Denoising Diffusion Probabilistic Models

Jonathan Ho, Ajay Jain, and Pieter Abbeel

UC Berkeley

NeurIPS 2020

Presented by Minho Park

Contribution

- Novel generative model which obtain sample quality similar to ProgressiveGAN.
  - A latent variable models training on a weighted variational bound.
- Connection between diffusion probabilistic models and denoising score matching with Langevin dynamics.
- Naturally admit a progressive lossy decompression scheme.

Introduction

- A class of latent variable models inspired by considerations from nonequilibrium thermodynamics.

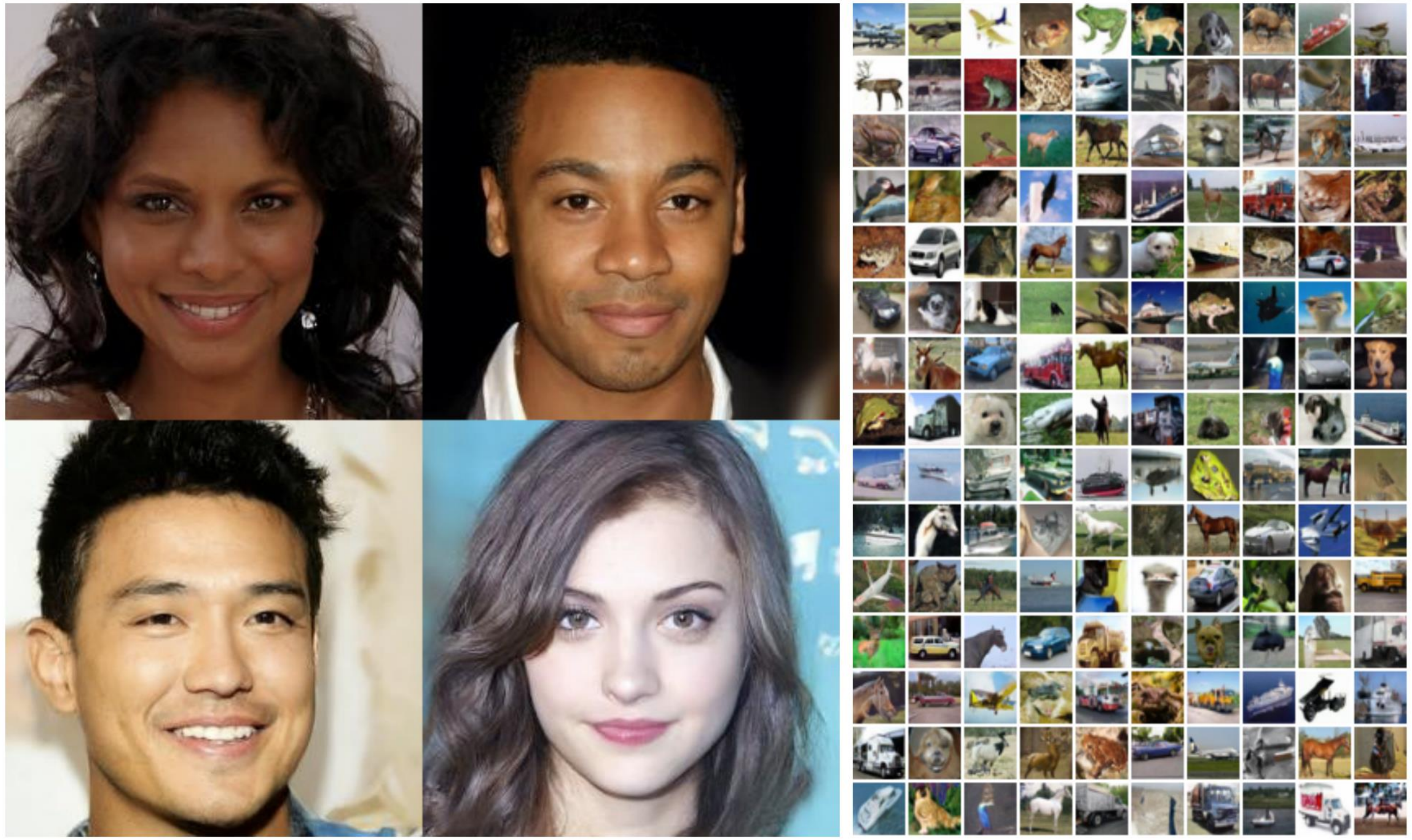
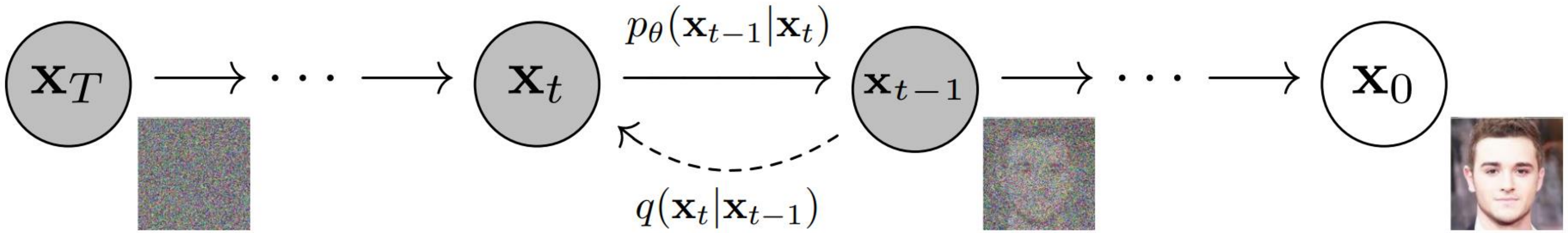


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

Background

- Notation
- Forward process: $q(x_{0:T})$ - predefined Markov chain distribution by $\mathcal{N}$.
    - $q(x_0) \coloneqq p_{data}(x_0)$.
- Reverse process: $p_\theta(x_{0:T})$ - Learnable distribution by $\mathcal{N}$.
- θ: Learnable parameters.



$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$

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

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.

