Generative Modeling with Diffusion Models

Juho Lee

Kim Jaechul Graduate School of AI, KAIST

Foundation and Future of Generative Models: Mathematics, Algorithms, and Applications @ CityU

Table of Contents

Generative models

Generative modeling, density estimation, . . . ,

$$x_1, \ldots, x_n \overset{\text{i.i.d.}}{\sim} p_{\text{data}}, \quad p_\theta \approx p_{\text{data}}? \tag{1}$$

Maximum likelihood training:

$$\begin{aligned} \theta^\star &= \arg\min_\theta D_{\mathrm{KL}}[p_{\text{data}} \| p_\theta], \\ \nabla_\theta D_{\mathrm{KL}}[p_{\text{data}} \| p_\theta] &= -\mathbb{E}_{p_{\text{data}}}[\nabla_\theta \log p_\theta(x)]. \end{aligned} \tag{2}$$

Generative models

How to build a model distribution $p_\theta(x)$?

- Variational AutoEncoder (VAE) (Kingma et al., 2014).
- Autoregressive models (Uria et al., 2016).
- Flow-based models (Rezende et al., 2015).
- Implicit models (Generative Adversarial Network (GAN) (Goodfellow et al., 2014)).

Variational autoencoders

The Evidence Lower BOund (ELBO) of VAE:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{\mathrm{KL}}[q_\phi(z|x)\|p_\theta(z)]. \tag{3}$$

- An encoder $q_\phi(z|x)$ compresses a data point x into a code z.
- A decoder $p_\theta(x|z)$ reconstructs x from z.
- The code z should be simple in a sense that $q_\phi(z|x)$ is similar to $p(z)$.

Autoregressive models

Let $x := [x_1, \ldots, x_d]^\top$ and $x_{1:j} := [x_1, \ldots, x_j]^\top$.

$$
\begin{aligned}
p_\theta(x) &= \prod_{j=1}^{d} p_\theta(x_j \,|\, x_{1:j-1}), \\
p_\theta(x_j \,|\, x_{1:j-1}) &= \mathcal{N}\left(x_j \,|\, f_\mu(x_{1:j-1}), f_\sigma^2(x_{1:j-1})\right).
\end{aligned}
\tag{4}
$$

- Exact likelihood computation.
- Slow generation and order dependency. Rarely used for images.

Flow-based models

Let $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$ be a bijection with an easily computable Jacobian.

$$
\begin{aligned}
z &\sim p_0(z), \quad x = f_\theta(z), \\
p_\theta(x) &= p_0(f_\theta^{-1}(x)) \left| \det \frac{\mathrm{d} f_\theta^{-1}(x)}{\mathrm{d}x} \right|
\end{aligned}
\tag{5}
$$

- Exact likelihood computation.
- Fast generation.
- Limited flexibility due to the bijection constraint on f_θ.

Implicit models

$f_\theta : \mathbb{R}^h \to \mathbb{R}^d$ is an unconstrained function.

$$z \sim p_0(z), \quad x = f_\theta(z). \tag{6}$$

- Cannot compute the likelihood $p_\theta(x)$.
- Requires adversarial training, which can be unstable.
- Hard to get the code z corresponding to a given x.

A better generative model?

- Minimal constraints on f_θ.
- No adversarial training.
- Moderately fast generation.
- Handy manipulation of the code x.

Table of Contents

Challenges in VAE

- The tension between the encoder and decoder:
  - The encoder $q_\phi(z|x)$ compresses x into a noise-like code z.
  - The decoder $p_\theta(x|z)$ reconstructs x from z.
- It is generally hard to learn powerful mappings that achieve both good compression and sample fidelity.

An idea: solve easier problems

- For the encoder, an easy way is to simply corrupt the input with noise.

$$z = \alpha x + \sigma\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I) \Rightarrow q(z|x) = \mathcal{N}(z|\alpha x, \sigma^2 I). \tag{7}$$

- In terms of code simplicity, we want to choose $\alpha = 0$ and $\sigma = 1$, but in that case, the decoder cannot reconstruct x from z.
- The main idea behind diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) is to break down this problem of corruption and reconstruction as a sequence of easier problems.

An idea: solve easier problems

Let $(\alpha_k, \sigma_k)_{k=1}^K$ be a sequence such that the log Signal-to-Noise Ratio (SNR) $\rho_k := \log(\alpha_k^2/\sigma_k^2)$ is non-increasing. Starting from $x_0 := x$, we decompose the corruption into smaller corruptions:

$$\begin{aligned} x_1 &= \alpha_1 x_0 + \sigma_1 \varepsilon_1, \\ x_2 &= \frac{\alpha_2}{\alpha_1} x_1 + \sqrt{\alpha_2^2 (e^{-\rho_2} - e^{-\rho_1})}\varepsilon_2, \\ &\vdots \\ x_K &= \frac{\alpha_K}{\alpha_{K-1}} x_{K-1} + \sqrt{\alpha_K^2 (e^{-\rho_K} - e^{-\rho_{K-1}})}\varepsilon_K, \end{aligned} \tag{8}$$

leading to $x_k \sim \mathcal{N}(\alpha_k x_0, \sigma_k^2 I)$ for $k = 1, \ldots, K$.

An idea: solve easier problems

The corruption sequence forms a forward process with a joint density,

$$q(x_{1:K}|x_0) = \prod_{k=1}^{K} q(x_k|x_{k-1}),$$
$$q(x_k|x_{k-1}) = \mathcal{N}(x_k|(\alpha_k/\alpha_{k-1})x_{k-1}, \alpha_k^2(e^{-\rho_k} - e^{-\rho_{k-1}})I). \tag{9}$$

To denoise (decode) the corrupted input, we build a sequence of denoisers,

$$p_\theta(x_{K-1}|x_K), p_\theta(x_{K-2}|x_{K-1}), \ldots, p_\theta(x_1|x_2), p_\theta(x|x_1), \tag{10}$$

defining a reverse process with a joint density,

$$p_\theta(x_{0:K}) = p_{\text{noise}}(x_K) \prod_{k=1}^{K} p_\theta(x_{k-1}|x_k), \tag{11}$$

where $p_{\text{noise}}(x_K)$ is a prior for the terminal noise.

Denoising diffusion probabilistic models

Denoising Diffusion Probabilistic Models (DDPM) (Ho et al., 2020), in a nutshell, is a VAE whose code z corresponds to the sequence of corrupted inputs $z = x_{1:K}$. It can thus be trained by maximizing the ELBO,

$$\log p_\theta(x) \geq \mathbb{E}_{q(x_{1:K}\,|\,x)}\left[\log \frac{p_\theta(x, x_{1:K})}{q(x_{1:K}\,|\,x)}\right] := -\mathcal{L}(\theta, x). \tag{12}$$

The negative ELBO can further be decomposed as,

$$\mathcal{L}(x, \theta) = \mathbb{E}_q\left[\log \frac{q(x_K\,|\,x)}{p_{\text{noise}}(x_K)} + \sum_{k=1}^{K} \log \frac{q(x_{k-1}\,|\,x_k, x)}{p_\theta(x_{k-1}\,|\,x_k)}\right]. \tag{13}$$

Denoising diffusion probabilistic models

The roles of the terms in the objective:

$$\mathcal{L}(x, \theta) = \mathbb{E}_q \left[\log \frac{q(x_K \,|\, x)}{p_{\text{noise}}(x_K)} + \sum_{k=1}^{K} \log \frac{q(x_{k-1} \,|\, x_k, x)}{p_\theta(x_{k-1} \,|\, x_k)} \right]. \tag{14}$$

- The first term measures the discrepancy between the prior noise distribution $p_{\text{noise}}(x_K)$ and the terminal noise distribution $q(x_K \,|\, x)$.
- The second term measures the divergence between the oracle decoder $q(x_{k-1} \,|\, x_k, x)$ and the intermediate decoder $p_\theta(x_{k-1} \,|\, x_k)$.

