Denoising Diffusion Probabilistic Models

Jonathan Ho, Ajay Jain, and Pieter Abbeel

UC Berkeley

NeurIPS 2020

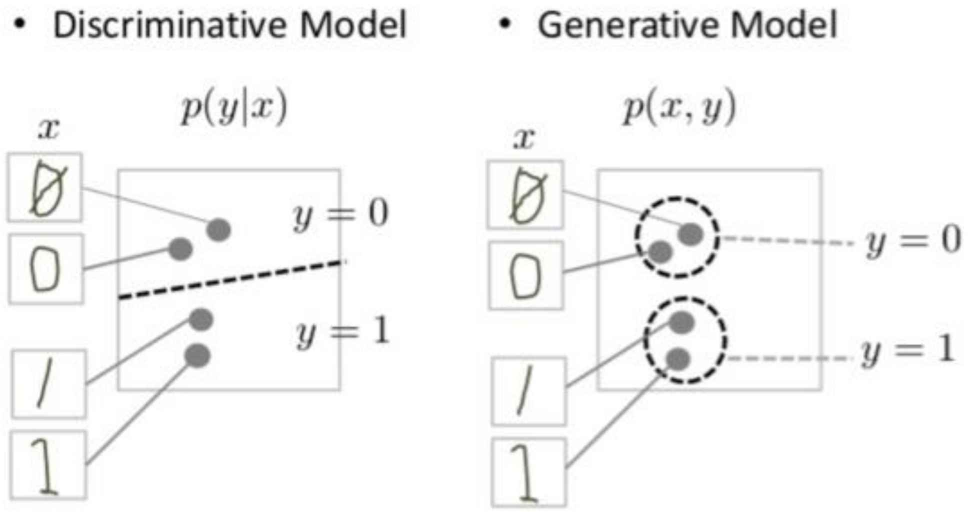
Presented by Minho Park

Contributions

- **Novel generative model** which obtain sample quality similar to ProgressiveGAN (SOTA).
  - A latent variable models training on a weighted variational bound.
- **Connection between diffusion probabilistic models and denoising score matching with Langevin dynamics.**
  - Score matching will be presented next week.
- Naturally admit a progressive lossy decompression scheme.

Generative Models

- A **generative model** is a statistical model of the joint distribution $\boldsymbol{P(X,Y)}$ on given observable variable X and target variable Y.
- A **discriminative model** is a model of the conditional probability $\boldsymbol{P(Y|X=x)}$ of the target Y, given an observation x.
  - And classifiers computed without using a probability model are also referred to loosely as “discriminative”.