Background

- Reverse process

$$p_\theta(x_{0:T}) \coloneqq p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)\,, \qquad p_\theta(x_{t-1}|x_t) \coloneqq \mathcal{N}\big(x_{t-1};\mu_\theta(x_t,t),\Sigma_\theta(x_t,t)\big)$$

- Forward process or diffusion process

$$q(x_{1:T}) \coloneqq \prod_{t=1}^{T} q(x_t|x_{t-1})\,, \qquad q(x_{t-1}|x_t) \coloneqq \mathcal{N}\big(x_t;\sqrt{1-\beta_t}x_{t-1},\beta_t I\big)$$

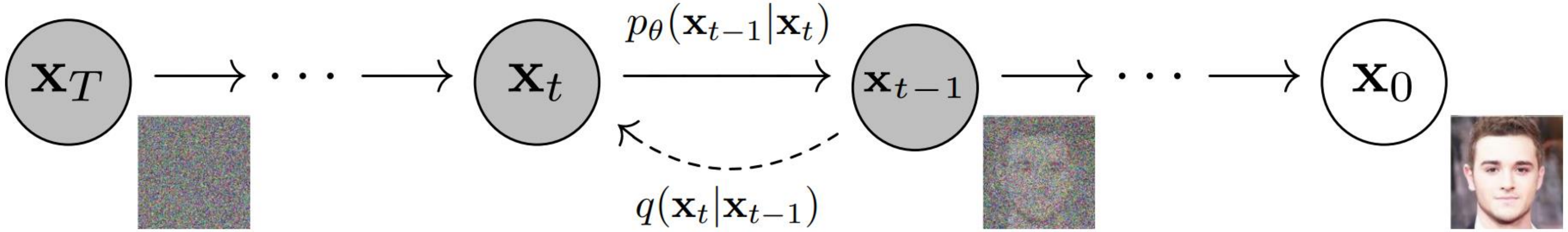

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$

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

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.

Background

- Optimizing the usual variational bound on negative log likelihood $\mathbb{E}_{x_0 \sim q}[-\log p_\theta(x_0)]$.

$$p_\theta(x) = \int p_\theta(x_{0:T}) dx_{1:T}$$

Marginal distribution

$$= \int p_\theta(x_{0:T}) \cdot \frac{q(x_{1:T}|x_0)}{q(x_{1:T}|x_0)} dx_{1:T}$$

$$= \int p_\theta(x_T) \cdot \frac{\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})} \cdot q(x_{1:T}|x_0) dx_{1:T}$$

$$= \int p_\theta(x_T) \cdot q(x_{1:T}|x_0) \cdot \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} dx_{1:T}$$

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.

Background

- Optimizing the usual variational bound on negative log likelihood $\mathbb{E}_{x_0 \sim q}[-\log p_\theta(x_0)]$.

$$\mathbb{E}_{x_0 \sim q}[-\log p_\theta(x_0)]$$

$$= \int -\log p_\theta(x_0) \cdot q(x_0) dx_0$$

$$= \int -\log\left(\int p_\theta(x_T) \cdot q(x_{1:T}|x_0) \cdot \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} dx_{1:T}\right) \cdot q(x_0) dx_0$$

$$= \int -\log\left(\mathbb{E}_{q(x_{1:T}|x_0)}\left[p_\theta(x_T) \cdot \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]\right) \cdot q(x_0) dx_0$$

$$\leq \int -\mathbb{E}_{q(x_{1:T}|x_0)} \log\left(p_\theta(x_T) \cdot \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right) \cdot q(x_0) dx_0$$

Jensen’s inequality

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.

Background

- Optimizing the usual variational bound on negative log likelihood $\mathbb{E}_{x_0\sim q}[-\log p_\theta(x_0)]$.

$$\int -\mathbb{E}_{q(x_{1:T}|x_0)} \log\left(p_\theta(x_T)\cdot\prod_{t=1}^{T}\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right)\cdot q(x_0)dx_0$$

$$= \int -\log\left(p_\theta(x_T)\cdot\prod_{t=1}^{T}\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right)\cdot \underbrace{q(x_0)\cdot q(x_{1:T}|x_0)}_{q(x_{0:T})}dx_0dx_{1:T}$$

$$= \mathbb{E}_{x_{0:T}\sim q}\left[-\log\left(p_\theta(x_T)\cdot\prod_{t=1}^{T}\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right)\right]$$

$$= \mathbb{E}_{x_{0:T}\sim q}\left[-\log p_\theta(x_T) - \sum_{t=1}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] \triangleq L$$

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.

Background

$$L = \mathbb{E}_{x_{0:T}\sim q}\left[-\log p_\theta(x_T) - \sum_{t=1}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right]$$

$$= \mathbb{E}_{x_{0:T}\sim q}\left[-\log p_\theta(x_T) - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} - \log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

$$= \mathbb{E}_{x_{0:T}\sim q}\left[-\log p_\theta(x_T) - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\cdot\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} - \log\frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right]$$

$$= \mathbb{E}_{x_{0:T}\sim q}\left[-\log\frac{p_\theta(x_T)}{q(x_T|x_0)} - \sum_{t=2}^{T}\log\frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} - \log p_\theta(x_0|x_1)\right]$$

$$= \mathbb{E}_{x_{0:T}\sim q}\left[D_{KL}\big(q(x_T|x_0)\parallel p_\theta(x_T)\big) + \sum_{t=2}^{T}D_{KL}\big(p_\theta(x_{t-1}|x_t)\parallel q(x_{t-1}|x_t,x_0)\big) - \log p_\theta(x_0|x_1)\right]$$

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.

Background

- Now the Evidence Lower BOund (ELBO) is tractable.

$$\mathbb{E}_{x_{0:T}\sim q}\left[\underbrace{D_{KL}\big(q(x_T|x_0)\parallel p_\theta(x_T)\big)}_{L_T}+\sum_{t=2}^{T}\underbrace{D_{KL}\big(p_\theta(x_{t-1}|x_t)\parallel q(x_{t-1}|x_t,x_0)\big)}_{L_{t-1}}-\underbrace{\log p_\theta(x_0|x_1)}_{L_0}\right]$$

- with $q(x_{t-1}|x_t,x_0)=\mathcal{N}\big(x_{t-1};\tilde{\mu}_t(x_t,x_0),\tilde{\beta}_t I\big)$

- where $\tilde{\mu}_t(x_t,x_0) \coloneqq \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{(1-\bar{\alpha}_t)}x_0+\frac{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}x_t$ and $\tilde{\beta}_t \coloneqq \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ (proof is started with Bayes' rule).