Table of Contents

What if $K \to \infty$?

- The key idea was to decompose $x \to z$ into small-sized steps.
- Can we take more steps with smaller sizes?
- $K \to \infty$ with infinitesimal step sizes?

A crash course on SDE

A Stochastic Differential Equation (SDE) on $\mathbb{R}^d$ has the following form,

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + g(t)\mathrm{d}W_t, \quad t \in [0, T]. \tag{15}$$

- $f : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ is called a drift coefficient.
- $g : [0, T] \to \mathbb{R}$ is called a diffusion coefficient.
- W_t is called a Wiener process, satisfying
  - W_t has independent increments.
  - $W_t - W_s \sim \mathcal{N}(0, t - s)$ for all $0 \leq s \leq t \leq T$.

A crash course on SDE

Just like the Ordinary Differential Equation (ODE), we can do something like,

$$X_t = X_0 + \underbrace{\int_0^t f(X_s, s)\mathrm{d}s}_{:=A} + \underbrace{\int_0^t g(s)\mathrm{d}W_s}_{:=B}. \tag{16}$$

- You can treat A as an ordinary integral, but technically it is a random variable (depending on ω from the underlying probability space).

$$A(\omega) = \int_0^t f(X_s(\omega), s)\mathrm{d}s, \quad \omega \in \Omega. \tag{17}$$

A crash course on SDE

Just like the ODE, we can do something like,

$$X_t = X_0 + \underbrace{\int_0^t f(X_s, s)\mathrm{d}s}_{:=A} + \underbrace{\int_0^t g(s)\mathrm{d}W_s}_{:=B}. \tag{18}$$

- B term requires an elaboration. You can roughly understand it as,

$$B(\omega) = \lim_{\Delta \to 0} \sum_{j \geq 0} g(t_j)[W_{t_{j+1}} - W_{t_j}](\omega), \tag{19}$$

where Δ is the largest length among the segments $\{[t_j, t_{j+1})\}_{j \geq 0}$.

A crash course on SDE

Without rigour, we see that B is Gaussian with parameters

$$\begin{aligned}
\mathbb{E}[B] &= \lim_{\Delta \to 0} \sum_{j \geq 0} g(t_j) \mathbb{E}[W_{t_{j+1}} - W_{t_j}] = 0, \\
\mathrm{Var}[B] &= \lim_{\Delta \to 0} \sum_{j \geq 0} g^2(t_j) \, \mathrm{Var}[W_{t_{j+1}} - W_{t_j}] \\
&= \lim_{\Delta \to 0} \sum_{j \geq 0} g^2(t_j)(t_{j+1} - t_j) \\
&= \int_0^t g^2(s) \mathrm{d}s.
\end{aligned} \tag{20}$$

A crash course on SDE

The solution X_t of Eq. (15) is generally intractable. To obtain a numerical approximation, divide $[0, T]$ into K bins and define for $k = 0, \ldots, K$,

$$t_k = k \cdot \delta \text{ where } \delta := T/K. \tag{21}$$

If we had only the drift term, that is, $\mathrm{d}X_t = f(X_t, t)\mathrm{d}t$, we consider a linear approximation of X_T around $X_{T-\delta} = X_{t_{K-1}}$:

$$X_T \approx X_{t_{K-1}} + \delta f(X_{t_{K-1}}, t_{K-1}). \tag{22}$$

Applying this approximation recursively, we get an iterative method,

$$X_{t_k} \approx X_{t_{k-1}} + \delta f(X_{t_{k-1}}, t_{k-1}) \text{ for } k = 1, \ldots, K. \tag{23}$$

A crash course on SDE

As we also have the Wiener process term. We know that

$$\int_{t_{k-1}}^{t_k} g(s)\mathrm{d}W_s \sim \mathcal{N}\left(0, \int_{t_{k-1}}^{t_k} g^2(s)\mathrm{d}s\right), \tag{24}$$

hence

$$\int_{t_{k-1}}^{t_k} g(s)\mathrm{d}W_s \approx \sqrt{\delta} g(t_{k-1})\varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I). \tag{25}$$

So the iterative approximation method for SDE becoe

$$X_{t_k} \approx X_{t_{k-1}} + \delta f(X_{t_{k-1}}, t_{k-1}) + \sqrt{\delta} g(t_{k-1})\varepsilon. \tag{26}$$

A crash course on SDE

- The law $\mathbb{P}$ of the process X is called a path measure of X.
- We assume that the time-marginals $\mathbb{P}_t$ admit a density p_t w.r.t. the Lebesgue measure.
- We also define joint and conditional densities, for instance,

$$\begin{aligned} \mathbb{P}(X_t \in A, X_s \in B) &= \iint_{A \times B} p_{t,s}(x,y)\mathrm{d}x\mathrm{d}y, \\ \mathbb{P}(X_t \in A | X_s = x) &= \int_A p_{t|s}(y|x)\mathrm{d}y. \end{aligned} \tag{27}$$

A crash course on SDE

A key question: how does p_t evolve over time?

Theorem 1 (Fokker-Planck Equation)

Let $(p_t)_{t\in[0,T]}$ be the time-marginal densities of X solving (15) . Then p_t solves the Fokker-Planck Equation (FPE),

$$\partial_t p_t(x) = -\nabla \cdot (f(x,t)p_t(x)) + \frac{g^2(t)}{2}\Delta p_t(x). \tag{28}$$

A crash course on SDE

Are there simple cases where p_t or at least $p_{t|s}$ is tractable?

Proposition 1 (Affine drifts)

Assume $f(x,t) = a(t)x$ in Eq. (15). Then for any $0 \leq s \leq t \leq T$,

$$X_t \,|\, X_s = x \sim \mathcal{N}(\mu_{t|s}(x), \sigma_{t|s}^2 I), \tag{29}$$

where

$$\begin{aligned} \mu_{t|s}(x) &= \exp\left(\int_s^t a(u)\mathrm{d}u\right) x, \\ \sigma_{t|s}^2 &= \int_s^t \exp\left(2\int_u^t a(v)\mathrm{d}v\right) g^2(u)\mathrm{d}u. \end{aligned} \tag{30}$$

From discrete to continuous time

Coming back to the discrete-time diffusion models,

$$q(x_k|x_{k-1}) = \mathcal{N}(x_k|(\alpha_k/\alpha_{k-1})x_{k-1}, \alpha_k^2(e^{-\rho_k} - e^{-\rho_{k-1}})I). \tag{31}$$

Map each step k to $t_k := k \cdot T/K$ in $[0, T]$. Let $\alpha : [0, T] \to \mathbb{R}$ and $\sigma : [0, T] \to [0, \infty)$ be differentiable functions where

$$\begin{aligned} &\alpha_k = \alpha(t_k), \quad \sigma_k = \sigma(t_k) \text{ for } k = 0, \ldots, K, \\ &\alpha(0) = 1, \quad \sigma(0) = 0. \end{aligned} \tag{32}$$

Then

$$\frac{\alpha_k}{\alpha_{k-1}} = \delta \frac{\alpha(t_{k-1} + \delta) - \alpha(t_{k-1})}{\delta \alpha(t_{k-1})} + 1 \approx \delta \frac{\dot{\alpha}(t_{k-1})}{\alpha(t_{k-1})} + 1, \tag{33}$$

where $\dot{f} = \frac{\mathrm{d}f}{\mathrm{d}t}$ denotes the time derivative of a function and $\delta := T/K$.

From discrete to continuous time

Similarly,

$$\begin{aligned}\alpha_k^2(e^{-\rho_k} - e^{-\rho_{k-1}}) &= \frac{\alpha_{k-1}^2(\sigma_k^2 - \sigma_{k-1}^2) - \sigma_{k-1}^2(\alpha_k^2 - \alpha_{k-1}^2)}{\alpha_{k-1}^2} \\ &\approx \delta\dot{\sigma^2}(t_{k-1}) - \delta\frac{\sigma^2(t_{k-1})}{\alpha^2(t_{k-1})}\dot{\alpha^2}(t_{k-1}) \\ &= -\delta\sigma^2(t_{k-1})\dot{\rho}(t_{k-1}).\end{aligned} \tag{34}$$

Here, $\alpha(t)$ and $\sigma(t)$ are properly chosen such that the log SNR $\rho(t) = \log(\alpha^2(t)/\sigma^2(t))$ is a non-increasing function.

