}_t, \mathbf{x}_0, \beta^\top \mathbf{I})$, $\sqrt{\alpha_t} \mathbf{x}_t$ và $\beta^\top \mathbf{I}$ trong

$$\text{đó } \mu^\top \mathbf{x}_t, \mathbf{x}_0 := \mathbf{x}_0 + \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t \quad (6)$$

$$\sqrt{\alpha_t} \mathbf{x}_t = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_t)}{1 - \bar{\alpha}_t} \mathbf{x}_t \quad \text{và} \quad \beta^\top \mathbf{I} = \frac{\beta^\top \mathbf{I}}{1 - \bar{\alpha}_t} \quad (7)$$

Do đó, tất cả các phân kỳ KL trong biểu thức (5) là sự so sánh giữa các Gaussian, do đó chúng có thể được tính toán theo kiểu Rao-Blackwellized với các biểu thức dạng đóng thay vì ước tính phương sai cao của Monte Carlo.

3 mô hình khuếch tán và bộ mã hóa tự động khử nhiễu

Các mô hình khuếch tán có thể dường như là một lớp hạn chế của các mô hình biến tiệm ẩn, nhưng chúng cho phép nhiều mức độ tự do trong việc thực hiện. Người ta phải chọn phương sai β_t của quy trình chuyển tiếp và kiến trúc mô hình cũng như tham số hóa phân bố Gaussian của quy trình ngược lại. Để hướng dẫn các lựa chọn của mình, chúng tôi thiết lập một kết nối rõ ràng mới giữa các mô hình khuếch tán và so sánh điểm khử nhiễu (Phần 3.2) dẫn đến mục tiêu ràng buộc biến thiên có trọng số, đơn giản hóa cho các mô hình khuếch tán (Phần 3.4). Cuối cùng, thiết kế mô hình của chúng tôi được chứng minh bằng sự đơn giản và kết quả thực nghiệm (Phần 4). Cuộc thảo luận của chúng tôi được phân loại theo các điều khoản của phương trình (5).

3.1 Quá trình chuyển tiếp và L_T

Chúng ta bỏ qua thực tế là các phương sai của quá trình chuyển tiếp β_t có thể học được bằng cách tham số hóa lại và thay vào đó sửa chúng thành các hằng số (xem Phần 4 để biết chi tiết). Do đó, trong quá trình triển khai của chúng tôi, q sau gần đúng không có tham số có thể học được, do đó L_T là một hằng số trong quá trình đào tạo và có thể bị bỏ qua.

3.2 Quy trình đảo ngược và $L_{1:T-1}$

Bây giờ chúng ta thảo luận về các lựa chọn của mình trong $p_\theta(\mathbf{x}_t | \mathbf{x}_0) = N(\mathbf{x}_t; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$ với $1 < t \leq T$. Đầu tiên, chúng ta đặt $\Sigma_\theta(\mathbf{x}_t, t) = \frac{1 - \bar{\alpha}_t}{1 - \bar{\alpha}_t} \beta_t^2 \mathbf{I}$ để đạt được kết quả tương tự. Lựa chọn đầu tiên là tối ưu cho $\mathbf{x}_0 \sim N(\mathbf{0}, \mathbf{I})$, và lựa chọn thứ hai là tối ưu cho \mathbf{x}_0 được đặt một cách xác định thành một điểm. Đây là hai lựa chọn cực trị tương ứng với giới hạn trên và giới hạn dưới của entropy quá trình ngược đối với dữ liệu có phương sai đơn vị tọa độ [53].

Thứ hai, để biểu thị giá trị trung bình $\mu_\theta(\mathbf{x}_t, t)$, chúng tôi đã xuất một tham số hóa cụ thể dựa trên phân tích L_t sau đây. Với $p_\theta(\mathbf{x}_t | \mathbf{x}_0) = N(\mathbf{x}_t; \mu_\theta(\mathbf{x}_t, t), \sigma_t^2 \mathbf{I})$, chúng ta có thể viết:

$$L_t = \text{phương trình } \frac{1}{2\sigma_t^2} \|\mu^\top \mathbf{x}_t - \mu_\theta(\mathbf{x}_t, t)\|^2 + C \quad (số 8)$$

trong đó C là hằng số không phụ thuộc vào θ . Vì vậy, chúng ta thấy rằng tham số hóa đơn giản nhất của μ_θ là một mô hình dự đoán $\mu^\top \mathbf{x}_t$, trung bình sau của quá trình thuận. Tuy nhiên, chúng ta có thể mở rộng phương trình (8) hơn nữa bằng cách tham số hóa lại phương trình (4) dưới dạng $\mathbf{x}_t(\mathbf{x}_0, \epsilon) = \sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$ với $N(\mathbf{0}, \mathbf{I})$ và áp dụng công thức hậu nghiệm của quá trình thuận (7):

$$L_t - C = \text{Ex}\theta, \quad \frac{1}{2\sigma_t^2} \|\mu^\top \mathbf{x}_t - \mu_\theta(\mathbf{x}_t, t)\|^2 + \frac{1}{\sqrt{1 - \bar{\alpha}_t}} (\mathbf{x}_t(\mathbf{x}_0, \epsilon) - \mu_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t))^2 \quad (9)$$

$$= \text{Ex}\theta, \quad \frac{1}{2\sigma_t^2} \|\frac{1}{\sqrt{\alpha_t}} (\mathbf{x}_t(\mathbf{x}_0, \epsilon) - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon) - \mu_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t)\|^2 \quad (10)$$

Algorithm 1 Training

```

1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   Take gradient descent step on
       $\nabla_\theta \|\epsilon - \epsilon_\theta(\sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \epsilon, t)\|^2$ 
6: until converged

```

Algorithm 2 Sampling

```

1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
4:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 
5: end for
6: return  $\mathbf{x}_0$ 

```

Equation (10) reveals that μ_θ must predict $\frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right)$ given \mathbf{x}_t . Since \mathbf{x}_t is available as input to the model, we may choose the parameterization

$$\mu_\theta(\mathbf{x}_t, t) = \tilde{\mu}_t \left(\mathbf{x}_t, \frac{1}{\sqrt{\alpha_t}} (\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t} \epsilon_\theta(\mathbf{x}_t)) \right) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) \quad (11)$$

where ϵ_θ is a function approximator intended to predict ϵ from \mathbf{x}_t . To sample $\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$ is to compute $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The complete sampling procedure, Algorithm 2, resembles Langevin dynamics with ϵ_θ as a learned gradient of the data density. Furthermore, with the parameterization (11), Eq. (10) simplifies to:

$$\mathbb{E}_{\mathbf{x}_0, \epsilon} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \|\epsilon - \epsilon_\theta(\sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \epsilon, t)\|^2 \right] \quad (12)$$

which resembles denoising score matching over multiple noise scales indexed by t [55]. As Eq. (12) is equal to (one term of) the variational bound for the Langevin-like reverse process (11), we see that optimizing an objective resembling denoising score matching is equivalent to using variational inference to fit the finite-time marginal of a sampling chain resembling Langevin dynamics.

To summarize, we can train the reverse process mean function approximator μ_θ to predict $\tilde{\mu}_t$, or by modifying its parameterization, we can train it to predict ϵ . (There is also the possibility of predicting \mathbf{x}_0 , but we found this to lead to worse sample quality early in our experiments.) We have shown that the ϵ -prediction parameterization both resembles Langevin dynamics and simplifies the diffusion model’s variational bound to an objective that resembles denoising score matching. Nonetheless, it is just another parameterization of $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)$, so we verify its effectiveness in Section 4 in an ablation where we compare predicting ϵ against predicting $\tilde{\mu}_t$.

3.3 Data scaling, reverse process decoder, and L_0

We assume that image data consists of integers in $\{0, 1, \dots, 255\}$ scaled linearly to $[-1, 1]$. This ensures that the neural network reverse process operates on consistently scaled inputs starting from the standard normal prior $p(\mathbf{x}_T)$. To obtain discrete log likelihoods, we set the last term of the reverse process to an independent discrete decoder derived from the Gaussian $\mathcal{N}(\mathbf{x}_0; \mu_\theta(\mathbf{x}_1, 1), \sigma_1^2 \mathbf{I})$:

$$p_\theta(\mathbf{x}_0 | \mathbf{x}_1) = \prod_{i=1}^D \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(\mathbf{x}_1, 1), \sigma_1^2) dx \quad (13)$$

$$\delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases} \quad \delta_-(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

where D is the data dimensionality and the i superscript indicates extraction of one coordinate. (It would be straightforward to instead incorporate a more powerful decoder like a conditional autoregressive model, but we leave that to future work.) Similar to the discretized continuous distributions used in VAE decoders and autoregressive models [34, 52], our choice here ensures that the variational bound is a lossless codelength of discrete data, without need of adding noise to the data or incorporating the Jacobian of the scaling operation into the log likelihood. At the end of sampling, we display $\mu_\theta(\mathbf{x}_1, 1)$ noiselessly.

3.4 Simplified training objective

With the reverse process and decoder defined above, the variational bound, consisting of terms derived from Eqs. (12) and (13), is clearly differentiable with respect to θ and is ready to be employed for

Thuật toán 1 Huấn luyện 1:

```

lặp lại 2:
x0 = q(x0) 3: t
Đồng phục(\{1, . . . , T\}) 4: N(0,
I)
5: Thực hiện bước giảm độ dốc trên 0
     0(  $\sqrt{\alpha^t} x_0 + \sqrt{1-\bar{\alpha}^t} \epsilon$ , t) 6: cho đến khi
hết tụ

```

Lấy mẫu thuật toán 2

```

1: xT = N(0, I) 2:
với t = T, . . . , 1 do 3:
z = N(0, I) nếu t > 1, ngược lại z = 0 4:
xt =  $\sqrt{1-\bar{\alpha}^t} \epsilon$  at  $\sqrt{\alpha^t} xT + \sqrt{1-\bar{\alpha}^t} \epsilon$  5: kết thúc
cho 6: trả về x0

```

Phương trình (10) cho thấy μ_θ phải dự đoán $\sqrt{1-\bar{\alpha}^t}$ đầu vào cho $\frac{1}{\sqrt{\alpha^t}}$, xT và $\frac{\beta_t}{\sqrt{1-\bar{\alpha}^t}}$ đã cho xT . Vì xT có sẵn dưới dạng tham số hóa

$$\frac{1}{\sqrt{1-\bar{\alpha}^t}} \text{ at } \frac{\beta_t}{\sqrt{1-\bar{\alpha}^t}} \text{ trong đó } \bar{\alpha}^t \text{ là một hàn công của } \bar{\alpha}^t \text{ và } \bar{\alpha}^t = \frac{\bar{\alpha}^{t-1} + \bar{\alpha}^t}{2} \text{ trong } xT = N(0, I). \text{ Việc lấy mẫu hoàn chỉnh để tính quy trình } xT = \sqrt{1-\bar{\alpha}^t} \text{, Thuật toán 2, giống với động lực học Langevin như gradient descent để học của mật độ dữ liệu.}$$

với tham số hóa (11), phương trình (10) đơn giản hóa thành:

$$\frac{1}{\sqrt{1-\bar{\alpha}^t}} \text{ at } \frac{\beta_t}{\sqrt{1-\bar{\alpha}^t}}$$

$$\mathbb{E}_{\mathbf{x}_0, \epsilon} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}^t)} \theta \left(\sqrt{\alpha^t} x_0 + \sqrt{1-\bar{\alpha}^t} \epsilon, t \right) \right]^2 \quad (12)$$

tương tự như việc so khớp điểm khứ nhiều trên nhiều thang đo nhiều được lập chỉ mục bởi t [55]. Như phương trình (12) bằng (một số hạng của) giới hạn biến phân cho quy trình ngược Langevin (11), chúng tôi thấy rằng việc tối ưu hóa một mục tiêu giống như đổi sánh điểm khứ nhiều tương đương với việc sử dụng suy luận biến phân để phù hợp với biến thời gian hữu hạn của một chuỗi lấy mẫu giống như động lực học Langevin.

Tóm lại, chúng ta có thể huấn luyện bộ xấp xỉ hàm trung bình quá trình ngược μ_θ để dự đoán μ^t hoặc bằng cách sửa đổi tham số hóa của nó, chúng ta có thể huấn luyện nó để dự đoán x_0 . (Cũng có khả năng dự đoán x_0 , nhưng chúng tôi nhận thấy điều này sớm dẫn đến chất lượng mẫu kém hơn trong các thử nghiệm của chúng tôi.) Chúng tôi đã chỉ ra rằng tham số hóa dự đoán vừa giống với động lực học Langevin vừa đơn giản hóa biến phân của mô hình khuếch tán bị ràng buộc với một mục tiêu giống với khứ nhiều kết hợp điểm số. Tuy nhiên, nó chỉ là một tham số hóa khác của $p_\theta(xt | 1xt)$, vì vậy chúng tôi xác minh tính hiệu quả của nó trong Phần 4 trong phép loại bỏ trong đó chúng tôi so sánh dự đoán với dự đoán μ^t .

3.3 Chia tỷ lệ dữ liệu, bộ giải mã quy trình ngược và L_0

Chúng tôi giả định rằng dữ liệu hình ảnh bao gồm các số nguyên trong $\{0, 1, \dots, 255\}$ được chia tỷ lệ tuyến tính thành $[1, 1]$. Điều này đảm bảo rằng quy trình đảo ngược của mạng thần kinh hoạt động trên các đầu vào có tỷ lệ nhất quán bắt đầu từ $p(xT)$ tiêu chuẩn bình thường trước đó. Để thu được khả năng ghi nhật ký rời rạc, chúng tôi đặt số hạng cuối cùng của quy trình ngược lại thành bộ giải mã rời rạc độc lập bắt nguồn từ Gaussian $N(x0; \mu_\theta(x1, 1), \sigma_1^2 \mathbf{I})$:

$$p_\theta(x0 | x1) = \frac{1}{D} \sum_{i=1}^D \delta(x_i - \delta_+(x_i)) \quad (13)$$

$$\delta_+(x) = \begin{cases} \infty & \text{if } x = 1 \\ x + \frac{1}{255} & \text{if } x < 1 \end{cases} \quad \delta(x) = \begin{cases} -\infty & \text{if } x = -1 \\ x - \frac{1}{255} & \text{if } x > -1 \end{cases}$$

Trong đó D là chiều dữ liệu và chỉ số trên i biểu thị việc trích xuất một tọa độ. (Thay vào đó, sẽ đơn giản hơn nếu kết hợp một bộ giải mã mạnh hơn như mô hình tự hồi quy có điều kiện, nhưng chúng tôi để việc đó cho công việc sau này.) Tương tự như các phân bố liên tục rời rạc được sử dụng trong bộ giải mã VAE và các mô hình tự hồi quy [34, 52], lựa chọn của chúng tôi ở đây đảm bảo rằng giới hạn biến phân là độ dài mà không mất dữ liệu của dữ liệu rời rạc, không cần thêm nhiễu vào dữ liệu hoặc kết hợp Jacobian của hoạt động chia tỷ lệ vào khả năng ghi nhật ký. Khi kết thúc lấy mẫu, chúng tôi hiển thị $\mu_\theta(x1, 1)$ một cách yên tĩnh.

3.4 Mục tiêu đào tạo đơn giản hóa

Với quy trình đảo ngược và bộ giải mã được xác định ở trên, giới hạn biến phân, bao gồm các thuật ngữ rút ra từ các phương trình (12) và (13), có khả vi rõ ràng đối với θ và sẵn sàng được sử dụng cho

Table 1: CIFAR10 results. NLL measured in bits/dim.

Model	IS	FID	NLL Test (Train)
Conditional			
EBM [11]	8.30	37.9	
JEM [17]	8.76	38.4	
BigGAN [3]	9.22	14.73	
StyleGAN2 + ADA (v1) [29]	10.06	2.67	
Unconditional			
Diffusion (original) [53]			≤ 5.40
Gated PixelCNN [59]	4.60	65.93	3.03 (2.90)
Sparse Transformer [7]			2.80
PixelINQ [43]	5.29	49.46	
EBM [11]	6.78	38.2	
NCSNv2 [56]	31.75		
NCSN [55]	8.87 ± 0.12	25.32	
SNGAN [39]	8.22 ± 0.05	21.7	
SNGAN-DDLS [4]	9.09 ± 0.10	15.42	
StyleGAN2 + ADA (v1) [29]	9.74 ± 0.05	3.26	
Ours (L , fixed isotropic Σ)	7.67 ± 0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46 ± 0.11	3.17	≤ 3.75 (3.72)

Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

Objective	IS	FID
$\tilde{\mu}$ prediction (baseline)		
L , learned diagonal Σ	7.28 ± 0.10	23.69
L , fixed isotropic Σ	8.06 ± 0.09	13.22
$\ \tilde{\mu} - \tilde{\mu}_\theta\ ^2$	-	-
ϵ prediction (ours)		
L , learned diagonal Σ	-	-
L , fixed isotropic Σ	7.67 ± 0.13	13.51
$\ \tilde{\epsilon} - \epsilon_\theta\ ^2$ (L_{simple})	9.46 ± 0.11	3.17

training. However, we found it beneficial to sample quality (and simpler to implement) to train on the following variant of the variational bound:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \right\|^2 \right] \quad (14)$$

where t is uniform between 1 and T . The $t = 1$ case corresponds to L_0 with the integral in the discrete decoder definition (13) approximated by the Gaussian probability density function times the bin width, ignoring σ_1^2 and edge effects. The $t > 1$ cases correspond to an unweighted version of Eq. (12), analogous to the loss weighting used by the NCSN denoising score matching model [55]. (L_T does not appear because the forward process variances β_t are fixed.) Algorithm 1 displays the complete training procedure with this simplified objective.

Since our simplified objective (14) discards the weighting in Eq. (12), it is a weighted variational bound that emphasizes different aspects of reconstruction compared to the standard variational bound [18, 22]. In particular, our diffusion process setup in Section 4 causes the simplified objective to down-weight loss terms corresponding to small t . These terms train the network to denoise data with very small amounts of noise, so it is beneficial to down-weight them so that the network can focus on more difficult denoising tasks at larger t terms. We will see in our experiments that this reweighting leads to better sample quality.

