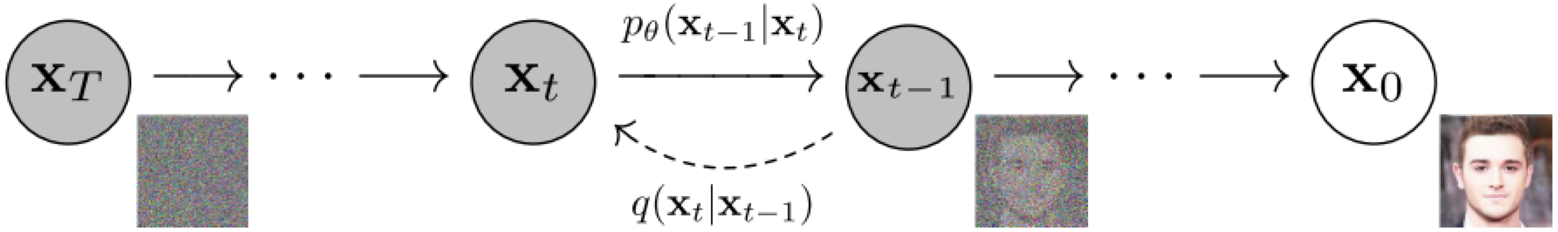
Microsoft

Introduction to Diffusion-Based Generative Models

Chang Liu
Microsoft Research AI4Science

Diffusion-Based Models

"Creating noise from data is easy; creating data from noise is generative modeling." [SSK+21]



[HJA20]

Discrete-time Markov chain.

- "Creating noise":

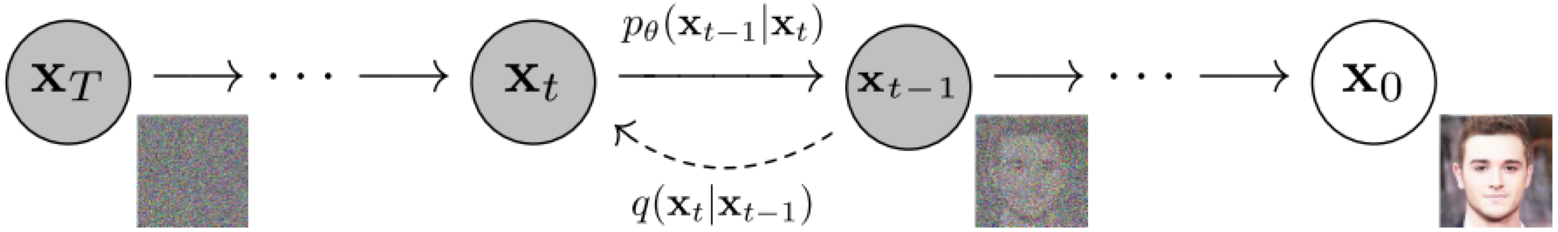
  A **diffusion process** that gradually transforms the data distr. to a noise distr. p_{noise}.

- "Creating data from noise":

  Learn the **reverse process** that gradually transforms the noise distr. p_{noise} to the data distr. "Denoising".

Diffusion-Based Models

"Creating noise from data is easy; creating data from noise is generative modeling." [SSK+21]



[HJA20]

Clarifications:

- Forward process: $q_0 \xrightarrow{q_{1|0}} q_1 \xrightarrow{q_{2|1}} \cdots \xrightarrow{q_{N|N-1}} q_N \approx p_{\text{noise}}$.
  - The terminal distribution is **tractable**: known and easy to (IID) sample.
  - Fixed: **additional information**, **step-by-step guidance**!
- Reverse process: $p_{\text{noise}} =: p_N \xrightarrow{p_{N-1|N}} p_{N-1} \xrightarrow{p_{N-2|N-1}} \cdots \xrightarrow{p_{0|1}} p_0 \approx q_0$.
  - "Reverse" means
    $p(x_{0:N}) = p_N(x_N)p(x_{N-1}|x_N)\cdots p(x_0|x_1) \quad = \quad q(x_{0:N}) = q_0(x_0)q(x_1|x_0)\cdots q(x_N|x_{N-1})$.
    - Principle of learning.
  - Distribution-to-distribution $q_0 \xrightarrow{\text{fwd}} q_N \xrightarrow{\text{rev}} p_0 = q_0$, **not point-to-point** $x_0 \xrightarrow{\text{fwd}} x_N \xrightarrow{\text{rev}} x_0' \neq x_0$.

DDPM
- **Evidence Lower BOund**
- **DDPM simple loss**
- DDPM variants

Cont.-time view:
- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching
- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Cont.-time improvements
- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood
- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Denoising Diffusion Probabilistic Model

[SWMG15, HJA20]

- Forward process:
  - $q_0 \coloneqq$ data distribution.
  - $q(x_i|x_{i-1}) \coloneqq \mathcal{N}\left(x_i|\sqrt{1-\beta_i}x_{i-1}, \beta_i I\right)$, where $\beta_i \in (0,1)$.

    $\beta_i = \frac{\beta_{\min}}{N} + \frac{i-1}{N-1}\left(\frac{\beta_{\max}}{N} - \frac{\beta_{\min}}{N}\right)$.

    ➔ $q(x_i|x_0) = \mathcal{N}\left(x_i|\sqrt{\alpha_i}x_0, (1-\alpha_i)I\right)$, $\alpha_i \coloneqq \prod_{j=1}^{i}\left(1-\beta_j\right)$.

    So $q(x_N|x_0) \approx \mathcal{N}(0, I)$ hence $q(x_N) \approx \mathcal{N}(0, I)$!!!
- Reverse process:
  - $p_N \coloneqq \mathcal{N}(0, I)$.
  - $p_\theta(x_{i-1}|x_i) \coloneqq \mathcal{N}\left(x_{i-1}|\mu_{\theta,i}(x_i), \Gamma_{\theta,i}(x_i)\right)$.
    - In the limit $\beta \to 0$, $p(x_{i-1}|x_i)$ has the same functional form as $q(x_i|x_{i-1})$ [SWMG15].
    - Easy to simulate.