From discrete to continuous time

Then the corruption density $q(x_k \,|\, x_{k-1})$ can be rewritten as,

$$X_{t_k} = X_{t_{k-1}} + \delta \frac{\dot{\alpha}(t_{k-1})}{\alpha(t_{k-1})} X_{t_{k-1}} + \sqrt{-\delta \sigma^2(t_{k-1}) \dot{\rho}(t_{k-1})} \cdot \varepsilon, \tag{35}$$

where $X_{t_k} = x_k$ and $X_{t_{k-1}} = x_{k-1}$. This is an approximate update step of the following SDE,

$$\mathrm{d}X_t = \underbrace{\frac{\dot{\alpha}(t)}{\alpha(t)} X_t}_{:= f(X_t, t)} \mathrm{d}t + \underbrace{\sqrt{-\sigma^2(t) \dot{\rho}(t)}}_{:= g(t)} \mathrm{d}W_t, \quad X_0 \sim p_{\text{data}}. \tag{36}$$

By Proposition 1, we recover the corruption equation we started from.

$$X_t \,|\, X_0 = x \sim \mathcal{N}(\alpha(t) x, \sigma^2(t) I), \tag{37}$$

The generative process via time reversal

- The forward process corrupts data into noise. For generation, we must reverse this process.

$$\mathrm{d}X_t = f(X_t, t)\mathrm{d}t + g(t)\mathrm{d}W_t, \quad X_0 \sim p_{\text{data}}. \tag{38}$$

- Let $\mathbb{P}^\star$ be the path measure of the forward process with time marginals $(p_t^\star)_{t\in[0,T]}$ and $p_0^\star = p_{\text{data}}$. Find the reverse-time SDE that starts from $p_T^\star$ and transform it back to $p_0^\star$.

$$\overleftarrow{X}_0 = X_T \sim p_T^\star \overset{\text{SDE}}{\Longrightarrow} \overleftarrow{X}_T = X_0 \sim p_0^\star = p_{\text{data}}, \tag{39}$$

where $\overleftarrow{f} := f(T-t)$ denotes the time-reversal.

The generative process via time reversal

Starting from the FPE, using the identity $\nabla p_t^\star = p_t^\star \nabla \log p_t^\star$,

$$\begin{aligned} \partial_t p_t^\star &= -\nabla \cdot (f p_t^\star) + \frac{g^2}{2} \nabla \cdot (p_t^\star \nabla \log p_t^\star) \\ &= -\nabla \cdot ((f - g^2 \nabla \log p_t^\star) p_t^\star) - \frac{g^2}{2} \Delta p_t^\star. \end{aligned} \tag{40}$$

Taking the change of variable $t \leftarrow T - t$,

$$\partial_t \overleftarrow{p}_t^\star = \nabla \cdot ((\overleftarrow{f} - \overleftarrow{g}^2 \nabla \log \overleftarrow{p}_t^\star) \overleftarrow{p}_t^\star) + \frac{\overleftarrow{g}^2}{2} \Delta \overleftarrow{p}_t^\star. \tag{41}$$

From Theorem 1, we realize that $\overleftarrow{p}_t^\star$ as the time-marginal of the SDE defining the reverse process (Anderson, 1982; Song, Sohl-Dickstein, et al., 2021):

$$\mathrm{d}\overleftarrow{X}_t = -(\overleftarrow{f} - \overleftarrow{g}^2 \nabla \log \overleftarrow{p}_t^\star)(\overleftarrow{X}_t, t)\mathrm{d}t + \overleftarrow{g}(t)\mathrm{d}\overleftarrow{W}_t, \quad \overleftarrow{X}_0 \sim p_T^\star. \tag{42}$$

Score-based generative models

To implement the reverse SDE as a generative model, two components are required.

- The initial distribution $p_T^\star$: this is generally unknown, but we can approximate it with a simple noise distribution p_{noise}.
- The score function $\nabla \log \overleftarrow{p}_t^\star$: this is generally intractable, so approximated via a neural network $\mathsf{s}_\theta(x, t)$.

How to learn the parameter θ of the score network s_θ?

Score-based generative models

Based on two choices, we construct a score-based generative model,

$$\mathrm{d}\overleftarrow{X}_t^\theta = -(\overleftarrow{f} - \overleftarrow{g}^2 \overleftarrow{\mathsf{s}}_\theta)(\overleftarrow{X}_t^\theta, t)\mathrm{d}t + \overleftarrow{g}(t)\mathrm{d}\overleftarrow{W}_t^\theta, \quad \overleftarrow{X}_0^\theta \sim p_{\text{noise}}. \tag{43}$$

Let $\overleftarrow{\mathbb{P}}^\theta$ be the path measure associated with $\overleftarrow{X}^\theta$. We want to minimize the KL divergence between them,

$$D_{\mathrm{KL}}[\overleftarrow{\mathbb{P}}^\star \| \overleftarrow{\mathbb{P}}^\theta] = \mathbb{E}_{\overleftarrow{\mathbb{P}}^\star}\left[\log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}^\star}{\mathrm{d}\overleftarrow{\mathbb{P}}^\theta}(\overleftarrow{X})\right]. \tag{44}$$

How to compute the KL divergence between the path measures?

Computing divergence between path measures

The KL divergence between two path measures can be decomposed into the KL divergence between the initial distributions and the expected KL divergence between the paths afterwards (Léonard, 2014)

$$D_{\mathrm{KL}}[\overleftarrow{\mathbb{P}}^{\star} \| \overleftarrow{\mathbb{P}}^{\theta}] = D_{\mathrm{KL}}[p_T^{\star} \| p_{\text{noise}}] + \mathbb{E}_{p_T^{\star}} \left[D_{\mathrm{KL}}[\overleftarrow{\mathbb{P}}^{\star}(\cdot | \overleftarrow{X}_0) \| \overleftarrow{\mathbb{P}}^{\theta}(\cdot | \overleftarrow{X}_0)] \right]. \tag{45}$$

The first term is irreducible, so we must focus on the second term.

Score matching for generative models

Theorem 2 (Girsanov Theorem, Informal)

Consider two SDEs,

$$\begin{aligned} \mathrm{d}X_t^a &= a(X_t, t)\mathrm{d}t + \sigma(t)\mathrm{d}W_t^a, \\ \mathrm{d}X_t^b &= b(X_t, t)\mathrm{d}t + \sigma(t)\mathrm{d}W_t^b, \end{aligned} \tag{46}$$

and let $\mathbb{P}^a$ and $\mathbb{P}^b$ denote the path measures of X^a and X^b, respectively. Assume that $h := (b-a)/\sigma$ satisfies $\mathbb{E}_{\mathbb{P}^a}[\exp(\frac{1}{2}\int_0^T h_t^2 \mathrm{d}t)] < \infty$. Then the Radon-Nikodym Derivative (RND) between the path measures is given as,

$$\log \frac{\mathrm{d}\mathbb{P}^b}{\mathrm{d}\mathbb{P}^a}(X) = \int_0^T h(X_t, t) \cdot \mathrm{d}W_t^a - \frac{1}{2}\int_0^T \|h\|^2(X_t, t)\mathrm{d}t. \tag{47}$$

Score matching for generative models

Applying Theorem 2 to $\overleftarrow{\mathbb{P}}$ and $\overleftarrow{\mathbb{P}}^\theta$,

$$\mathbb{E}_{\overleftarrow{\mathbb{P}}}\left[\log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}}{\mathrm{d}\overleftarrow{\mathbb{P}}^\theta}(\overleftarrow{X})\right] = \frac{1}{2}\mathbb{E}_{\overleftarrow{\mathbb{P}}}\left[\int_0^T g^2(t)\|\overleftarrow{\mathsf{s}}_\theta(\overleftarrow{X}_t, t) - \nabla \log \overleftarrow{p}_t^\star(\overleftarrow{X}_t)\|^2 \mathrm{d}t\right]. \quad (48)$$

That is, minimizing the divergence between the path measure amounts to matching the score network s_θ to the true score function $\nabla \log p_t$. The integral along the time axis can be written as an expectation, so we write

$$\mathcal{L}_{\mathsf{SM}}(\theta; g) := \frac{1}{2}\mathbb{E}_{\mathcal{U}(0,T),p_t^\star}\left[g^2(t)\|\mathsf{s}_\theta(X_t, t) - \nabla \log p_t^\star(X_t)\|^2\right], \quad (49)$$

where $\mathcal{U}(0, T)$ denotes a uniform distribution over $[0, T]$.

