Denoising Diffusion Probabilistic Models

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2020)*. 6840–6851.

2025년 7월 24일

이규원

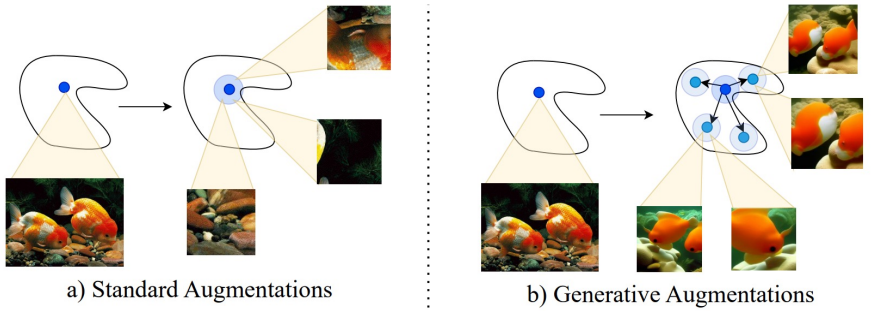
Department of Computer Science and Engineering
Chung-Ang University

Index

- Backgrounds
- Motivation
- Methods
- Experimental Results
- Conclusions

Backgrounds

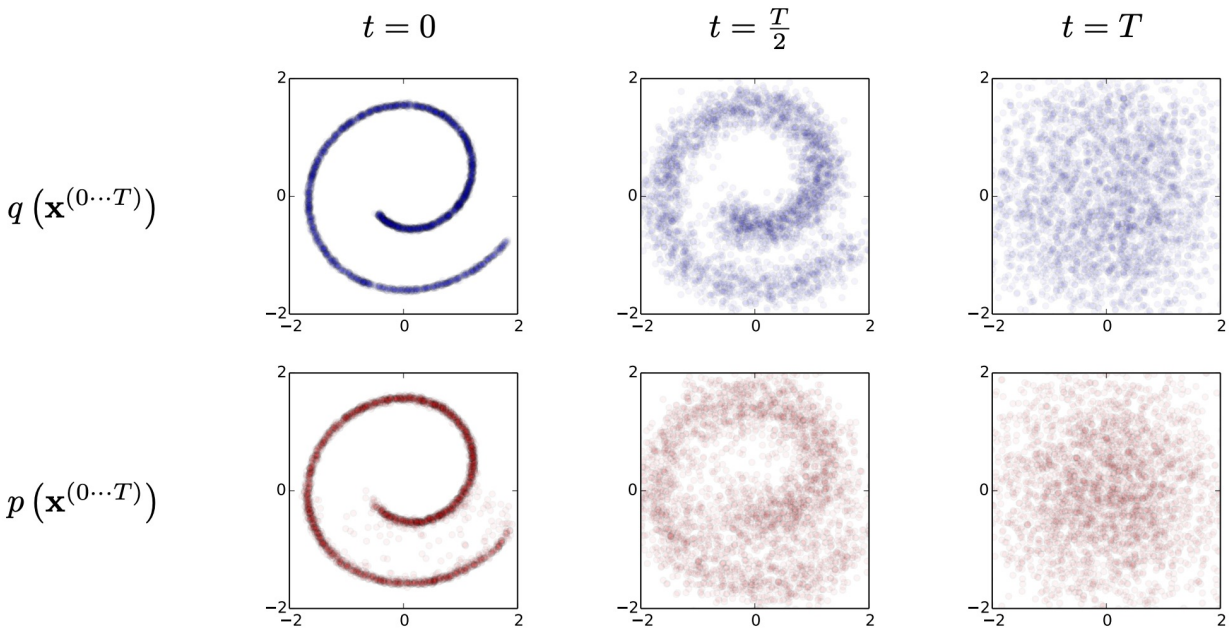
- **Generative models**
  - Models that learn the data generation process and generate new samples
- **Applications of generative models**
  - Creative work (e.g., art, music)
  - Data augmentation – when the dataset is small



Ayromlou, Sana, et al. "Can Generative Models Improve Self-Supervised Representation Learning?." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. No. 2. 2025.