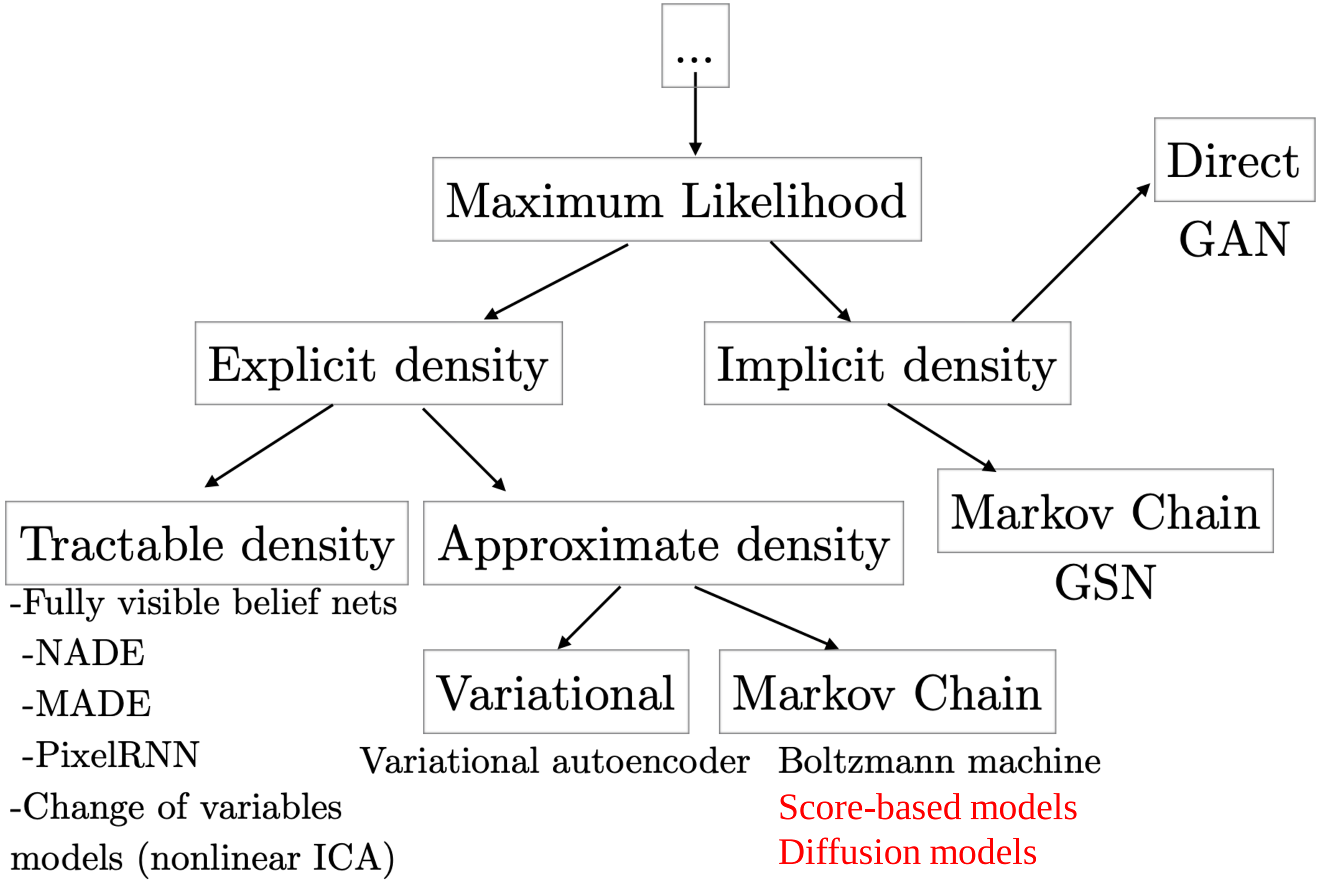


Discriminative and generative models of handwritten digits.

Jebara, Tony. Machine Learning: Discriminative and Generative. The Springer International Series in Engineering and Computer Science. 2004.
Google developers documents. https://developers.google.com/machine-learning/gan/generative

Generative Models in Computer Vision

- Generative models that work via the principle of maximum likelihood.

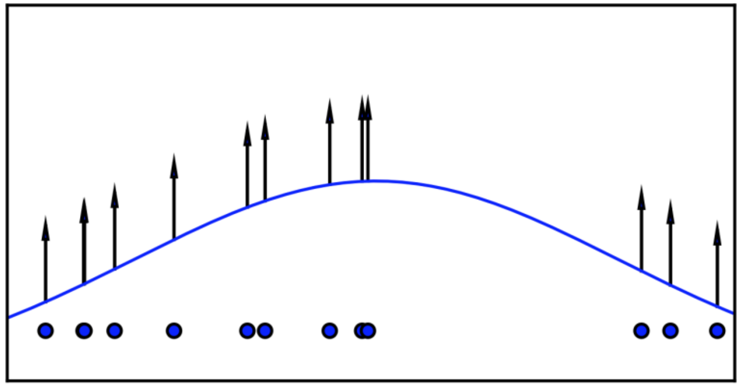


A taxonomy of deep generative models.

Goodfellow, Ian. "Nips 2016 tutorial: Generative adversarial networks." *arXiv preprint arXiv:1701.00160.* 2016.