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *International Conference on Machine Learning*. PMLR, 2015.

Train DDPM

$$L_T = D_{KL}\left(q(x_T|x_0) \parallel p_\theta(x_T)\right)$$

- $q(x_T|x_0)$ converges to standard Gaussian distribution.
- We assume $p_\theta(x_T)$ as the standard Gaussian distribution.
- ⇒ Do not have to train θ in this term.

Train DDPM

$$L_{t-1} = D_{KL}\left(p_\theta(x_{t-1}|x_t) \parallel q(x_{t-1}|x_t, x_0)\right)$$

- $q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \underbrace{\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{(1-\bar{\alpha}_t)}x_0 + \frac{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}x_t}_{\text{Estimate with } \mu_\theta(x_t, t).}, \underbrace{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}_{\text{All parameters are given.}} I\right)$

$$\therefore \Sigma_\theta = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

- Reparametrize with $x_t(x_0, t) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ for $\epsilon \sim \mathcal{N}(0, I)$ and estimate ϵ_θ then minimize L_{t-1} can be same as

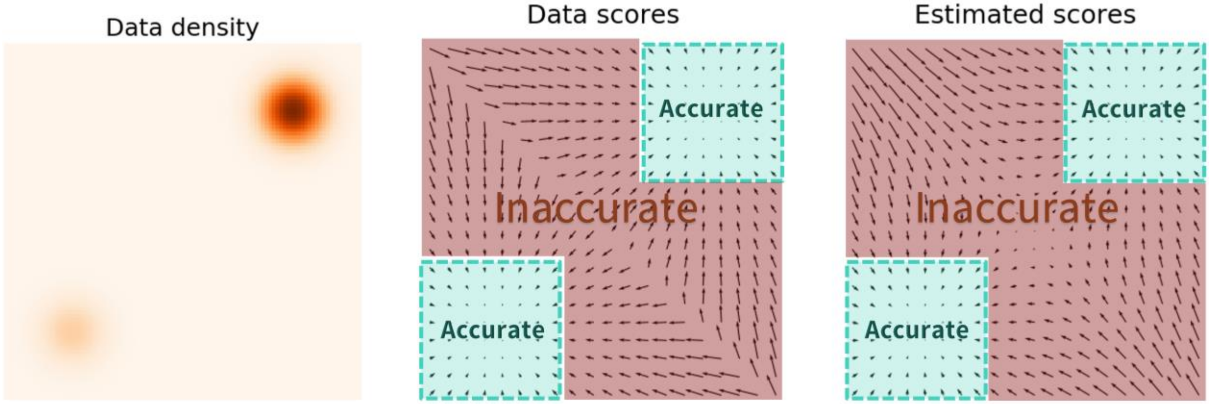
$$\mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right]$$

- which resembles denoising score matching over multiple noise scales indexed by t.

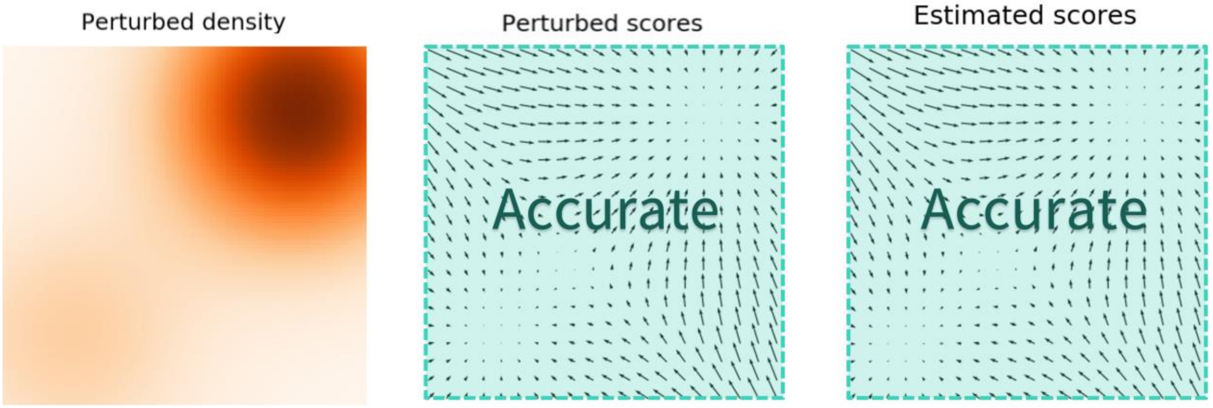
Recap Score-base Generative Model

- We estimate $p_{data}(x)$ to our dataset $\{x_1, \dots x_N\} \sim p_{data}(x)$.

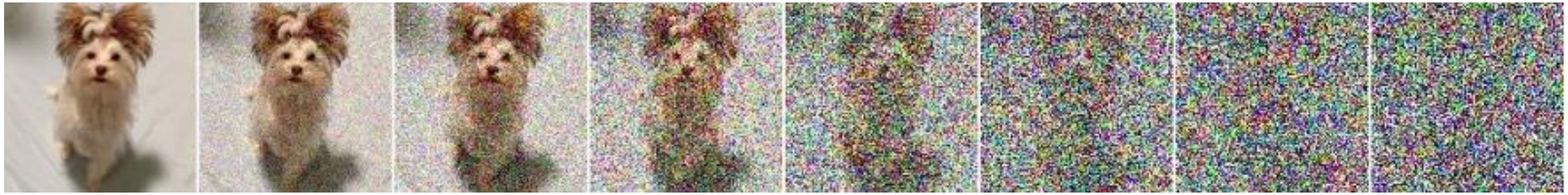
$$\arg\min_{\theta} \mathbb{E}_{p_{data}}\left[\frac{1}{2}\|\nabla_x \log p_{data}(x) - s_\theta(x)\|_2^2\right] \approx \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}[\frac{1}{2}\|s_\theta(x)\|_2^2 + \mathrm{trace}(\nabla_x s_\theta(x))]$$



Estimated scores are only accurate in high density regions.