Backgrounds - Generative Models

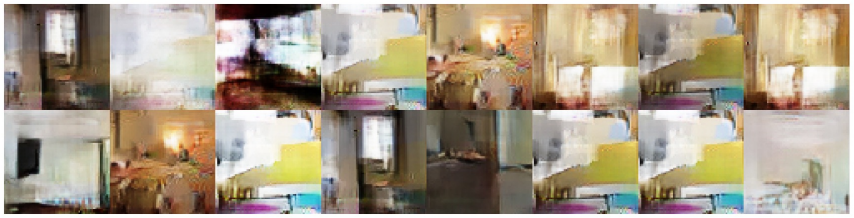
- **Example: Diffusion(Sohl-Dickstein et al., 2015)**
  - Data structure is gradually destroyed via a fixed forward diffusion process
  - A learnable reverse process is trained to reconstruct the data distribution



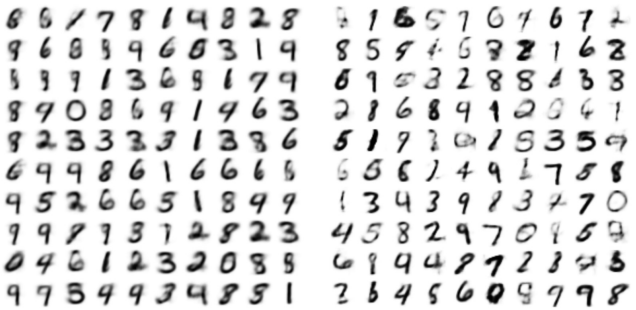
Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In International Conference on Machine Learning, pages 2256–2265, 2015.

Motivation

- **Limitations of Diffusion(Sohl-Dickstein et al., 2015)**
  - Sample quality inferior to GANs or VAEs at the time

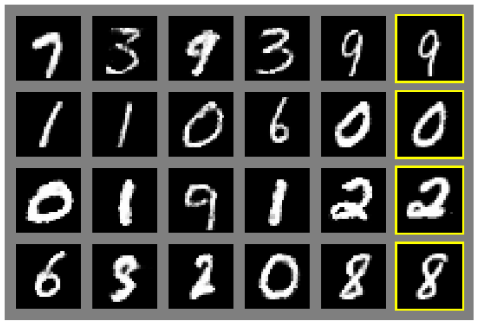


- **Limitation of previous methods**
  - VAE: Tends to produce blurry outputs
  - GAN: prone to mode collapse



(a) 2-D latent space (b) 5-D latent space

VAE



GAN

Methods - Overview

- **Idea**
  - Fixed diffusion rate (β) scheduling
    - Uses a linear variance schedule from small to moderate noise levels
    - Empirically shown to perform as well as or better than learned schedules
  - Simplified objective via ε-prediction
    - Resembles denoising score matching(training), Langevin dynamics(sampling)
    - Leads to stable training and significantly improved sample quality
  - Discretized gaussian decoder for images
    - Models pixel values as integer bins with integrated Gaussian likelihood

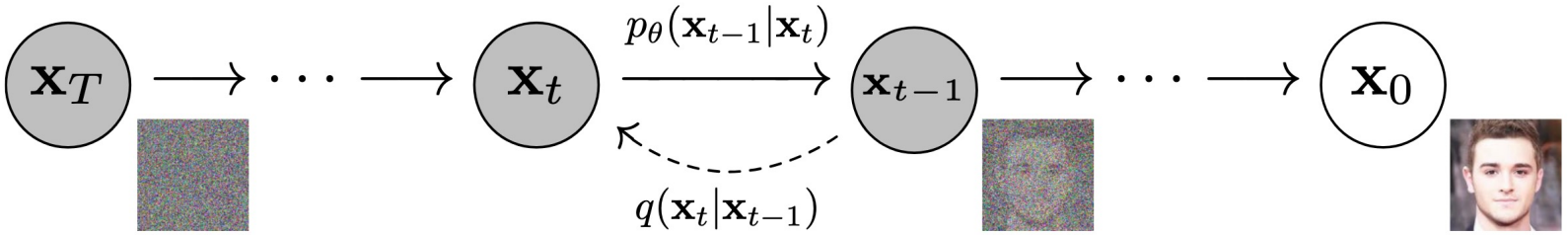


Figure 2: The directed graphical model considered in this work.