4 Experiments

We set $T = 1000$ for all experiments so that the number of neural network evaluations needed during sampling matches previous work [53, 55]. We set the forward process variances to constants increasing linearly from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$. These constants were chosen to be small relative to data scaled to $[-1, 1]$, ensuring that reverse and forward processes have approximately the same functional form while keeping the signal-to-noise ratio at \mathbf{x}_T as small as possible ($L_T = D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| \mathcal{N}(\mathbf{0}, \mathbf{I})) \approx 10^{-5}$ bits per dimension in our experiments).

To represent the reverse process, we use a U-Net backbone similar to an unmasked PixelCNN++ [52, 48] with group normalization throughout [66]. Parameters are shared across time, which is specified to the network using the Transformer sinusoidal position embedding [60]. We use self-attention at the 16×16 feature map resolution [63, 60]. Details are in Appendix B.

4.1 Sample quality

Table 1 shows Inception scores, FID scores, and negative log likelihoods (lossless codelengths) on CIFAR10. With our FID score of 3.17, our unconditional model achieves better sample quality than most models in the literature, including class conditional models. Our FID score is computed with respect to the training set, as is standard practice; when we compute it with respect to the test set, the score is 5.24, which is still better than many of the training set FID scores in the literature.

Bảng 1: Kết quả CIFAR10. NLL được đo bằng bit/dim.

Người mẫu	LÀ	Kiểm tra FID NLL (Tàu hỏa)	Bảng 2: Đảo ngược CIFAR10 vô điều kiện
có điều kiện			
EBM [11]	8,30	37,9	tham số hóa quá trình và loại bỏ mục tiêu đào tạo. Các mục tiêu tổng không ổn định
JEM [17]	8,76	38,4	huấn luyện và tạo ra các mẫu kém với điểm số nằm ngoài phạm vi.
BigGAN [3]	9,22	14,73	
StyleGAN2 + ADA (v1) [29]	10,06	2,67	
vô điều kiện			
Khuêch tán (bản gốc) [53]			dự đoán μ^- (dường cơ sở)
PixelCNN có công [59]	4,60	65,93	L , đường chéo đã học Σ $7,28 \pm 0,10$ 23,69
Máy biến áp thừa thoát [7]			L , dâng hướng cố định Σ $8,66 \pm 0,09$ 13,22
PixelION [43] 49,46	5,29		$\mu^- \mu^- \theta^2$ - -
EBM [11] 38,2	6,78		
NCSNv2 [56] 31,75			dự đoán (của chúng tôi)
NCSN [55] $8.87 \pm 0,12$ 25,32			L , đường chéo đã học Σ - -
SNGAN [39] $8.22 \pm 0,05$ 21,7			L , dâng hướng cố định Σ $7,67 \pm 0,13$ 13,51
SNGAN-DDLS [4] $9.09 \pm 0,10$ 15,42			θ^2 (Đơn giản) $9,46 \pm 0,11$ 3,17
StyleGAN2 + ADA (v1) [29] $9.74 \pm 0,05$ 3,26			Của chúng tôi (L , dâng hướng) $7,67 \pm 0,13$ 13,51 3,70 (3,69)
Của chúng tôi (L_{simple}) $9,46 \pm 0,11$ 3,17			Của chúng tôi (L_{simple}) $9,46 \pm 0,11$ 3,75 (3,72)

đào tạo. Tuy nhiên, chúng tôi nhận thấy việc đào tạo về chất lượng mẫu (và thực hiện đơn giản hơn) sẽ có lợi cho biến thể sau đây của biến đổi:

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t) \right\|^2 \right] \quad (14)$$

trong đó t đồng nhất giữa 1 và T . Trường hợp $t = 1$ tương ứng với L_0 với tích phân trong định nghĩa bộ giải mã rời rạc (13) được tính gần đúng bằng hàm mật độ xác suất Gaussian nhân với chiều rộng thùng, bỏ qua σ_1^2 và hiệu ứng biên. Các trường hợp $t > 1$ tương ứng với phiên bản không có trọng số của phương trình (12), tương tự như trọng số tồn thắt được sử dụng bởi mô hình so khớp điểm khứ nhiều NCSN [55]. (LT không xuất hiện vì phương sai của quá trình chuyển tiếp bt đã có định.) Thuật toán 1 hiển thị hoàn thành quy trình đào tạo với mục tiêu đơn giản hóa này.

Vì mục tiêu đơn giản hóa của chúng tôi (14) loại bỏ trọng số trong biểu thức (12), nó là một biến thể có trọng số ràng buộc nhấn mạnh các khía cạnh khác nhau của việc xây dựng lại so với biến thể tiêu chuẩn ràng buộc [18, 22]. Đặc biệt, việc thiết lập quy trình khuêch tán của chúng tôi trong Phần 4 giúp đơn giản hóa mục tiêu với các điều khoản giảm cân tương ứng với t nhỏ. Những thuật ngữ này huấn luyện mạng cách khử nhiễu dữ liệu với lượng nhiễu rất nhỏ, do đó sẽ có lợi khi giảm trọng lượng của chúng để mạng có thể tập trung vào các nhiệm vụ khử nhiễu khó khăn hơn với số lượng t lớn hơn. Chúng ta sẽ thấy trong các thí nghiệm của mình rằng điều này việc cân nhắc lại sẽ dẫn đến chất lượng mẫu tốt hơn.

4 thí nghiệm

Chúng tôi đặt $T = 1000$ cho tất cả các thử nghiệm sao cho số lượng đánh giá mạng lưới thần kinh cần thiết trong quá trình lấy mẫu phù hợp với công việc trước đó [53, 55]. Chúng tôi đặt phương sai của quá trình chuyển tiếp thành hằng số tăng tuyến tính từ $\beta_1 = 10^{-4}$ đến $\beta_T = 0.02$. Các hằng số này được chọn có giá trị nhỏ so với dữ liệu được chia tỷ lệ thành $[1, 1]$, đảm bảo rằng các quy trình đảo ngược và chuyển tiếp có khoảng cách năng lượng tự trong khi vẫn giữ tỷ lệ tin hiệu trên tạp âm ở \mathbf{x}_T càng nhỏ càng tốt ($LT = D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| \mathcal{N}(\mathbf{0}, \mathbf{I})) \approx 10,5$ bit cho mỗi chiều trong các thử nghiệm của chúng tôi).

Để biểu diễn quá trình ngược lại, chúng tôi sử dụng đường trực U-Net tương tự như PixelCNN++ không được che mờ [52, 48] với sự chuẩn hóa nhóm trong suốt [66]. Các thông số được chia sẻ theo thời gian, được chỉ định vào mạng bằng cách nhúng vị trí hình sin của Máy biến áp [60]. Chúng tôi sử dụng sự chú ý tại độ phân giải ban đầu đặc trưng 16×16 [63, 60]. Chi tiết có trong Phụ lục B.

4.1 Chất lượng mẫu

Bảng 1 cho thấy điểm khởi đầu, điểm FID và khả năng ghi nhận ký âm (độ dài mã không mất dữ liệu) trên CIFAR10. Với điểm FID là 3,17, mô hình vô điều kiện của chúng tôi đạt được chất lượng mẫu tốt hơn so với hầu hết các mô hình trong tài liệu, bao gồm cả các mô hình có điều kiện của lớp. Điểm FID của chúng tôi được tính bằng tông trọng tập huấn luyện, cũng như thực hành tiêu chuẩn; khi chúng tôi tính toán nó theo tập kiểm tra, điểm là 5,24, vẫn tốt hơn nhiều điểm FID của tập huấn luyện trong tài liệu.



Figure 3: LSUN Church samples. FID=7.89



Figure 4: LSUN Bedroom samples. FID=4.90



Hình 3: Mẫu nhà thờ LSUN. FID=7,89



Hình 4: Mẫu phòng ngủ LSUN. FID=4,90

Algorithm 3 Sending \mathbf{x}_0

```

1: Send  $\mathbf{x}_T \sim q(\mathbf{x}_T | \mathbf{x}_0)$  using  $p(\mathbf{x}_T)$ 
2: for  $t = T - 1, \dots, 2, 1$  do
3:   Send  $\mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_{t+1}, \mathbf{x}_0)$  using  $p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})$ 
4: end for
5: Send  $\mathbf{x}_0$  using  $p_\theta(\mathbf{x}_0 | \mathbf{x}_1)$ 

```

Algorithm 4 Receiving

```

1: Receive  $\mathbf{x}_T$  using  $p(\mathbf{x}_T)$ 
2: for  $t = T - 1, \dots, 1, 0$  do
3:   Receive  $\mathbf{x}_t$  using  $p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})$ 
4: end for
5: return  $\mathbf{x}_0$ 

```

We find that training our models on the true variational bound yields better codelengths than training on the simplified objective, as expected, but the latter yields the best sample quality. See Fig. 1 for CIFAR10 and CelebA-HQ 256 × 256 samples, Fig. 3 and Fig. 4 for LSUN 256 × 256 samples [71], and Appendix D for more.

4.2 Reverse process parameterization and training objective ablation

In Table 2, we show the sample quality effects of reverse process parameterizations and training objectives (Section 3.2). We find that the baseline option of predicting $\tilde{\mu}$ works well only when trained on the true variational bound instead of unweighted mean squared error, a simplified objective akin to Eq. (14). We also see that learning reverse process variances (by incorporating a parameterized diagonal $\Sigma_\theta(\mathbf{x}_t)$ into the variational bound) leads to unstable training and poorer sample quality compared to fixed variances. Predicting ϵ , as we proposed, performs approximately as well as predicting $\tilde{\mu}$ when trained on the variational bound with fixed variances, but much better when trained with our simplified objective.

4.3 Progressive coding

Table 1 also shows the codelengths of our CIFAR10 models. The gap between train and test is at most 0.03 bits per dimension, which is comparable to the gaps reported with other likelihood-based models and indicates that our diffusion model is not overfitting (see Appendix D for nearest neighbor visualizations). Still, while our lossless codelengths are better than the large estimates reported for energy based models and score matching using annealed importance sampling [11], they are not competitive with other types of likelihood-based generative models [7].

Since our samples are nonetheless of high quality, we conclude that diffusion models have an inductive bias that makes them excellent lossy compressors. Treating the variational bound terms $L_1 + \dots + L_T$ as rate and L_0 as distortion, our CIFAR10 model with the highest quality samples has a rate of **1.78** bits/dim and a distortion of **1.97** bits/dim, which amounts to a root mean squared error of 0.95 on a scale from 0 to 255. More than half of the lossless codelength describes imperceptible distortions.

Progressive lossy compression We can probe further into the rate-distortion behavior of our model by introducing a progressive lossy code that mirrors the form of Eq. (5): see Algorithms 3 and 4, which assume access to a procedure, such as minimal random coding [19, 20], that can transmit a sample $\mathbf{x} \sim q(\mathbf{x})$ using approximately $D_{KL}(q(\mathbf{x}) \| p(\mathbf{x}))$ bits on average for any distributions p and q , for which only p is available to the receiver beforehand. When applied to $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, Algorithms 3 and 4 transmit $\mathbf{x}_T, \dots, \mathbf{x}_0$ in sequence using a total expected codelength equal to Eq. (5). The receiver,

Thuật toán 3 Gửi \mathbf{x}_0 1: Gửi

```

 $\mathbf{x}_T \sim q(\mathbf{x}_T | \mathbf{x}_0)$  sử dụng  $p(\mathbf{x}_T)$  2: với
 $t = T - 1, \dots, 2, 1$  do 3:
Gửi  $\mathbf{x}_t \sim q(\mathbf{x}_t | \mathbf{x}_{t+1}, \mathbf{x}_0)$  sử dụng  $p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})$  4: end
for
5: Gửi  $\mathbf{x}_0$  sử dụng  $p_\theta(\mathbf{x}_0 | \mathbf{x}_1)$ 

```

Thuật toán 4 Nhận

```

1: Nhận  $\mathbf{x}_T$  sử dụng  $p(\mathbf{x}_T)$ 
2: với  $t = T - 1, \dots, 1,$ 
 $\dots, 0$  do 3: Nhận  $\mathbf{x}_t$  sử dụng  $p_\theta(\mathbf{x}_t | \mathbf{x}_{t+1})$ 
4: end for
5: return  $\mathbf{x}_0$ 

```

Chúng tôi thấy rằng việc huấn luyện các mô hình của chúng tôi về giới hạn biến phân thực sự mang lại độ dài mã tốt hơn so với huấn luyện về mục tiêu đơn giản hóa, như mong đợi, nhưng mục tiêu sau mang lại chất lượng mẫu tốt nhất. Xem Hình 1 để biết các mẫu CIFAR10 và CelebA-HQ 256 × 256 , Hình 3 và Hình 4 về các mẫu LSUN 256 × 256 [71] và Phụ lục D để biết thêm.

4.2 Tham số hóa ngược quá trình và loại bỏ mục tiêu đào tạo

Trong Bảng 2, chúng tôi trình bày tác động chất lượng mẫu của việc tham số hóa quy trình ngược và mục tiêu đào tạo (Phần 3.2). Chúng tôi thấy rằng tùy chọn cơ sở để dự đoán μ^* chỉ hoạt động tốt khi được huấn luyện về giới hạn biến phân thực sự thay vì sai số bình phương trung bình không có trọng số, một mục tiêu đơn giản hóa giống như phương trình. (14). Chúng tôi cũng thấy rằng việc học các phương sai của quá trình ngược lại (bằng cách kết hợp đường chéo được tham số hóa $\Sigma\theta(x_t)$ vào giới hạn biến phân) dẫn đến việc đào tạo không ổn định và chất lượng mẫu kém hơn so với các phương sai cố định. Dự đoán , như chúng tôi đã đề xuất, thực hiện xấp xỉ cũng như dự đoán μ^* khi được huấn luyện về giới hạn biến phân với các phương sai cố định, nhưng tốt hơn nhiều khi được huấn luyện với mục tiêu đơn giản hóa của chúng tôi.

4.3 Mã hóa lũy tiến

Bảng 1 cũng cho thấy độ dài mã của các mô hình CIFAR10 của chúng tôi. Khoảng cách giữa đào tạo và kiểm tra tối đa là 0,03 bit cho mỗi chiều, có thể so sánh với khoảng cách được báo cáo với các mô hình dựa trên khả năng khác và chỉ ra rằng mô hình khuếch tán của chúng tôi không phù hợp quá mức (xem Phụ lục D để biết hình ảnh trực quan lân cận gần nhất). Tuy nhiên, mặc dù độ dài mã không mất dữ liệu của chúng tôi tốt hơn so với các ước tính lớn được báo cáo cho các mô hình dựa trên năng lượng và so khớp điểm bằng cách lấy mẫu tầm quan trọng đã được ủ [11], nhưng chúng không cạnh tranh với các loại mô hình tổng quát dựa trên khả năng khác [7].

Vì các mẫu của chúng tôi dù sao cũng có chất lượng cao nên chúng tôi kết luận rằng các mô hình khuếch tán có độ lệch cảm ứng khiến chúng trở thành máy nén hao tuyệt vời. Xử lý các thuật ngữ ràng buộc biến thiên $L_1 + \dots + LT$ là tốc độ và L_0 là độ mèo, mô hình CIFAR10 của chúng tôi với các mẫu chất lượng cao nhất có tốc độ 1,78 bit/độ mờ và độ mèo là 1,97 bit/độ mờ, tương đương với giá trị trung bình gốc sai số bình phương 0,95 trên thang điểm từ 0 đến 255. Hơn nữa độ dài mã không mất dữ liệu mô tả các biến dạng không thể nhận thấy.

Nén tồn hao lũy tiến Chúng ta có thể thăm dò sâu hơn về hành vi biến dạng tốc độ của mô hình bằng cách giới thiệu một mã tồn hao lũy tiến phản chiếu dạng biểu thức. (5): xem Thuật toán 3 và 4, giả định quyền truy cập vào một thủ tục, chẳng hạn như mã hóa ngẫu nhiên tối thiểu [19, 20], có thể truyền một mẫu $x \sim q(x)$ bằng cách sử dụng khoảng $D_{KL}(q(x) \| p(x))$ trung bình các bit đối với bất kỳ phân bố p và q nào, trong đó chỉ có p có sẵn cho người nhận trước đó. Khi áp dụng cho $\mathbf{x}_0 \sim q(\mathbf{x}_0)$, Thuật toán 3 và 4 truyền $\mathbf{x}_T, \dots, \mathbf{x}_0$ theo trình tự sử dụng tổng chiều dài mã dự kiến bằng biểu thức. (5). Người nhận,

at any time t , has the partial information \mathbf{x}_t fully available and can progressively estimate:

$$\mathbf{x}_0 \approx \hat{\mathbf{x}}_0 = (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\mathbf{x}_t)) / \sqrt{\bar{\alpha}_t} \quad (15)$$

due to Eq. (4). (A stochastic reconstruction $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ is also valid, but we do not consider it here because it makes distortion more difficult to evaluate.) Figure 5 shows the resulting rate-distortion plot on the CIFAR10 test set. At each time t , the distortion is calculated as the root mean squared error $\sqrt{\|\mathbf{x}_0 - \hat{\mathbf{x}}_0\|^2/D}$, and the rate is calculated as the cumulative number of bits received so far at time t . The distortion decreases steeply in the low-rate region of the rate-distortion plot, indicating that the majority of the bits are indeed allocated to imperceptible distortions.

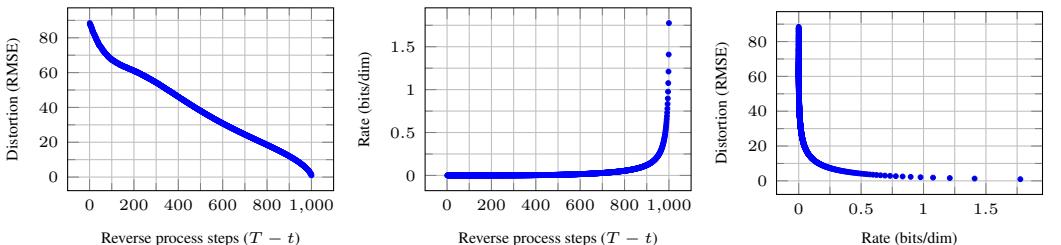


Figure 5: Unconditional CIFAR10 test set rate-distortion vs. time. Distortion is measured in root mean squared error on a $[0, 255]$ scale. See Table 4 for details.