Denoising Diffusion Probabilistic Model

- Training: $\underset{\theta}{\operatorname{argmin}} \mathrm{KL}\big(q(x_{0:N}) \| p_\theta(x_{0:N})\big) = \underset{\theta}{\operatorname{argmin}} -\mathbb{H}[q_0] - \mathbb{E}_{q_0(x_0)}[\mathrm{ELBO}_\theta(x_0)]$, [SWMG15]

$$\mathrm{ELBO}_\theta(x_0) \coloneqq \mathbb{E}_{q(x_{1:N}|x_0)}[\log p_\theta(x_0, x_{1:N}) - \log q(x_{1:N}|x_0)].$$

  - Step-by-step supervision: $\mathrm{ELBO}_\theta(x_0) =$
$-\sum_{i=2}^{N} \underbrace{\mathbb{E}_{q(x_i|x_0)} \mathrm{KL}\big(q(x_{i-1}|x_i, x_0) \| p_\theta(x_{i-1}|x_i)\big)}_{=:L_{i-1}(x_0)} - \underbrace{\mathrm{KL}\big(q(x_N|x_0) \| p_N(x_N)\big)}_{\text{const.}} + \underbrace{\mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0|x_1)]}_{\text{handle separately}}.$
  - Let $p_\theta(x_{i-1}|x_i) = \mathcal{N}\big(x_{i-1}|\mu_{\theta,i}(x_i), \gamma_i^2 I\big)$:

    ➔ $L_{i-1}(x_0) = \mathbb{E}_{q(x_i|x_0)}\left[\frac{1}{2\gamma_i^2}\left\|\tilde{\mu}_i(x_i, x_0) - \mu_{\theta,i}(x_i)\right\|^2\right] + \text{const.}$
  - [HJA20: DDPM] Let $\mu_{\theta,i}(x_i) = \frac{1}{\sqrt{1-\beta_i}}\left(x_i - \frac{\beta_i}{\sqrt{1-\alpha_i}}\epsilon_{\theta,i}(x_i)\right)$:

    ➔ $L_{i-1}(x_0) = \frac{\beta_i^2}{2\gamma_i^2(1-\beta_i)(1-\alpha_i)} \mathbb{E}_{p(\epsilon)}\left\|\epsilon - \epsilon_{\theta,i}\big(x_i(x_0, \epsilon)\big)\right\|^2 + \text{const.}$,
    where $x_i(x_0, \epsilon) \coloneqq \sqrt{\alpha_i} x_0 + \sqrt{1 - \alpha_i}\epsilon$.

    ➔ DDPM simple loss $\mathbb{E}_{q_0(x_0)} \mathbb{E}_{\mathrm{U}(i|\{1,\ldots,N\})} \mathbb{E}_{p(\epsilon)}\left\|\epsilon - \epsilon_{\theta,i}\big(x_i(x_0, \epsilon)\big)\right\|^2$:
    Better generation results.

$O(1)$ (w.r.t i) evaluation and backpropagation cost, since $q(x_i|x_0)$ can be sampled in $O(1)$.

DDPM

- Evidence Lower BOund
- DDPM simple loss
- **DDPM variants**

Cont.-time view:

- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching

- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time techniques:

- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Cont.-time improvements

- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood

- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Denoising Diffusion Implicit Models [SME21]

- Define the **forward process** as:

  No longer Markov

  $q_{\tilde{\sigma}}(x_{1:N}|x_0) = q(x_N|x_0) \prod_{i=2}^{N} q_{\tilde{\sigma}}(x_{i-1}|x_i, x_0)$,
  where $q(x_N|x_0) \coloneqq \mathcal{N}\left(\sqrt{\alpha_N} x_0, (1-\alpha_N) I\right)$, and

  $q_{\tilde{\sigma}}(x_{i-1}|x_i, x_0) \coloneqq \mathcal{N}\left(\tilde{\mu}_i\left(x_i, x_0, \tilde{\sigma}_i^2\right), \tilde{\sigma}_i^2 I\right)$, where $\tilde{\mu}_i\left(x_i, x_0, \tilde{\sigma}_i^2\right) \coloneqq \sqrt{\alpha_{i-1}} x_0 + \sqrt{1 - \alpha_{i-1} - \tilde{\sigma}_i^2} \, \frac{x_i - \sqrt{\alpha_i} x_0}{\sqrt{1-\alpha_i}}$.

  - ➔ $q_{\tilde{\sigma}}(x_i|x_0) = \mathcal{N}\left(\sqrt{\alpha_i} x_0, (1-\alpha_i) I\right), \forall i$:
    Recovers DDPM $q(x_i|x_0)$ (though not $q(x_{0:N})$) **for any $\tilde{\sigma}_i$ schedule**: Additional degree of freedom!

- DDPM $\epsilon_{\theta,i}$ can be used to predict x_0:

  $x_{0\theta,i}(x_i) \coloneqq \frac{x_i - \sqrt{1-\alpha_i}\,\epsilon_{\theta,i}(x_i)}{\sqrt{\alpha_i}}$ (recall fwd. proc. $x_i = \sqrt{\alpha_i} x_0 + \sqrt{1-\alpha_i}\,\epsilon_i$).

  ➔ Define **reverse model** using DDPM $\epsilon_{\theta,i}$: $p_\theta(x_{i-1}|x_i) \coloneqq q_{\tilde{\sigma}}\left(x_{i-1}|x_i, x_{0\theta,i}(x_i)\right)$.

  - $\tilde{\sigma}_i^2 = \tilde{\beta}_i = \frac{1-\alpha_{i-1}}{1-\alpha_i} \beta_i$ and $\gamma_i = \tilde{\sigma}_i$ ➔ Recover DDPM reverse model & DDPM loss.

- Efficient data generation:
  - Smaller $\tilde{\sigma}_i$ allows coarser $\{i_0 = 0, i_1, \dots, i_K = N\}$: **Accelerate generation by fewer layers**!
  - If $\tilde{\sigma}_i = 0$, $p(x_0|x_N)$ is a **deterministic map**: This is an **implicit** generative model (like a GAN).

Diffusion Model for Likelihood Estimation [KSPH21]

- SOTA likelihoods on image, outperforming autoregressive models (previous SOTA).
- Forward process: $q(x_i|x_0) = \mathcal{N}\left(x_i|\sqrt{\alpha_i}x_0, \sigma_i^2 I\right)$.