Methods - Overview

- **Forward process (Diffusion)**
  - Gradually adds Gaussian noise to data across T steps
  - Uses a fixed noise schedule over time
  - $q(x_{1:T}|x_0) \coloneqq \prod_{t=1}^{T} q(x_t|x_{t-1}), \qquad q(x_t|x_{t-1}) \coloneqq \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathrm{I}\right)$
- **Reverse process (Denoising)**
  - $p_\theta(x_{0:T}) \coloneqq p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \qquad p_\theta\left(x_{t-1}|x_t\right) \coloneqq \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$
  - The model learns the reverse transitions to reconstruct data step-by-step

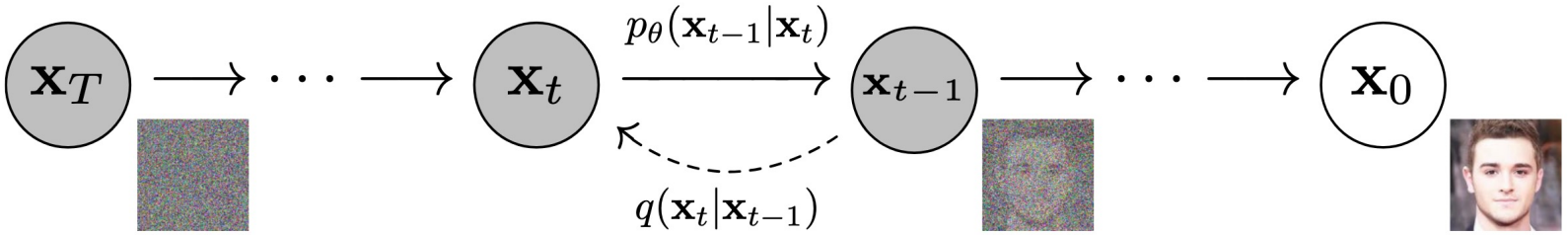


Figure 2: The directed graphical model considered in this work.

Methods

- **Forward process and $\mathbf{L_T}$**
  - The forward process yields the prior x_t, which can be viewed as noisy or random
  - Since β follows a fixed noise schedule, the L_T term is constant and ignored during training
- **Reverse process and $\mathbf{L_{1:T-1}}$**
  - The reverse process is trained to accurately predict the posterior x_{t-1} from x_t
  - The L_{t-1}term is reformulated as an ε prediction loss
- **Data scaling, reverse process decoder, and $\mathbf{L_0}$**
  - The model should generate a high-quality final output L_0

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] = \mathbb{E}_q\left[-\log p(x_t) - \sum_{t\geq 1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}\right] \qquad p_\theta(x_{0:T}) \coloneqq p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

$$\mathbb{E}_q\left[D_{KL}\big(q(x_T|x_0) \parallel p(x_T)\big) + \sum_{t>1} D_{KL}\big(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)\big) - \log p_\theta(x_0|x_1)\right]$$