Score matching for generative models

- The weight g can be replaced by an arbitrary weight $w(t)$ without chaning the optimal solution (i.e., $\mathsf{s}_\theta(X_t, t) = \nabla \log p_t^\star(X_t)$ for all $t \in [0, T]$).

$$\mathcal{L}_{\mathrm{SM}}(\theta; w) = \frac{1}{2}\mathbb{E}_{\mathcal{U}(0,T),p_t^\star}\left[w^2(t)\|\mathsf{s}_\theta(X_t, t) - \nabla \log p_t^\star(X_t)\|^2\right]. \quad (50)$$

- For instance, Ho et al. (2020) suggested to set $w(t) = 1$ for all $t \in [0, T]$, leading to better quality for image generative models.
- However, Eq. (50) is still intractable due to the score $\nabla \log p_t^\star(x)$.

Conditional score matching

The marginal score is an expectation of the conditional score:

$$\nabla \log p_t^\star(x) = \mathbb{E}_{p^\star_{0|t}} \left[\nabla \log p^\star_{t|0}(x \,|\, X_0) \right]. \tag{51}$$

Based on this observation,

$$\begin{aligned}
&\mathbb{E}_{p_t^\star} \left[\| \mathsf{s}_\theta(X_t, t) - \nabla \log p_t^\star(X_t) \|^2 \right] \\
&\overset{\mathrm{c}}{=} \int p_t^\star(x_t) \left(\| \mathsf{s}_\theta(x_t, t) \|^2 - \mathsf{s}_\theta(x_t, t) \cdot \nabla \log p_t(x_t) \right) \mathrm{d}x_t \\
&\overset{\mathrm{c}}{=} \int p^\star_{t|0}(x_t) p_{\mathsf{data}}(x_0) \left(\| \mathsf{s}_\theta(x_t, t) \|^2 - \mathsf{s}_\theta(x_t, t) \cdot \nabla \log p^\star_{t|0}(x_t \,|\, x_0) \right) \mathrm{d}x_t \mathrm{d}x_0 \\
&\overset{\mathrm{c}}{=} \mathbb{E}_{p^\star_{t|0} p_{\mathsf{data}}} \left[\| \mathsf{s}_\theta(X_t, t) - \nabla \log p^\star_{t|0}(X_t \,|\, X_0) \|^2 \right],
\end{aligned} \tag{52}$$

where $\overset{\mathrm{c}}{=}$ denotes the equality up to a constant.

Conditional score matching

Since $p^{\star}_{t|0}$ is Gaussian, its score has a simple analytic form:

$$\nabla \log p^{\star}_{t|0}(x_t \,|\, x_0) = -\frac{x_t - \alpha(t) x_0}{\sigma^2(t)}. \tag{53}$$

The conditional score matching loss is thus defined as,

$$\mathcal{L}_{\mathsf{CSM}}(\theta; w) = \frac{1}{2}\mathbb{E}_{\mathcal{U}(0,T), p^{\star}_{t|0} p_{\mathsf{data}}} \left[w^2(t) \left\| \mathsf{s}_\theta(X_t, t) + \frac{X_t - \alpha(t) X_0}{\sigma^2(t)} \right\|^2 \right]. \tag{54}$$

Conditional score matching

A step of training:

- Draw $t \sim \mathcal{U}(0, T)$.
- Draw X_0 from training dataset and corrupt by $X_t = \alpha(t)X_0 + \sigma(t)\varepsilon_t$.
- Compute $w^2(t)\nabla_\theta \left\| \mathsf{s}_\theta(X_t, t) + \frac{X_t - \alpha(t)X_0}{\sigma^2(t)} \right\|^2$ and update.

This training procedure is highly scalable because,

- Everything involved in unbiased gradient estimation is tractable.
- Simulation-free: we don't need to simulate any SDE during the training. No backpropagation through numerical SDE solvers.

Parameterizations for score matching

We may also use alternative parameterizations for the score network s_θ.

- Noise prediction model: since $\nabla \log p^\star_{t|0}(x_t \,|\, x_0) = -\frac{\varepsilon_t}{\sigma(t)}$, we can parameterize $\mathsf{s}_\theta(x,t) = -\frac{\mathsf{e}_\theta(x,t)}{\sigma(t)}$ and minimize

$$\frac{1}{2}\mathbb{E}_{\mathcal{U}(0,T),p^\star_{t|0}p_{\text{data}}}\left[w^2(t)\|\mathsf{e}_\theta(X_t,t)-\varepsilon_t\|^2\right]. \tag{55}$$

  Ho et al. (2020) proposed to use this one with $w(t) = 1$.

- Denoiser: parameterize $\mathsf{s}_\theta(x,t) = -\frac{x_t - \alpha(t)\mathsf{x}_\theta(x,t)}{\sigma^2(t)}$ and minimize

$$\frac{1}{2}\mathbb{E}_{\mathcal{U}(0,T),p^\star_{t|0}p_{\text{data}}}\left[w^2(t)\|\mathsf{x}_\theta(X_t,t)-X_0\|^2\right]. \tag{56}$$

Examples of score-based generative models

Existing score-based or diffusion-based generative models can be described as a special case of the framework we discussed so far, with varying choices of the functions $\alpha(t)$ and $\sigma(t)$ (Song, Sohl-Dickstein, et al., 2021).

- Variance Exploding SDE (VE-SDE): set $\alpha(t) = 0$, leading to

$$\mathrm{d}X_t = \sqrt{\dot{\sigma^2}(t)}\mathrm{d}W_t. \tag{57}$$

  Song and Ermon (2019) proposed a discrete time version of VE-SDE.

- Variance Preserving SDE (VP-SDE): set $\alpha(t) = e^{-\frac{1}{2}\int_0^t \beta(s)\mathrm{d}s}$ for some function β, and $\sigma^2(t) = 1 - \alpha^2(t)$. This lead to,

$$\mathrm{d}X_t = -\frac{1}{2}\beta(t)X_t\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}W_t, \tag{58}$$

  which is a continuous-time extension of DDPM (Ho et al., 2020).

Sampling from score-based generative models

Once trained s_θ, to generate a new sample, we have solve the SDE,

$$\mathrm{d}\overleftarrow{X}_t^\theta = -(\overleftarrow{f} - \overleftarrow{g}^2\overleftarrow{\mathsf{s}}_\theta)(\overleftarrow{X}_t^\theta, t)\mathrm{d}t + \overleftarrow{g}(t)\mathrm{d}\overleftarrow{W}_t^\theta, \quad \overleftarrow{X}_0^\theta \sim p_{\text{noise}}. \tag{59}$$

This requires a numerical solver, for instance, the linear approximation method (Euler-Maruyama solver):

$$\overleftarrow{X}_{t_k}^\theta = \overleftarrow{X}_{t_{k-1}}^\theta - \delta(\overleftarrow{f} - \overleftarrow{g}^2\overleftarrow{\mathsf{s}}_\theta)(\overleftarrow{X}_{t_{k-1}}^\theta, t_{k-1}) + \sqrt{\delta}\overleftarrow{g}(t_{k-1})\varepsilon_{k-1}. \tag{60}$$

Each step of the numerical solver requires a forward pass through a neural network s_θ, so reducing the Number of Function Evaluations (NFE) is crucial for practical use.