  ➔ $q(x_j|x_i) = \mathcal{N}\left(\sqrt{\frac{\alpha_j}{\alpha_i}}x_i, \sigma_j^2\left(1 - \frac{\xi_j}{\xi_i}\right)I\right)$ $(j > i)$.
  - Signal-to-noise ratio: $\xi_i \coloneqq \alpha_i/\sigma_i^2$ decreasing in i, s.t. $q(x_N|x_0)$ & $q(x_N) \approx \mathcal{N}(0, I)$.
- Reverse process:
  - Take $p(x_i|x_j) \coloneqq q\left(x_i|x_j, x_0 = \hat{x}_{\theta,j}(x_j)\right)$. Alternatively, use $\hat{x}_{\theta,j}(x_j) \leftarrow \frac{x_j - \sigma_j \hat{\epsilon}_{\theta,j}(x_j)}{\sqrt{\alpha_j}}$.

    It predicts x_0 from x_j, while DDPM model predicts x_{j-1} from x_j.
- Loss:
  - $L_{i-1}(x_0) = \frac{1}{2}(\xi_{i-1} - \xi_i)\mathbb{E}_{p(\epsilon)}\left\|x_0 - \hat{x}_{\theta,i}\left(\sqrt{\alpha_i}x_0 + \sigma_i\epsilon\right)\right\|^2 = \frac{1}{2}\left(\frac{\xi_{i-1}}{\xi_i} - 1\right)\mathbb{E}_{p(\epsilon)}\left\|\epsilon - \hat{\epsilon}_{\theta,i}\left(\sqrt{\alpha_i}x_0 + \sigma_i\epsilon\right)\right\|^2$.
  - Also optimize noise schedule: let $\sigma_i^2 = \mathrm{sigm}(\eta_i)$, $\alpha_i = 1 - \sigma_i^2$, DDPM's choice.

    $L_{i-1}(x_0) = \frac{1}{2}(e^{\eta_i - \eta_{i-1}} - 1)\mathbb{E}_{p(\epsilon)}\left\|\epsilon - \hat{\epsilon}_{\theta,i}\left(\sqrt{\alpha_i}x_0 + \sigma_i\epsilon\right)\right\|^2$.

Optimal Reverse Process [BLZZ22]

- Optimal reverse process to minimize DDPM loss (ELBO):
  $p^*(x_{0:N}) = p(x_N) \prod_{i=1}^{N} p^*(x_{i-1}|x_i)$, where $p^*(x_{i-1}|x_i) \coloneqq \mathcal{N}\left(x_{i-1}|\mu_i^*(x_i), \gamma_i^{*2} I\right)$,
  - $\mu_i^*(x_i) = \tilde{\mu}_i\left(x_i, \frac{1}{\sqrt{\alpha_i}}(x_i + (1-\alpha_i)\nabla \log q_{\tilde{\sigma}}(x_i)), \tilde{\sigma}_i^2\right)$, ➔ $\nabla \log q_{\tilde{\sigma}}(x_i) \approx -\frac{\epsilon_{\theta,i}(x_i)}{\sqrt{1-\alpha_i}}$ recovers DDPM.
  - $\gamma_i^{*2} = \tilde{\sigma}_i^2 + \left(\sqrt{\frac{1-\alpha_i}{1-\beta_i}} - \sqrt{1-\alpha_{i-1}-\tilde{\sigma}_i^2}\right)^2 \left(1 - \frac{1-\alpha_i}{d}\mathbb{E}_{q_{\tilde{\sigma}}(x_i)}\|\nabla \log q_{\tilde{\sigma}}(x_i)\|^2\right)$.
  - Bound: $\tilde{\sigma}_i^2 \le \gamma_i^{*2} \le \tilde{\sigma}_i^2 + \left(\sqrt{\frac{1-\alpha_i}{1-\beta_i}} - \sqrt{1-\alpha_{i-1}-\tilde{\sigma}_i^2}\right)^2$.
    - Used to clip stochastic estimate of γ_i^{*2}.
- Optimizing shortened diffusion process.
  - $\mathrm{KL}\left(q_{\tilde{\sigma}}(x_{0:N})\|p^*(x_{0:N})\right) = \frac{d}{2}\sum_{i=2}^{N} \log \frac{\gamma_i^{*2}}{\tilde{\sigma}_i^2} + \mathrm{C}$.
  - Choose $\{i'\} \subset \{1, \ldots, N\}$ to minimize:
    $\mathrm{KL}\left(q_{\tilde{\sigma}}(x_0, \{x_{i'}\})\|p^*(x_0, \{x_{i'}\})\right) = \frac{d}{2}\sum_{i'=2}^{K} \log \frac{\gamma_{i'-1|i'}^{*2}}{\tilde{\sigma}_{i'-1|i'}^2} + \mathrm{C}.$, by least-cost-path dynamic programming.

Optimal Reverse Process [BLS+22]

- Extension to covariance matrix:
  - Reverse: $p^*(x_{i-1}|x_i) := \mathcal{N}\left(x_{i-1}|\mu_i^*(x_i), \mathrm{Diag}\left(\boldsymbol{\gamma}_i^{*2}(x_i)\right)\right)$,
    - $\mu_i^*(x_i)$ is the same.
    - $\boldsymbol{\gamma}_i^{*2}(x_i) = \tilde{\sigma}_i^2 \mathbf{1} + \frac{1-\alpha_i}{\alpha_i}\left(\sqrt{\alpha_{i-1}} - \sqrt{\frac{\alpha_i}{1-\alpha_i}}\sqrt{1-\alpha_{i-1}-\tilde{\sigma}_i^2}\right)^2 \left(1 - \frac{1}{d}\mathrm{diag}\left(\mathrm{Cov}_{q_{\tilde{\sigma}}(x_0|x_i)}[\epsilon(x_i|x_0)]\right)\right)$.
  - $\mathrm{diag}\left(\mathrm{Cov}_{q_{\tilde{\sigma}}(x_0|x_i)}[\epsilon(x_i|x_0)]\right) = \underbrace{\mathbb{E}_{q_{\tilde{\sigma}}(x_0|x_i)}[\epsilon(x_i|x_0)^2]}_{\text{Train another model } h_{\theta,i}(x_i) \text{ for this: } \min_\theta \mathbb{E}_i \mathbb{E}_{x_0} \mathbb{E}_{x_i|x_0} \left\| h_{\theta,i}(x_i) - \epsilon(x_i|x_0)^2 \right\|^2.} - \underbrace{\mathbb{E}_{q_{\tilde{\sigma}}(x_0|x_i)}[\epsilon(x_i|x_0)]^2}_{\text{Estimated by DDPM } \epsilon_{\theta,i}(x_i)^2.}$.
  - Error in $\epsilon_{\theta,i}(x_i)$ is amplified in estimating $\boldsymbol{\gamma}_i^{*2}(x_i)$: it is squared.
    - Use a third model $g_{\phi,i}(x_i)$ to estimate $\left(\epsilon_{\theta,i}(x_i) - \epsilon(x_i|x_0)\right)^2$. Error not amplified.