Progressive generation We also run a progressive unconditional generation process given by progressive decompression from random bits. In other words, we predict the result of the reverse process, $\hat{\mathbf{x}}_0$, while sampling from the reverse process using Algorithm 2. Figures 6 and 10 show the resulting sample quality of $\hat{\mathbf{x}}_0$ over the course of the reverse process. Large scale image features appear first and details appear last. Figure 7 shows stochastic predictions $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ with \mathbf{x}_t frozen for various t . When t is small, all but fine details are preserved, and when t is large, only large scale features are preserved. Perhaps these are hints of conceptual compression [18].

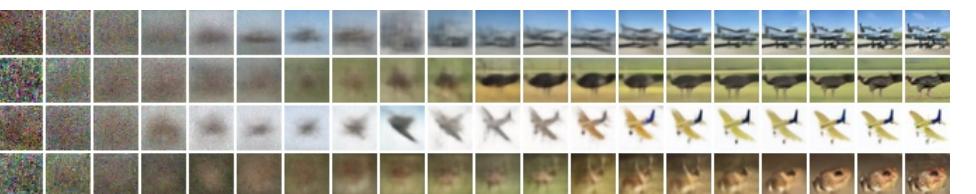


Figure 6: Unconditional CIFAR10 progressive generation ($\hat{\mathbf{x}}_0$ over time, from left to right). Extended samples and sample quality metrics over time in the appendix (Figs. 10 and 14).



Figure 7: When conditioned on the same latent, CelebA-HQ 256×256 samples share high-level attributes. Bottom-right quadrants are \mathbf{x}_t , and other quadrants are samples from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

Connection to autoregressive decoding Note that the variational bound (5) can be rewritten as:

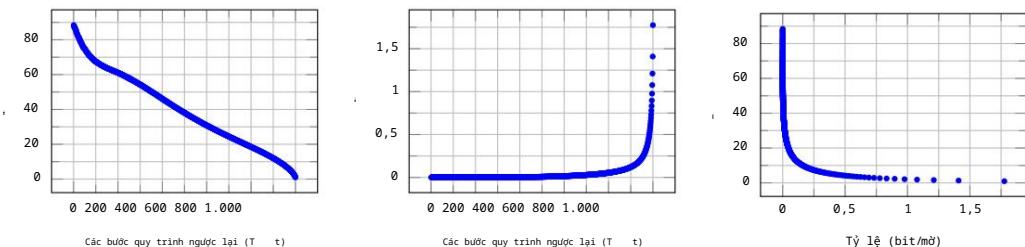
$$L = D_{\text{KL}}(q(\mathbf{x}_T) \| p(\mathbf{x}_T)) + \mathbb{E}_q \left[\sum_{t \geq 1} D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \right] + H(\mathbf{x}_0) \quad (16)$$

(See Appendix A for a derivation.) Now consider setting the diffusion process length T to the dimensionality of the data, defining the forward process so that $q(\mathbf{x}_t|\mathbf{x}_0)$ places all probability mass on \mathbf{x}_0 with the first t coordinates masked out (i.e. $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ masks out the t^{th} coordinate), setting $p(\mathbf{x}_T)$ to place all mass on a blank image, and, for the sake of argument, taking $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ to

tại bất kỳ thời điểm t nào, có sẵn một phần thông tin \mathbf{x}_t và có thể ước tính dần dần:

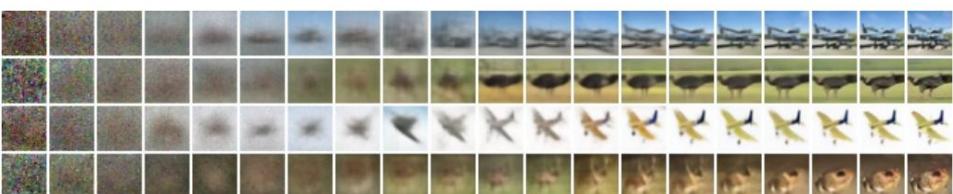
$$\mathbf{x}_0 \approx \hat{\mathbf{x}}_0 = \mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\mathbf{x}_t) / \sqrt{\bar{\alpha}_t} \quad (15)$$

do phương trình. (4). (Việc tái tạo ngẫu nhiên $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ cũng hợp lệ, nhưng chúng tôi không xem xét nó ở đây vì nó làm cho việc đánh giá độ biến dạng trở nên khó khăn hơn.) Hình 5 cho thấy biểu đồ tần số lẻ-độ biến dạng thu được trên bộ kiểm tra CIFAR10. Tại mỗi thời điểm t , độ méo được tính bằng sai số bình phương trung bình gốc $\mathbf{x}_0 - \hat{\mathbf{x}}_0$ và tốc độ được tính bằng số bit tích lũy nhận được cho đến thời điểm t . Độ méo giảm mạnh ở vùng tốc độ thấp của biểu đồ độ méo tốc độ, cho thấy rằng phần lớn các bit thực sự được phân bổ cho các biến dạng không thể nhận thấy.



Hình 5: Độ méo tốc độ của bộ kiểm tra CIFAR10 vô điều kiện theo thời gian. Độ méo được đo bằng sai số bình phương trung bình gốc trên thang đo $[0, 255]$. Xem Bảng 4 để biết chi tiết.

Tạo lũy tiến Chúng tôi cũng chạy quy trình tạo lũy tiến vô điều kiện được đưa ra bằng cách giải nén lũy tiến từ các bit ngẫu nhiên. Nói cách khác, chúng tôi dự đoán kết quả của quy trình ngược lại, $\hat{\mathbf{x}}_0$, trong khi lấy mẫu từ quy trình ngược lại bằng Thuật toán 2. Hình 6 và 10 cho thấy chất lượng mẫu thu được là $\hat{\mathbf{x}}_0$ trong suốt quá trình ngược lại. Các đặc điểm của hình ảnh tần số lẻ lớn xuất hiện đầu tiên và các chi tiết xuất hiện sau cùng. Hình 7 cho thấy các dự đoán ngẫu nhiên $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ với \mathbf{x}_t cố định cho các t khác nhau. Khi t nhỏ, tất cả trừ các chi tiết nhỏ được giữ nguyên và khi t lớn, chỉ các đặc điểm quy mô lớn được giữ nguyên. Có lẽ đây là những gợi ý về việc nén khái niệm [18].



Hình 6: Tạo lũy tiến CIFAR10 vô điều kiện ($\hat{\mathbf{x}}_0$ theo thời gian, từ trái sang phải). Các mẫu mở rộng và số liệu chất lượng mẫu theo thời gian trong phần phụ lục (Hình 10 và 14).



Hình 7: Khi được điều hòa trên cùng một mức tiềm ẩn, các mẫu CelebA-HQ 256×256 có chung các thuộc tính cấp cao. Các góc phần tư dưới cùng bên phải là \mathbf{x}_t và các góc phần tư khác là mẫu từ $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

Kết nối với giải mã tự hồi quy Lưu ý rằng giới hạn biến phân (5) có thể được viết lại thành:

$$L = D_{\text{KL}}(q(\mathbf{x}_T) \| p(\mathbf{x}_T)) + \mathbb{E}_q \left[\sum_{t \geq 1} D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \right] + H(\mathbf{x}_0) \quad (16)$$

(Xem Phụ lục A để biết đạo hàm.) Bây giờ hãy xem xét việc đặt độ dài quá trình khuếch tán T theo chiều của dữ liệu, xác định quá trình chuyển tiếp sao cho $q(\mathbf{x}_t|\mathbf{x}_0)$ đặt tất cả khối lượng xác suất lên \mathbf{x}_0 với tần số t đầu tiên bị che đi (tức là $q(\mathbf{x}_t|\mathbf{x}_0)$ che giấu tần số t), đặt $p(\mathbf{x}_T)$ để đặt toàn bộ khối lượng lên một ảnh trắng, và, để tranh luận, lấy $p_\theta(\mathbf{x}_t|\mathbf{x}_0)$ thành

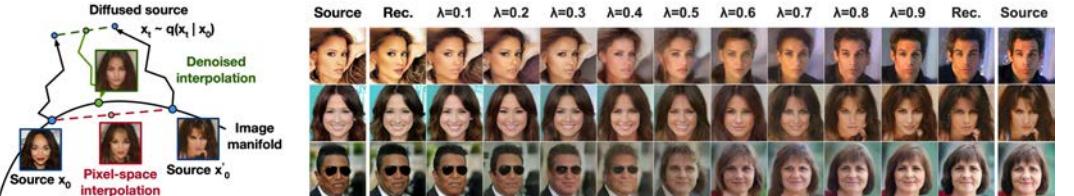


Figure 8: Interpolations of CelebA-HQ 256x256 images with 500 timesteps of diffusion.

be a fully expressive conditional distribution. With these choices, $D_{KL}(q(\mathbf{x}_T) \parallel p(\mathbf{x}_T)) = 0$, and minimizing $D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$ trains p_θ to copy coordinates $t+1, \dots, T$ unchanged and to predict the t^{th} coordinate given $t+1, \dots, T$. Thus, training p_θ with this particular diffusion is training an autoregressive model.

We can therefore interpret the Gaussian diffusion model (2) as a kind of autoregressive model with a generalized bit ordering that cannot be expressed by reordering data coordinates. Prior work has shown that such reorderings introduce inductive biases that have an impact on sample quality [38], so we speculate that the Gaussian diffusion serves a similar purpose, perhaps to greater effect since Gaussian noise might be more natural to add to images compared to masking noise. Moreover, the Gaussian diffusion length is not restricted to equal the data dimension; for instance, we use $T = 1000$, which is less than the dimension of the $32 \times 32 \times 3$ or $256 \times 256 \times 3$ images in our experiments. Gaussian diffusions can be made shorter for fast sampling or longer for model expressiveness.

4.4 Interpolation

We can interpolate source images $\mathbf{x}_0, \mathbf{x}'_0 \sim q(\mathbf{x}_0)$ in latent space using q as a stochastic encoder, $\mathbf{x}_t, \mathbf{x}'_t \sim q(\mathbf{x}_t | \mathbf{x}_0)$, then decoding the linearly interpolated latent $\bar{\mathbf{x}}_t = (1 - \lambda)\mathbf{x}_0 + \lambda\mathbf{x}'_0$ into image space by the reverse process, $\bar{\mathbf{x}}_0 \sim p(\mathbf{x}_0 | \bar{\mathbf{x}}_t)$. In effect, we use the reverse process to remove artifacts from linearly interpolating corrupted versions of the source images, as depicted in Fig. 8 (left). We fixed the noise for different values of λ so \mathbf{x}_t and \mathbf{x}'_t remain the same. Fig. 8 (right) shows interpolations and reconstructions of original CelebA-HQ 256×256 images ($t = 500$). The reverse process produces high-quality reconstructions, and plausible interpolations that smoothly vary attributes such as pose, skin tone, hairstyle, expression and background, but not eyewear. Larger t results in coarser and more varied interpolations, with novel samples at $t = 1000$ (Appendix Fig. 9).

5 Related Work

While diffusion models might resemble flows [9, 46, 10, 32, 5, 16, 23] and VAEs [33, 47, 37], diffusion models are designed so that q has no parameters and the top-level latent \mathbf{x}_T has nearly zero mutual information with the data \mathbf{x}_0 . Our ϵ -prediction reverse process parameterization establishes a connection between diffusion models and denoising score matching over multiple noise levels with annealed Langevin dynamics for sampling [55, 56]. Diffusion models, however, admit straightforward log likelihood evaluation, and the training procedure explicitly trains the Langevin dynamics sampler using variational inference (see Appendix C for details). The connection also has the reverse implication that a certain weighted form of denoising score matching is the same as variational inference to train a Langevin-like sampler. Other methods for learning transition operators of Markov chains include infusion training [2], variational walkback [15], generative stochastic networks [1], and others [50, 54, 36, 42, 35, 65].

By the known connection between score matching and energy-based modeling, our work could have implications for other recent work on energy-based models [67–69, 12, 70, 13, 11, 41, 17, 8]. Our rate-distortion curves are computed over time in one evaluation of the variational bound, reminiscent of how rate-distortion curves can be computed over distortion penalties in one run of annealed importance sampling [24]. Our progressive decoding argument can be seen in convolutional DRAW and related models [18, 40] and may also lead to more general designs for subscale orderings or sampling strategies for autoregressive models [38, 64].



Hình 8: Nội suy của hình ảnh CelebA-HQ 256x256 với 500 dấu thời gian khuếch tán.

là một phân phối có điều kiện biến cảm đầy đủ. Với những lựa chọn này, $D_{KL}(q(\mathbf{x}_T) \parallel p(\mathbf{x}_T)) = 0$, và cực tiểu hóa $D_{KL}(q(\mathbf{x}_t | \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_t | \mathbf{x}_0))$ huấn luyện p_θ để sao chép tọa độ $t+1, \dots, T$ không thay đổi và dự đoán tọa độ t cho trước $t+1, \dots, T$. Do đó, việc huấn luyện p_θ với sự khuếch tán đặc biệt này đang huấn luyện một mô hình tự hồi quy.

Do đó, chúng ta có thể hiểu mô hình khuếch tán Gaussian (2) là một loại mô hình tự hồi quy với thứ tự bit tổng quát không thể biểu thị bằng cách sắp xếp lại tọa độ dữ liệu. Công việc trước đây đã chỉ ra rằng việc sắp xếp lại như vậy tạo ra các sai lệch quy nạp có ảnh hưởng đến chất lượng mẫu [38], vì vậy chúng tôi suy đoán rằng sự khuếch tán Gaussian phục vụ một mục đích tương tự, có lẽ sẽ có tác dụng lớn hơn vì nhiều Gaussian có thể được thêm vào hình ảnh một cách tự nhiên hơn so với che giấu tiếng ồn. Hơn nữa, độ dài khuếch tán Gaussian không bị giới hạn bằng kích thước dữ liệu; chẳng hạn, chúng tôi sử dụng $T = 1000$, nhỏ hơn kích thước của hình ảnh $32 \times 32 \times 3$ hoặc $256 \times 256 \times 3$ trong các thử nghiệm của chúng tôi.

Khuếch tán Gaussian có thể được thực hiện ngắn hơn để lấy mẫu nhanh hoặc dài hơn để biểu đạt mô hình.

4.4 Nội suy

có thể nội suy ảnh nguồn x_0 , x thành không gian ảnh x_t , $\theta = q(x_0)$ trong không gian tiềm ẩn sử dụng q làm bộ mã hóa ngẫu nhiên. Chúng ta trình $t = q(x_t | x_0)$, sau đó giải mã tiềm ẩn nội suy tuyến tính của ảnh nguồn, như được mô tả trong Hình 8 (trái). Chúng tôi đã sửa chữa cho các giá trị khác nhau của λ để x_t và x vẫn giữ nguyên. Hình 8 (phải) hiển thị các nội suy và tái tạo của hình ảnh CelebA-HQ 256×256 ban đầu ($t = 500$). Quá trình ngược lại tạo ra các bản tái tạo chất lượng cao và các phép nội suy hợp lý giúp thay đổi các thuộc tính một cách mượt mà như tư thế, tông màu da, kiểu tóc, biến cảm và phòng nền chứ không phải kính mắt. T lớn hơn dẫn đến các phép nội suy thô hơn và đa dạng hơn, với các mẫu mới ở mức $t = 1000$ (Phụ lục Hình 9).

5 công việc liên quan

Trong khi các mô hình khuếch tán có thể giống với các dòng [9, 46, 10, 32, 5, 16, 23] và VAE [33, 47, 37], các mô hình khuếch tán được thiết kế sao cho q không có tham số và x_T tiềm ẩn cấp cao nhất gần như có không thông tin lẫn nhau với dữ liệu x_0 . Việc tham số hóa quá trình đảo ngược dự đoán của chúng tôi thiết lập mối liên hệ giữa các mô hình khuếch tán và so khớp điểm khứ nhiều trên nhiều mức nhiều với động lực học Langevin đã được lấy mẫu [55, 56]. Tuy nhiên, các mô hình khuếch tán thường nhận đánh giá khả năng ghi nhận ký đơn giản và quy trình đào tạo huấn luyện rõ ràng bộ lấy mẫu động lực học Langevin bằng cách sử dụng suy luận biến phân (xem Phụ lục C để biết chi tiết). Mối liên hệ này cũng có hàm ý ngược lại rằng một dạng khớp điểm khứ nhiều có trọng số nhất định cũng giống như suy luận biến phân để huấn luyện một bộ lấy mẫu giống Langevin. Các phương pháp khác để học các toán tử chuyển tiếp của chuỗi Markov bao gồm đào tạo truyền tải [2], quay lại biến thể [15], mạng ngẫu nhiên tổng quát [1] và các phương pháp khác [50, 54, 36, 42, 35, 65].

Bằng mối liên hệ đã biết giữa việc so sánh điểm số và mô hình hóa dựa trên năng lượng, công việc của chúng tôi có thể có ý nghĩa đối với công việc gần đây về các mô hình dựa trên năng lượng [67–69, 12, 70, 13, 11, 41, 17, 8]. Các đường cong biến dạng tốc độ của chúng tôi được tính toán theo thời gian trong một đánh giá về giới hạn biến thiên, gợi nhớ đến cách các đường cong biến dạng tốc độ có thể được tính toán dựa trên các hình phạt biến dạng trong một lần lấy mẫu tầm quan trọng đã [24]. Đôi số giải mã lũy tiến của chúng tôi có thể được nhìn thấy trong DRAW tích chập và các mô hình liên quan [18, 40] và cũng có thể dẫn đến các thiết kế tổng quát hơn cho thứ tự quy mô phụ hoặc chiến lược lấy mẫu cho các mô hình tự hồi quy [38, 64].