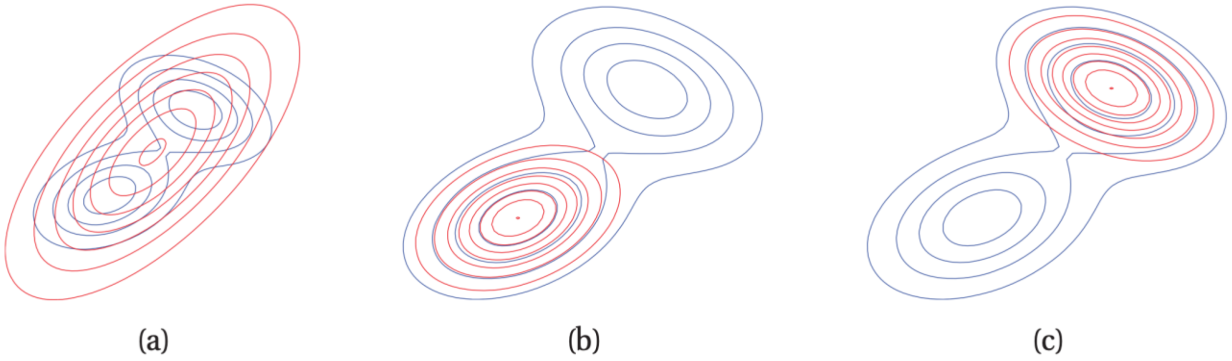
Maximum Likelihood Estimation

- MLE process: $\theta^*_{MLE} = \underset{\theta}{\operatorname{argmax}} P(\mathcal{D}|\theta)$
  - Log likelihood: $\log P(\mathcal{D}|\theta) = \mathbb{E}_{x \sim p_{data}}[\log p_{model}(x;\theta)]$
- KL divergence between data generating distribution and the model
  - $\theta^*_{MLE} = \underset{\theta}{\operatorname{argmin}} D_{KL}\big(p_{data}(x) \parallel p_{model}(x;\theta)\big)$
  - $= \underset{\theta}{\operatorname{argmin}} \mathbb{E}_{x \sim p_{data}}[\log p_{data}(x) - \log p_{model}(x;\theta)] = \underset{\theta}{\operatorname{argmax}} \mathbb{E}_{x \sim p_{data}}[\log p_{model}(x;\theta)]$



$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\arg\max}\, \mathbb{E}_{x \sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

Maximum likelihood estimation process.



Optimization results using forward (a) / reverse (b), (c) KL divergence

Goodfellow, Ian. "Nips 2016 tutorial: Generative adversarial networks." *arXiv preprint arXiv:1701.00160.* 2016.
Murphy, Kevin P. Machine learning: a probabilistic perspective. MIT press, 2012.

Diffusion Models: Notation

- $p(x_{0:T})$: Joint distribution of $x_0, x_1, \dots, x_T$
- q: Real distribution (p_{data})
- p_θ: Modelled distribution parameterized by θ ($p_{model}(\cdot\,;\theta)$)
- Forward process (diffusion process): $q(x_t|x_{t-1}) \triangleq \mathcal{N}\left(x_t; \sqrt{\alpha_t}x_{t-1}, 1-\alpha_t I\right)$ Scheduled by α_t and $\beta_t = 1-\alpha_t$
  - Then, $q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I\right)$ where $\bar{\alpha}_t \triangleq \prod_{s=1}^{t}\alpha_t$.

Derived from Mathematical Induction and the definition of the Gaussian distribution.

- Reverse process (denoising process): $p_\theta(x_{t-1}|x_t)$

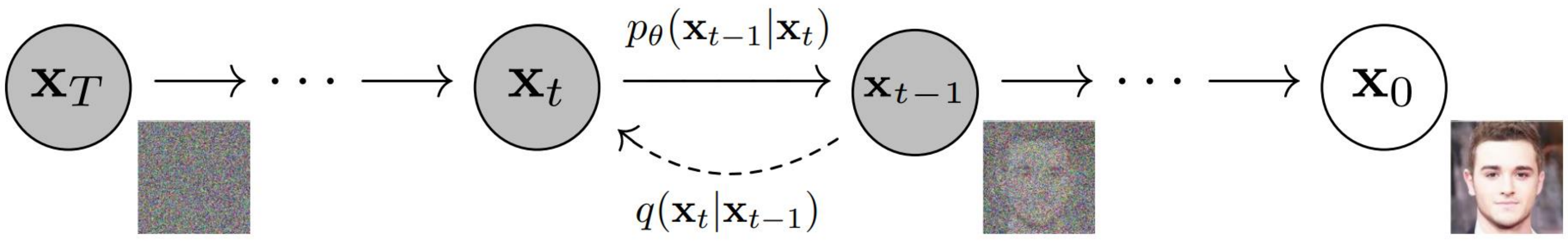


Figure 2: The directed graphical model considered in this work.

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *ICML.* 2015.

Diffusion Models: Maximum Likelihood Estimation

- Let's maximize the log-likelihood $\mathbb{E}_{x_0 \sim q}[\log p_\theta(x_0)]$.

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

Marginal distribution

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

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

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

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

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *ICML.* 2015.

Diffusion Models: Maximum Likelihood Estimation

- Let's maximize the log-likelihood $\mathbb{E}_{x_0\sim q}[\log p_\theta(x_0)]$.

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

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

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

Previous slide

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

Evidence Lower BOund (ELBO)
or Variational bound

Jensen's inequality

Equality holds iff $p_\theta(x_T) \cdot \prod_{t=1}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}$ is constant for all $x_{1:T}$.

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *ICML.* 2015.