DDPM
- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:
- **Diffusion process**
- **VP SDE: Cont.-time DDPM**
- **Training**

Interlude: Score Matching
- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time improvements
- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood
- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Continuous-Time Diffusion Process

- Diffusion Process

$i \in \{0, \dots, N\}$ Let $N \to \infty$:	⬄	$t := i\frac{T}{N} \in [0, T]$.
x_i	⬄	$x_t := x_{i=Nt/T}$
$x_{i+1} = x_i + f_i(x_i)$	⬄	ODE: **Flow**, (deterministic) Dynamics $\mathrm{d}x_t = f_t(x_t)\,\mathrm{d}t$, where $f_t := (N/T) f_{i=Nt/T}$.
$x_{t+h} \sim \mathcal{N}(x_t, hI)$, or $x_{t+h} = x_t + \sqrt{h}\,\epsilon$	⬄	Standard **Brownian motion** (Wiener process) $\mathrm{d}x_t = \mathrm{d}B_t$.
$x_{i+1} \sim \mathcal{N}\left(x_i + f_i(x_i), g_i^2 I\right)$, or $x_{i+1} = x_i + f_i(x_i) + g_i \epsilon$	⬄	SDE: **Diffusion process** (Itô process, No-jump Markov process) $\mathrm{d}x_t = f_t(x_t)\,\mathrm{d}t + g_t\,\mathrm{d}B_t$, where $g_t := \sqrt{N/T}\, g_{i=Nt/T}$.

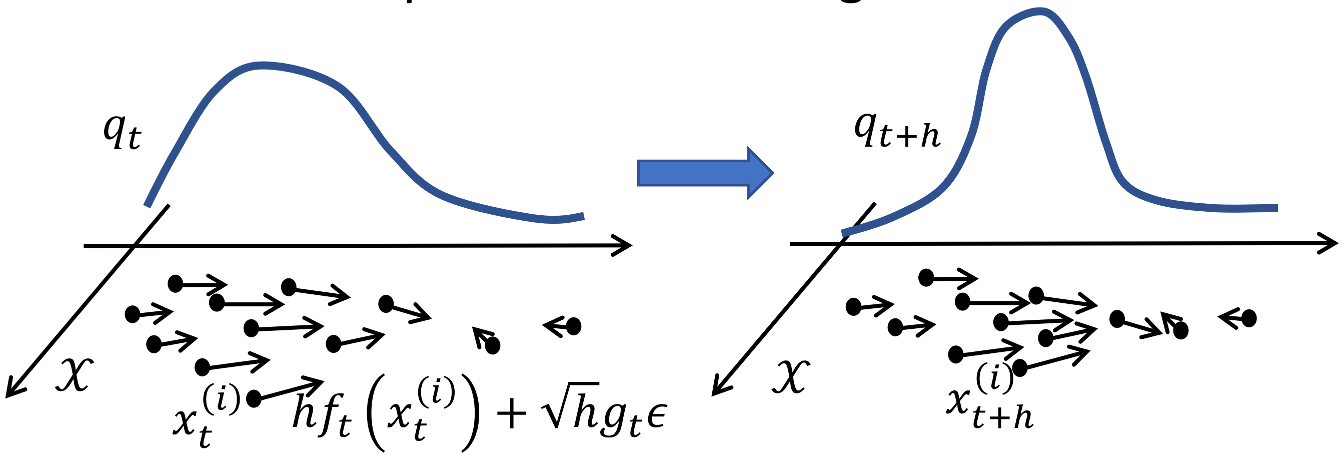
Continuous-Time Diffusion Process

- Diffusion Process and Distribution Evolution/Path

Under the diffusion process

$$\mathrm{d}x_t = f_t(x_t)\,\mathrm{d}t + g_t\,\mathrm{d}B_t,$$

distribution of particles is evolving:



Fokker-Planck Equation (Kolmogorov forward equation):

$$\partial_t q_t = -\nabla \cdot (q_t f_t) + \frac{g_t^2}{2}\nabla^2 q_t.$$

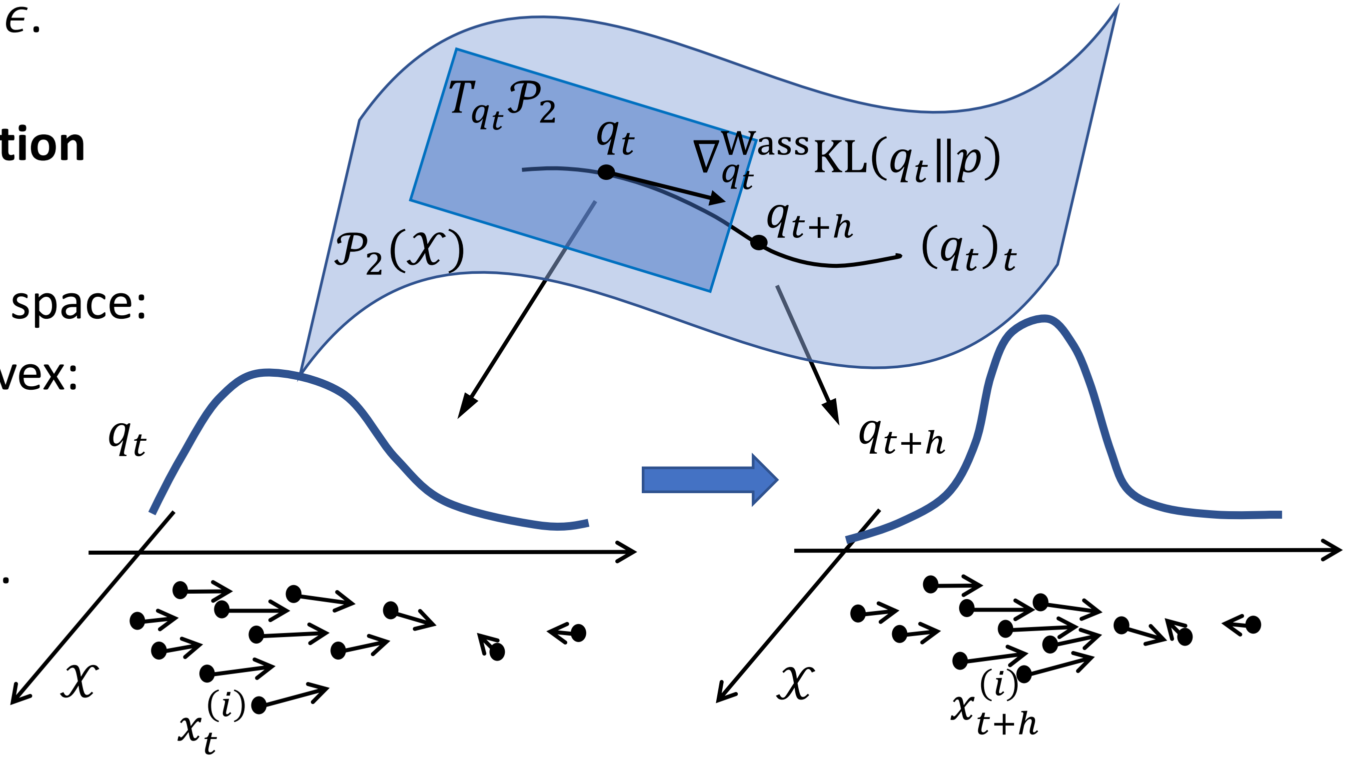
Liu, C., & Zhu, J. (2022). Geometry in sampling methods: A review on manifold MCMC and particle-based variational inference methods. *Advancements in Bayesian Methods and Implementations*, 47, 239.