6 Conclusion

We have presented high quality image samples using diffusion models, and we have found connections among diffusion models and variational inference for training Markov chains, denoising score matching and annealed Langevin dynamics (and energy-based models by extension), autoregressive models, and progressive lossy compression. Since diffusion models seem to have excellent inductive biases for image data, we look forward to investigating their utility in other data modalities and as components in other types of generative models and machine learning systems.

Broader Impact

Our work on diffusion models takes on a similar scope as existing work on other types of deep generative models, such as efforts to improve the sample quality of GANs, flows, autoregressive models, and so forth. Our paper represents progress in making diffusion models a generally useful tool in this family of techniques, so it may serve to amplify any impacts that generative models have had (and will have) on the broader world.

Unfortunately, there are numerous well-known malicious uses of generative models. Sample generation techniques can be employed to produce fake images and videos of high profile figures for political purposes. While fake images were manually created long before software tools were available, generative models such as ours make the process easier. Fortunately, CNN-generated images currently have subtle flaws that allow detection [62], but improvements in generative models may make this more difficult. Generative models also reflect the biases in the datasets on which they are trained. As many large datasets are collected from the internet by automated systems, it can be difficult to remove these biases, especially when the images are unlabeled. If samples from generative models trained on these datasets proliferate throughout the internet, then these biases will only be reinforced further.

On the other hand, diffusion models may be useful for data compression, which, as data becomes higher resolution and as global internet traffic increases, might be crucial to ensure accessibility of the internet to wide audiences. Our work might contribute to representation learning on unlabeled raw data for a large range of downstream tasks, from image classification to reinforcement learning, and diffusion models might also become viable for creative uses in art, photography, and music.

Acknowledgments and Disclosure of Funding

This work was supported by ONR PECASE and the NSF Graduate Research Fellowship under grant number DGE-1752814. Google's TensorFlow Research Cloud (TFRC) provided Cloud TPUs.

References

- [1] Guillaume Alain, Yoshua Bengio, Li Yao, Jason Yosinski, Eric Thibodeau-Laufer, Saizheng Zhang, and Pascal Vincent. GSNs: generative stochastic networks. *Information and Inference: A Journal of the IMA*, 5(2):210–249, 2016.
- [2] Florian Bordes, Sina Honari, and Pascal Vincent. Learning to generate samples from noise through infusion training. In *International Conference on Learning Representations*, 2017.
- [3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.
- [4] Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your GAN is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020.
- [5] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583, 2018.
- [6] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad, and Pieter Abbeel. PixelSNAIL: An improved autoregressive generative model. In *International Conference on Machine Learning*, pages 863–871, 2018.
- [7] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

6 Kết luận

Chúng tôi đã trình bày các mẫu hình ảnh chất lượng cao bằng cách sử dụng mô hình khuếch tán và chúng tôi đã tìm thấy mối liên hệ giữa các mô hình khuếch tán và suy luận biến phân để huấn luyện chuỗi Markov, khop điểm khứ nhiều và động lực học Langevin đã được ủ (và các mô hình dựa trên năng lượng theo phần mở rộng), mô hình tự hồi quy và tốn hao lũy tiến nén. Vì các mô hình khuếch tán dường như có độ lệch quy nạp tuyệt vời đối với dữ liệu hình ảnh, nên chúng tôi mong muốn nghiên cứu tiện ích của chúng trong các phương thức dữ liệu khác và như các thành phần trong các loại mô hình tổng quát và hệ thống máy học khác.

Tác động rộng hơn

Công việc của chúng tôi về các mô hình khuếch tán có phạm vi tương tự như công việc hiện có trên các loại mô hình tổng quát sâu khác, chẳng hạn như nỗ lực cải thiện chất lượng mẫu của GAN, dòng chảy, mô hình tự hồi quy, v.v. Bài viết của chúng tôi thể hiện sự tiên bộ trong việc biến các mô hình khuếch tán trở thành một công cụ hữu ích nói chung trong nhóm kỹ thuật này, do đó, nó có thể dùng để khuếch đại bất kỳ tác động nào mà các mô hình sinh sản đã (và sẽ có) đối với thế giới rộng lớn hơn.

Thật không may, có rất nhiều cách sử dụng độc hại nổi tiếng của các mô hình tổng quát. Kỹ thuật tạo mẫu có thể được sử dụng để tạo ra hình ảnh và video giả mạo về các nhân vật nổi tiếng nhằm mục đích chính trị. Mặc dù các hình ảnh giả đã được tạo thủ công từ rất lâu trước khi có các công cụ phần mềm, nhưng các mô hình tổng hợp như của chúng tôi giúp quá trình này trở nên dễ dàng hơn. May mắn thay, các hình ảnh do CNN tạo ra hiện có những sai sót nhỏ cho phép phát hiện [62], nhưng những cải tiến trong các mô hình tổng hợp có thể khiến việc này trở nên khó khăn hơn. Các mô hình sáng tạo cũng phản ánh những thành kiến trong bộ dữ liệu mà chúng được đào tạo. Vì nhiều bộ dữ liệu lớn được thu thập từ internet bằng hệ thống tự động nên khó có thể loại bỏ những thành kiến này, đặc biệt là khi hình ảnh không được gán nhãn. Nếu các mẫu từ các mô hình tổng quát được đào tạo trên các bộ dữ liệu này sinh sôi này nở trên internet, thì những thành kiến này sẽ chỉ được cung cấp thêm.

Mặt khác, các mô hình phổ biến có thể hữu ích cho việc nén dữ liệu, khi dữ liệu có độ phân giải cao hơn và khi lưu lượng truy cập internet toàn cầu tăng lên, có thể rất quan trọng để đảm bảo khả năng truy cập Internet cho nhiều đối tượng. Công việc của chúng tôi có thể đóng góp vào việc học biểu diễn trên dữ liệu thô chưa được gắn nhãn cho một loạt các nhiệm vụ tiếp theo, từ phân loại hình ảnh đến học tăng cường và các mô hình khuếch tán cũng có thể trở nên khả thi cho các mục đích sử dụng sáng tạo trong nghệ thuật, nhiếp ảnh và âm nhạc.

Lời cảm ơn và tiết lộ tài trợ

Công trình này được hỗ trợ bởi ONR PECASE và Học bổng nghiên cứu sau đại học của NSF theo số cấp DGE-1752814. Đám mê nghiên cứu TensorFlow (TFRC) của Google đã cung cấp Cloud TPU.

Người giới thiệu

- [1] Guillaume Alain, Yoshua Bengio, Li Yao, Jason Yosinski, Eric Thibodeau-Laufer, Saizheng Zhang và Pascal Vincent. GSN: mạng ngẫu nhiên tổng quát. Thông tin và suy luận: Tạp chí của IMA, 5(2):210–249, 2016.
- [2] Florian Bordes, Sina Honari và Pascal Vincent. Học cách tạo mẫu từ tiếng ồn thông qua truyền dịch đào tạo. Trong Hội nghị quốc tế về đại diện học tập, 2017.
- [3] Andrew Brock, Jeff Donahue và Karen Simonyan. Đào tạo GAN quy mô lớn để có độ chính xác cao tự nhiên tổng hợp hình ảnh. Trong Hội nghị quốc tế về đại diện học tập, 2019.
- [4] Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao và Yoshua Bengio. GAN của bạn bí mật là một mô hình dựa trên năng lượng và bạn nên sử dụng phương pháp lấy mẫu tiềm ẩn do bộ phân biệt đối xử điều khiển. bản in trước arXiv arXiv:2003.06060, 2020.
- [5] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt và David K Duvenaud. Phương trình vi phân thông thường thần kinh. Trong Những tiên bộ trong hệ thống xử lý thông tin thần kinh, trang 6571–6583, 2018.
- [6] Xi Chen, Nikhil Mishra, Mostafa Rohaninejad và Pieter Abbeel. PixelSNAIL: Một mô hình tạo sinh tự hồi quy được cải tiến. Trong Hội nghị quốc tế về Học máy, trang 863–871, 2018.
- [7] Rewon Child, Scott Gray, Alec Radford và Ilya Sutskever. Tạo chuỗi dài với thưa thoát máy biến áp. bản in trước arXiv arXiv:1904.10509, 2019.

- [8] Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. Residual energy-based models for text generation. *arXiv preprint arXiv:2004.11714*, 2020.
- [9] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.
- [10] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016.
- [11] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems*, pages 3603–3613, 2019.
- [12] Ruiqi Gao, Yang Lu, Junpei Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning generative ConvNets via multi-grid modeling and sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9155–9164, 2018.
- [13] Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7518–7528, 2020.
- [14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.
- [15] Anirudh Goyal, Nan Rosemary Ke, Surya Ganguli, and Yoshua Bengio. Variational walkback: Learning a transition operator as a stochastic recurrent net. In *Advances in Neural Information Processing Systems*, pages 4392–4402, 2017.
- [16] Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.
- [17] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.
- [18] Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In *Advances In Neural Information Processing Systems*, pages 3549–3557, 2016.
- [19] Prahladh Harsha, Rahul Jain, David McAllester, and Jaikumar Radhakrishnan. The communication complexity of correlation. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 10–23. IEEE, 2007.
- [20] Marton Havasi, Robert Peharz, and José Miguel Hernández-Lobato. Minimal random code learning: Getting bits back from compressed model parameters. In *International Conference on Learning Representations*, 2019.
- [21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017.
- [22] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.
- [23] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, 2019.
- [24] Sicong Huang, Alireza Makhzani, Yanshuai Cao, and Roger Grosse. Evaluating lossy compression rates of deep generative models. In *International Conference on Machine Learning*, 2020.
- [25] Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves, and Koray Kavukcuoglu. Video pixel networks. In *International Conference on Machine Learning*, pages 1771–1779, 2017.
- [26] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In *International Conference on Machine Learning*, pages 2410–2419, 2018.
- [27] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.
- [28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages
- [8] Đặng Yuntian, Anton Bakhtin, Myle Ott, Arthur Szlam và Marc'Aurelio Ranzato. Các mô hình dựa trên năng lượng để tạo văn bản. bản in trước arXiv arXiv:2004.11714, 2020.
- [9] Laurent Dinh, David Krueger, và Yoshua Bengio. NICE: Ước lượng các thành phần độc lập phi tuyến tính. bản in trước arXiv arXiv:1410.8516, 2014.
- [10] Laurent Dinh, Jascha Sohl-Dickstein và Samy Bengio. Ước tính mật độ bằng Real NVP. arXiv bản in trước arXiv:1605.08803, 2016.
- [11] Yilun Du và Igor Mordatch. Tạo và mô hình hóa ngầm với các mô hình dựa trên năng lượng. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, trang 3603–3613, 2019.
- [12] Ruiqi Gao, Yang Lu, Junpei Chu, Song-Chun Zhu, và Ying Nian Wu. Học các ConvNet tổng quát thông qua mô hình hóa và lấy mẫu trên nhiều lứa. Trong *Ký yếu của Hội nghị IEEE về Thị giác máy tính và Nhận dạng mẫu*, trang 9155–9164, 2018.
- [13] Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai và Ying Nian Wu. Ước tính tương phản dòng chảy của các mô hình dựa trên năng lượng. Trong *Ký yếu của Hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu*, trang 7518–7528, 2020.
- [14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville và Yoshua Bengio. Mạng lưới đối nghịch sáng tạo. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, trang 2672–2680, 2014.
- [15] Anirudh Goyal, Nan Rosemary Ke, Surya Ganguli và Yoshua Bengio. Quay lại biến thể: Học toán tử chuyển tiếp như một mạng hồi quy ngẫu nhiên. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, trang 4392–4402, 2017.
- [16] Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt và David Duvenaud. FFJORD: Động lực liên tục dạng tự do cho các mô hình thế hệ có thể đảo ngược có thể mở rộng. Trong *Hội nghị quốc tế về đại diện học tập*, 2019.
- [17] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi và Kevin Swersky. Bộ phân loại của bạn bí mật là một mô hình dựa trên năng lượng và bạn nên coi nó như một mô hình. Trong *Hội nghị quốc tế về đại diện học tập*, 2020.
- [18] Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka và Daan Wierstra. Hướng tới nén khái niệm. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, trang 3549–3557, 2016.
- [19] Prahladh Harsha, Rahul Jain, David McAllester và Jaikumar Radhakrishnan. Sự phức tạp trong giao tiếp của mối tương quan. Trong *Hội nghị thường niên lần thứ 22 của IEEE về độ phức tạp tính toán (CCC'07)*, trang 10–23. IEEE, 2007.
- [20] Marton Havasi, Robert Peharz và José Miguel Hernández-Lobato. Học mã ngẫu nhiên tối thiểu: Lấy lại bit từ các tham số mô hình nén. Trong *Hội nghị quốc tế về đại diện học tập*, 2019.
- [21] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, và Sepp Hochreiter. Các GAN được huấn luyện theo quy tắc cập nhật theo thang thời gian hai sê hội tụ về trạng thái cân bằng Nash cục bộ. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, trang 6626–6637, 2017.
- [22] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, và Alexander Lerchner. beta-VAE: Học các khái niệm trực quan cơ bản với khung biến thể bị ràng buộc. Trong *Hội nghị quốc tế về đại diện học tập*, 2017.
- [23] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, và Pieter Abbeel. Flow++: Cải thiện các mô hình sinh học dựa trên dòng chảy với thiết kế kiến trúc và giải lượng tử biến phân. Trong *Hội nghị quốc tế về học máy*, 2019.
- [24] Sicong Huang, Alireza Makhzani, Yanshuai Cao, và Roger Grosse. Đánh giá tốc độ nén tổn hao của các mô hình sinh sâu. Trong *Hội nghị quốc tế về học máy*, 2020.
- [25] Nal Kalchbrenner, Aaron van den Oord, Karen Simonyan, Ivo Danihelka, Oriol Vinyals, Alex Graves và Koray Kavukcuoglu. Mạng pixel video. Trong *Hội nghị quốc tế về học máy*, trang 1771–1779, 2017.
- [26] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman và Koray Kavukcuoglu. Tổng hợp âm thanh thần kinh hiệu quả. Trong *Hội nghị quốc tế về học máy*, trang 2410–2419, 2018.
- [27] Tero Karras, Timo Aila, Samuli Laine, và Jaakko Lehtinen. Sự phát triển dần dần của GAN để cải thiện chất lượng, tính ổn định và sự thay đổi. Trong *Hội nghị quốc tế về đại diện học tập*, 2018.
- [28] Tero Karras, Samuli Laine và Timo Aila. Kiến trúc trình tạo dựa trên kiểu cho các mạng đối thủ tổng quát. Trong *Ký yếu của Hội nghị IEEE về Thị giác máy tính và Nhận dạng mẫu*, các trang

4401–4410, 2019.

- [29] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676v1*, 2020.
- [30] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.
- [31] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- [32] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pages 10215–10224, 2018.
- [33] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [34] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.
- [35] John Lawson, George Tucker, Bo Dai, and Rajesh Ranganath. Energy-inspired models: Learning with sampler-induced distributions. In *Advances in Neural Information Processing Systems*, pages 8501–8513, 2019.
- [36] Daniel Levy, Matt D. Hoffman, and Jascha Sohl-Dickstein. Generalizing Hamiltonian Monte Carlo with neural networks. In *International Conference on Learning Representations*, 2018.
- [37] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. BIVA: A very deep hierarchy of latent variables for generative modeling. In *Advances in Neural Information Processing Systems*, pages 6548–6558, 2019.
- [38] Jacob Menick and Nal Kalchbrenner. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *International Conference on Learning Representations*, 2019.
- [39] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.
- [40] Alex Nichol. VQ-DRAW: A sequential discrete VAE. *arXiv preprint arXiv:2003.01599*, 2020.
- [41] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of MCMC-based maximum likelihood learning of energy-based models. *arXiv preprint arXiv:1903.12370*, 2019.
- [42] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information Processing Systems*, pages 5233–5243, 2019.
- [43] Georg Ostrovski, Will Dabney, and Remi Munos. Autoregressive quantile networks for generative modeling. In *International Conference on Machine Learning*, pages 3936–3945, 2018.
- [44] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. WaveGlow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.
- [45] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847, 2019.
- [46] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.
- [47] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014.
- [48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.
- [49] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 901–909, 2016.
- [50] Tim Salimans, Diederik Kingma, and Max Welling. Markov Chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226, 2015.

4401–4410, 2019.