Diffusion Models: Deriving the ELBO

- Understanding equality of the ELBO w.r.t. KL divergence.

$$\log p_\theta(x_0) = \log\left(\mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]\right)$$

$$\geq \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log\left(\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right)\right]$$

- Then,

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

$$= \mathbb{E}_{x_{1:T}\sim q(x_{1:T}|x_0)}\left[\log\left(\frac{q(x_{1:T}|x_0)}{p_\theta(x_{1:T}|x_0)}\right)\right]$$

$$= D_{KL}\big(q(x_{1:T}|x_0) \parallel p_\theta(x_{1:T}|x_0)\big) \geq 0$$

Equality holds iff $q(x_{1:T}|x_0) = p_\theta(x_{1:T}|x_0)$ is constant for all $x_{1:T}$

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *ICML.* 2015.

Diffusion Models: Training Objective

- Let's minimize the negative log-likelihood $-\mathbb{E}_{x_0 \sim q}[\log p_\theta(x_0)]$.

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

Previous slide

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

Equation (3) in the paper

Assumption of the Markov Chain

$$q(x_{t-1}|x_t, x_0) = \frac{q(x_t|x_{t-1}, \cancel{x_0})q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

$$= \frac{\frac{1}{\sqrt{1-\alpha_t} \cdot \sqrt{2\pi}} \exp\left(-\frac{\left\|x_t - \sqrt{\alpha_t}x_{t-1}\right\|^2}{2(1-\alpha_t)}\right) \cdot \frac{1}{\sqrt{1-\bar{\alpha}_{t-1}} \cdot \sqrt{2\pi}} \exp\left(-\frac{\left\|x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0\right\|^2}{2(1-\bar{\alpha}_{t-1})}\right)}{\frac{1}{\sqrt{1-\bar{\alpha}_t} \cdot \sqrt{2\pi}} \exp\left(-\frac{\left\|x_t - \sqrt{\bar{\alpha}_t}x_0\right\|^2}{2(1-\bar{\alpha}_t)}\right)}$$

$$= \mathcal{N}\left(x_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t, \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t I\right)$$

1. We cannot change it to the KL divergence form directly.

2. We don't know $q(x_{t-1}|x_t)$.

Therefore, we change it into a gaussian distribution $q(x_{t-1}|x_t, x_0)$, and make it in the form of KL divergence.

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *ICML.* 2015.

Diffusion Models: Training Objective

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

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

Handle the $q(x_1|x_0)$

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

Desired term

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

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

Sohl-Dickstein, Jascha, et al. "Deep unsupervised learning using nonequilibrium thermodynamics." *ICML.* 2015.

Forward Process and L_T

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

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

Reverse Process and $L_{1:T-1}$

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

- $q(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{(1-\bar{\alpha}_t)}x_0 + \frac{\sqrt{\bar{\alpha}_t}(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}x_t, \overbrace{\underline{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}}^{\sigma_t^2} I\right)$
- $p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$

Estimate 1. $\mu_\theta(x_t, t)$
2. $x_0(x_t, t)$
3. $\epsilon_\theta(x_t, t)$

Estimate $\Sigma_\theta(x_t, t)$ or use σ_t^2

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

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

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

Simplified Training Objective

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

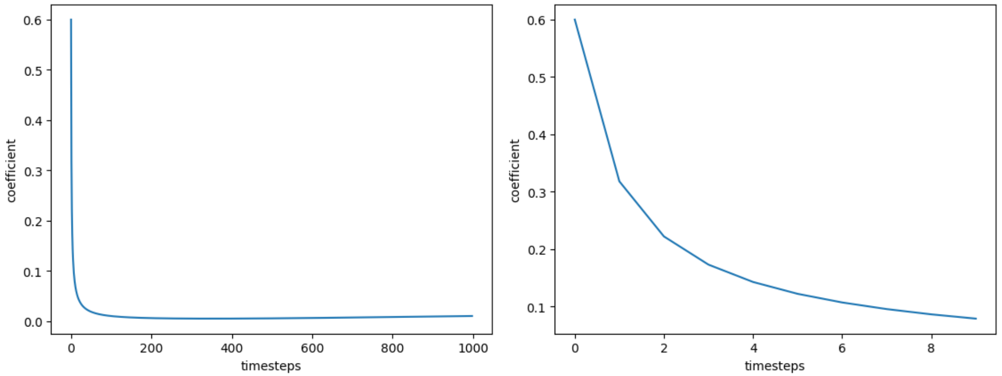
$$L_{t-1} = \mathbb{E}_{x_0,\epsilon}\left[\cancel{\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}}\left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t\right)\right\|^2\right]$$