Continuous-Time Diffusion Process

- Langevin Dynamics: A common diffusion process.

$$\mathrm{d}x_t = \nabla \log p(x_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t.$$

  - When $q_t = p$, we have FPE $\partial_t q_t = 0$: keeps p stationary.
  - Simulation: $x_{t+h} \leftarrow x_t + h \nabla \log p(x_t) + \sqrt{2h}\,\epsilon$.
  - Only requires an **unnormalized density function / energy function** of p: $\nabla \log p(x) = \nabla \log \frac{\tilde{p}(x)}{Z} = \nabla \log \tilde{p}(x)$.
  - Gradient flow of $\mathrm{KL}(\cdot\,\|p)$ on the Wasserstein space:
    - Exponential convergence if $\mathrm{KL}(\cdot\,\|p)$ is convex: e.g., when p is log-concave.
  - Riemannian-manifold version: $\mathrm{d}x_t = G^{-1}\nabla \log p\,\mathrm{d}t + \nabla \cdot G^{-1} + \sqrt{2G^{-1}}\,\mathrm{d}B_t$.



Liu, C., & Zhu, J. (2022). Geometry in sampling methods: A review on manifold MCMC and particle-based variational inference methods. *Advancements in Bayesian Methods and Implementations*, 47, 239.

Continuous-Time Diffusion Process

- Equivalent Flow of a Diffusion Process

Dynamics | **Distribution Evolution (FPE)**

Diffusion Process $\mathrm{d}x_t = f_t(x_t)\,\mathrm{d}t + g_t\,\mathrm{d}B_t$. $\Rightarrow$ $\partial_t q_t = -\nabla\cdot(q_t f_t) + \frac{g_t^2}{2}\nabla^2 q_t$.

$=$

Equivalent Flow $\mathrm{d}x_t = \left(f_t(x_t) - \frac{g_t^2}{2}\nabla\log q_t(x_t)\right)\mathrm{d}t$. $\Rightarrow$ $\partial_t q_t = -\nabla\cdot\left(q_t\left(f_t - \frac{g_t^2}{2}\nabla\log q_t\right)\right)$.

- Langevin dynamics $\mathrm{d}x_t = \nabla\log p(x_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$

➔ $\mathrm{d}x_t = \nabla\log p(x_t)\,\mathrm{d}t - \nabla\log q_t(x_t)\,\mathrm{d}t$

Particle-Based Variational Inference

- Blob [CZW+18]: $\nabla\log p\left(x_t^{(j)}\right) - \left(\sum_j \nabla_{x^{(i)}} K_{ij}\right)/\left(\sum_k K_{ik}\right) - \sum_j\left(\nabla_{x^{(i)}} K_{ij}\right)/\sum_k K_{jk}$.
- Gradient Flow with Smoothed Density / Function [LZC+19]: $\nabla\log p\left(x_t^{(j)}\right) + \begin{cases} -\left(\sum_j \nabla_{x^{(i)}} K_{ij}\right)/\left(\sum_k K_{ik}\right) \\ \sum_{j,k}(K^{-1})_{ik}\nabla_{x^{(j)}} K_{kj} \end{cases}$.
- Stein Variational Gradient Descent [LW16]: $x_{t+h}^{(i)} \leftarrow x_t^{(i)} + h\left[\sum_j K_{ij}\nabla\log p\left(x_t^{(j)}\right) + \sum_j \nabla_{x_t^{(j)}} K_{ij}\right]$.

Liu, C., & Zhu, J. (2022). Geometry in sampling methods: A review on manifold MCMC and particle-based variational inference methods. *Advancements in Bayesian Methods and Implementations*, 47, 239.

VP SDE: Continuous-Time DDPM [SSK+21]

- Diffusion-Process Interpretation of DDPM:

$i \in \{0, \dots, N\}$ ⇔ $t := i\frac{T}{N} \in [0, T]$.

Let $N \to \infty$:

x_i ⇔ $x_t := x_{i=Nt/T}$

DDPM: | Variance-Preserving SDE:

$x_i = \sqrt{1-\beta_i}\, x_{i-1} + \sqrt{\beta_i}\epsilon_i,$ ⇔ $\mathrm{d}x_t = -\frac{\beta_t}{2} x_t \,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}B_t,$ $\quad t \in [0, T]$.

