Understanding Diffusion Models in Two Perspectives

Junoh Kang

Computer Vision Laboratory
ECE, Seoul National University
junoh.kang@snu.ac.kr

Contents

1. Denoising Diffusion Probabilistic Models [3]
2. Score-Based Generative Modeling through Stochastic Differential Equations [9]
3. Strong and Weak Points of Diffusion Models.

Denoising Diffusion Probabilistic Models [3]: Minimizing Negative Log-Likelihood

DDPM

Overview

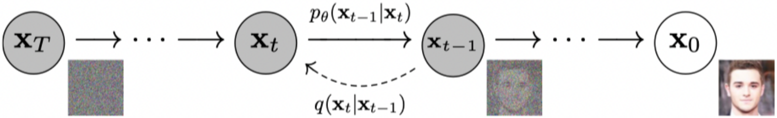


Figure 2: The directed graphical model considered in this work.

DDPM

Forward Process

- Forward Process is a Markov chain that gradually perturbs images to Gaussian distribution.

$$\mathbf{x}_t \perp\!\!\!\perp \mathbf{x}_{0:t-1}, \tag{1}$$

$$q(\mathbf{x}_0) := \mathrm{P}_{data}(\mathbf{x}_0), \tag{2}$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathrm{I}). \tag{3}$$

DDPM

Backward Process

- Backward Process is also a Markov chain that gradually denoises noises from perturbs images.

$$\mathbf{x}_t \perp\!\!\!\perp \mathbf{x}_{T:t+1}, \tag{4}$$

$$p_\theta(\mathbf{x}_T) := \mathcal{N}(\mathbf{x}_T; 0, \mathrm{I}), \tag{5}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = ???.$$

Lemma 1

When β_t is small for $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathrm{I})$, its reverse conditional distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is also a Gaussian:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx \mathcal{N}(\mathbf{x}_{t-1}; \frac{1}{\sqrt{1-\beta_t}}(\mathbf{x}_t + \beta_t \nabla \log q(\mathbf{x}_t)), \beta_t \mathrm{I}). \tag{6}$$

- It is reasonable to parametrize $\nabla \log q(\mathbf{x}_t)$ by neural network.

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \frac{1}{\sqrt{1-\beta_t}}(\mathbf{x}_t + \beta_t s_\theta(\mathbf{x}_t, t)), \beta_t \mathrm{I}). \tag{7}$$

DDPM

Objective

- The objective of DDPM is to minimize negative log-likelihood.

$$\mathbb{E}_{\mathbf{x}_0 \sim q} \left[-\log p_\theta(\mathbf{x}_0) \right]. \tag{8}$$

DDPM

Objective

$$
\begin{aligned}
\mathbb{E}_{\mathbf{x}_0 \sim q}\left[-\log p_\theta(\mathbf{x}_0)\right] &= \mathbb{E}_{\mathbf{x}_0 \sim q}\left[-\log \int p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}\right] && (9) \\
&= \mathbb{E}_{\mathbf{x}_0 \sim q}\left[-\log \int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] && (10) \\
&\leq \mathbb{E}_{\mathbf{x}_0 \sim q}\left[-\int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \log \frac{p_\theta(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] && (11) \\
&= \mathbb{E}_{\mathbf{x}_0 \sim q}\left[\mathbb{E}_{\mathbf{x}_{1:T|0} \sim q}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right]\right] && (12) \\
&= \mathbb{E}_{\mathbf{x}_{0:T} \sim q}\left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right]. && (13)
\end{aligned}
$$

DDPM

Objective

- Forward Process:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{14}$$

$$= q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) \tag{15}$$

$$= q(\mathbf{x}_1|\mathbf{x}_0) \prod_{t=2}^{T} \frac{q(\mathbf{x}_t|\mathbf{x}_0)q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} \tag{16}$$

$$= q(\mathbf{x}_T|\mathbf{x}_0) \prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0). \tag{17}$$

- Backward Process:

$$p_\theta(\mathbf{x}_{T:0}) = p_\theta(\mathbf{x}_T) \prod_{t=T}^{1} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t). \tag{18}$$

DDPM

Objective

- The surrogate of negative log-likelihood is

$$\mathbb{E}_{\mathbf{x}_{0:T}\sim q}\left[\log\frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})}\right] \tag{19}$$

$$= \mathbb{E}_{\mathbf{x}_{0:T}\sim q}\left[\frac{q(\mathbf{x}_T|\mathbf{x}_0)\prod_{t=2}^{T} q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)\prod_{t=T}^{1} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}\right] \tag{20}$$

$$= D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)||p_\theta(\mathbf{x}_T)) + \mathbb{E}_q\left[-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right] + \sum_{t=2}^{T} D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)). \tag{21}$$

DDPM

Objective

- $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ and $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ are Gaussian distributions.