Sampling by solving ODEs

We can manipulate the same FPE a bit differently,

$$\begin{aligned}\partial_t p_t &= -\nabla_z \cdot (f p_t) + \frac{g^2}{2} \nabla \cdot (p_t \nabla \log p_t) \\ &= -\nabla \cdot \left(\left(f - \frac{g^2}{2} \nabla \log p_t \right) p_t \right).\end{aligned} \tag{61}$$

With a change of variable, we get another reverse process,

$$\partial_t \overleftarrow{p}_t = \nabla \cdot \left(\left(\overleftarrow{f} - \frac{\overleftarrow{g}^2}{2} \nabla \log \overleftarrow{p}_t \right) \overleftarrow{p}_t \right), \tag{62}$$

which corresponds to an ODE,

$$\mathrm{d}\overleftarrow{X}_t = -\left(\overleftarrow{f} - \frac{\overleftarrow{g}^2}{2} \nabla \log \overleftarrow{p}_t \right) (\overleftarrow{X}_t, t) \mathrm{d}t. \tag{63}$$

This is called a Probability Flow ODE (PF-ODE) (Song, Sohl-Dickstein, et al., 2021).

Sampling by solving ODEs

Both the reverse SDE and PF-ODE share the same FPEs, so the time marginals of them are the same.

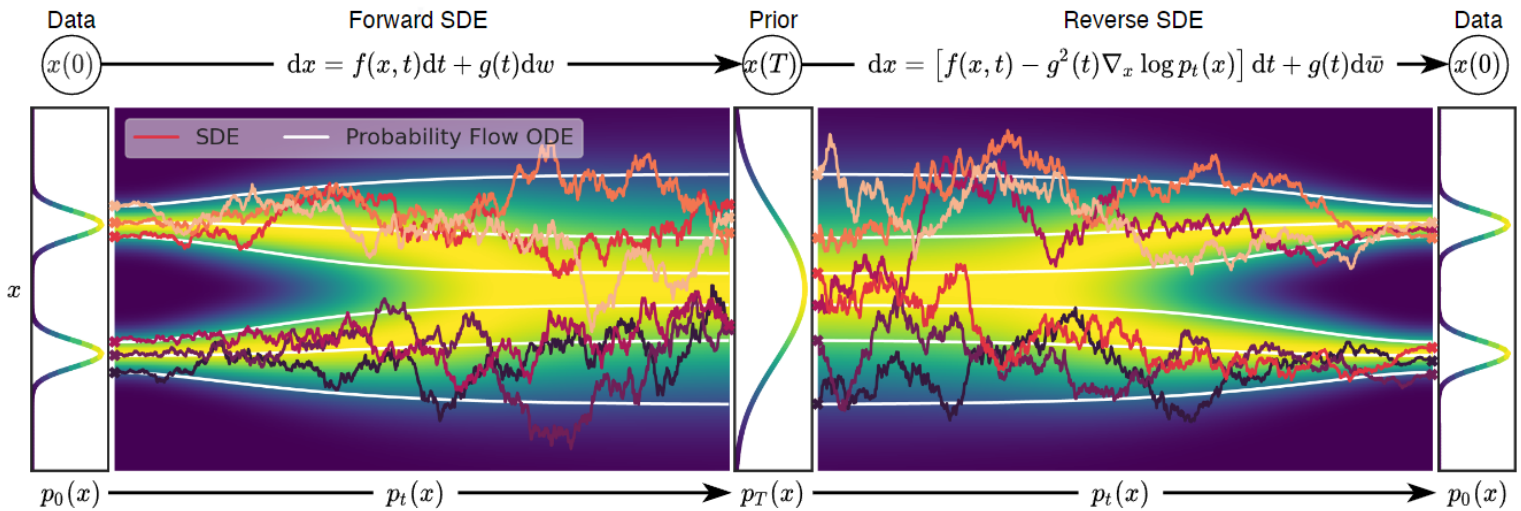

Figure 1: Figure from Song, Sohl-Dickstein, et al. (2021).

Sampling by solving ODEs

PF-ODE vs. reverse SDE?

- A single step of PF-ODE: take a deterministic step with f and s_θ.

$$\overleftarrow{X}^\theta_{t_k} = \overleftarrow{X}^\theta_{t_{k-1}} - \delta\left(\overleftarrow{f} - \frac{\overleftarrow{g}^2}{2}\overleftarrow{\mathsf{s}}_\theta\right)(\overleftarrow{X}^\theta_{t_{k-1}}, t_{k-1}). \tag{64}$$

- A single step of reverse SDE: take a double-length step with s_θ in the reverse SDE, then correct it with Gaussian noise.

$$\overleftarrow{X}^\theta_{t_k} = \overleftarrow{X}^\theta_{t_{k-1}} - \delta\left(\overleftarrow{f} - \overleftarrow{g}^2\overleftarrow{\mathsf{s}}_\theta\right)(\overleftarrow{X}^\theta_{t_{k-1}}, t_{k-1}) + \sqrt{\delta}\overleftarrow{g}(t_{k-1})\varepsilon_{k-1}. \tag{65}$$

In general, the reverse SDE takes longer to converge but produces higher-quality samples. On the other hand, while the PF-ODE may require fewer steps for convergence, its sample quality over time is typically inferior to that of the reverse SDE.

Table of Contents

Latent diffusion models

Vahdat et al. (2021) proposes Latent Score-based Generative Model (LSGM), where the SDE is defined on a latent space.

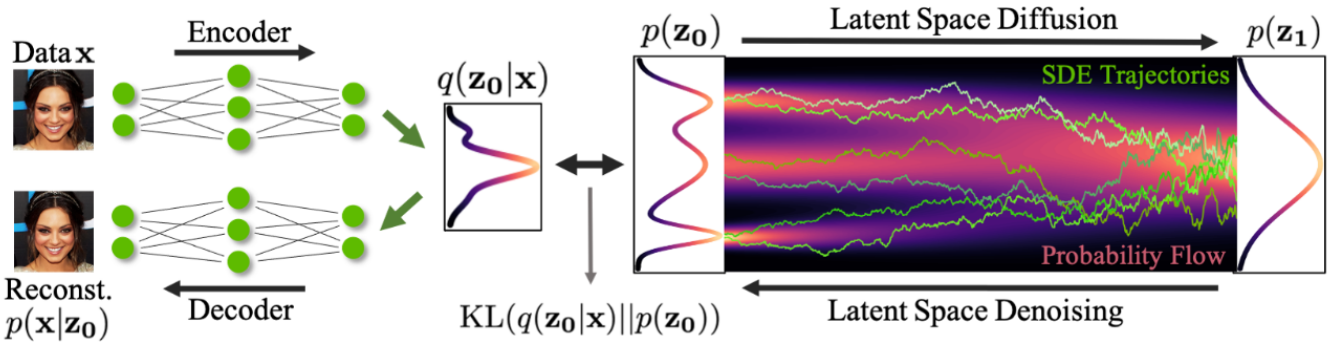

Figure 2: Vahdat et al. (2021).

Latent diffusion models

Rombach et al. (2022) introduces Latent Diffusion Model (LDM), a discrete variant of LSGM enhanced with various engineering optimizations to enable scalability, forming the foundation of Stable Diffusion.