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

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})	$\mathbf{9.46\pm0.11}$	**3.17**

$$\text{coefficient} = \frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}$$




```
import numpy as np
import matplotlib.pyplot as plt

betas = np.linspace(1e-4, 0.02, 1000)
alphas = 1 - betas
alphas_cumprod = np.cumprod(alphas)
sigmas = (1 - alphas_cumprod[:-1]) / (1 -
alphas_cumprod[1:]) * betas[1:]
coeff = betas[1:] ** 2 / (2 * sigmas *
alphas[1:] * (1 - alphas_cumprod[1:]))
```

Data Scaling, Reverse Process Decoder, and L_0

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

- 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 $\mathcal{N}(x_0; \mu_\theta(x_1, 1), \sigma_1^2 I)$.

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

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

Clipping into $[-1, 1]$

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.
  - E.g., autoregressive models.

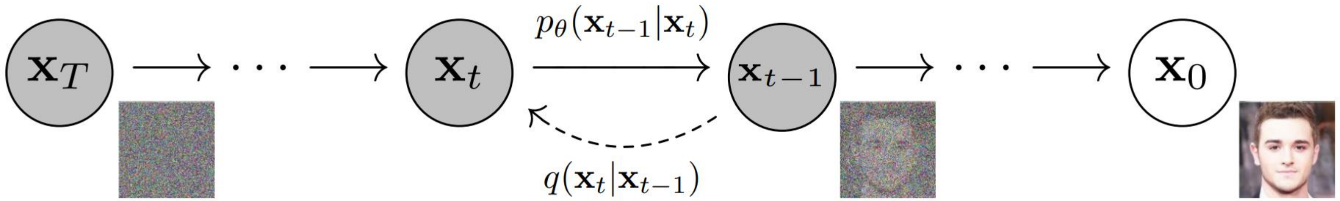


Figure 2: The directed graphical model considered in this work.

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 $\mathbf{x}_0$

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.

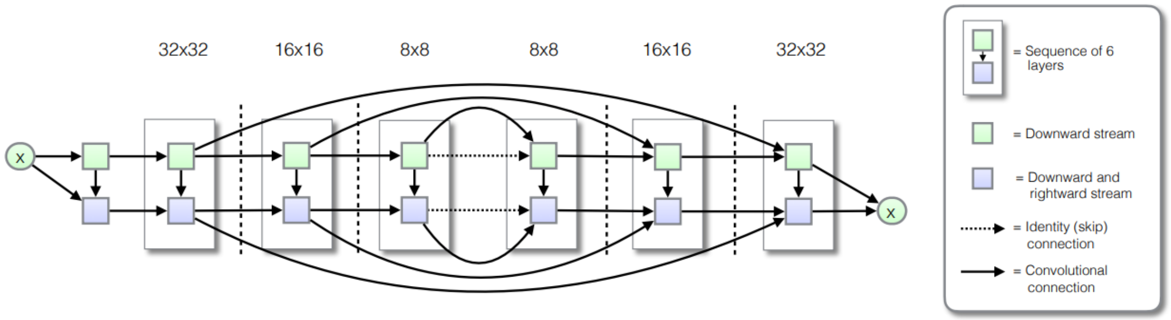


Figure 2: Like van den Oord et al. (2016c), our model follows a two-stream (downward, and downward+rightward) convolutional architecture with residual connections; however, there are two significant differences in connectivity. First, our architecture incorporates downsampling and up-sampling, such that 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.
https://github.com/lucidrains/denoising-diffusion-pytorch/blob/aab2b04e95a266568ff768c8175d2c9bc1b66d4e/denoising_diffusion_pytorch/denoising_diffusion_pytorch.py#L267