- [29] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen và Timo Aila. Đào tạo các mạng đối thủ tổng quát với dữ liệu hạn chế. bản in trước arXiv arXiv:2006.06676v1, 2020.
- [30] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen và Timo Aila. Phân tích và cải thiện chất lượng hình ảnh của StyleGAN. Trong Ký yếu của Hội nghị IEEE/CVF về Thị giác máy tính và Nhận dạng mẫu, trang 8110-8119, 2020.
- [31] Diederik P Kingma và Jimmy Ba. Adam: Một phương pháp tối ưu hóa ngẫu nhiên. Trong Hội nghị quốc tế về đại diện học tập, 2015.
- [32] Diederik P Kingma và Prafulla Dhariwal. Glow: Đông sinh ra với các kết cấu 1x1 không thể đảo ngược. TRONG Những tiến bộ trong Hệ thống xử lý thông tin thần kinh, trang 10215-10224, 2018.
- [33] Diederik P Kingma và Max Welling. Tự động mã hóa các Bayes biến thể. bản in trước arXiv arXiv:1312.6114, 2013.
- [34] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever và Max Welling. Cải thiện suy luận biến phân với luồng tự hồi quy nghịch đảo. Trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, trang 4743-4751, 2016.
- [35] John Lawson, George Tucker, Bùi Đại, và Rajesh Ranganath. Các mô hình lấy cảm hứng từ năng lượng: Học tập với các phân phối do bộ lấy mẫu tạo ra. Trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, trang 8501-8513, 2019.
- [36] Daniel Levy, Matt D. Hoffman và Jascha Sohl-Dickstein. Khái quát hóa Hamiltonian Monte Carlo với mạng lưới thần kinh. Trong Hội nghị quốc tế về đại diện học tập, 2018.
- [37] Lars Maaløe, Marco Fraccaro, Valentin Liévin và Ole Winther. BIVA: Một hệ thống phân cấp rất sâu sắc của các biến tiềm ẩn cho mô hình tổng quát. Trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, trang 6548-6558, 2019.
- [38] Jacob Menick và Nal Kalchbrenner. Tạo hình ảnh có độ trung thực cao với mạng pixel tỷ lệ phụ và nâng cấp đa chiều. Trong Hội nghị quốc tế về đại diện học tập, 2019.
- [39] Takeru Miyato, Toshiki Kataoka, Masanori Koyama và Yuichi Yoshida. Chuẩn hóa quang phổ cho các mạng đối nghịch tổng quát. Trong Hội nghị quốc tế về đại diện học tập, 2018.
- [40] Alex Nichol. VQ-DRAW: VAE rời rạc tuần tự. bản in trước arXiv arXiv:2003.01599, 2020.
- [41] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu và Ying Nian Wu. Về giải phẫu khả năng học tập tối đa dựa trên MCMC của các mô hình dựa trên năng lượng. bản in trước arXiv arXiv:1903.12370, 2019.
- [42] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, và Ying Nian Wu. Học MCMC ngắn hạn không hội tụ, không liên tục theo mô hình dựa trên năng lượng. Trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, trang 5233-5243, 2019.
- [43] Georg Ostrovski, Will Dabney và Remi Munos. Mạng lương tử tự hồi quy cho mô hình tổng quát. Trong Hội nghị quốc tế về Học máy, trang 3936-3945, 2018.
- [44] Ryan Prenger, Rafael Valle và Bryan Catanzaro. WaveGlow: Mạng tổng hợp giọng nói dựa trên dòng chảy. Trong ICASSP 2019-2019 Hội nghị quốc tế IEEE về Âm học, Xử lý giọng nói và tín hiệu (ICASSP), trang 3617-3621. IEEE, 2019.
- [45] Ali Razavi, Aaron van den Oord, và Oriol Vinyals. Tạo ra hình ảnh đa dạng có độ trung thực cao với VQ-VAE-2. Trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, trang 14837-14847, 2019.
- [46] Danilo Rezende và Shakir Mohamed. Suy luận biến phân với các luồng chuẩn hóa. Trong quốc tế Hội nghị về Học máy, trang 1530-1538, 2015.
- [47] Danilo Jimenez Rezende, Shakir Mohamed và Daan Wierstra. Lan truyền ngược ngẫu nhiên và suy luận gần đúng trong các mô hình sinh sâu. Trong Hội nghị quốc tế về Học máy, trang 1278-1286, 2014.
- [48] Olaf Ronneberger, Philipp Fischer và Thomas Brox. U-Net: Mạng tích chập để phân đoạn hình ảnh y sinh. Trong Hội nghị quốc tế về máy tính hình ảnh y tế và can thiệp có sự hỗ trợ của máy tính, trang 234-241. Mùa xuân, 2015.
- [49] Tim Salimans và Durk P Kingma. Chuẩn hóa trọng số: Việc tái tham số hóa đơn giản để tăng tốc quá trình đào tạo mạng lưới thần kinh sâu. Trong Những tiến bộ trong hệ thống xử lý thông tin thần kinh, trang 901-909, 2016.
- [50] Tim Salimans, Diederik Kingma, và Max Welling. Chuỗi Markov Monte Carlo và suy luận biến phân: Thu hẹp khoảng cách. Trong Hội nghị quốc tế về Học máy, trang 1218-1226, 2015.

- [51] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.
- [52] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.
- [53] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015.
- [54] Jiaming Song, Shengjia Zhao, and Stefano Ermon. A-NICE-MC: Adversarial training for MCMC. In *Advances in Neural Information Processing Systems*, pages 5140–5150, 2017.
- [55] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.
- [56] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020.
- [57] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [58] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *International Conference on Machine Learning*, 2016.
- [59] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems*, pages 4790–4798, 2016.
- [60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.
- [61] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011.
- [62] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot...for now. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.
- [63] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.
- [64] Auke J Wiggers and Emiel Hoogeboom. Predictive sampling with forecasting autoregressive models. *arXiv preprint arXiv:2002.09928*, 2020.
- [65] Hao Wu, Jonas Köhler, and Frank Noé. Stochastic normalizing flows. *arXiv preprint arXiv:2002.06707*, 2020.
- [66] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.
- [67] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644, 2016.
- [68] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7093–7101, 2017.
- [69] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. Learning descriptor networks for 3d shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8629–8638, 2018.
- [70] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Learning energy-based spatial-temporal generative convnets for dynamic patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [71] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
- [72] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- [51] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford và Xi Chen. Cải tiến kỹ thuật để huấn luyện gans. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, trang 2234–2242, 2016.
- [52] Tim Salimans, Andrej Karpathy, Xi Chen, và Diederik P Kingma. PixelCNN++: Cải thiện PixelCNN với khả năng kết hợp logistic rời rạc và các sửa đổi khác. Trong *Hội nghị quốc tế về đại diện học tập*, 2017.
- [53] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan và Surya Ganguli. Học tập không giám sát sâu bằng cách sử dụng nhiệt động lực học không cân bằng. Trong *Hội nghị quốc tế về Học máy*, trang 2256–2265, 2015.
- [54] Jiaming Song, Shengjia Zhao, và Stefano Ermon. A-NICE-MC: Huấn luyện đối kháng cho MCMC. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, trang 5140–5150, 2017.
- [55] Yang Song và Stefano Ermon. Mô hình hóa tổng quát bằng cách ước tính độ dốc của phân phối dữ liệu. TRONG *Những tiến bộ trong Hệ thống xử lý thông tin thần kinh*, trang 11895–11907, 2019.
- [56] Yang Song và Stefano Ermon. Cải tiến các kỹ thuật để đào tạo các mô hình tổng quát dựa trên điểm số. *arXiv* bản in trước *arXiv:2006.09011*, 2020.
- [57] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior và Koray Kavukcuoglu. WaveNet: Một mô hình tổng quát cho âm thanh thô. *bản in trước arXiv arXiv:1609.03499*, 2016.
- [58] Aaron van den Oord, Nal Kalchbrenner, và Koray Kavukcuoglu. Mạng lưới thần kinh tái phát pixel. *Hội nghị quốc tế về học máy*, 2016.
- [59] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves và Koray Kavukcuoglu. Tạo hình ảnh có điều kiện với bộ giải mã PixelCNN. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, trang 4790–4798, 2016.
- [60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser và Illia Polosukhin. Sự chú ý là tất cả những gì bạn cần. Trong *Những tiến bộ trong hệ thống xử lý thông tin thần kinh*, trang 5998–6008, 2017.
- [61] Pascal Vincent. Kết nối giữa khớp điểm và bộ mã hóa tự động khử nhiễu. *tính toán thần kinh*, 23(7):1661–1674, 2011.
- [62] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, và Alexei A Efros. Hình ảnh do CNN tạo ra dễ dàng được phát hiện một cách đáng ngạc nhiên...vào thời điểm hiện tại. Trong *Ký yếu của Hội nghị IEEE về Thị giác máy tính và Nhận dạng mẫu*, năm 2020.
- [63] Xiaolong Wang, Ross Girshick, Abhinav Gupta, và Kaiming He. Mạng lưới thần kinh phi cục bộ. Trong *Ký yếu của Hội nghị IEEE về Thị giác máy tính và Nhận dạng mẫu*, trang 7794–7803, 2018.
- [64] Auke J Wiggers và Emiel Hoogeboom. Lấy mẫu dự đoán với các mô hình tự hồi quy dự báo. *bản in trước arXiv arXiv:2002.09928*, 2020.
- [65] Hảo Ngô, Jonas Köhler, và Frank Noé. Dòng chảy bình thường hóa ngũ nhiên. *bản in trước arXiv arXiv:2002.06707*, 2020.
- [66] Yuxin Wu và Kaiming He. Bình thường hóa nhóm. Trong *Ký yếu của Hội nghị Châu Âu về Máy tính Tâm nhìn (ECCV)*, trang 3–19, 2018.
- [67] Jianwen Xie, Yang Lu, Song-Chun Zhu, và Yinnian Wu. Một lý thuyết về convnet tạo sinh. Trong *quốc tế Hội nghị về Học máy*, trang 2635–2644, 2016.
- [68] Jianwen Xie, Song-Chun Zhu, và Ying Nian Wu. Tổng hợp các mô hình động bằng mạng tạo sinh không gian-thời gian. Trong *Ký yếu của Hội nghị IEEE về Thị giác máy tính và Nhận dạng mẫu*, trang 7093–7101, 2017.
- [69] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, và Ying Nian Wu. Mạng mô tả học tập để tổng hợp và phân tích hình dạng 3D. Trong *Ký yếu của Hội nghị IEEE về Thị giác máy tính và Nhận dạng mẫu*, trang 8629–8638, 2018.
- [70] Jianwen Xie, Song-Chun Zhu, và Ying Nian Wu. Học các phương pháp tổng hợp không gian-thời gian dựa trên năng lượng cho các mô hình động. *Giao dịch của IEEE về Phân tích Mẫu và Trí tuệ Máy*, 2019.
- [71] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff và Jianxiong Xiao. LSUN: Xây dựng bộ dữ liệu hình ảnh quy mô lớn bằng cách sử dụng deep learning với con người trong vòng lặp. *bản in trước arXiv arXiv:1506.03365*, 2015.
- [72] Sergey Zagoruyko và Nikos Komodakis. Mạng dư rộng. *bản in trước arXiv arXiv:1605.07146*, 2016.

Extra information

LSUN FID scores for LSUN datasets are included in Table 3. Scores marked with * are reported by StyleGAN2 as baselines, and other scores are reported by their respective authors.

Table 3: FID scores for LSUN 256×256 datasets

Model	LSUN Bedroom	LSUN Church	LSUN Cat
ProgressiveGAN [27]	8.34	6.42	37.52
StyleGAN [28]	2.65	4.21*	8.53*
StyleGAN2 [30]	-	3.86	6.93
Ours (L_{simple})	6.36	7.89	19.75
Ours (L_{simple} , large)	4.90	-	-

Thông tin bổ sung

Điểm LSUN FID cho bộ dữ liệu LSUN được bao gồm trong Bảng 3. Điểm được StyleGAN2 đánh dấu làm đường cơ sở và các điểm khác được báo cáo bởi các tác giả tương ứng của chúng.

Bảng 3: Điểm FID cho bộ dữ liệu LSUN 256×256

Người mẫu	Phòng ngủ LSUN	Nhà thờ LSUN	Mèo LSUN
ProgressiveGAN [27]	8,34	6,42	37,52
StyleGAN [28]	2,65	4,21	8,53
StyleGAN2 [30]	-	3,86	6,93
Của chúng tôi (L_{simple})	6,36	7,89	19:75
Của chúng tôi (L_{large} , lớn)	4,90	-	-

Progressive compression Our lossy compression argument in Section 4.3 is only a proof of concept, because Algorithms 3 and 4 depend on a procedure such as minimal random coding [20], which is not tractable for high dimensional data. These algorithms serve as a compression interpretation of the variational bound (5) of Sohl-Dickstein et al. [53], not yet as a practical compression system.

Nén lũy tiến Đôi số nén mát của chúng tôi trong Phần 4.3 chỉ là bằng chứng về khái niệm, bởi vì Thuật toán 3 và 4 phụ thuộc vào một quy trình như mã hóa ngẫu nhiên tối thiểu [20], đó là không thể xử lý được đối với dữ liệu có chiều cao. Các thuật toán này phục vụ như một cách diễn giải nén của ràng buộc biến thể (5) của Sohl-Dickstein et al. [53], chưa phải là một hệ thống nén thực tế.

Table 4: Unconditional CIFAR10 test set rate-distortion values (accompanies Fig. 5)

Reverse process time ($T - t + 1$)	Rate (bits/dim)	Distortion (RMSE [0, 255])
1000	1.77581	0.95136
900	0.11994	12.02277
800	0.05415	18.47482
700	0.02866	24.43656
600	0.01507	30.80948
500	0.00716	38.03236
400	0.00282	46.12765
300	0.00081	54.18826
200	0.00013	60.97170
100	0.00000	67.60125

Bảng 4: Các giá trị độ biến dạng tốc độ của bộ thử nghiệm CIFAR10 vô điều kiện (kèm theo Hình 5)

Thời gian xử lý ngược ($T - t + 1$)	Tốc độ (bit/dim)	Độ méo (RMSE [0, 255])
1000	1.77581	0,95136
900	0.11994	12.02277
800	0.05415	18.47482
700	0.02866	24.43656
600	0.01507	30.80948
500	0.00716	38.03236
400	0.00282	46.12765
300	0.00081	54.18826
200	0.00013	60.97170
100	0.00000	67.60125

A Extended derivations

Below is a derivation of Eq. (5), the reduced variance variational bound for diffusion models. This material is from Sohl-Dickstein et al. [53]; we include it here only for completeness.

$$L = \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right] \quad (17)$$

$$= \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \quad (18)$$

$$= \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t > 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right] \quad (19)$$

$$= \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t > 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \cdot \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right] \quad (20)$$

$$= \mathbb{E}_q \left[-\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} - \sum_{t > 1} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \quad (21)$$

A Đạo hàm mở rộng

Dưới đây là một dẫn xuất của phương trình (5), giới hạn biến thiên phương sai giảm cho các mô hình khuếch tán. Cái này tài liệu là từ Sohl-Dickstein et al. [53]; chúng tôi chỉ đưa nó vào đây để đảm bảo đầy đủ.

$$L = \text{Phương trình } \frac{\frac{p_\theta(\mathbf{x}_0:\mathbf{T})}{\log q(\mathbf{x}_1:\mathbf{T}|\mathbf{x}_0)}}{\log q(\mathbf{x}_1:\mathbf{T}|\mathbf{x}_0)} \quad (17)$$

$$= \text{Phương trình } \log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \quad (18)$$

$$= \text{Phương trình } \log p(\mathbf{x}_T) - \sum_{t > 1} \log \frac{p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \cdot \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \quad (19)$$

$$= \text{Phương trình } \log p(\mathbf{x}_T) - \sum_{t > 1} \log \frac{p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \cdot \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} - \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \quad (20)$$

$$= \text{Phương trình } \frac{p(\mathbf{x}_T)}{\log q(\mathbf{x}_T|\mathbf{x}_0)} - \sum_{t > 1} \log \frac{p_\theta(\mathbf{x}_t|\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)} \quad (21)$$

$$= \mathbb{E}_q \left[D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T)) + \sum_{t>1} D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0 | \mathbf{x}_1) \right] \quad (22)$$

The following is an alternate version of L . It is not tractable to estimate, but it is useful for our discussion in Section 4.3.

$$L = \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \right] \quad (23)$$

$$= \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_{t-1} | \mathbf{x}_t)} \cdot \frac{q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)} \right] \quad (24)$$

$$= \mathbb{E}_q \left[-\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T)} - \sum_{t \geq 1} \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_{t-1} | \mathbf{x}_t)} - \log q(\mathbf{x}_0) \right] \quad (25)$$

$$= D_{\text{KL}}(q(\mathbf{x}_T) \| p(\mathbf{x}_T)) + \mathbb{E}_q \left[\sum_{t \geq 1} D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) \right] + H(\mathbf{x}_0) \quad (26)$$

$$= \text{Phương trình } D_{\text{KL}}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T)) + \sum_{t>1} D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0 | \mathbf{x}_1) \quad (22)$$

Sau đây là phiên bản thay thế của L . Nó không thể ước tính được nhưng nó hữu ích cho cuộc thảo luận của chúng ta trong Phần 4.3.

$$L = \text{phương trình} \quad \log p(\mathbf{x}_T) \quad \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_t | \mathbf{x}_{t-1})} \quad (23)$$

$$= \text{phương trình} \quad \log p(\mathbf{x}_T) \quad \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_{t-1} | \mathbf{x}_t)} \cdot \frac{q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)} \quad (24)$$

$$= \text{phương trình} \quad \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T)} \quad \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)}{q(\mathbf{x}_T)} \quad (25)$$

$$= D_{\text{KL}}(q(\mathbf{x}_T) \| p(\mathbf{x}_T)) + \mathbb{E}_q \quad D_{\text{KL}}(q(\mathbf{x}_{t-1} | \mathbf{x}_t) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) \quad + H(\mathbf{x}_0) \quad (26)$$

B Experimental details

Our neural network architecture follows the backbone of PixelCNN++ [52], which is a U-Net [48] based on a Wide ResNet [72]. We replaced weight normalization [49] with group normalization [66] to make the implementation simpler. Our 32×32 models use four feature map resolutions (32×32 to 4×4), and our 256×256 models use six. All models have two convolutional residual blocks per resolution level and self-attention blocks at the 16×16 resolution between the convolutional blocks [6]. Diffusion time t is specified by adding the Transformer sinusoidal position embedding [60] into each residual block. Our CIFAR10 model has 35.7 million parameters, and our LSUN and CelebA-HQ models have 114 million parameters. We also trained a larger variant of the LSUN Bedroom model with approximately 256 million parameters by increasing filter count.

We used TPU v3-8 (similar to 8 V100 GPUs) for all experiments. Our CIFAR model trains at 21 steps per second at batch size 128 (10.6 hours to train to completion at 800k steps), and sampling a batch of 256 images takes 17 seconds. Our CelebA-HQ/LSUN (256^2) models train at 2.2 steps per second at batch size 64, and sampling a batch of 128 images takes 300 seconds. We trained on CelebA-HQ for 0.5M steps, LSUN Bedroom for 2.4M steps, LSUN Cat for 1.8M steps, and LSUN Church for 1.2M steps. The larger LSUN Bedroom model was trained for 1.15M steps.