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \frac{1}{\sqrt{1-\beta_t}}(\mathbf{x}_t + \beta_t \nabla \log q(\mathbf{x}_t|\mathbf{x}_0)), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \mathrm{I}), \tag{22}$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \frac{1}{\sqrt{1-\beta_t}}(\mathbf{x}_t + \beta_t s_\theta(\mathbf{x}_t, t)), \beta_t \mathrm{I}). \tag{23}$$

- Therefore, the surrogate of negative log-likelihood becomes

$$\sum_{t=2}^{T} \lambda_t ||s_\theta(\mathbf{x}_t, t) - \nabla \log q(\mathbf{x}_t|\mathbf{x}_0)||_2^2 + C. \tag{24}$$

DDPM

Objective

- For $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 - \sqrt{1-\bar{\alpha}_t}\epsilon$ for $\epsilon \sim \mathcal{N}(0, \mathrm{I})$,

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathrm{I}) \tag{25}$$

$$= (2\pi(1-\bar{\alpha}_t))^{-d/2}\exp(-\frac{1}{2(1-\bar{\alpha}_t)}||\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0||^2). \tag{26}$$

$$\therefore \nabla \log q(\mathbf{x}_t|\mathbf{x}_0) = -\frac{1}{1-\bar{\alpha}_t}(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0) = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon. \tag{27}$$

- For $s_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)$, the objective (24) becomes

$$\sum_{t=2}^{T}\mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{\lambda_t}{1-\bar{\alpha}_t}||\epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 - \sqrt{1-\bar{\alpha}_t}\epsilon, t) - \epsilon||_2^2\right] \tag{28}$$

$$= (T-1)\mathbb{E}_{\mathbf{x}_0,t,\epsilon}\left[\frac{\lambda_t}{1-\bar{\alpha}_t}||\epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 - \sqrt{1-\bar{\alpha}_t}\epsilon, t) - \epsilon||_2^2\right]. \tag{29}$$

Sampling algorithm:

$$\mathbf{x}_T \sim \mathcal{N}(0, \mathrm{I}), \tag{30}$$

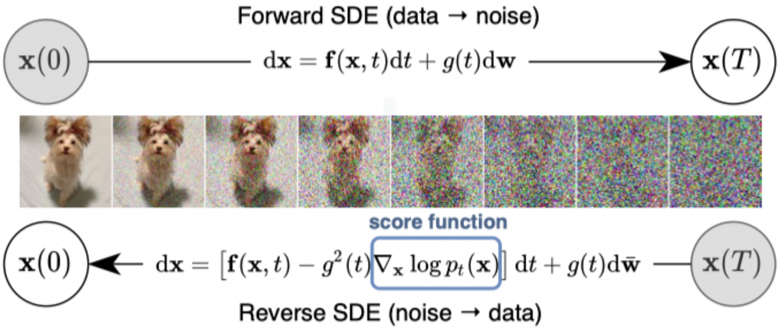
$$\mathbf{x}_{t-1}|\mathbf{x}_t \sim \mathcal{N}(\frac{1}{\sqrt{1-\beta_t}}(\mathbf{x}_t + \beta_t s_\theta(\mathbf{x}_t, t)), \beta_t \mathrm{I}). \tag{31}$$

- The assumption, small β_t, in Lemma 1 is required to properly model the backward distribution. This leads to slow sampling speed.

Score-Based Generative Modeling through Stochastic Differential Equations [9]: Matching Marginal Distributions

Score-Based Generative Models

Overview



Score-Based Generative Models

Forward SDE

- Forward SDE diffuses data distribution to Gaussian distribution

$$d\mathbf{x}_t = f(t)\mathbf{x}_t dt + g(t)d\mathbf{w}_t, \mathbf{x}_0 \sim P_{data}, \quad (32)$$

  where $\mathbf{w}_t$ is Brownian process.

- A solution of (32), $\{\mathbf{x}_t\}_{t=0}^{T}$, can be treated as a sample from the joint distribution $\{p_t\}_{t=0}^{T}$.
- Learning joint distribution is difficult and the region of interest is $\mathbf{x}_0 \sim p_0$. Therefore, authors detour to learn marginal distribution.