Qualitative Results

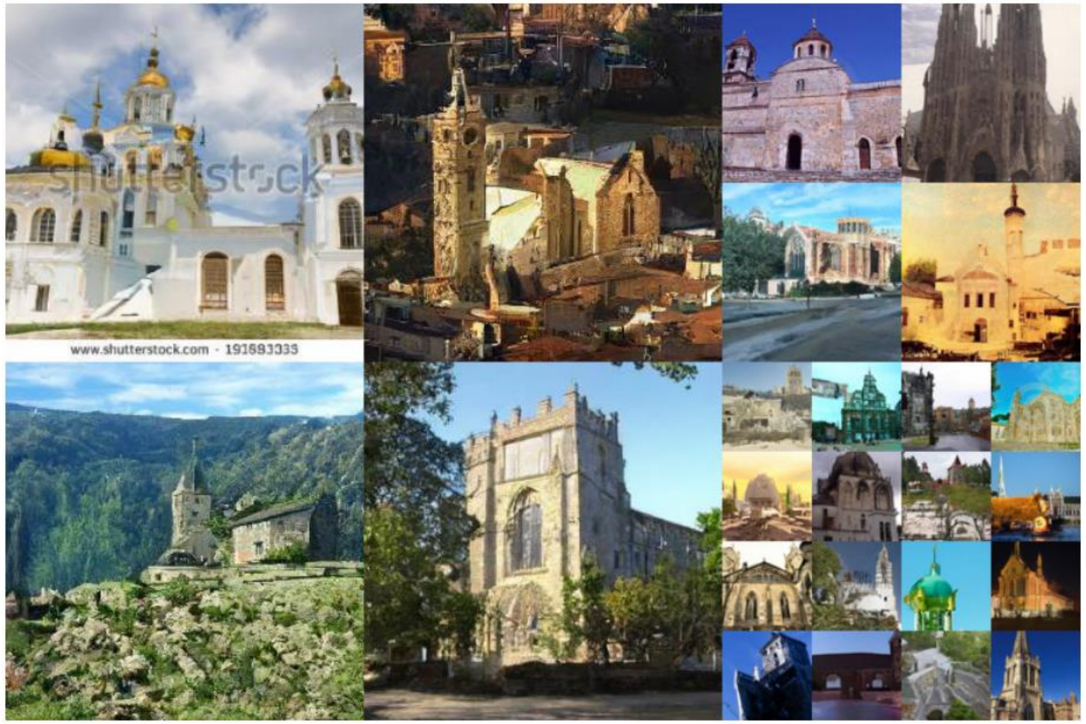


Figure 3: LSUN Church samples. FID=7.89

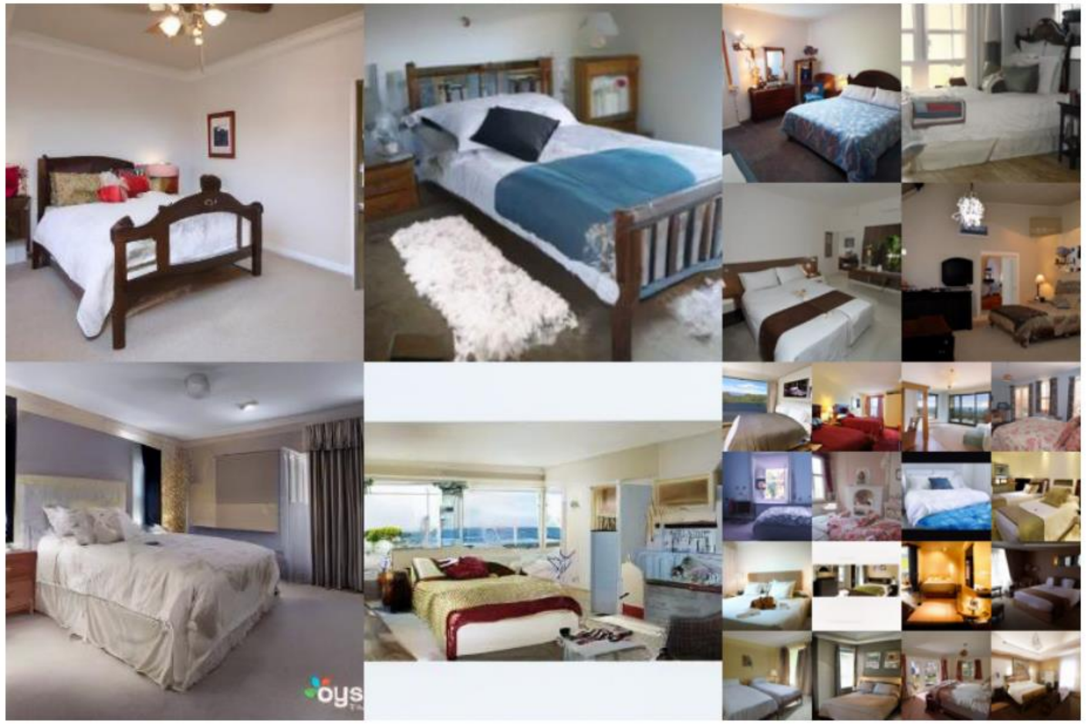


Figure 4: LSUN Bedroom samples. FID=4.90

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)

Progressive Coding: Rate-Distortion Theory

- Rate (bits/dim): $L_t + \cdots + L_T$
- Distortion (RMSE): $\sqrt{\|x_0 - \hat{x}_0\|^2 / D}$