Apart from an initial choice of hyperparameters early on to make network size fit within memory constraints, we performed the majority of our hyperparameter search to optimize for CIFAR10 sample quality, then transferred the resulting settings over to the other datasets:

- We chose the β_t schedule from a set of constant, linear, and quadratic schedules, all constrained so that $L_T \approx 0$. We set $T = 1000$ without a sweep, and we chose a linear schedule from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$.
- We set the dropout rate on CIFAR10 to 0.1 by sweeping over the values $\{0.1, 0.2, 0.3, 0.4\}$. Without dropout on CIFAR10, we obtained poorer samples reminiscent of the overfitting artifacts in an unregularized PixelCNN++ [52]. We set dropout rate on the other datasets to zero without sweeping.
- We used random horizontal flips during training for CIFAR10; we tried training both with and without flips, and found flips to improve sample quality slightly. We also used random horizontal flips for all other datasets except LSUN Bedroom.
- We tried Adam [31] and RMSProp early on in our experimentation process and chose the former. We left the hyperparameters to their standard values. We set the learning rate to 2×10^{-4} without any sweeping, and we lowered it to 2×10^{-5} for the 256×256 images, which seemed unstable to train with the larger learning rate.

B Chi tiết thí nghiệm

Kiến trúc mạng thần kinh của chúng tôi tuân theo xương sống của PixelCNN++ [52], là U-Net [48] dựa trên Wide ResNet [72]. Chúng tôi đã thay thế chuẩn hóa trọng số [49] bằng chuẩn hóa nhóm [66] để làm cho việc thực hiện đơn giản hơn. Các mô hình 32×32 của chúng tôi sử dụng bốn độ phân giải bốn lần đặc điểm (32×32 đến 4×4) và các mô hình 256×256 của chúng tôi sử dụng sáu độ phân giải. Tất cả các mô hình đều có hai khối dư tích chập cho mỗi mức độ phân giải và các khối tự chú ý ở độ phân giải 16×16 giữa các khối chập [6]. Thời gian khuếch tán t được xác định bằng cách thêm vị trí hình sin của Máy biến áp nhúng [60] vào mỗi khối dư. Mô hình CIFAR10 của chúng tôi có 35,7 triệu tham số và các mô hình LSUN và CelebA-HQ của chúng tôi có 114 triệu tham số. Chúng tôi cũng đã đào tạo một biến thể lớn hơn của mô hình Phòng ngủ LSUN với khoảng 256 triệu tham số bằng cách tăng số lượng bộ lọc.

Chúng tôi đã sử dụng TPU v3-8 (tương tự 8 GPU V100) cho tất cả các thử nghiệm. Mô hình CIFAR của chúng tôi huấn luyện với tốc độ 21 bước mỗi giây ở kích thước lô 128 (10,6 giờ để huấn luyện cho đến khi hoàn thành ở 800 nghìn bước) và lấy mẫu một loạt 256 hình ảnh mất 17 giây. Các mô hình CelebA-HQ/LSUN (256^2) của chúng tôi huấn luyện ở tốc độ 2,2 bước mỗi giây ở kích thước lô 64 và lấy mẫu một lô gồm 128 hình ảnh mất 300 giây. Chúng tôi đã huấn luyện trên CelebA-HQ với 0,5 triệu bước, Phòng ngủ LSUN với 2,4 triệu bước, LSUN Cat với 1,8 triệu bước và LSUN Church với 1,2 triệu bước. Mô hình Phòng ngủ LSUN lớn hơn đã được huấn luyện cho 1,15 triệu bước.

Ngoài lựa chọn ban đầu về siêu tham số để làm cho kích thước mạng phù hợp với giới hạn bộ nhớ, chúng tôi đã thực hiện phần lớn tìm kiếm siêu tham số để tối ưu hóa chất lượng mẫu CIFAR10, sau đó chuyển cài đặt kết quả sang các bộ dữ liệu khác:

- Chúng tôi đã chọn lịch trình β_t từ một tập hợp các lịch trình không đối, tuyến tính và bậc hai, tất cả đều bị ràng buộc sao cho $L_T \approx 0$. Chúng tôi đặt $T = 1000$ mà không quét và chúng tôi đã chọn lịch trình tuyến tính từ $\beta_1 = 10^{-4}$ đến $\beta_T = 0.02$.
- Chúng tôi đặt tỷ lệ bỏ học trên CIFAR10 là 0.1 bằng cách quét qua các giá trị $\{0.1, 0.2, 0.3, 0.4\}$. Không bị loại bỏ trên CIFAR10, chúng tôi đã thu được các mẫu kém hơn gợi nhớ đến các tạo phầm được trang bị quá mức trong PixelCNN++ không được chuẩn hóa [52]. Chúng tôi đặt tỷ lệ bỏ học trên các tập dữ liệu khác về 0 mà không cần quét.
- Chúng tôi sử dụng các cú lật ngang nhiên trong quá trình huấn luyện cho CIFAR10; chúng tôi đã thử huấn luyện cả khi có và không có lật và nhận thấy các lật có thể cải thiện chất lượng mẫu một chút. Chúng tôi cũng sử dụng thao tác lật ngang nhiên cho tất cả các tập dữ liệu khác trừ Phòng ngủ LSUN.
- Chúng tôi đã thử Adam [31] và RMSProp ngay từ đầu trong quá trình thử nghiệm và đã chọn cái trước. Chúng tôi để các siêu tham số ở giá trị tiêu chuẩn của chúng. Chúng tôi đặt tốc độ học thành 2×10^{-4} mà không cần quét và chúng tôi hạ nó xuống 2×10^{-5} cho các hình ảnh 256×256 , có vẻ không ổn định khi huấn luyện với tốc độ học lớn hơn.

- We set the batch size to 128 for CIFAR10 and 64 for larger images. We did not sweep over these values.
- We used EMA on model parameters with a decay factor of 0.9999. We did not sweep over this value.

Final experiments were trained once and evaluated throughout training for sample quality. Sample quality scores and log likelihood are reported on the minimum FID value over the course of training. On CIFAR10, we calculated Inception and FID scores on 50000 samples using the original code from the OpenAI [51] and TTUR [21] repositories, respectively. On LSUN, we calculated FID scores on 50000 samples using code from the StyleGAN2 [30] repository. CIFAR10 and CelebA-HQ were loaded as provided by TensorFlow Datasets (<https://www.tensorflow.org/datasets>), and LSUN was prepared using code from StyleGAN. Dataset splits (or lack thereof) are standard from the papers that introduced their usage in a generative modeling context. All details can be found in the source code release.

C Discussion on related work

Our model architecture, forward process definition, and prior differ from NCSN [55, 56] in subtle but important ways that improve sample quality, and, notably, we directly train our sampler as a latent variable model rather than adding it after training post-hoc. In greater detail:

1. We use a U-Net with self-attention; NCSN uses a RefineNet with dilated convolutions. We condition all layers on t by adding in the Transformer sinusoidal position embedding, rather than only in normalization layers (NCSNv1) or only at the output (v2).
2. Diffusion models scale down the data with each forward process step (by a $\sqrt{1 - \beta_t}$ factor) so that variance does not grow when adding noise, thus providing consistently scaled inputs to the neural net reverse process. NCSN omits this scaling factor.
3. Unlike NCSN, our forward process destroys signal ($D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel \mathcal{N}(\mathbf{0}, \mathbf{I})) \approx 0$), ensuring a close match between the prior and aggregate posterior of \mathbf{x}_T . Also unlike NCSN, our β_t are very small, which ensures that the forward process is reversible by a Markov chain with conditional Gaussians. Both of these factors prevent distribution shift when sampling.
4. Our Langevin-like sampler has coefficients (learning rate, noise scale, etc.) derived rigorously from β_t in the forward process. Thus, our training procedure directly trains our sampler to match the data distribution after T steps: it trains the sampler as a latent variable model using variational inference. In contrast, NCSN’s sampler coefficients are set by hand post-hoc, and their training procedure is not guaranteed to directly optimize a quality metric of their sampler.

D Samples

Additional samples Figure 11, 13, 16, 17, 18, and 19 show uncurated samples from the diffusion models trained on CelebA-HQ, CIFAR10 and LSUN datasets.

Latent structure and reverse process stochasticity During sampling, both the prior $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and Langevin dynamics are stochastic. To understand the significance of the second source of noise, we sampled multiple images conditioned on the same intermediate latent for the CelebA 256 × 256 dataset. Figure 7 shows multiple draws from the reverse process $\mathbf{x}_0 \sim p_\theta(\mathbf{x}_0|\mathbf{x}_t)$ that share the latent \mathbf{x}_t for $t \in \{1000, 750, 500, 250\}$. To accomplish this, we run a single reverse chain from an initial draw from the prior. At the intermediate timesteps, the chain is split to sample multiple images. When the chain is split after the prior draw at $\mathbf{x}_{T=1000}$, the samples differ significantly. However, when the chain is split after more steps, samples share high-level attributes like gender, hair color, eyewear, saturation, pose and facial expression. This indicates that intermediate latents like \mathbf{x}_{750} encode these attributes, despite their imperceptibility.

Coarse-to-fine interpolation Figure 9 shows interpolations between a pair of source CelebA 256 × 256 images as we vary the number of diffusion steps prior to latent space interpolation. Increasing the number of diffusion steps destroys more structure in the source images, which the

- Chúng tôi đặt kích thước lô thành 128 cho CIFAR10 và 64 cho hình ảnh lớn hơn. Chúng tôi đã không quét qua những giá trị.
- Chúng tôi đã sử dụng EMA trên các tham số mô hình với hệ số phân rã là 0,9999. Chúng tôi đã không quét qua giá trị này.

Các thí nghiệm cuối cùng được huấn luyện một lần và được đánh giá trong suốt quá trình huấn luyện về chất lượng mẫu. Điểm chất lượng mẫu và khả năng ghi nhận ký được báo cáo dựa trên giá trị FID tối thiểu trong quá trình đào tạo. Trên CIFAR10, chúng tôi đã tính điểm Inception và FID trên 50000 mẫu bằng cách sử dụng mã gốc từ kho lưu trữ OpenAI [51] và TTUR [21] tương ứng. Trên LSUN, chúng tôi đã tính điểm FID trên 50000 mẫu bằng mã từ kho lưu trữ StyleGAN2 [30]. CIFAR10 và CelebA-HQ đã được tải do Bộ dữ liệu TensorFlow cung cấp (<https://www.tensorflow.org/datasets>), và LSUN đã được chuẩn bị bằng mã từ StyleGAN. Việc phân chia tập dữ liệu (hoặc thiểu chúng) là tiêu chuẩn từ các bài báo giới thiệu cách sử dụng chúng trong bối cảnh mô hình hóa tổng quát. Tất cả các chi tiết có thể được tìm thấy trong bản phát hành mã nguồn.

C Thảo luận về công việc liên quan

Kiến trúc mô hình, định nghĩa quy trình chuyển tiếp và trước đó của chúng tôi khác với NCSN [55, 56] theo những cách tinh tế nhưng quan trọng giúp cải thiện chất lượng mẫu và đáng chú ý là chúng tôi trực tiếp đào tạo bộ lấy mẫu của mình như một mô hình biến tiệm ẩn thay vì thêm nó sau khi đào tạo sau-hoc. Chi tiết hơn:

1. Chúng tôi sử dụng U-Net với sự chú ý đến bản thân; NCSN sử dụng RefineNet với các kết cấu giãn nở. Chúng tôi điều hòa tất cả các lớp trên t bằng cách thêm vào những vị trí hình sin của Máy biến áp, thay vì chỉ trong các lớp chuẩn hóa (NCSNv1) hoặc chỉ ở đầu ra (v2).
2. Các mô hình khuếch tán thu nhỏ dữ liệu theo từng bước quy trình chuyển tiếp (theo hệ số $\sqrt{1 - \beta_t}$) để phương sai không tăng khi thêm nhiều, do đó cung cấp đầu vào có tỷ lệ nhất quán cho quy trình đào ngược mạng nơ-ron. NCSN bỏ qua hệ số tỷ lệ này.
3. Không giống như NCSN, quy trình chuyển tiếp của chúng tôi sẽ hủy tín hiệu ($DKL(q(x_T|x_0) \parallel N(0, I)) \approx 0$), đảm bảo sự khớp chặt chẽ giữa phần trước và phần sau tổng hợp của x_T . Cũng không giống như NCSN, β_t của chúng tôi rất nhỏ, điều này đảm bảo rằng quá trình chuyển tiếp có thể đảo ngược được bởi chuỗi Markov với Gaussian có điều kiện. Cả hai yếu tố này đều ngăn cản sự dịch chuyển phân phối khi lấy mẫu.
4. Bộ lấy mẫu giống Langevin của chúng tôi có các hệ số (tốc độ học, thang nhiễu, v.v.) được lấy chính xác từ β_t trong quy trình chuyển tiếp. Do đó, quy trình đào tạo của chúng tôi trực tiếp huấn luyện bộ lấy mẫu của chúng tôi để khớp với phân phối dữ liệu sau T bước: nó huấn luyện bộ lấy mẫu như một mô hình biến tiệm ẩn bằng cách sử dụng suy luận biến phân. Ngược lại, các hệ số của bộ lấy mẫu của NCSN được thiết lập thủ công và quy trình đào tạo của họ không được đảm bảo để tối ưu hóa trực tiếp do chất lượng của bộ lấy mẫu.

Mẫu D

Các mẫu bổ sung Hình 11, 13, 16, 17, 18 và 19 hiển thị các mẫu chưa được xử lý từ các mô hình khuếch tán được đào tạo trên bộ dữ liệu CelebA-HQ, CIFAR10 và LSUN.

Cấu trúc tiệm ẩn và tính ngẫu nhiên của quá trình đảo ngược. Trong quá trình lấy mẫu, cả x_T trước đó động lực $N(0, I)$ và Langevin là ngẫu nhiên. Để hiểu tầm quan trọng của nguồn nhiễu thứ hai, chúng tôi đã lấy mẫu nhiều hình ảnh được điều chỉnh trên cùng một mức tiệm ẩn trung gian cho tập dữ liệu CelebA 256 × 256. Hình 7 cho thấy nhiều lần rút ra từ quá trình ngược lại $x_0 \sim p_\theta(x_0|x_T)$ có chung x_T tiệm ẩn cho $t \in \{1000, 750, 500, 250\}$. Để thực hiện điều này, chúng tôi chạy một chuỗi đảo ngược duy nhất từ lần rút ban đầu từ lần rút trước đó. Tại các dấu thời gian trung gian, chuỗi được chia thành nhiều hình ảnh mẫu. Khi chuỗi được tách ra sau lần rút trước ở $x_T = 1000$, các mẫu sẽ khác nhau đáng kể. Tuy nhiên, khi chuỗi được phân chia sau nhiều bước hơn, các mẫu sẽ có chung các thuộc tính cấp cao như giới tính, màu tóc, kính mắt, độ bão hòa, tư thế và nét mặt. Điều này chỉ ra rằng các tiệm ẩn trung gian như x_{750} mã hóa các thuộc tính này, mặc dù chúng không thể được nhận ra.

Nội suy từ thô đến mịn Hình 9 cho thấy các phép nội suy giữa một cặp hình ảnh CelebA 256 × 256 nguồn khi chúng tôi thay đổi số bước khuếch tán trước khi nội suy không giàn tiệm ẩn. Việc tăng số bước khuếch tán sẽ phá hủy nhiều cấu trúc hơn trong ảnh nguồn, điều này

model completes during the reverse process. This allows us to interpolate at both fine granularities and coarse granularities. In the limiting case of 0 diffusion steps, the interpolation mixes source images in pixel space. On the other hand, after 1000 diffusion steps, source information is lost and interpolations are novel samples.