Estimated scores are accurate everywhere for the noise-perturbed data distribution due to reduced low data density regions.



Song, Yang, and Stefano Ermon. "Generative modeling by estimating gradients of the data distribution." *Advances in Neural Information Processing Systems* 32 (2019).

Train DDPM

$$L_0 = \mathbb{E}_{x_{0:T}\sim q}[-\log p_\theta(x_0|x_1)]$$

- We assume that image data consists of integers in $\{0, 1, \ldots, 255\}$ scaled linearly to $[-1, 1]$.
- Set the last term of the reverse process to an independent discrete decoder derived from the Gaussian $\mathcal{N}(x_0; \mu_\theta(x_1, 1), \sigma_1^2 I)$.

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

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

In practice, this term is optimized by MSE Loss.
Furthermore, there is no independent decoder.

Inference Algorithm

- Although training can be done by single step, **inference can not be done by singe step.**

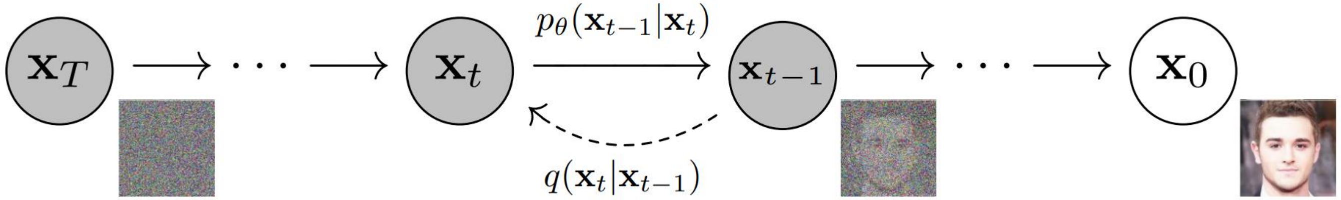
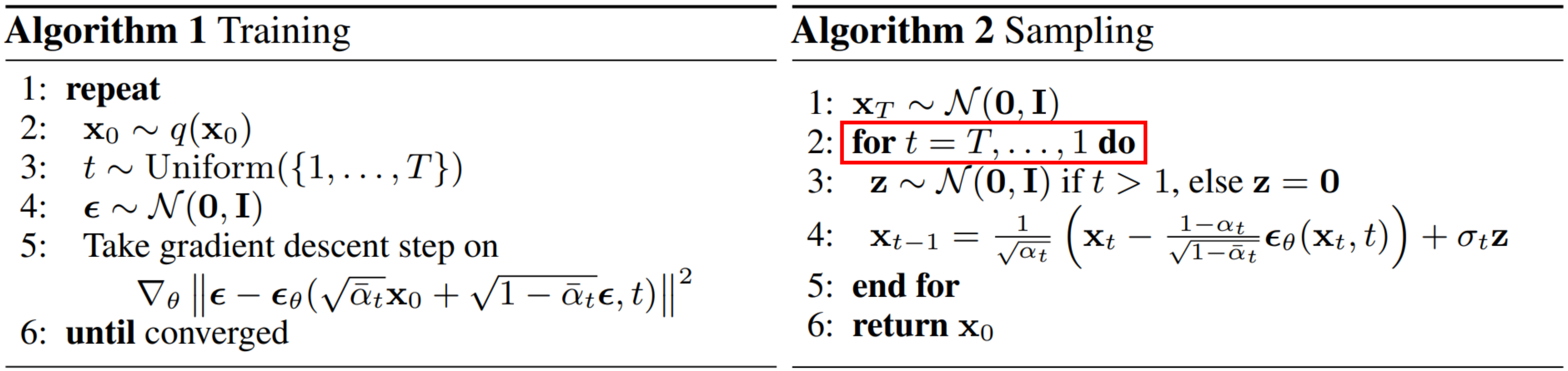

$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$

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



Algorithm 1 Training

```
repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
      $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
until converged
```

Algorithm 2 Sampling

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

Simplified Training Objective

- Simplified objective aggregates L_{t-1} and L_0.
- Ignoring weight of t and discrete Gaussian distribution.

$$L_{simple}(\theta) \coloneqq \mathbb{E}_{t,x_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right]$$

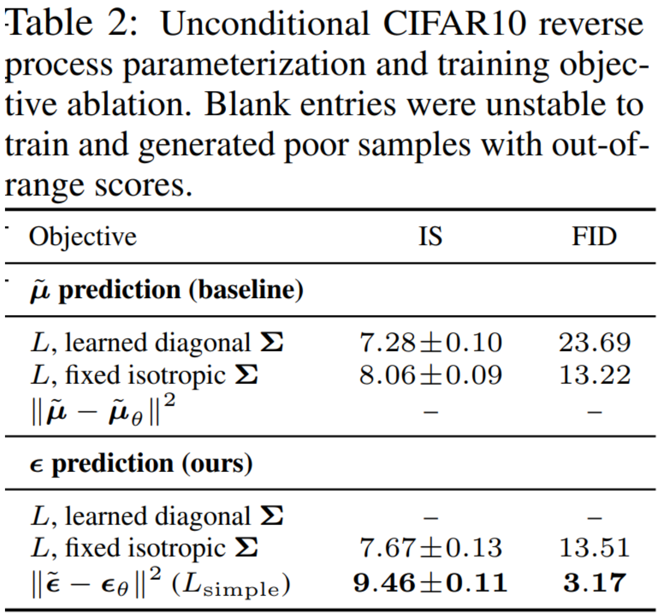


Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

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

Architecture

- U-Net backbone similar to an unmasked PixelCNN++.
- Parameter are shared across time, which is specified to the network using the Transformer sinusoidal positional embedding.
- Use self-attention at the 16×16 feature map resolution.