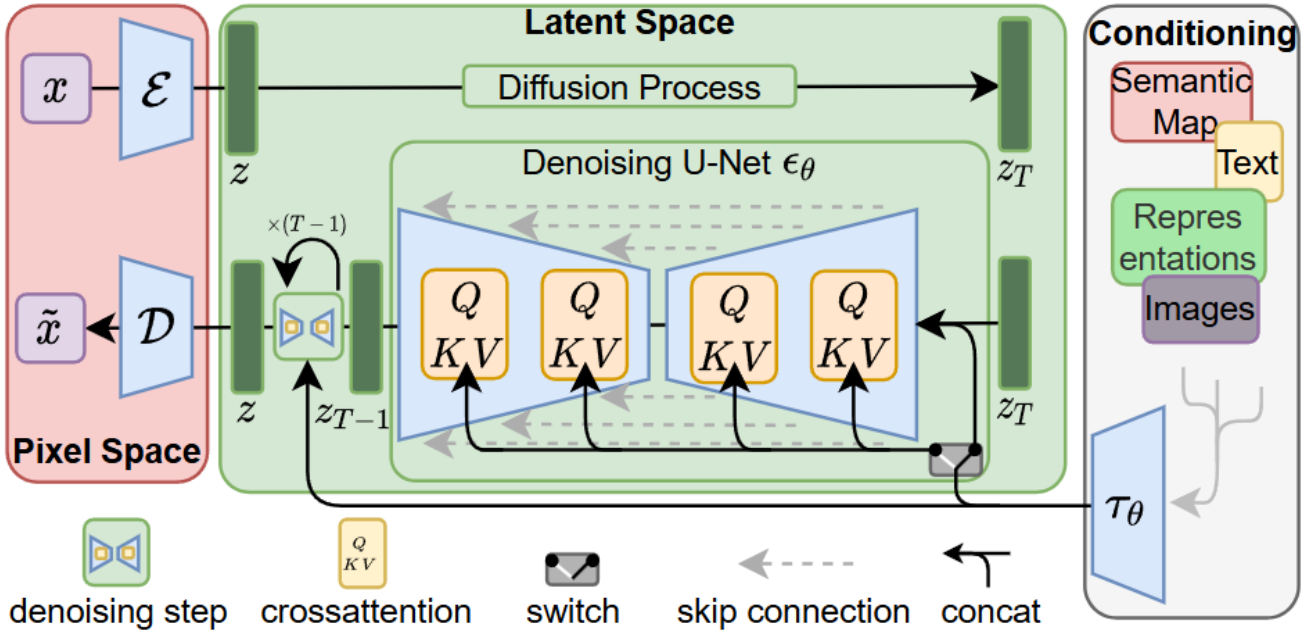

Figure 3: Rombach et al. (2022).

Latent diffusion models

The forward process of LDM is defined as,

$$x \sim p_{\text{data}}, \quad z_0 = \mathcal{E}(x), \quad q(z_k \,|\, z_0) = \mathcal{N}(z_k \,|\, \alpha_k z_0, \sigma_k^2 I), \tag{66}$$

where $\dim(z_k) \ll \dim(x)$. The backward process to generate a new sample is then

$$\begin{aligned} &\overleftarrow{z}_0 \sim p_{\text{noise}}, \\ &\overleftarrow{z}_k = \overleftarrow{z}_{k-1} - \delta(\overleftarrow{f} - \overleftarrow{g}^2 \overleftarrow{\mathsf{s}}_\theta)(\overleftarrow{z}_{k-1}, t_{k-1}) + \sqrt{\delta}\overleftarrow{g}(t_{k-1})\varepsilon_{k-1}, \\ &x = \mathcal{D}(\overleftarrow{z}_K). \end{aligned} \tag{67}$$

Training LDM

- Vahdat et al. (2021) derive a variational inference algorithm that jointly learns the autoencoder $(\mathcal{E}, \mathcal{D})$ and the score network. However, this method suffers from learning instability.
- Rombach et al. (2022) introduces an alternative strategy that pretrains the autoencoder and keeps it fixed during the training of the score network. This method has been shown to significantly improve stability, scalability, and generalization.

Advantages of LDM

Working in pixel space wastes computation on imperceptible details. In the latent space, these details are already encoded, allowing the model to focus solely on semantic compression.

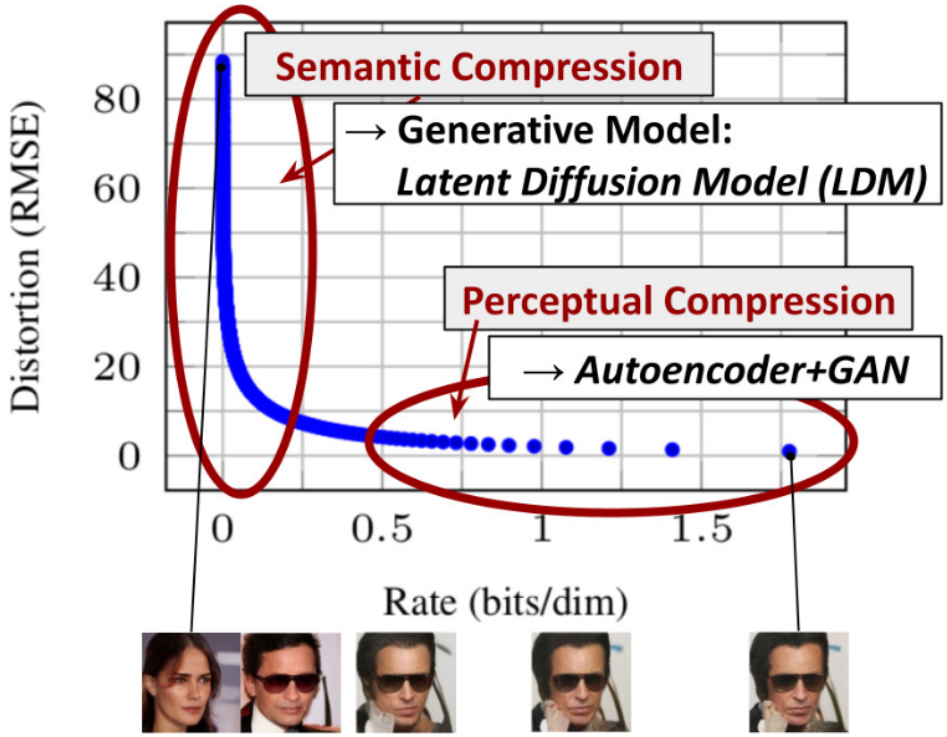

Figure 4: Rombach et al. (2022).

Advantages of LDM

Many real-world applications require conditioning on some information y (e.g., class conditional, text-to-image, image-to-image translation).

- Unconditional Diffusion Model (DM):

$$\nabla_z \log p_{t_k}(z_k) \approx \mathsf{s}_\theta(z_k, t_k). \tag{68}$$

- Conditional DM:

$$\nabla_z \log p_{t_k}(z_k \,|\, y) \approx \mathsf{s}_\theta(z_k, t_k, y). \tag{69}$$

Advantages of LDM

LDMs allow more flexible manipulation of conditioning by learning a shared latent space that seamlessly integrates both images and conditioning information.