L_T $L_{1:T-1}$ L_0

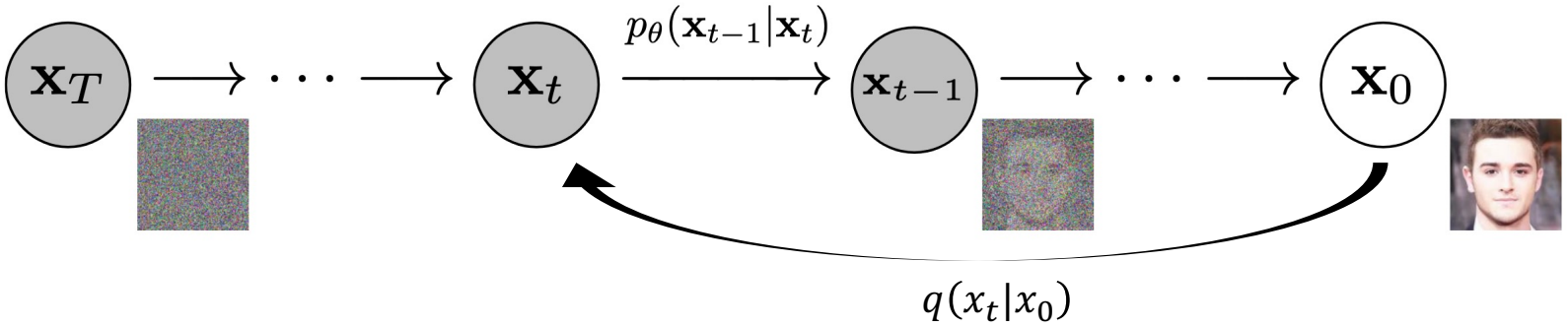
Methods

- **Forward process and $\mathbf{L_T}$**
  - Since β follows a fixed noise schedule, the L_T term is constant and ignored during training
  - $q(x_{1:T}|x_0) \coloneqq \prod_{t=1}^{T} q(x_t|x_{t-1}), \qquad q(x_t|x_{t-1}) \coloneqq \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathrm{I}\right)$
  - Can be computed in closed form at an arbitrary timestep $q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathrm{I}\right)$
  - $x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon_{t-1}$ $\qquad \alpha_t = 1-\beta_t, \alpha_t = \prod_{i=1}^{t}\alpha_i$

$$= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\epsilon}_{t-2}$$

$$= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

  - $q(x_t|x_0) \coloneqq \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathrm{I}\right)$



Methods

- **Reverse process and $\mathbf{L_{1:T-1}}$**
  - The reverse process is trained to accurately predict the posterior x_{t-1} from x_t
  - $q(x_{t-1}|x_t, x_0) \coloneqq \mathcal{N}\left(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \mathrm{I}\right)$
    - where $\tilde{\mu}_t(x_t, x_0) \coloneqq \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t} x_0 + \frac{\sqrt{\alpha_t}(1-\alpha_{t-1})}{1-\alpha_t} x_t$ and $\tilde{\beta}_t \coloneqq \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t$
  - $p_\theta(x_{t-1}|x_t) \coloneqq \mathcal{N}\left(x_{t-1};\ \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$ for $1 < t \leq T$
    - $\Sigma_\theta(x_t, t) = \sigma_t^2 \mathrm{I},\ \ \sigma_t^2 = \beta_t \ or\ \tilde{\beta}_t$
  - Parameterization is restructured to accurately predict the $\tilde{\mu}_t$
  - Reformulated as a noise prediction problem: ε-prediction

$$\mathbb{E}_q \left[D_{KL}\left(q(x_T|x_0) \parallel p(x_T)\right) + \sum_{t>1} D_{KL}\left(q(x_{t-1}|x_t, x_0) \parallel p_\theta(x_{t-1}|x_t)\right) - \log p_\theta(x_0|x_1) \right]$$

L_T $\qquad$ $L_{1:T-1}$ $\qquad$ L_0

Methods

- **Reverse process and $\mathbf{L_{1:T-1}}$**
  - The reverse process is trained to accurately predict the posterior x_{t-1} from x_t
  - $q(x_{t-1}|x_t, x_0) \coloneqq \mathcal{N}\left(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \mathrm{I}\right)$
    - where $\tilde{\mu}_t(x_t, x_0) \coloneqq \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t} x_0 + \frac{\sqrt{\alpha_t}(1-\alpha_{t-1})}{1-\alpha_t} x_t$ and $\tilde{\beta}_t \coloneqq \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t$

$$q(x_t|x_{t-1}) \coloneqq \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathrm{I}\right)$$
$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathrm{I}\right)$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{x}_0)}{q(\mathbf{x}_t, \mathbf{x}_0)} \frac{q(\mathbf{x}_{t-1}, \mathbf{x}_0)}{q(\mathbf{x}_{t-1}, \mathbf{x}_0)} = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1}, \mathbf{x}_0)}{q(\mathbf{x}_t, \mathbf{x}_0)} \frac{q(\mathbf{x}_0)}{q(\mathbf{x}_0)} = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$$

$$\propto \exp\left(-\frac{1}{2}\left(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$= \exp\left(-\frac{1}{2}\left(\frac{\mathbf{x}_t^2 - 2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1} + \alpha_t\mathbf{x}_{t-1}^2}{\beta_t} + \frac{\mathbf{x}_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\mathbf{x}_{t-1} + \bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}_{t-1}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)\mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0)\right)\right)$$

Weng, Lilian. (Jul 2021). What are diffusion models? Lil'Log. https://lilianweng.github.io/posts/2021-07-11-diffusion-models/.