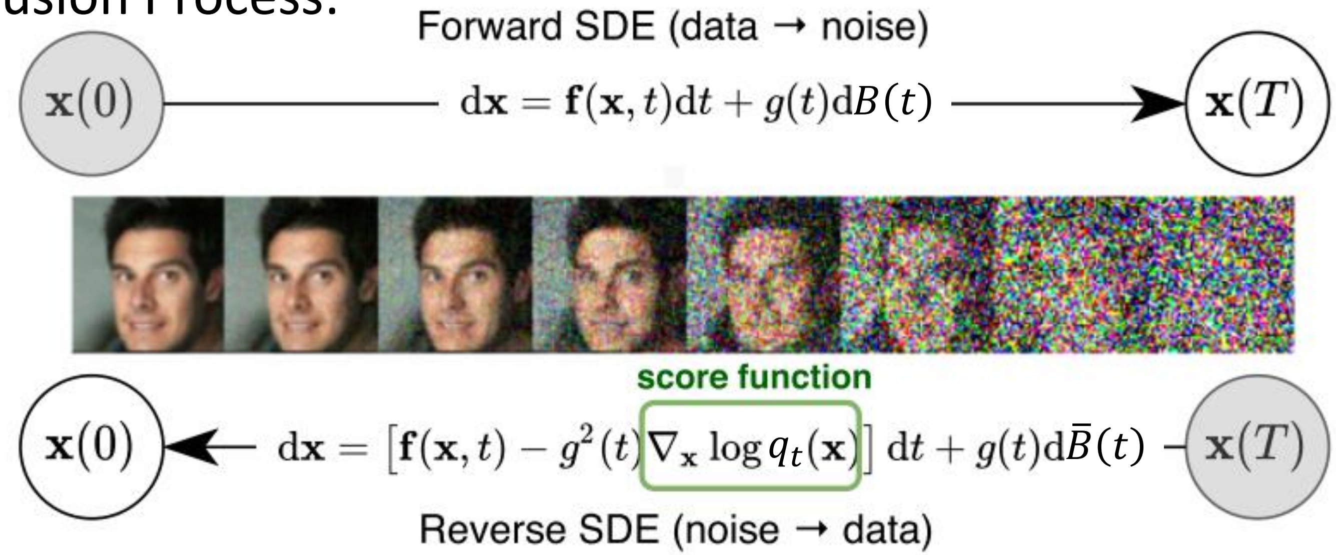
β_i ⇔ $\beta_t := (N/T)\beta_{i=Nt/T}$.

- Variance-Preserving: $\Sigma_{q_t} = I + e^{-\int_0^t \beta_s \,\mathrm{d}s}(\Sigma_{q_0} - I) \equiv I$ if $\Sigma_{q_0} = I$.
- Understanding VP SDE:
  - Langevin dynamics targeting $\mathcal{N}(0, I)$: $\quad \mathrm{d}x_t = \nabla \log \mathcal{N}(x_t|0, I)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t = -x_t + \sqrt{2}\,\mathrm{d}B_t$.
  - Time dilation $\mathrm{d}t \to \frac{\beta_t}{2}\mathrm{d}t$ [WWJ16]: $\quad \mathrm{d}x_t = -\frac{\beta_t}{2} x_t \,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}B_t$ (or, take $G^{-1} = \frac{\beta_t}{2} I$).

    Exponential convergence on $[0, \infty]$ ➔ Convergence on $[0, T]$.

VP SDE: Continuous-Time DDPM [SSK+21]

- Reverse Diffusion Process:



$\bar{B}_t$: reverse Brownian motion. In reverse time $\bar{t} := T - t$,

$\mathrm{d}x_t = \tilde{f}_t(x_t)\,\mathrm{d}t + g_t\,\mathrm{d}\bar{B}_t \quad \Leftrightarrow \quad \mathrm{d}x_{T-\bar{t}} = -\tilde{f}_{T-\bar{t}}(x_{T-\bar{t}})\,\mathrm{d}\bar{t} + g_{T-\bar{t}}\,\mathrm{d}B_{\bar{t}},$

$x_{t-h} = x_t - h\tilde{f}_t(x_t) + \sqrt{h}\,g_t\epsilon.$

  - Reverse process (generation):
    Only need a score model $s_{\theta,t}(x)$ targeting the **score function** $\nabla \log q_t(x)$.

- Learning: $\min_\theta \mathbb{E}_{q_t(x)} \left\| s_{\theta,t}(x) - \nabla \log q_t(x) \right\|^2$ for every $t \in [0, T]$.

But $\nabla \log q(x)$ is unknown!
Only data from $q(x)$ available.

DDPM
- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:
- Diffusion process
- VP SDE: Cont.-time DDPM
- Training
- VE SDE: cont.-time NCSN

Interlude: Score Matching
- **Denoising score-matching**
- **NCSN**

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Cont.-time improvements
- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood
- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Interlude: Score Matching

This is $\left\|\nabla^{\mathrm{Wass}}_{q_t} \mathrm{KL}(q_t \| p)\right\|^2$.

- Learn a **score model** $s_\theta(x)$ that targets the data score function $\nabla \log q(x)$:

$$\min_\theta \underbrace{\mathbb{E}_{q(x)} \|s_\theta(x) - \nabla \log q(x)\|^2}_{=: D_{\mathrm{Fisher}}(q(x) \| p_\theta(x))}.$$

But $\nabla \log q(x)$ is unknown!
Only data from $q(x)$ available.

  - Alternative way to learning an energy-based model. Data generation by Langevin dynamics using $s_\theta(x)$.
  - If data follows Boltzmann distribution $q(x) \propto e^{-E(x)}$, then $s_\theta(x)$ learns the **force field** $-\nabla E(x)$!
- Score Matching [Hyv05]:

$$D_{\mathrm{Fisher}}(q(x) \| p_\theta(x)) = \mathbb{E}_{q(x)} \|s_\theta(x)\|^2 - 2\mathbb{E}_{q(x)}[s_\theta(x) \cdot \nabla \log q(x)] + \mathbb{E}_{q(x)} \|\nabla \log q(x)\|^2,$$

$$\mathbb{E}_{q(x)}[s_\theta(x) \cdot \nabla \log q(x)] = \int_{\mathcal{X}} s_\theta(x) \cdot \nabla q(x) \,\mathrm{d}x = \int_{\mathcal{X}} \nabla \cdot \big(q(x) s_\theta(x)\big) \,\mathrm{d}x - \int_{\mathcal{X}} q(x) \nabla \cdot s_\theta(x) \,\mathrm{d}x,$$