Algorithm 3 Sending $\mathbf{x}_0$

Send $\mathbf{x}_T \sim q(\mathbf{x}_T|\mathbf{x}_0)$ using $p(\mathbf{x}_T)$
for $t = T-1, \ldots, 2, 1$ **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 $t = T-1, \ldots, 1, 0$ **do**
  Receive $\mathbf{x}_t$ using $p_\theta(\mathbf{x}_t|\mathbf{x}_{t+1})$
end for
return $\mathbf{x}_0$

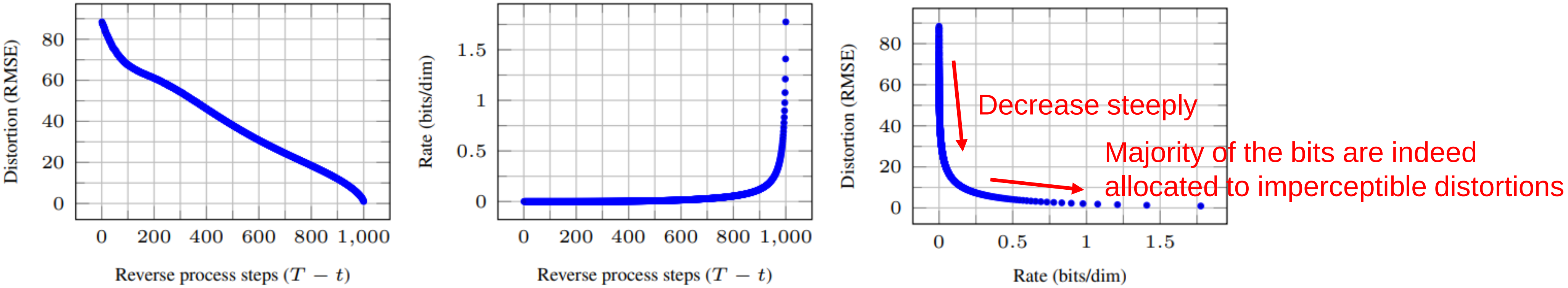


Figure 5: Unconditional CIFAR10 test set rate-distortion vs. time. Distortion is measured in root mean squared error on a [0, 255] scale. See Table 4 for details.

Progressive Coding: Estimate $\hat{x}_0$ directly.

- We can directly estimate $\hat{x}_0$ for each timestep using $\hat{x}_0 = \left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t)\right)/\sqrt{\bar{\alpha}_t}$.

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).

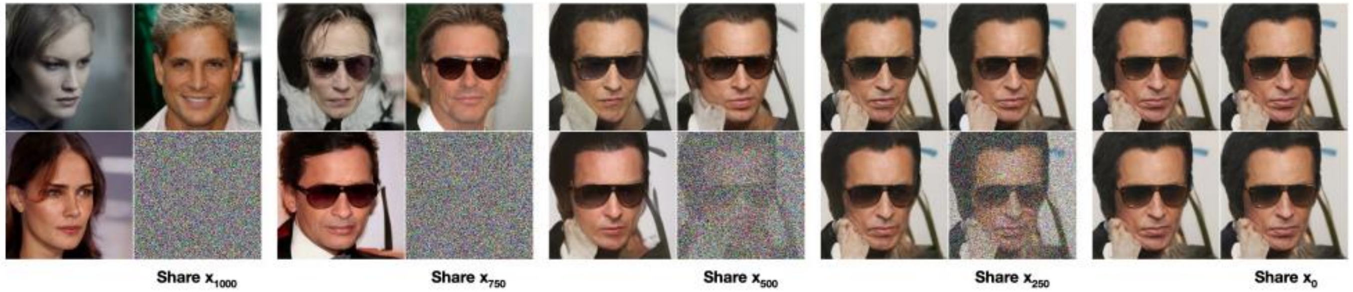


Figure 7: When conditioned on the same latent, CelebA-HQ 256 × 256 samples share high-level attributes. Bottom-right quadrants are $\mathbf{x}_t$, and other quadrants are samples from $p_\theta(\mathbf{x}_0|\mathbf{x}_t)$.