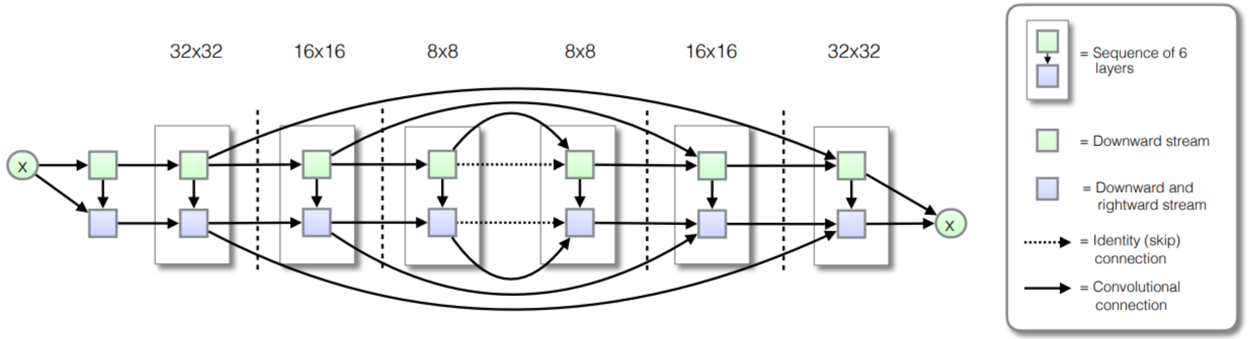

32x32
16x16
8x8
8x8
16x16
32x32
x
x
= Sequence of 6 layers
= Downward stream
= Downward and rightward stream
= Identity (skip) connection
= Convolutional connection

Figure 2: Like van den Oord et al. (2016c), our model follows a two-stream (downward, and downward+rightward) convolutional architecture with residual connections; however, there are two significant differences in connectivity. First, our architecture incorporates downsampling and upsampling, such that the inner parts of the network operate over larger spatial scale, increasing computational efficiency. Second, we employ long-range skip-connections, such that each k-th layer provides a direct input to the $(K-k)$-th layer, where K is the total number of layers in the network. The network is grouped into sequences of six layers, where most sequences are separated by downsampling or upsampling.

Salimans, Tim, et al. "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications." *arXiv preprint arXiv:1701.05517* (2017).

Quantitative Results

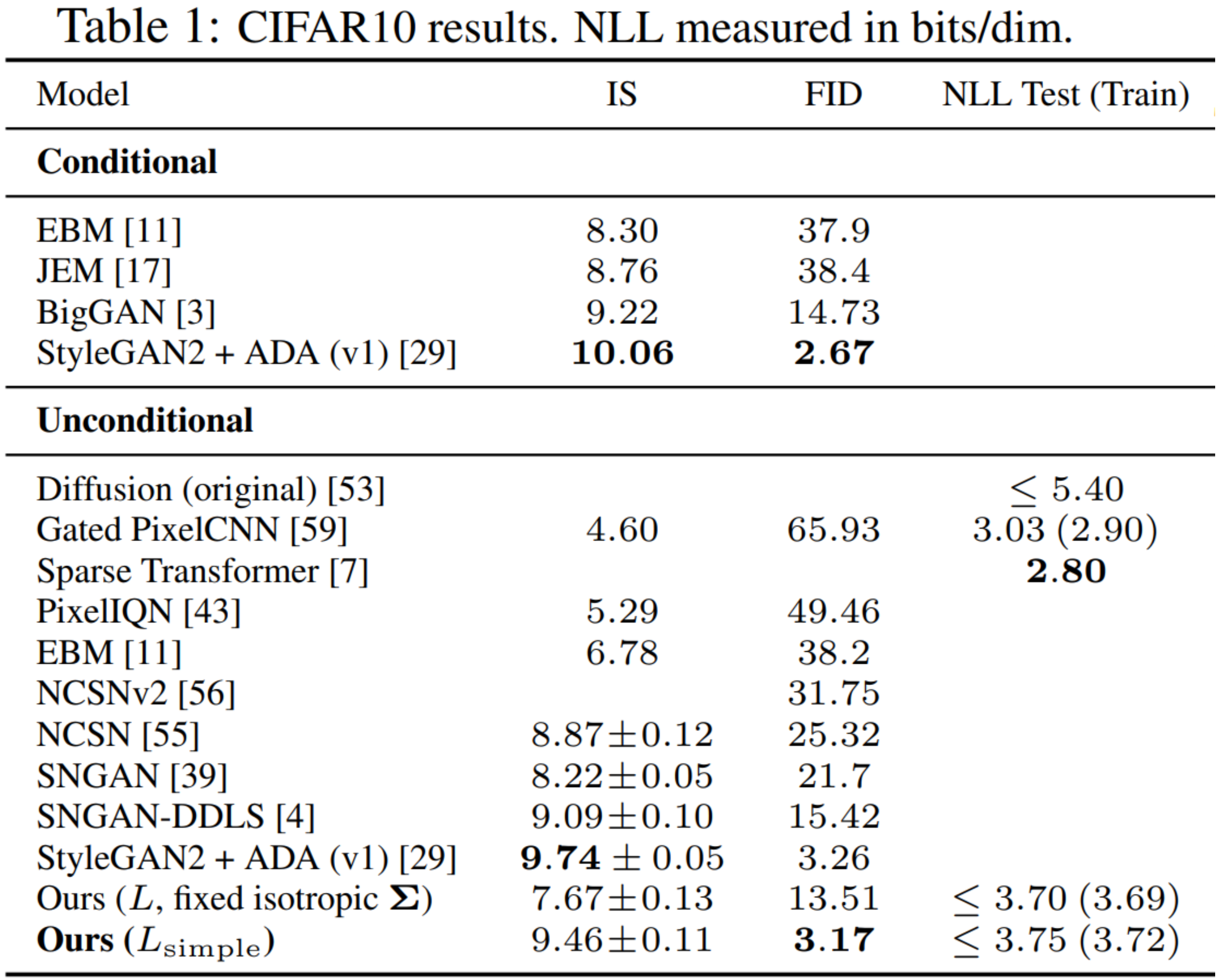


Table 1: CIFAR10 results. NLL measured in bits/dim.