$\int_{\mathcal{X}} \nabla \cdot \big(q(x) s_\theta(x)\big) \,\mathrm{d}x = \oint_{\partial \mathcal{X}} q(x) s_\theta(x) \cdot \mathrm{d}\vec{S} = 0$, if $s_\theta \in L^2(\mathcal{X})$, $q(x) \in H_0^1(\mathcal{X})$ (compactly supp. 1-Sobolev fns.).

$$\Rightarrow D_{\mathrm{Fisher}}(q(x) \| p_\theta(x)) = \underbrace{\mathbb{E}_{q(x)}[\|s_\theta(x)\|^2 + 2\nabla \cdot s_\theta(x)]}_{=: D_{\mathrm{SM}}(q(x) \| p_\theta(x))} + \mathbb{E}_{q(x)} \|\nabla \log q(x)\|^2,$$

Only requires data from $q(x)$!

$$\operatorname*{argmin}_\theta D_{\mathrm{Fisher}}(q(x) \| p_\theta(x)) = \operatorname*{argmin}_\theta D_{\mathrm{SM}}(q(x) \| p_\theta(x)).$$

Interlude: Score Matching

When data distributes on a low-dimensional manifold in $\mathcal{X}$, $\nabla_x \log q(x)$ is ill-defined.

➔ Consider $q_\sigma(\tilde{x}) := \int q(x)\, q_\sigma(\tilde{x}|x)\, \mathrm{d}x$, where $q_\sigma(\tilde{x}|x)$ is typically $\mathcal{N}\left(\tilde{x} \middle| x, \sigma^2 I_{\dim(\mathcal{X})}\right)$.

- Score Matching [Hyv05]:

$$\underbrace{\mathbb{E}_{q_\sigma(\tilde{x})}\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x})\|^2}_{D_{\text{Fisher}}(q_\sigma \| p_\theta)} = \underbrace{\mathbb{E}_{q_\sigma(\tilde{x})}[\|s_\theta(\tilde{x})\|^2 + 2\nabla_{\tilde{x}} \cdot s_\theta(\tilde{x})]}_{D_{\text{SM}}(q_\sigma \| p_\theta)} + \mathbb{E}_{q_\sigma(\tilde{x})}\|\nabla_{\tilde{x}} \log q_\sigma(\tilde{x})\|^2.$$

- Denoising Score Matching [Vin11]:

$D_{\text{Fisher}}(q_\sigma \| p_\theta) = \mathbb{E}_{q_\sigma(\tilde{x})}\|s_\theta(\tilde{x})\|^2 - 2\mathbb{E}_{q_\sigma(\tilde{x})}[s_\theta(\tilde{x}) \cdot \nabla_{\tilde{x}} \log q_\sigma(\tilde{x})] + \text{const.}$

Fisher identity: $\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}) = \int \frac{1}{q_\sigma(\tilde{x})} \nabla_{\tilde{x}} q_\sigma(x, \tilde{x})\, \mathrm{d}x = \int q_\sigma(x|\tilde{x}) \nabla_{\tilde{x}} \log q_\sigma(x, \tilde{x})\, \mathrm{d}x = \mathbb{E}_{q_\sigma(x|\tilde{x})}[\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)]$,

so 2nd term = $\mathbb{E}_{q_\sigma(\tilde{x})}\left[s_\theta(\tilde{x}) \cdot \mathbb{E}_{q_\sigma(x|\tilde{x})}[\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)]\right] = \mathbb{E}_{q_\sigma(\tilde{x}) q_\sigma(x|\tilde{x})}[s_\theta(\tilde{x}) \cdot \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)] = \mathbb{E}_{q(x) q_\sigma(\tilde{x}|x)}[s_\theta(\tilde{x}) \cdot \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)]$:

➔ Introduce $D_{\text{DSM}_\sigma}(q \| p_\theta) := \mathbb{E}_{q(x)} \mathbb{E}_{q_\sigma(\tilde{x}|x)} \|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)\|^2$.

➔ $D_{\text{DSM}_\sigma}(q \| p_\theta) = D_{\text{SM}}(q_\sigma \| p_\theta) + \mathbb{E}_{q(x)} \mathbb{E}_{q_\sigma(\tilde{x}|x)} \|\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)\|^2$, if $s_\theta \in L^2(\mathcal{X})$, $q_\sigma(\tilde{x}|x) \in H_0^1(\mathcal{X})$ for q-a.e. x.

➔ $\operatorname*{argmin}_\theta D_{\text{Fisher}}(q_\sigma \| p_\theta) = \operatorname*{argmin}_\theta D_{\text{SM}}(q_\sigma \| p_\theta) = \operatorname*{argmin}_\theta D_{\text{DSM}_\sigma}(q \| p_\theta)$.

Interlude: Score Matching

- Why called "denoising":
  - For Gaussian $p_\sigma(\tilde{x}|x)$: $\tilde{x} = x + \sigma\epsilon$, $\epsilon \sim p(\epsilon)$,

    $D_{\mathrm{DSM}_\sigma}(q \| p_\theta) = \mathbb{E}_{q(x)} \mathbb{E}_{p_\sigma(\tilde{x}|x)} \left\| s_\theta(\tilde{x}) + \frac{\tilde{x}-x}{\sigma^2} \right\|^2 = \mathbb{E}_{q(x)} \mathbb{E}_{p(\epsilon)} \left\| s_\theta(x + \sigma\epsilon) + \frac{\epsilon}{\sigma} \right\|^2.$
    - $s_\theta(\tilde{x})$ targets $-\frac{\epsilon}{\sigma}$ ➔ Noise-predicting model $\epsilon_\theta(\tilde{x}) = -\sigma s_\theta(\tilde{x})$!
  - Connection to Denoising Auto-Encoder:
    - Auto-Encoder: $\min_\theta \mathbb{E}_{q(x)} \left\| \mathrm{dec}_\theta\left(\mathrm{enc}_\theta(x)\right) - x \right\|^2$.
    - Denoising Auto-Encoder [VLBM08]:

    $\mathbb{E}_{q(x)} \mathbb{E}_{p_\sigma(\tilde{x}|x)} \left\| \mathrm{dec}_\theta\left(\mathrm{enc}_\theta(\tilde{x})\right) - x \right\|^2 = \sigma^4 \mathbb{E}_{q(x)} \mathbb{E}_{p_\sigma(\tilde{x}|x)} \left\| \frac{\mathrm{dec}_\theta(\mathrm{enc}_\theta(\tilde{x})) - \tilde{x}}{\sigma^2} + \frac{\tilde{x}-x}{\sigma^2} \right\|^2$:

    ➔ $\frac{\mathrm{dec}_\theta(\mathrm{enc}_\theta(\tilde{x})) - \tilde{x}}{\sigma^2}$ ⇔ score model $s_\theta(\tilde{x})$! [Vin11, AB14].