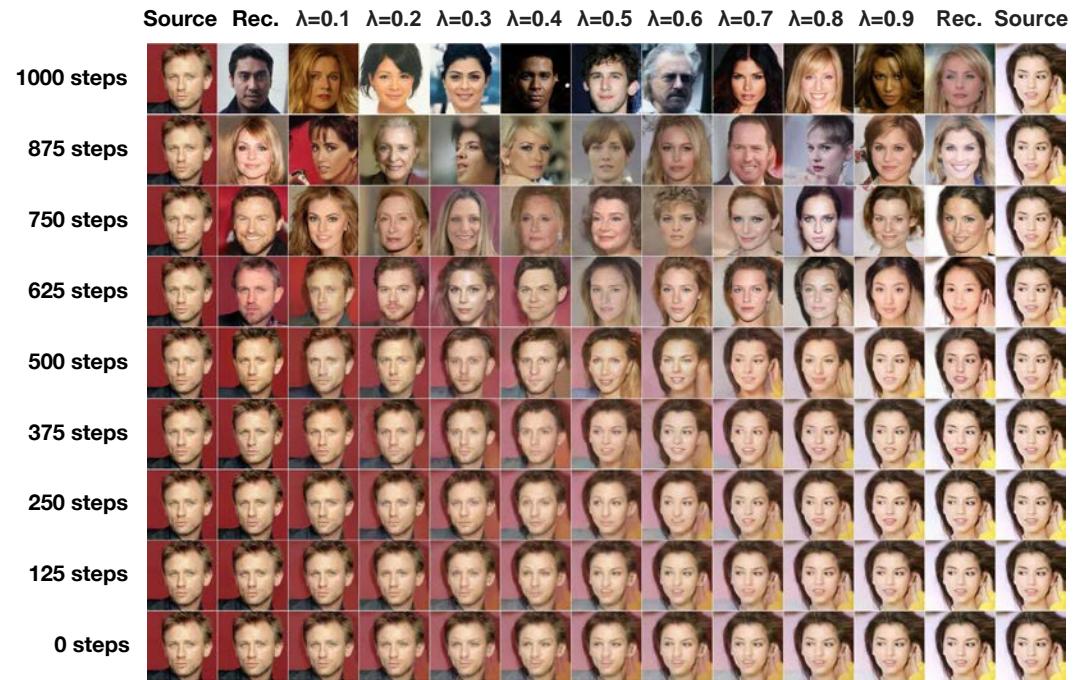
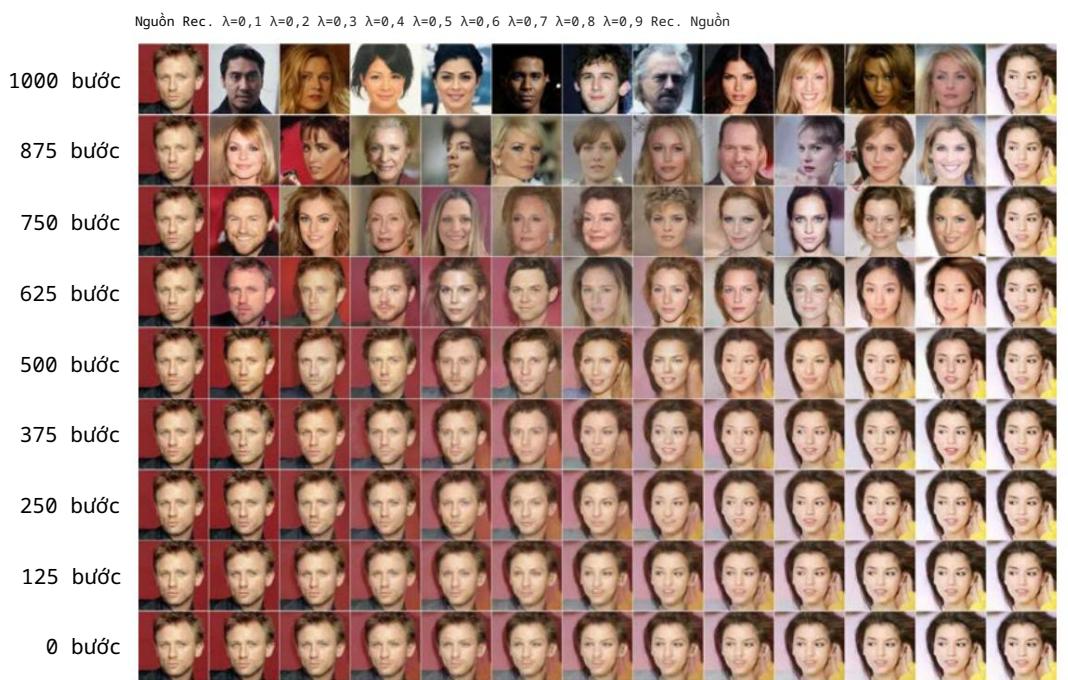


Figure 9: Coarse-to-fine interpolations that vary the number of diffusion steps prior to latent mixing.

mô hình hoàn thành trong quá trình ngược lại. Điều này cho phép chúng ta nội suy ở cả độ chi tiết mịn và độ chi tiết thô. Trong trường hợp giới hạn 0 bước khuếch tán, phép nội suy sẽ trộn các hình ảnh nguồn trong không gian pixel. Mặt khác, sau 1000 bước khuếch tán, thông tin nguồn sẽ bị mất và các phép nội suy là các mẫu mới.



Hình 9: Các phép nội suy từ thô đến mịn thay đổi số bước khuếch tán trước khi trộn tiêm ẩn.

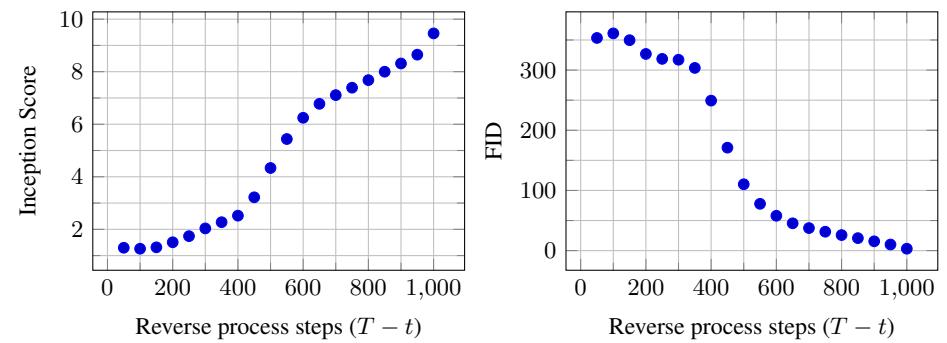
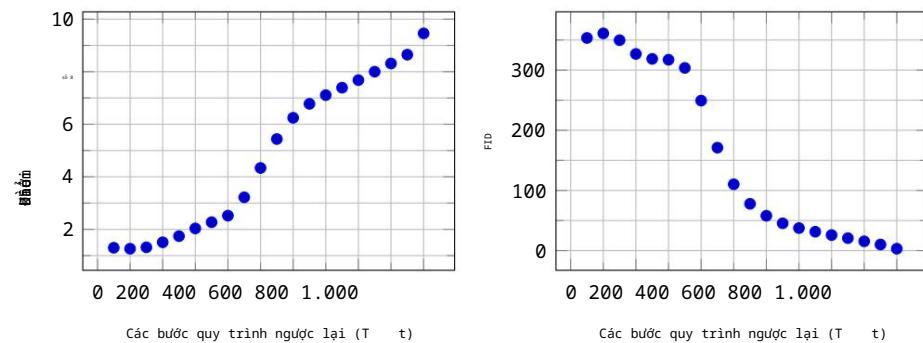


Figure 10: Unconditional CIFAR10 progressive sampling quality over time



Hình 10: Chất lượng lấy mẫu lũy tiến CIFAR10 vô điều kiện theo thời gian



Figure 11: CelebA-HQ 256 × 256 generated samples



Hình 11: Mẫu được tạo CelebA-HQ 256 × 256



(a) Pixel space nearest neighbors



(b) Inception feature space nearest neighbors

Figure 12: CelebA-HQ 256×256 nearest neighbors, computed on a 100×100 crop surrounding the faces. Generated samples are in the leftmost column, and training set nearest neighbors are in the remaining columns.



(a) Không gian pixel lân cận gần nhất



(b) Không gian tính năng khởi đầu lân cận gần nhất

Hình 12: CelebA-HQ 256×256 hàng xóm gần nhất, được tính toán trên phần cắt 100×100 bao quanh các mặt. Các mẫu được tạo nằm ở cột ngoài cùng bên trái và tập huấn luyện lân cận gần nhất nằm ở các cột còn lại.

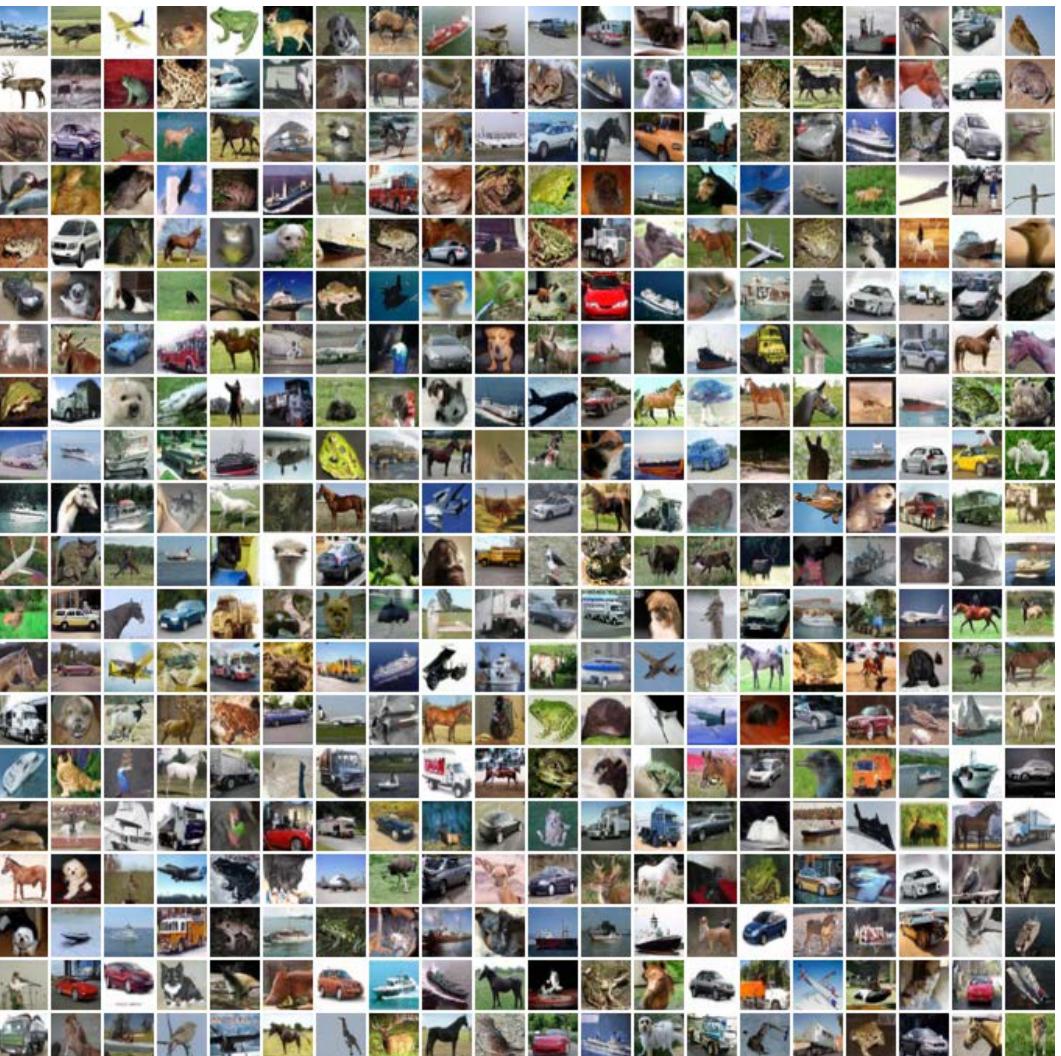
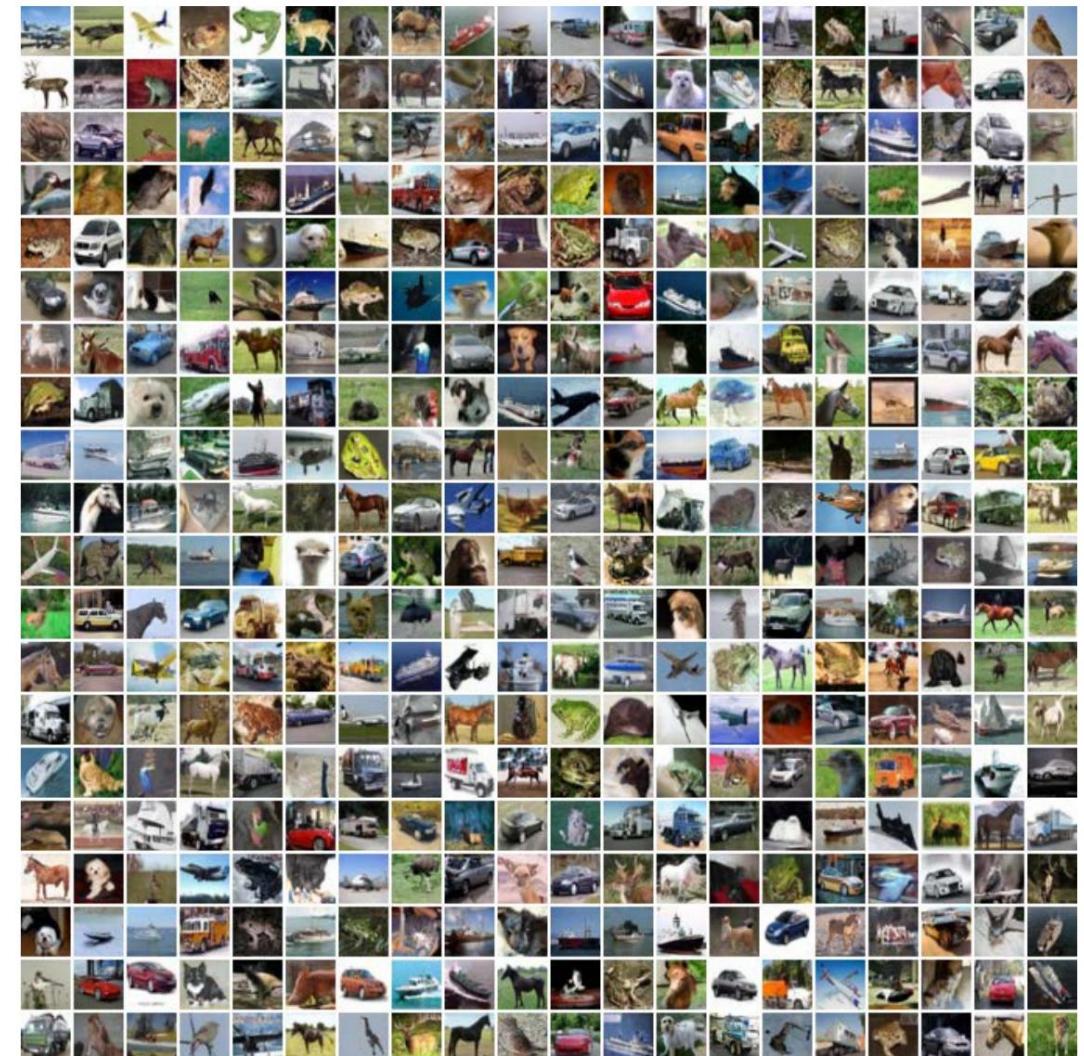


Figure 13: Unconditional CIFAR10 generated samples



Hình 13: Các mẫu được tạo CIFAR10 vô điều kiện

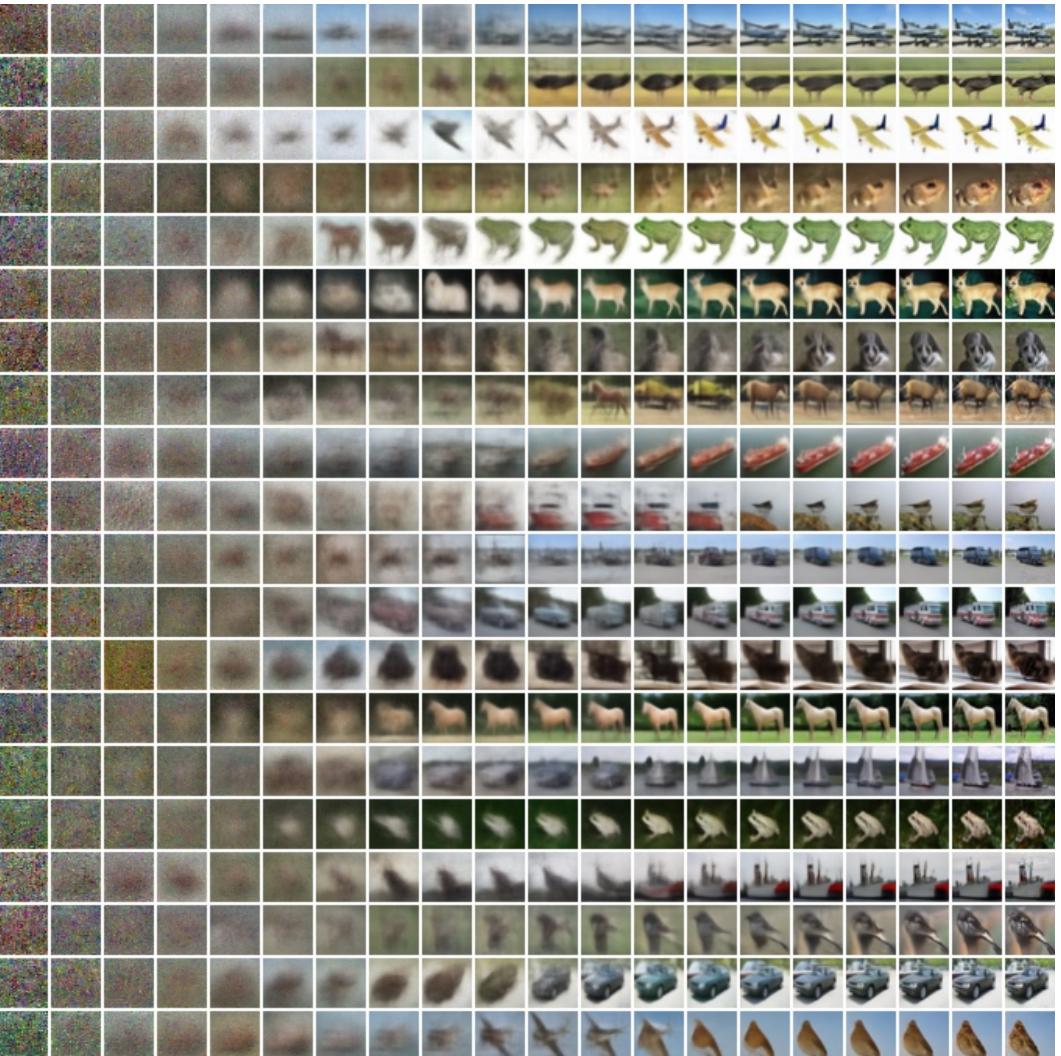
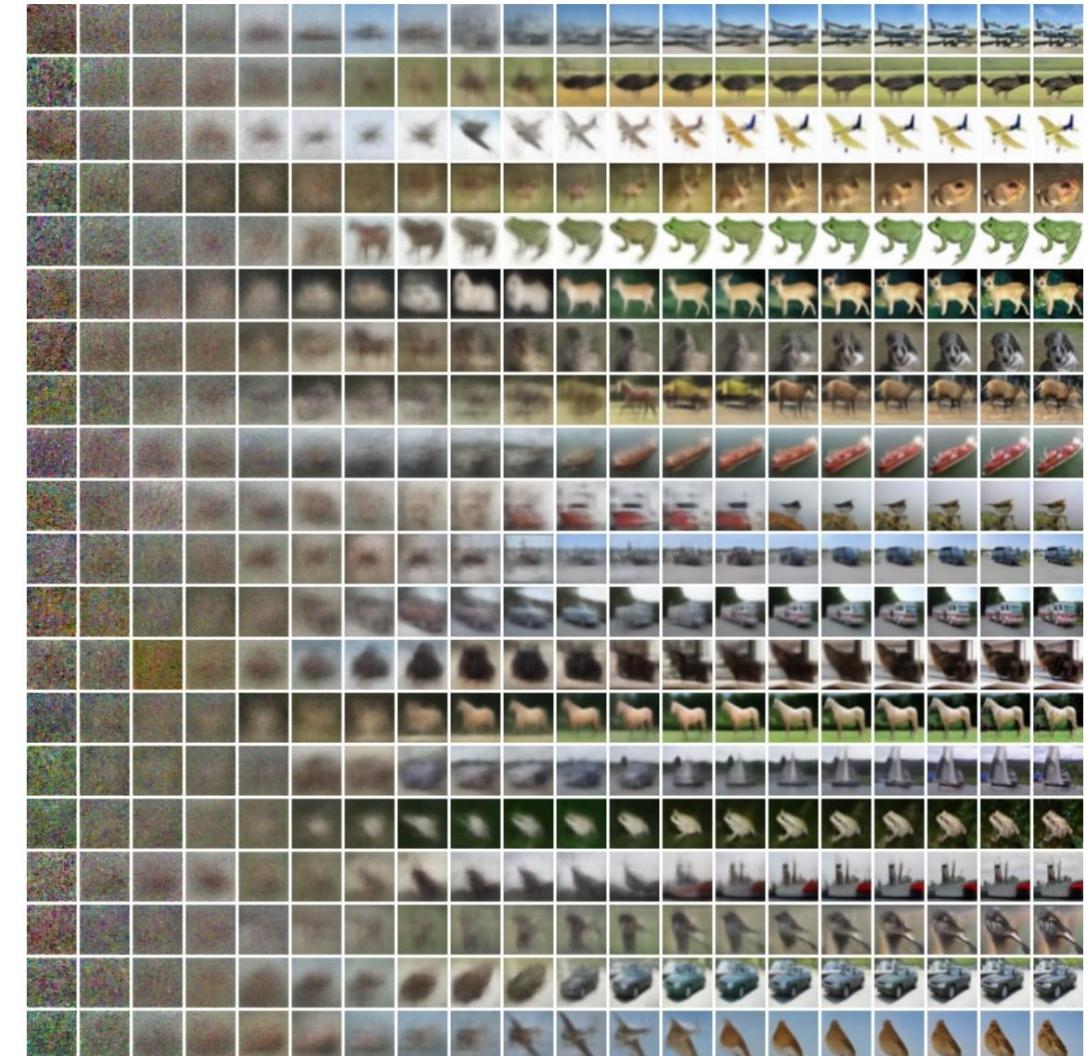
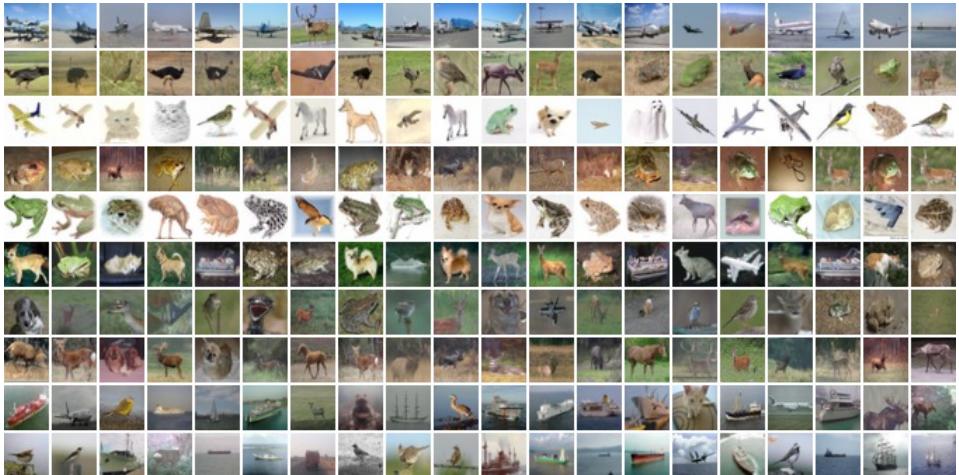