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**
PixelIQN [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]	$\mathbf{9.74} \pm 0.05$	3.26	
Ours (L, fixed isotropic $\mathbf{\Sigma}$)	7.67 ± 0.13	13.51	≤ 3.70 (3.69)
Ours (L_{simple})	9.46 ± 0.11	**3.17**	≤ 3.75 (3.72)

Qualitative Results

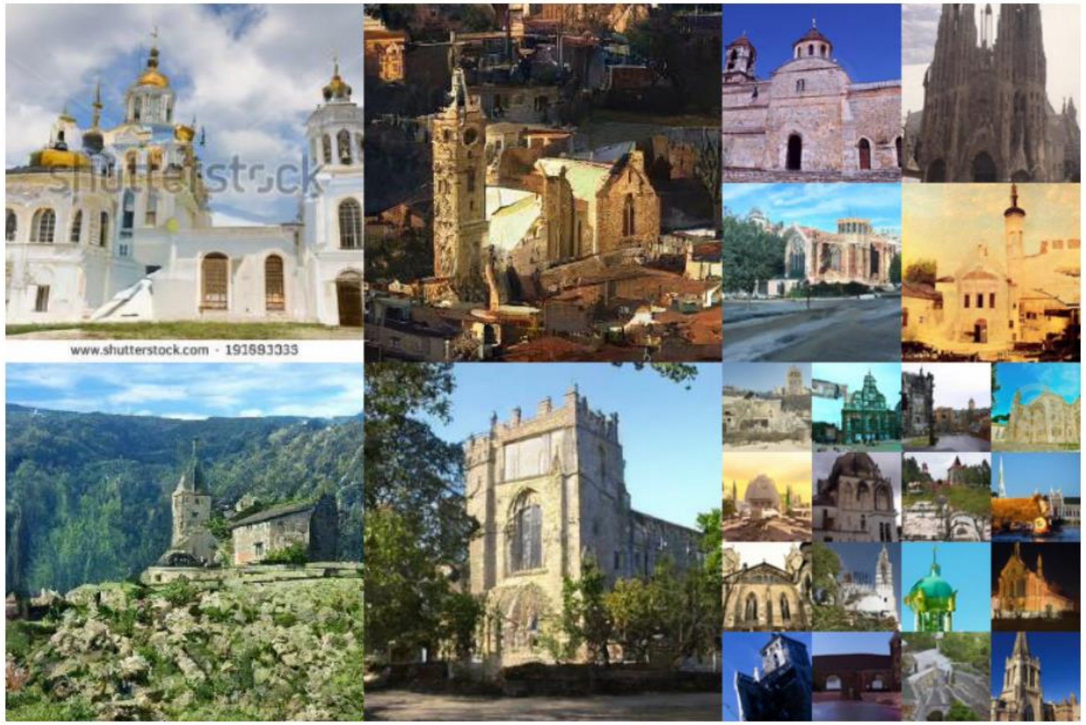


Figure 3: LSUN Church samples. FID=7.89

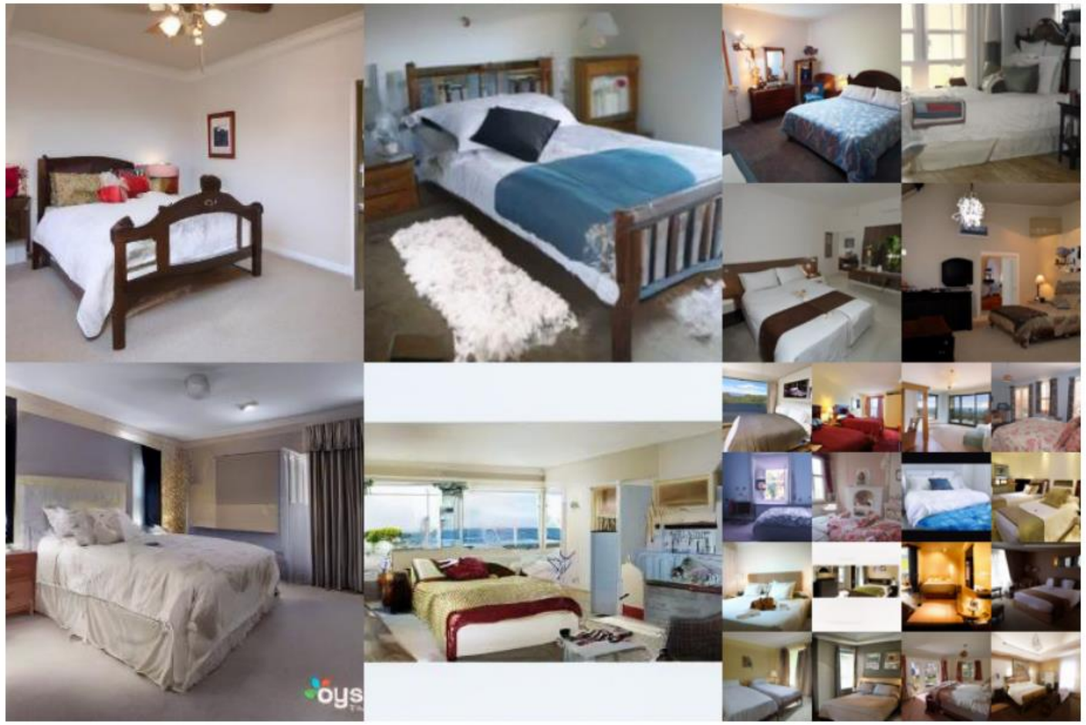
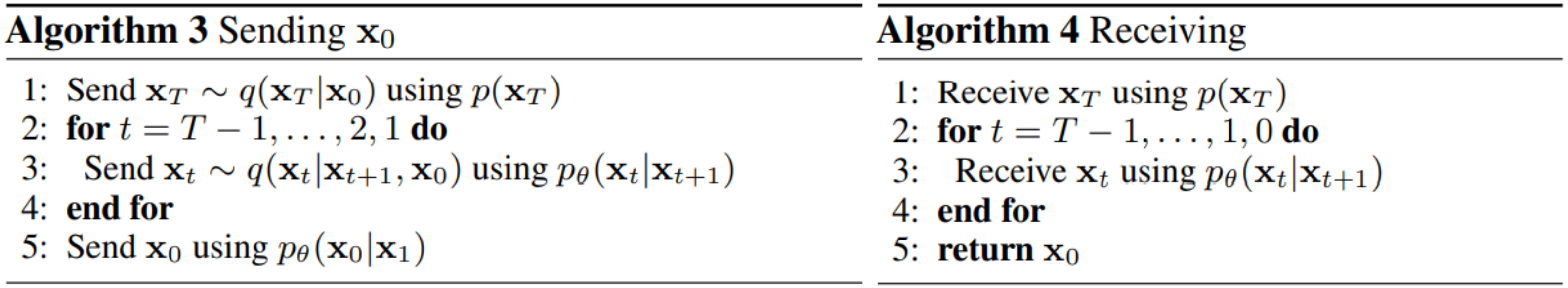