Interpolation

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

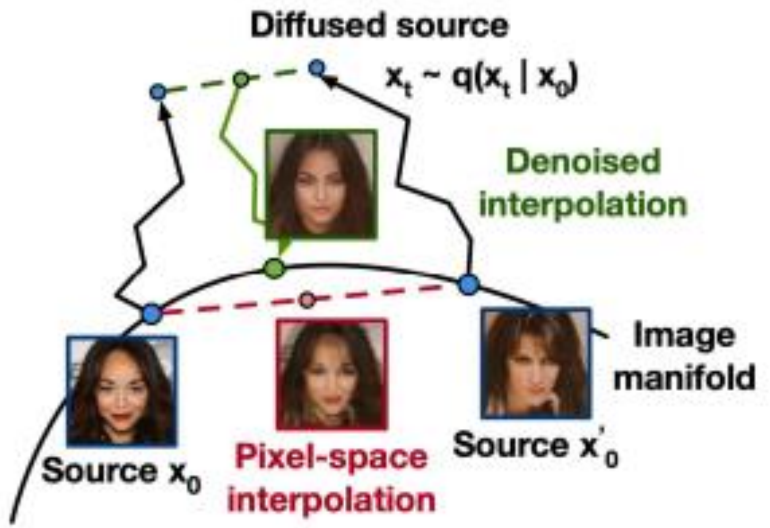


Illustration of interpolation with diffusion models

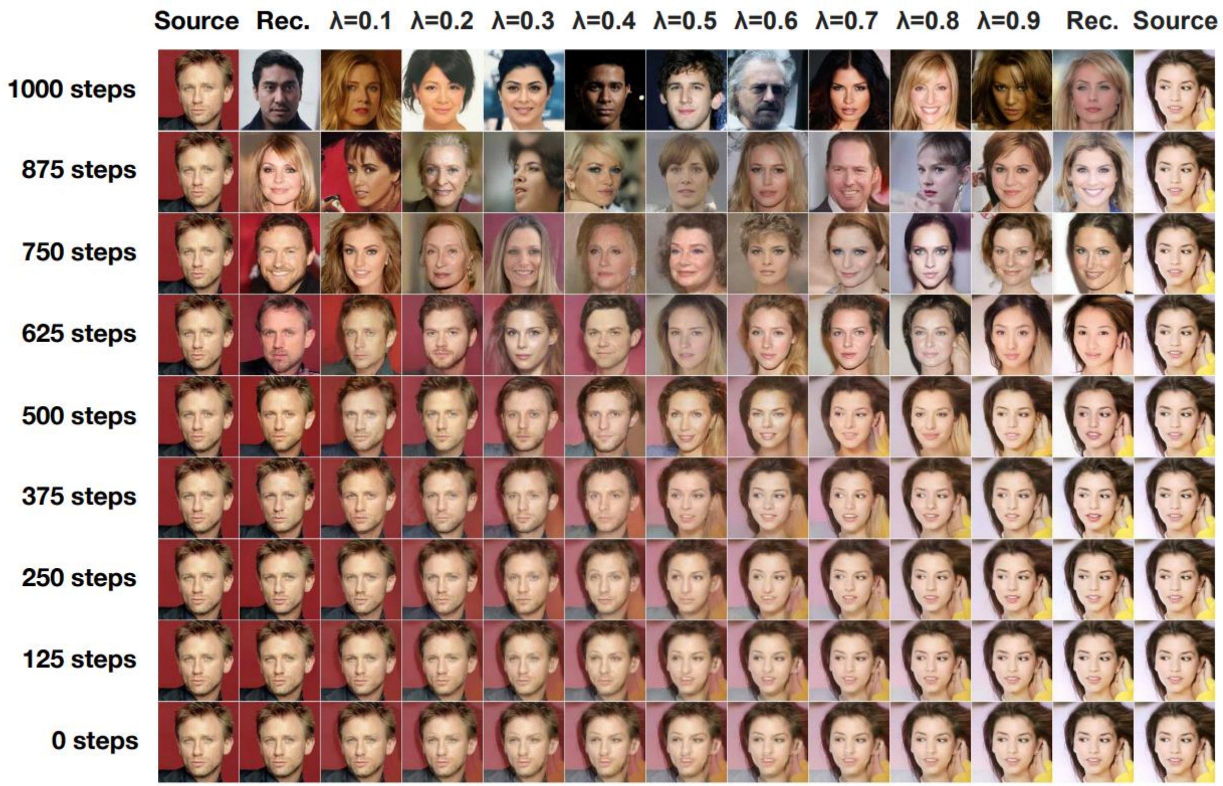


Figure 9: Coarse-to-fine interpolations that vary the number of diffusion steps prior to latent mixing.

Contributions

- **Novel generative model** which obtain sample quality similar to ProgressiveGAN (SOTA).
  - A latent variable models training on a weighted variational bound.
- **Connection between diffusion probabilistic models and denoising score matching with Langevin dynamics.**
  - Score matching will be presented next week.
- Naturally admit a progressive lossy decompression scheme.