Figure 14: Unconditional CIFAR10 progressive generation



Hình 14: Thé hệ lũy tiến CIFAR10 vô điều kiện

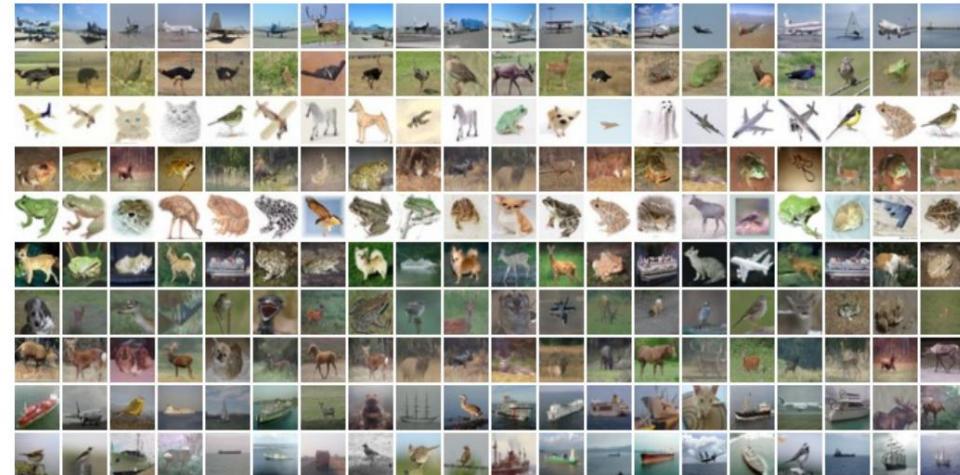


(a) Pixel space nearest neighbors



(b) Inception feature space nearest neighbors

Figure 15: Unconditional CIFAR10 nearest neighbors. Generated samples are in the leftmost column, and training set nearest neighbors are in the remaining columns.



(a) Không gian pixel lân cận gần nhất



(b) Không gian tính năng khồi đầu lân cận gần nhất

Hình 15: Hàng xóm gần nhất CIFAR10 vô điều kiện. Các mẫu được tạo nằm ở cột ngoài cùng bên trái và tập huấn luyện lân cận gần nhất nằm ở các cột còn lại.



Figure 16: LSUN Church generated samples. FID=7.89



Hình 16: Các mẫu do LSUN Church tạo ra. FID=7,89

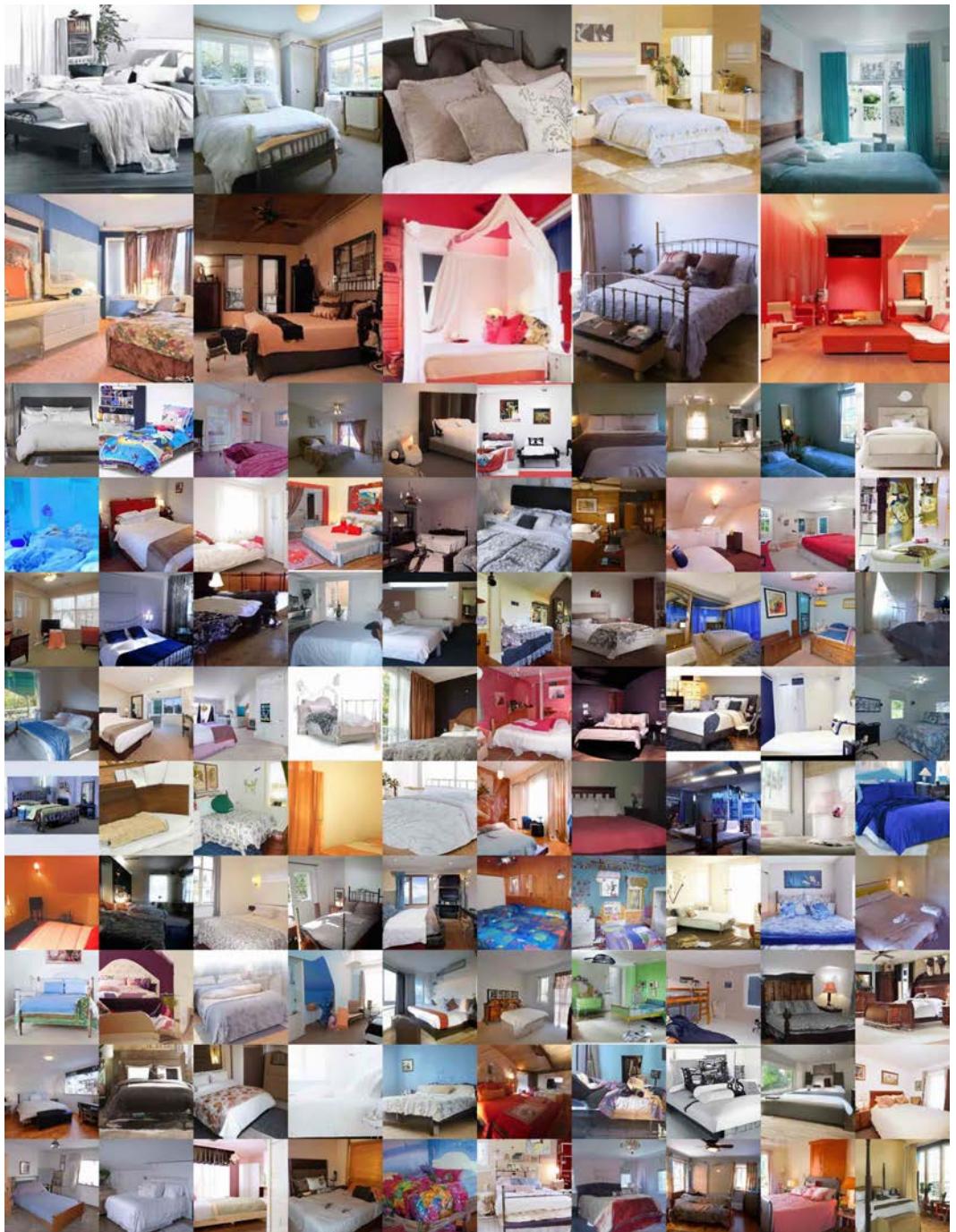


Figure 17: LSUN Bedroom generated samples, large model. FID=4.90



Hình 17: LSUN Bedroom tạo mẫu, mô hình lớn. FID=4,90

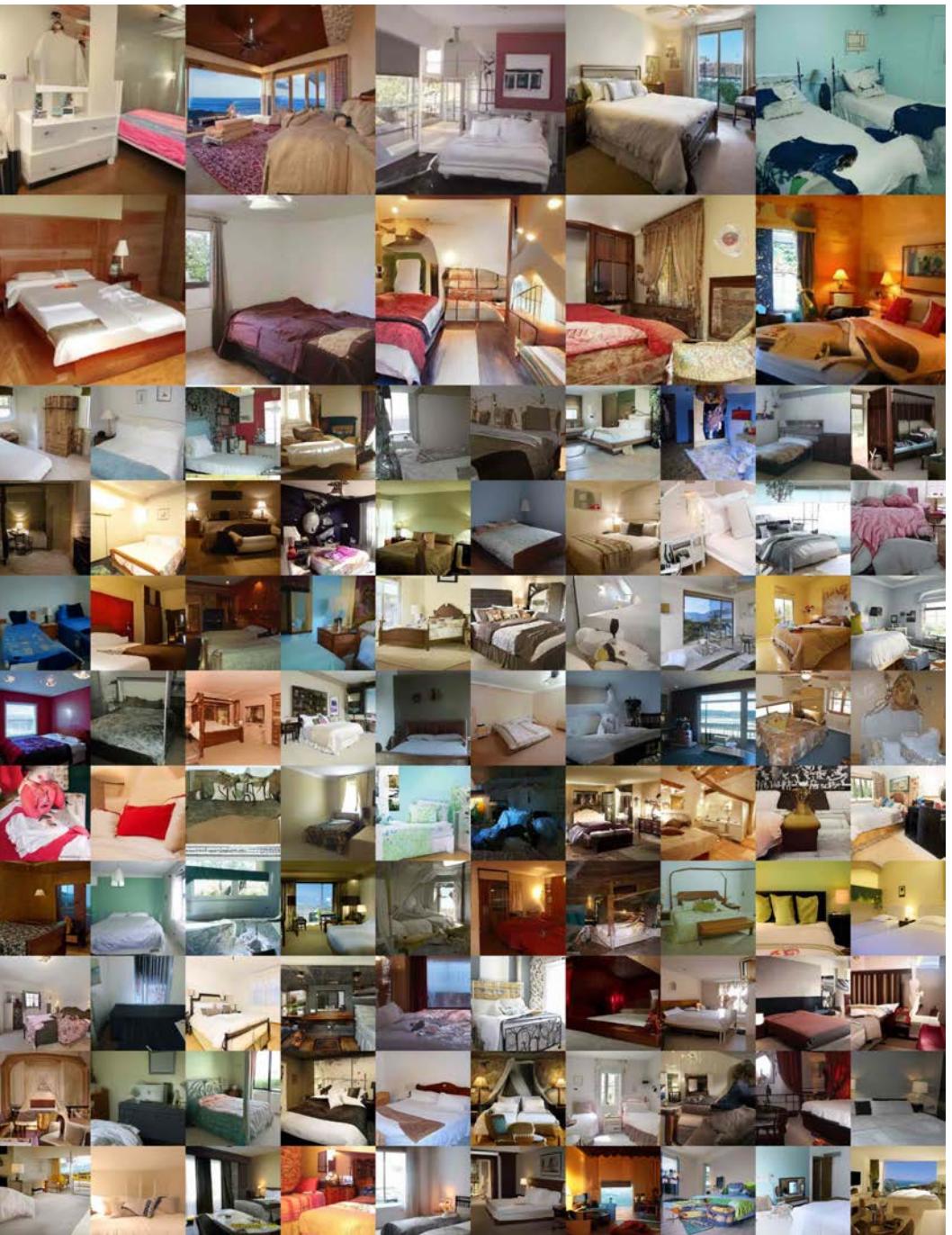
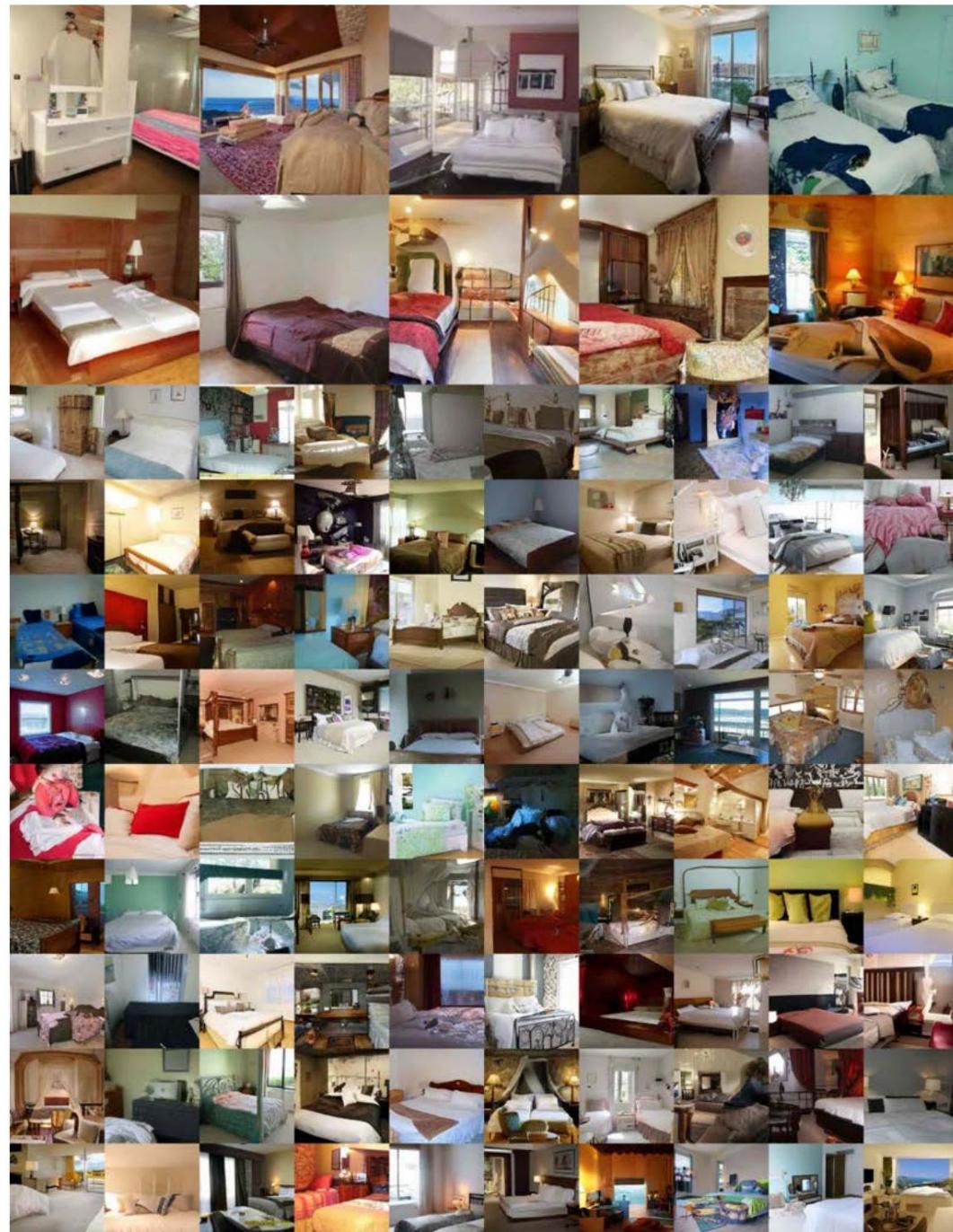


Figure 18: LSUN Bedroom generated samples, small model. FID=6.36



Hình 18: LSUN Bedroom tạo mẫu, mô hình nhỏ. FID=6,36



Figure 19: LSUN Cat generated samples. FID=19.75



Hình 19: LSUN Cat tạo mẫu. FID=19,75