Figure 4: LSUN Bedroom samples. FID=4.90

Progressive Coding

- Progressive lossy compression.

- Distortion (RMSE): $\sqrt{\|x_0 - \hat{x}_0\|^2/D}$ where $\hat{x}_0 = \left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t)\right)/\sqrt{\bar{\alpha}_t}$



Algorithm 3 Sending $\mathbf{x}_0$

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

Algorithm 4 Receiving

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

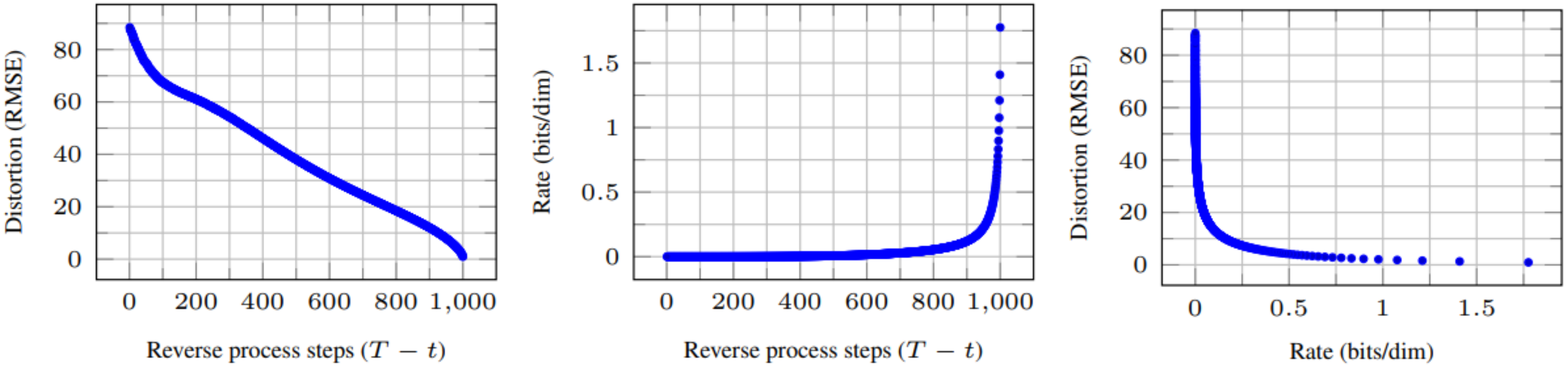


Figure 5: Unconditional CIFAR10 test set rate-distortion vs. time. Distortion is measured in root mean squared error on a $[0, 255]$ scale. See Table 4 for details.

Progressive Coding

- Progressive generation.

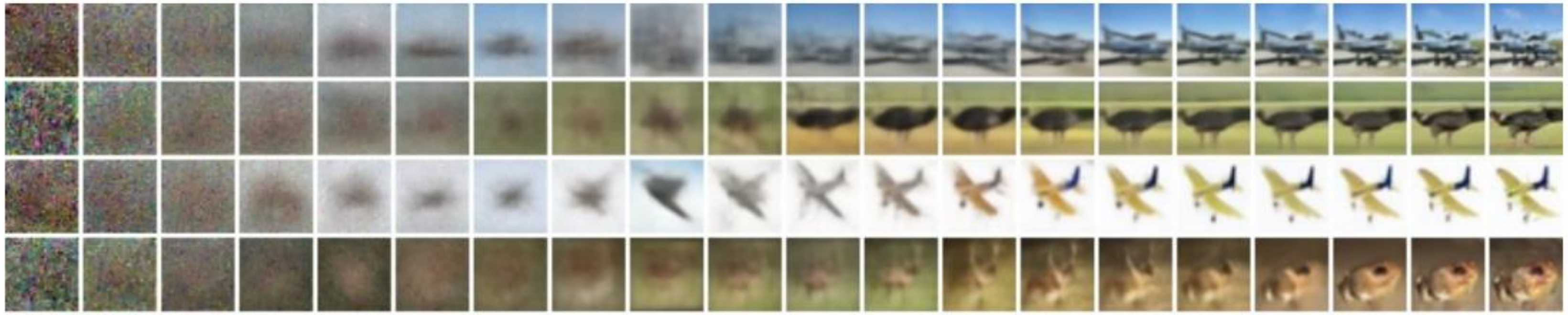


Figure 6: Unconditional CIFAR10 progressive generation ($\hat{\mathbf{x}}_0$ over time, from left to right). Extended samples and sample quality metrics over time in the appendix (Figs. 10 and 14).