Methods

- **Reverse process and $\mathbf{L_{1:T-1}}$**
  - The reverse process is trained to accurately predict the posterior x_{t-1} from x_t
  - $q(x_{t-1}|x_t, x_0) \coloneqq \mathcal{N}\left(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t \mathrm{I}\right)$
    - where $\tilde{\mu}_t(x_t, x_0) \coloneqq \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\alpha_t} x_0 + \frac{\sqrt{\alpha_t}(1-\alpha_{t-1})}{1-\alpha_t} x_t$ and $\tilde{\beta}_t \coloneqq \frac{1-\alpha_{t-1}}{1-\alpha_t}\beta_t$

$$q(x_t|x_{t-1}) \coloneqq \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathrm{I}\right)$$
$$q(x_t|x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathrm{I}\right)$$

$$
\begin{aligned}
q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= \frac{q(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{x}_0)}{q(\mathbf{x}_t, \mathbf{x}_0)} \frac{q(\mathbf{x}_{t-1}, \mathbf{x}_0)}{q(\mathbf{x}_{t-1}, \mathbf{x}_0)} = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1}, \mathbf{x}_0)}{q(\mathbf{x}_t, \mathbf{x}_0)} \frac{q(\mathbf{x}_0)}{q(\mathbf{x}_0)} = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} \\
&\propto \exp\Big(-\frac{1}{2}\big(\frac{(\mathbf{x}_t - \sqrt{\alpha_t}\mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0)^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big(\frac{\mathbf{x}_t^2 - 2\sqrt{\alpha_t}\mathbf{x}_t\mathbf{x}_{t-1} + \alpha_t\mathbf{x}_{t-1}^2}{\beta_t} + \frac{\mathbf{x}_{t-1}^2 - 2\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\mathbf{x}_{t-1} + \bar{\alpha}_{t-1}\mathbf{x}_0^2}{1-\bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0)^2}{1-\bar{\alpha}_t}\big)\Big) \\
&= \exp\Big(-\frac{1}{2}\big((\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}})\mathbf{x}_{t-1}^2 - (\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0)\big)\Big) \\
\tilde{\beta}_t &= 1/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) = 1/(\frac{\alpha_t - \bar{\alpha}_t + \beta_t}{\beta_t(1-\bar{\alpha}_{t-1})}) = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t \\
\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) &= (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)/(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}) \\
&= (\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0)\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t \\
&= \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0
\end{aligned}
$$

Weng, Lilian. (Jul 2021). What are diffusion models? Lil'Log. https://lilianweng.github.io/posts/2021-07-11-diffusion-models/.

Methods

- **Reverse process and $\mathbf{L_{1:T-1}}$**
  - Parameterization is restructured to accurately predict the $\tilde{\mu}_t$
    - Only the μ_θ-dependent term is extracted from the KL divergence between Gaussians
  - Reformulated as a noise prediction problem: ε-prediction
    - Expressing x_{t-1} in this form yields a process similar to Langevin dynamics
    - resembles denoising score matching over multiple noise scales indexed by t

$$\mathrm{KL}(q\|p) = \frac{1}{2}\left[\log\frac{|\Sigma_p|}{|\Sigma_q|} - d + \mathrm{tr}(\Sigma_p^{-1}\Sigma_q) + (\mu_p - \mu_q)^\mathsf{T}\Sigma_p^{-1}(\mu_p - \mu_q)\right]$$