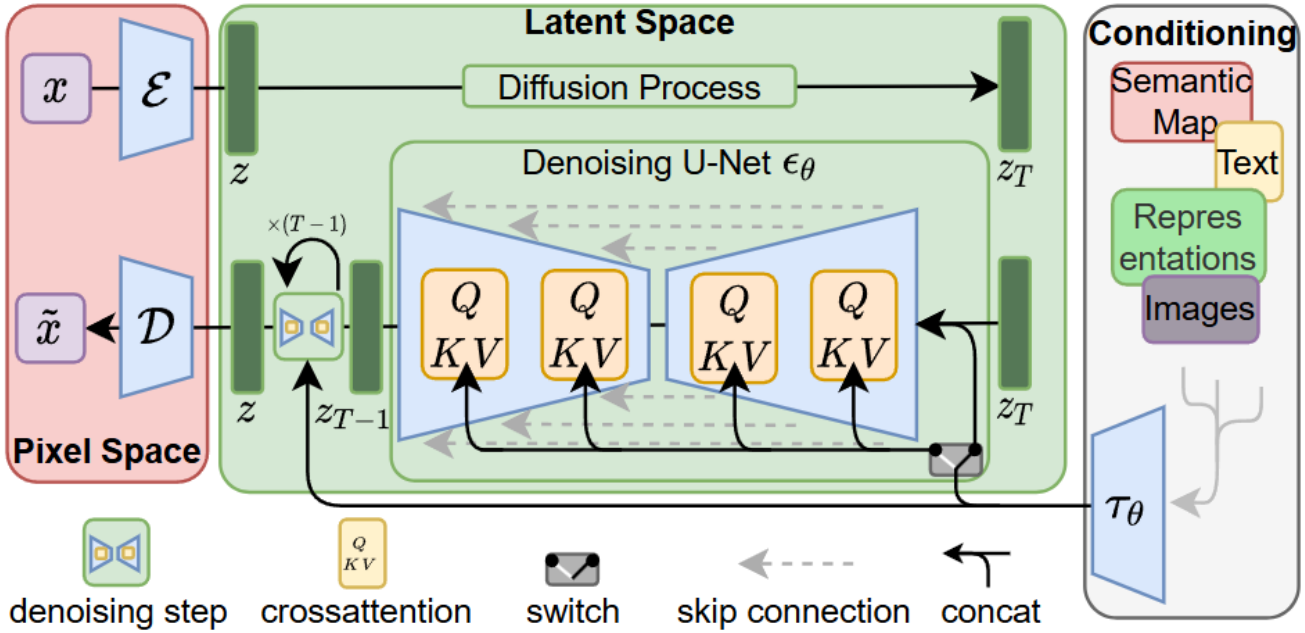

Figure 5: Rombach et al. (2022).

Table of Contents

The sampling problem

- So far, we have discussed generative modeling, where i.i.d. samples from a target distribution are given as a training set.
- In many scenarios (especially in Bayesian inference), we are often interested in sampling from a target distribution specified by an unnormalized density,

$$\pi(x) = \frac{\gamma(x)}{Z}, \tag{70}$$

where the normalization constant Z is typically unknown.

The sampling problem

- Typical approaches to sample from a unnormalized density $\gamma(x)$?
- MCMC (Hamiltonian Monte Carlo (HMC) (Neal, 2010) when γ is differentiable), without elaboration, may suffer mixing for high-dimensional and multi-modal distributions.
- Vanila MCMC is similar to "vanila" VAE in a sense that it is trying to "directly" sample from the target distribution.

Annealed importance sampling

- Annealed Importance Sampling (AIS) (Neal, 2001) introduces a sequence of annealed distributions and sample from a path that gradually transitions from an easy one to the target.
- Neal (2001) proposed a geometric annealing path as an example,

$$\pi_k(x) \propto \pi_0(x)^{1-\beta_k} \gamma(x)^{\beta_k}, \tag{71}$$

  where π_0 is a easy to simulate density and $0 = \beta_0 < \cdots < \beta_K = 1$.
- Similar to diffusion models, AIS breaks down the complex sampling problems into a sequence of simpler problems.

Diffusions for learned samplers

- The idea behind AIS is related to diffusion or score-based models.
- Based on the connection, many works (Doucet et al., 2022; Vargas et al., 2023; Richter et al., 2024; Chen et al., 2024) proposed learning diffusion models for sampling from an unnormalized density.

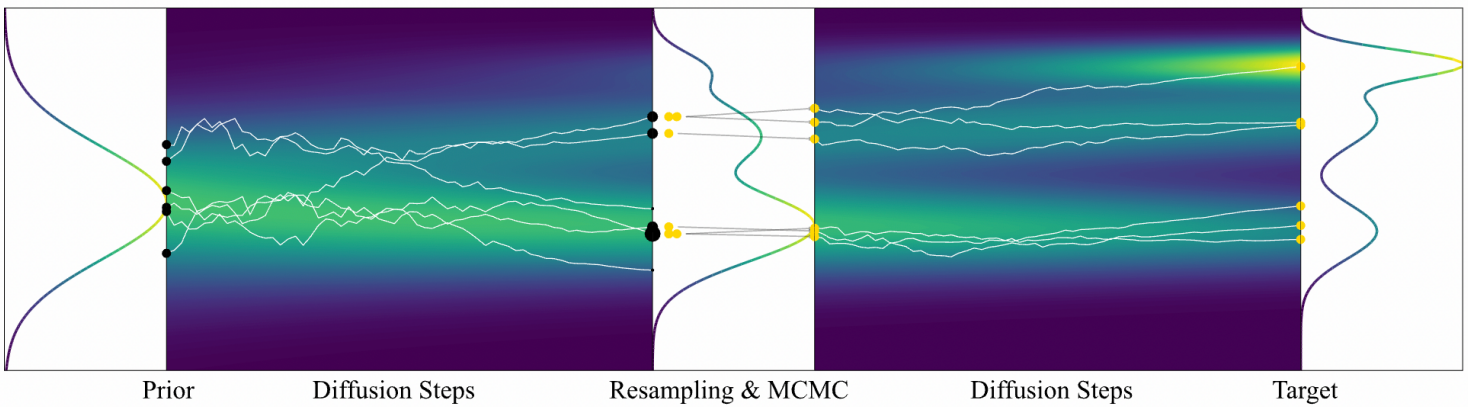

Figure 6: Figure from Chen et al. (2024).

Denoising diffusion samplers

- Denoising Diffusion Sampler (DDS) (Vargas et al., 2023) adapts VP-SDE (continuous-time DDPM) for the sampling problem.
- Take the time-reversal of the forward process, starting from π, as a target process.

$$\begin{aligned} \mathrm{d}X_t &= f(X_t, t)\mathrm{d}t + g(t)\mathrm{d}W_t, \quad X_0 \sim \pi, \\ \mathrm{d}\overleftarrow{X}_t &= -(\overleftarrow{f} - \overleftarrow{g}^2 \nabla \log \overleftarrow{p}_t^\star)(\overleftarrow{X}_t, t)\mathrm{d}t + \overleftarrow{g}(t)\mathrm{d}\overleftarrow{W}_t, \quad \overleftarrow{X}_0 \sim p_T^\star. \end{aligned} \tag{72}$$

- The sampling process is parameterized as,

$$\mathrm{d}\overleftarrow{X}_t^\theta = -(\overleftarrow{f} - \overleftarrow{g}^2 \overleftarrow{\mathsf{s}}_\theta)(\overleftarrow{X}_t^\theta, t)\mathrm{d}t + \overleftarrow{g}(t)\mathrm{d}\overleftarrow{W}_t^\theta, \quad \overleftarrow{X}_0^\theta \sim p_{\text{noise}}. \tag{73}$$

The objective for DDS

Let $\overleftarrow{\mathbb{P}}^\star$ and $\overleftarrow{\mathbb{P}}^\theta$ be path measures of the target and sampling processes, respectively. In generative modeling, we could minimize

$$D_{\mathrm{KL}}[\overleftarrow{\mathbb{P}}^\star \| \overleftarrow{\mathbb{P}}^\theta] = \mathbb{E}_{\overleftarrow{\mathbb{P}}^\star}\left[\log \frac{\mathrm{d}\overleftarrow{\mathbb{P}}^\star}{\mathrm{d}\overleftarrow{\mathbb{P}}^\theta}\right], \tag{74}$$

whose unbiased estimate could effectively be estimated as below,

$$\overleftarrow{X}_T \sim p_{\text{data}}, \quad \overleftarrow{X}_t \,|\, \overleftarrow{X}_T \sim p^\star_{T-t|0} \text{ for } t \in [0, T]. \tag{75}$$

However, in the sampling problem, we cannot do the first step, because we cannot sample from π!