Score-Based Generative Models

Backward SDE/ODE

- Backward SDE/ODE matches marginal distribution of forward SDE. (This can be proven by Fokker-Plank equation)

$$d\mathbf{x}_t = \left[f(t)\mathbf{x}_t dt - g^2(t)\nabla \log p_t(\mathbf{x}_t)\right] dt + g(t)d\bar{\mathbf{w}}_t, \quad \mathbf{x}_T \sim \mathcal{N}(0, \mathrm{I}), \tag{33}$$

$$d\mathbf{x}_t = \left[f(t)\mathbf{x}_t dt - \frac{1}{2}g^2(t)\nabla \log p_t(\mathbf{x}_t)\right] dt, \quad \mathbf{x}_T \sim \mathcal{N}(0, \mathrm{I}), \tag{34}$$

  where $\bar{\mathbf{w}}_t$ is reverse-time Brownian process.
- Note that the sampling at the boundary ($t = T$) is simple.
- Since $f(\cdot)$ and $g(\cdot)$ are given, the only unknown component of (33) and (34) is $\nabla \log p_t(\mathbf{x}_t)$, which is known as a score function.
- It is reasonable to paramerize a score function with a neural network, $s_\theta(\mathbf{x}_t, t)$.

Score-Based Generative Models

Objective

- The objective of score-based generative models is to learn score function:

$$\int_0^T \lambda_t \mathbb{E}_{\mathbf{x}_t} \left[||s_\theta(\mathbf{x}_t, t) - \nabla \log p_t(\mathbf{x}_t)||_2^2 \right] dt. \tag{35}$$

- Impossible to train since $\nabla \log p_t(\mathbf{x}_t)$ in (35) is intractable!
- With equivalent equation, training the network is feasible.

$$\int_0^T \lambda_t \mathbb{E}_{\mathbf{x}_0} \left[\mathbb{E}_{\mathbf{x}_t|\mathbf{x}_0} \left[||s_\theta(\mathbf{x}_t, t) - \nabla \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)||_2^2 \right] \right] dt + C. \tag{36}$$

Score-Based Generative Models

Objective

- Variance-Exploding (VE) SDE

$$d\mathbf{x}_t = \sigma d\mathbf{w}_t, \tag{37}$$

$$\mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0, t\sigma^2) \tag{38}$$

- Variance-Preserving (VP) SDE

$$d\mathbf{x}_t = -\beta\mathbf{x}_t dt + \sigma d\mathbf{w}_t \tag{39}$$

$$\mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}(e^{-\beta t}\mathbf{x}_0, \frac{1-e^{-2\beta t}}{2\beta}\sigma^2) \tag{40}$$

Score-Based Generative Models

Objective

- For both cases, $\mathbf{x}_t = \gamma_t \mathbf{x}_0 - \sigma_t \epsilon$ for $\epsilon \sim \mathcal{N}(0, \mathrm{I})$.
- As in DDPM, $\nabla \log p_{t|0}(\mathbf{x}_t|\mathbf{x}_0) = \frac{\epsilon}{\sigma_t}$.
- The objective (36) becomes

$$\int_0^T \frac{\lambda_t}{\sigma_t^2} \mathbb{E}_{\mathbf{x}_0} \left[\mathbb{E}_{\epsilon} \left[||\epsilon_\theta(\gamma_t \mathbf{x}_0 - \sigma_t \epsilon, t) - \epsilon||_2^2 \right] \right] dt \tag{41}$$

$$= T \mathbb{E}_{\mathbf{x}_0, t, \epsilon} \left[\frac{\lambda_t}{\sigma_t^2} ||\epsilon_\theta(\gamma_t \mathbf{x}_0 - \sigma_t \epsilon, t) - \epsilon||_2^2 \right] dt. \tag{42}$$

In the perspective of solving ODE by Euler method,

$$d\mathbf{x}_t = f_\theta(\mathbf{x}_t)dt, \mathbf{x}_T = \mathbf{x}_T \tag{43}$$

$$\mathbf{x}_0 = \mathbf{x}_T + \int_T^0 f_\theta(\mathbf{x}_t)dt \tag{44}$$

$$= \mathbf{x}_T + \sum_{i=N}^{1} \int_{t_i}^{t_{i-1}} f_\theta(\mathbf{x}_t)dt \tag{45}$$

$$= \mathbf{x}_T + \sum_{i=N}^{1} (t_{i-1} - t_i) f_\theta(\mathbf{x}_{t_i}) + O(|t_{i-1} - t_i|^2) \tag{46}$$

- Requirement of discretizations for precise approximation on integral causes slow sampling speed.

Summary

- The objective of DDPM is to minimize the surrogate of the negative log-likelihood.
- The objective of score-based generative models is to match marginal distribution of forward SDE and backward SDE/ODE.
- The slow speed of DDPM is due to assumption, $\beta_t << 1$ in Lemma 1.
- The slow speed of score-based generative models originates from the discretizations which minimize errors in integral.
- Even two works have different motivations, but their objectives are the same: learn score function by a neural network.