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

Methods

- **Reverse process and $\mathbf{L_{1:T-1}}$**
  - Parameterization is restructured to accurately predict the $\tilde{\mu}_t$
    - Only the μ_θ-dependent term is extracted from the KL divergence between Gaussians
  - Reformulated as a noise prediction problem: ε-prediction
    - Expressing x_{t-1} in this form yields a process similar to Langevin dynamics
    - resembles denoising score matching over multiple noise scales indexed by t

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

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

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

Methods

- **Reverse process and $\mathbf{L_{1:T-1}}$**
  - Parameterization is restructured to accurately predict the $\tilde{\mu}_t$
    - Only the μ_θ-dependent term is extracted from the KL divergence between Gaussians
  - Reformulated as a noise prediction problem: ε-prediction
    - Expressing x_{t-1} in this form yields a process similar to Langevin dynamics
    - resembles denoising score matching over multiple noise scales indexed by t
  - Performs better when trained without weighting on timestep
    - Original weighting puts large weight on small t (less noise), focusing on easy denoising

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

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

Methods

- **Data scaling, reverse process decoder, and $\mathbf{L_0}$**
  - Original image data (0–255) is linearly scaled to [−1, 1]
    - Matches standard normal prior and stabilizes training
  - Final step $p_\theta(x_0|x_1)$ modeled using a discrete decoder
    - Each dimension decoded from Gaussian mean $\mu_\theta(x_1, 1)$
  - L_0 term $\log p_\theta(x_0|x_1)$
    - Measures final reconstruction accuracy

Evaluation Metrics

- **Inception Score (IS)**
  - Measures sharpness and diversity of generated samples
  - Computed using KL divergence between p(y|x) and p(y) from Inception model
  - Higher is better
- **Fréchet Inception Distance (FID)**
  - Measures similarity between real and generated data distributions
  - Uses Gaussian statistics of Inception feature embeddings
  - Lower is better
- **Negative Log-Likelihood (NLL, bits/dim)**
  - Measures how well the model explains real data
  - Interpreted as lossless codelength in bits per dimension
  - Lower is better

Experimental Results

- **DDPM achieves better FID than most other conditional models**
- **Training objective trade-off**
  - Variational bound → better NLL
  - Simplified objective (ε-prediction) → better FID
- **A large portion of NLL arises from the distortion term $\mathbf{L_0}$**
  - Half of the total codelength is spent on imperceptible details (RMSE ≈ 0.95 on [0, 255] scale)
- **Diffusion models have an inductive bias for perceptually faithful reconstructions**
  - Act as effective **lossy compressors**

Rate $\mathbf{L_{1:T}}$: amount of information needed to remove noise

Distortion $\mathbf{L_0}$: reconstruction error of the final output

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)

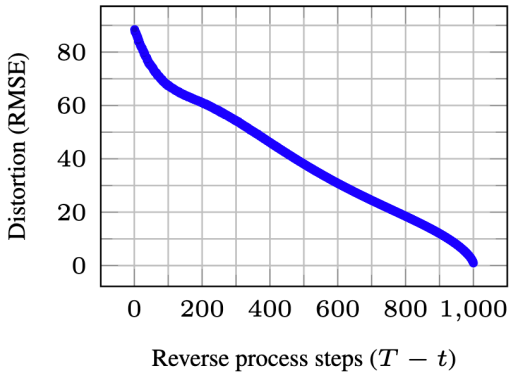
Experimental Results

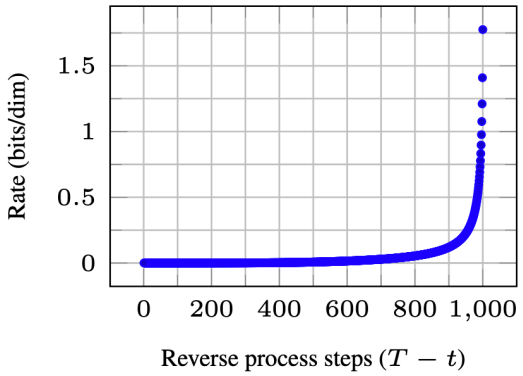
Table 2: Unconditional CIFAR10 reverse process parameterization and training objective ablation. Blank entries were unstable to train and generated poor samples with out-of-range scores.