The objective for DDS

To facilitate learning, we introduce a reference process,

$$\mathrm{d}X_t^\circ = f(X_t^\circ, t)\mathrm{d}t + g(t)\mathrm{d}W_t^\circ, \quad X_0^\circ \sim p_0^\circ, \tag{76}$$

Here, p_0° is chosen such that $\nabla \log p_t^\circ$ is analytic for all $t \in [0, T]$. For instance, in DDS, we choose

$$f(x,t) = -\frac{\beta(t)}{2}x, \quad g(t) = \sigma\sqrt{\beta(t)}, \quad p_0^\circ(x) = \mathcal{N}(x|0, \sigma^2 I), \tag{77}$$

leading to

$$\begin{aligned} &p_t^\circ(x) = \mathcal{N}(x|0,\sigma^2 I) \text{ for all } t \in [0,T], \quad \overleftarrow{f} - \overleftarrow{g}^2 \nabla \log \overleftarrow{p}_t^\circ = -\overleftarrow{f}, \\ &\mathrm{d}\overleftarrow{X}_t^\circ = \overleftarrow{f}(\overleftarrow{X}_t^\circ, t)\mathrm{d}t + \overleftarrow{g}(t)\mathrm{d}\overleftarrow{W}_t^\circ, \quad \overleftarrow{X}_0^\circ \sim p_T^\circ = p_0^\circ. \end{aligned} \tag{78}$$

The objective for DDS

With the help of the reference process, we can write

$$D_{\mathrm{KL}}[\mathbb{P}^\theta \| \mathbb{P}^\star] = D_{\mathrm{KL}}[\mathbb{P}^\theta \| \mathbb{P}^\circ] - \mathbb{E}_{\mathbb{P}^\theta}\left[\log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^\circ}\right]. \tag{79}$$

Here, we have

$$\log \frac{\mathrm{d}\mathbb{P}^\star}{\mathrm{d}\mathbb{P}^\circ}(X^\theta) = \log \frac{\pi(X_0^\theta)}{p_0^\circ(X_0^\theta)} = \log \frac{\pi(\overleftarrow{X}_T^\theta)}{p_0^\circ(\overleftarrow{X}_T^\theta)}. \tag{80}$$

The first KL term can be computed with Theorem 2,

$$\begin{aligned} D_{\mathrm{KL}}[\mathbb{P}^\theta \| \mathbb{P}^\circ] &= D_{\mathrm{KL}}[\overleftarrow{\mathbb{P}}^\theta \| \overleftarrow{\mathbb{P}}^\circ] \\ &= \frac{1}{2}\mathbb{E}_{\overleftarrow{\mathbb{P}}^\theta}\left[\int_0^T \overleftarrow{g}(t)^2 \|\overleftarrow{\mathsf{s}}_\theta(\overleftarrow{X}^\theta, t) - \nabla \log \overleftarrow{p}_t^\circ(\overleftarrow{X}^\theta)\|^2\right]. \end{aligned} \tag{81}$$

The objective for DDS

We can consider a parameterization,

$$\mathsf{u}_\theta(x,t) := g(t)(\mathsf{s}_\theta(x,t) - \nabla \log p_t^\circ(x)) = \sigma\sqrt{\beta(t)}(\mathsf{s}_\theta(x,t) + x/\sigma^2), \quad (82)$$

leading to the overall objective,

$$D_{\mathrm{KL}}[\mathbb{P}^\theta \| \mathbb{P}^\star] = \mathbb{E}_{\overleftarrow{\mathbb{P}}^\theta}\left[\frac{1}{2}\int_0^T \|\overleftarrow{\mathsf{u}}_\theta(\overleftarrow{X}_t^\theta, t)\|^2 \mathrm{d}t + \log \frac{\pi(\overleftarrow{X}_T^\theta)}{p_0^\circ(\overleftarrow{X}_T)}\right]. \quad (83)$$

The corresponding objective is actually a special instance of a Schrödinger bridge problem (Schrödinger, 1931; Schrödinger, 1932; Léonard, 2013), an entropy-regularized optimal transport problem.

Table of Contents

Diffusion models are everywhere!

- Generative models for various domains:
  - Image synthesis.
  - Molecule, protein, and drug discovery.
  - Improving on language models.
- Learned samplers.
  - Sampling from Boltzmann distributions.
  - Approximate Bayesian inference.
  - Solving optimal transport problems (e.g., distribution to distribution translation problems).

Key takeaways

Two things to learn from the methods discussed today:

- Break down a complex problem into easier problems.
- Tricks to keep learning tractable and scalable, for instance, conditional matching.

References I

- Anderson, B. D. O. (1982). "Reverse-time diffusion equation models". In: *Stochastic Processes and their Applications* 12.3, pp. 313–326 (cit. on p. 32).
- Chen, J. et al. (2024). "Sequential controlled Langevin diffusions". In: *Proceedings of The 13th International Conference on Learning Representations (ICLR 2025)* (cit. on p. 60).
- Doucet, A. et al. (2022). "Score-based diffusion meets annealed importance sampling". In: *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)* (cit. on p. 60).
- Goodfellow, I. et al. (2014). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27 (NIPS 2014)* (cit. on p. 4).
- Ho, J., A. Jain, and P. Abbeel (2020). "Denoising diffusion probabilistic models". In: *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)* (cit. on pp. 12, 15, 38, 42, 43).
- Kingma, D. P. and M. Welling (2014). "Auto-encoding variational Bayes". In: *Proceedings of The 2nd International Conference on Learning Representations (ICLR 2014)* (cit. on p. 4).
- Léonard, C. (2013). "A survey of the Schrödinger problem and some of its connections with optimal transport". In: *Discrete and Continuous Dynamical Systems* 34.4, pp. 1533–1574 (cit. on p. 65).
- — (2014). "Some properties of path measures". In: *Séminaire de Probabilités XLVI*. Vol. 2123. Lecture Notes in Mathematics. Springer, pp. 207–230 (cit. on p. 35).
- Neal, R. M. (2001). "Annealed importance sampling". In: *Statistics and Computing* 11.2, pp. 125–139 (cit. on p. 59).
- — (2010). "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo* 54, pp. 113–162 (cit. on p. 58).

References II

- Rezende, D. and S. Mohamed (2015). "Variational inference with normalizing flows". In: *Proceedings of The 32th International Conference on Machine Learning (ICML 2015)* (cit. on p. 4).
- Richter, L. and J. Berner (2024). "Improved sampling via learned diffusions". In: *Proceedings of The 12th International Conference on Learning Representations (ICLR 2024)* (cit. on p. 60).
- Rombach, R. et al. (2022). "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 50, 52, 53, 55).
- Schrödinger, Erwin (1931). "Über die Umkehrung der Naturgesetze". In: *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse*, pp. 144–153 (cit. on p. 65).
- — (1932). "Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique". In: *Annales de l'Institut Henri Poincaré* 2.4, pp. 269–310 (cit. on p. 65).
- Sohl-Dickstein, J. et al. (2015). "Deep unsupervised learning using nonequilibrium thermodynamics". In: *Proceedings of The 32th International Conference on Machine Learning (ICML 2015)* (cit. on p. 12).
- Song, Y. and S. Ermon (2019). "Generative modeling by estimating gradients of the data distribution". In: *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)* (cit. on p. 43).
- Song, Y., J. Sohl-Dickstein, et al. (2021). "Score-based generative modeling through stochastic differential equations". In: *Proceedings of The 9th International Conference on Learning Representations (ICLR 2021)* (cit. on pp. 32, 43, 45, 46).
- Uria, B. et al. (2016). "Neural autorgressive distribution estimation". In: *Journal of Machine Learning Research* 17.205, pp. 1–37 (cit. on p. 4).

References III

- Vahdat, A., K. Kreis, and J. Kautz (2021). “Score-based generative modeling in latent space”. In: *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)* (cit. on pp. 49, 52).
- Vargas, F., W. Grathwohl, and A. Doucet (2023). “Denoising diffusion samplers”. In: *Proceedings of The 11th International Conference on Learning Representations (ICLR 2023)* (cit. on pp. 60, 61).