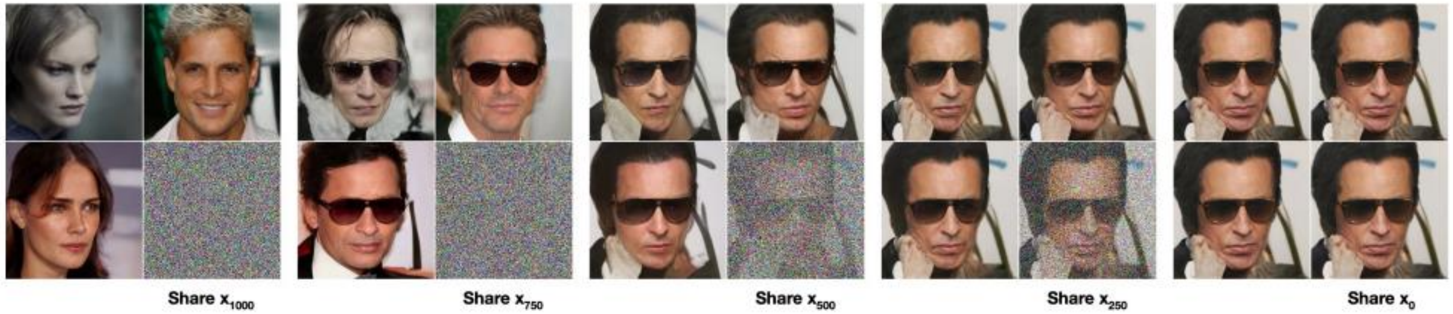


Figure 7: When conditioned on the same latent, CelebA-HQ 256 × 256 samples share high-level attributes. Bottom-right quadrants are $\mathbf{x}_t$, and other quadrants are samples from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

Interpolation

- Large t results in coarser and more varied interpolations, with novel samples at $t = 1000$.

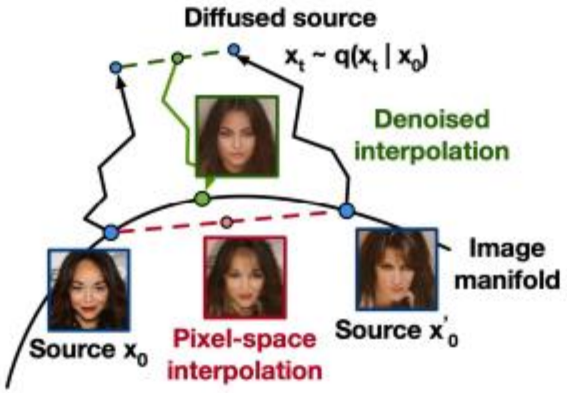


Illustration of interpolation with diffusion models

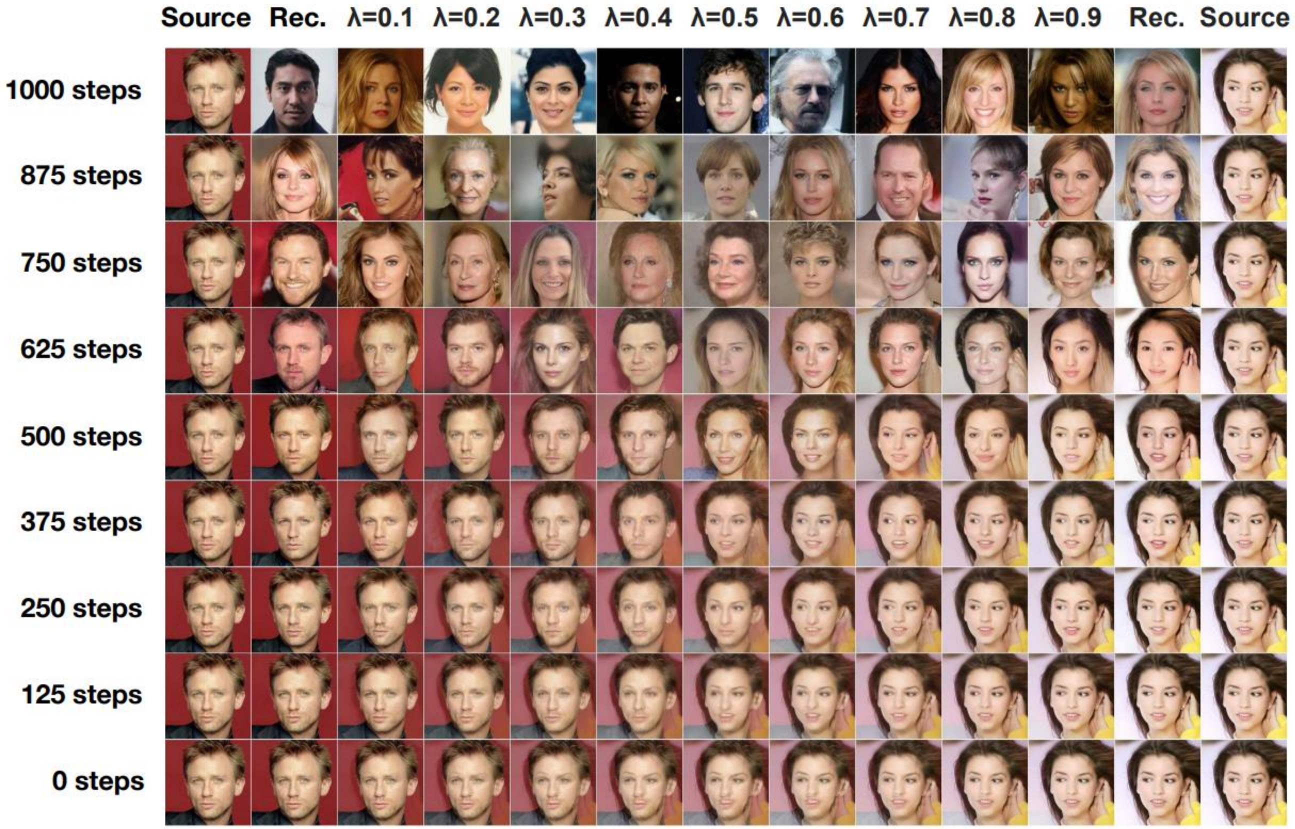


Figure 9: Coarse-to-fine interpolations that vary the number of diffusion steps prior to latent mixing.

Contribution

- Novel generative model which obtain sample quality similar to ProgressiveGAN.
  - A latent variable models training on a weighted variational bound.
- Connection between diffusion probabilistic models and denoising score matching with Langevin dynamics.
- Naturally admit a progressive lossy decompression scheme.