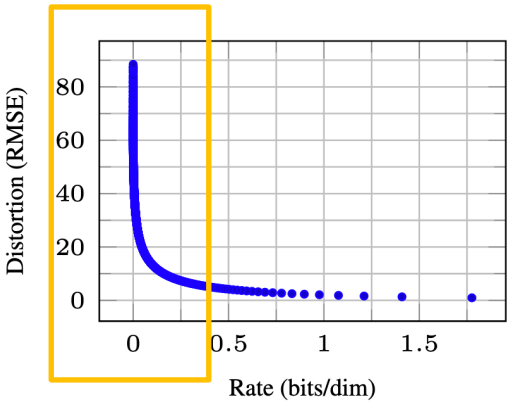
Objective	IS	FID
$\tilde{\mu}$ prediction (baseline)		
L, learned diagonal Σ	7.28 ± 0.10	23.69
L, fixed isotropic Σ	8.06 ± 0.09	13.22
$\|\|\tilde{\mu} - \tilde{\mu}_\theta\|\|^2$	–	–
ϵ prediction (ours)		
L, learned diagonal Σ	–	–
L, fixed isotropic Σ	7.67 ± 0.13	13.51
$\|\|\tilde{\epsilon} - \epsilon_\theta\|\|^2$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

- **Progressive lossy compression**
  - Predicting $\widetilde{\mu_t}$: works well only with full variational bound
  - Predicting ε: works similarly with variational bound, but performs better with simplified objective
  - Fixed variance yields more stable and better results

Experimental Results





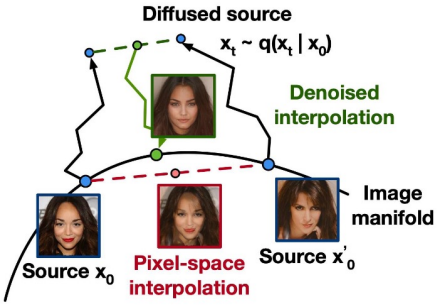


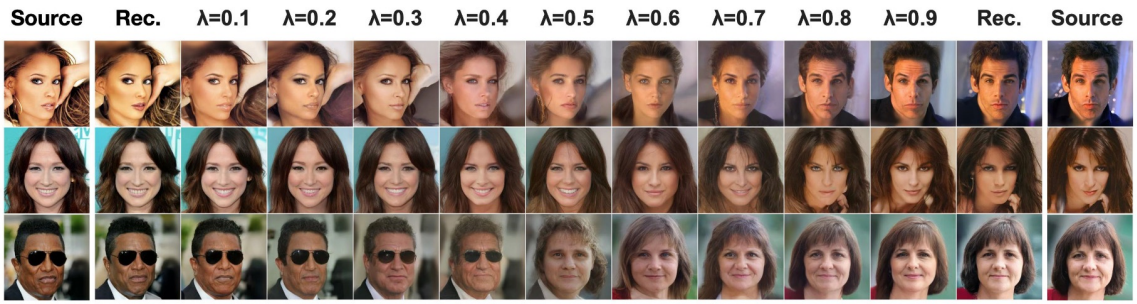
- **Progressive lossy compression**
  - Distortion: RMSE between predicted and ground-truth x_0
  - Rate: Cumulative number of bits up to time t
  - Steep drop in distortion at low rates and diminishing returns as rate increases further
  - Diffusion models act as **perceptually efficient lossy compressors**

Experimental Results

- **Progressive generation (visual interpretation)**
  - Early steps recover coarse elements → shape, pose, background
  - Later steps refine edges, textures, and high-frequency features

Experimental Results





- **Interpolation**
  - Encode two images x_0, x_0' into noisy latents x_t, x_t'
  - Interpolate in latent space and decode interpolated latent $\bar{x}_t$ to $\bar{x}_0$
  - High ttt can yield novel samples not seen in training

Conclusions

- **Introduces ε-parameterization for simplified and stable training**
  - Enables connection to denoising score matching
- **Demonstrates that diffusion models can rival GANs in image quality**
  - Opens a new direction for likelihood-based generative models

Appendix

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

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

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

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

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