Components to Implement Diffusion Models

- Training
  - Choice of forward SDE: VP SDE, VE SDE,*etc.*.
  - What should model predict? Denoiser $\mathbb{E}[\mathbf{x}_0|\mathbf{x}_t]$, or noise ϵ.
  - Choice of weights, λ_t.
- Sampling
  - Choice of SDE/ODE solvers: Euler, Heun's, Runge-Kutta, *etc.*.
  - Discretization methods: practically small $|t_{i-1} - t_i|$ for small i (when data is near image manifold) yields better quality of samples.

Strong and Weak Points of Diffusion Models.

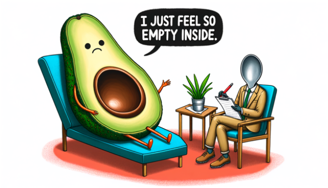
Diffusion Models vs GANs [2]

Table 1: Comparisons between diffusion models and GANs.

	Diffusion Models	GANs
Objective	explicit	implicit
Optimization	minimization	minimax
Sampling speed	NFE $>>$ 1	NFE=1
Mode coverage	high	low

Strong Points of Diffusion Models

Training stability



An illustration of an avocado sitting in a therapist's chair, saying 'I just feel so empty inside' with a pit-sized hole in its center. The therapist, a spoon, scribbles notes.

A 2D animation of a folk music band composed of anthropomorphic autumn leaves, each playing traditional bluegrass instruments, amidst a rustic forest setting dappled with the soft light of a harvest moon.

Figure 1: Image generated by DALL-E 3 [*]. Training stability of diffusion models enables training on a large scale dataset.

[*] https://openai.com/dall-e-3

Strong Points of Diffusion Models

Controllable generation

- Suppose we only have unconditional score function, $\nabla \log p_t(\mathbf{x}_t)$.
- Still we can generate conditional sample $\mathbf{x}_0|y$.
- To generate $\mathbf{x}_0|y$, we have to solve backward ODE as following:

$$d\mathbf{x}_t = \left[f(t)\mathbf{x}_t dt - \frac{1}{2} g^2(t) \nabla \log p_t(\mathbf{x}_t|\mathbf{y}) \right] dt, \quad \mathbf{x}_T \sim \mathcal{N}(0, \mathrm{I}) \tag{47}$$

- The conditional score function can be calculated by

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log \frac{p_t(\mathbf{x}_t) p_t(\mathbf{y}|\mathbf{x}_t)}{p_t(\mathbf{y})} \tag{48}$$

$$= \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)}_{\text{unconditional score fucntion}} + \underbrace{\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t)}_{\text{external information}} \tag{49}$$

Strong Points of Diffusion Models

Bi-directional ODE solving

- Generating samples by ODE makes the sampling path deterministic. Moreover, solving in (image $\rightarrow$ latent) direction is also feasible. These properties are useful for many tasks.
- *e.g.*, for the I2I task, many calculate the latent of the source image and give it as a boundary condition of target sampling ODE. Moreover, cycle consistency is guaranteed theoretically.

Weak Points of Diffusion Models

Slow sampling speed

- When solving ODE, small $|t_{i-1} - t_i|$ is required to calculate following integral precisely, which leads to slow generation.

$$\int_{t_i}^{t_{i-1}} f_\theta(\mathbf{x}_t)dt \tag{50}$$

- To accelerate generation, accurate integral for large $|t_{i-1} - t_i|$ is required.
  1. Advanced inference algorithms

$$\int_{t_i}^{t_{i-1}} f_\theta(\mathbf{x}_t)dt \approx (t_{i-1} - t_i)h(f_\theta(\mathbf{x}_t)) \tag{51}$$

     - *e.g.*, Euler [7], Heun's method [4], PNDM [5], GENIE [1]
  2. Distillation algorithms

$$\int_{t_i}^{t_{i-1}} f_\theta(\mathbf{x}_t)dt \approx h_\phi(\mathbf{x}_{t_i}) \tag{52}$$

     - *e.g.*, Progressive disillation [6], Consistency models [8]

Reference I

Tim Dockhorn, Arash Vahdat, and Karsten Kreis.
GENIE: Higher-order denoising diffusion solvers.
In *NeurIPS*, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio.
Generative adversarial nets.
In *NeurIPS*, 2014.

Jonathan Ho, Ajay Jain, and Pieter Abbeel.
Denoising diffusion probabilistic models.
In *NeurIPS*, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine.
Elucidating the design space of diffusion-based generative models.
NeurIPS, 2022.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao.
Pseudo numerical methods for diffusion models on manifolds.
In *ICLR*, 2022.

Tim Salimans and Jonathan Ho.
Progressive distillation for fast sampling of diffusion models.
In *ICLR*, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon.
Denoising diffusion implicit models.
In *ICLR*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever.
Consistency models.
In *ICML*, 2023.

Reference II

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole.
Score-based generative modeling through stochastic differential equations.
In *ICLR*, 2021.