    ➔ DAE has a generative modeling utility.

Interlude: Score Matching

Typically $\sigma_i = \sigma_{\min}(\sigma_{\max}/\sigma_{\min})^{\frac{i-1}{N-1}}$.

- Noise Conditional Score Networks (**NCSN**) [SE19]:
  - **Annealed** perturbation: $\sigma_{\max} = \sigma_1 > \cdots > \sigma_N = \sigma_{\min}$, s.t.
    - $q_{\sigma_{\max}}(x) \approx \mathcal{N}(x|0, \sigma_{\max}^2 I)$ to explore the sample space,
    - $q_{\sigma_{\min}}(x) \approx q(x)$ to approach to the data distribution.
  - Score model $s_\theta(x, \sigma)$: also depends on σ.
    - Extrapolates to $\nabla \log q(x) = \nabla \log q_0(x)$.
    - Allow **Annealed Langevin Dynamics**: Explore for all modes + Correct the shape near each.

      $x \leftarrow x + \alpha_i s_\theta(x, \sigma_i) + \sqrt{2\alpha_i}\epsilon_i$. ($\alpha_i \propto \sigma_i^2$ to fix SNR)
  - Learning: Denoising Score Matching for all steps.

    $$\min_\theta \frac{1}{N} \sum_{i=1}^N \lambda_i D_{\mathrm{DSM}_{\sigma_i}}(q \| p_\theta).$$
    - Choose $\lambda_i \propto 1/\mathbb{E}\left\|\nabla_{\tilde{x}} \log q_{\sigma_i}(\tilde{x}|x)\right\|^2 \propto \sigma_i^2$ to fix $\lambda_i D_{\mathrm{DSM}_{\sigma_i}}$ scale.

VP SDE: Continuous-Time DDPM [SSK+21]

- Learning: $\min_\theta \mathbb{E}_{q_t(x)} \left\| s_{\theta,t}(x) - \nabla \log q_t(x) \right\|^2$ for every $t \in [0, T]$.

  ➔ Denoising Score Matching for each step,

$$D_{\mathrm{DSM}}(\theta) \coloneqq \mathbb{E}_{\mathrm{U}(t|[0,1])} \left[\lambda_t \underbrace{\mathbb{E}_{q_0(x)} \mathbb{E}_{q_{t|0}(\tilde{x}|x)} \left\| s_{\theta,t}(\tilde{x}) - \nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x) \right\|^2}_{D_{\mathrm{DSM}_{q_{t|0}}}\left(q_0 \| \tilde{p}_{\theta,t}\right)} \right].$$

  - The noising distribution $q_{t|0}(\tilde{x}|x)$ is available and a Gaussian:

    $q_{t|0}(\tilde{x}|x) = \mathcal{N}(\tilde{x} \mid \varsigma_t x, (1 - \varsigma_t^2) I)$ ⇔ DDPM $q(x_i|x_0) = \mathcal{N}\left(\tilde{x} \mid \sqrt{\alpha_i} x, (1 - \alpha_i) I\right)$,

    $\varsigma_t \coloneqq e^{-\frac{1}{2} \int_0^t \beta_s \, \mathrm{d}s}$.

  - Choosing $\lambda_t \propto 1 / \mathbb{E} \left\| \nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x) \right\|^2$ ⇔ DDPM simple loss!

DDPM
- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:
- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching
- Denoising score-matching
- NCSN

- **VE SDE: cont.-time NCSN**

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Cont.-time improvements
- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood
- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

VE SDE: Continuous-Time NCSN [SSK+21]

- Diffusion-Process Interpretation of NCSN:

$i \in \{0, \dots, N\}$ ⇔ $t := i\frac{T}{N} \in [0, T]$.

Let $N \to \infty$:

x_i ⇔ $x_t := x_{i=Nt/T}$

NCSN:

$x_{i-1} \sim \mathcal{N}(x_0, \sigma_{i-1}^2 I), x_i \sim \mathcal{N}(x_0, \sigma_i^2 I)$

➔ $x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\epsilon_i$ ⇔

Variance-Exploding SDE:

$\mathrm{d}x_t = \sqrt{(\sigma_t^2)'}\,\mathrm{d}B_t, \qquad t \in [0, T]$.

σ_i ⇔ $\sigma_t := \sigma_{i=Nt/T}$.

- Variance-Exploding: $\Sigma_{q_t} = \sigma_t^2 I + (\Sigma_{q_0} - \sigma_0^2 I) \to \infty$ when $t \to \infty$.
- Understanding VE SDE: Time-dilated Brownian motion.

VE SDE: Continuous-Time NCSN [SSK+21]

- Learning: Denoising Score Matching for each step,

$$D_{\mathrm{DSM}}(\theta) \coloneqq \mathbb{E}_{\mathrm{U}(t|[0,1])}\left[\lambda_t \underbrace{\mathbb{E}_{q_0(x)} \mathbb{E}_{q_{t|0}(\tilde{x}|x)} \left\| s_{\theta,t}(\tilde{x}) - \nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x) \right\|^2}_{D_{\mathrm{DSM}_{q_{t|0}}}(q_0 \| \tilde{p}_{\theta,t})}\right].$$

  - The noising distribution $q_{t|0}(\tilde{x}|x)$ is available and a Gaussian:

    $q_{t|0}(\tilde{x}|x) = \mathcal{N}(\tilde{x} \mid x, (\sigma_t^2 - \sigma_0^2)I)$ $\Leftrightarrow$ NCSN $q_{\sigma_i}(\tilde{x}|x) = \mathcal{N}(\tilde{x} \mid x, \sigma_i^2 I)$.

  - Choosing $\lambda_t \propto 1/\mathbb{E}\left\|\nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x)\right\|^2$ $\Leftrightarrow$ NCSN loss!

Score Model and Noise-Predicting Model

- Learning: Denoising Score Matching for each step,

$$D_{\mathrm{DSM}}(\theta) \coloneqq \mathbb{E}_{\mathrm{U}(t|[0,1])}\Big[\lambda_t \underbrace{\mathbb{E}_{q_0(x)}\mathbb{E}_{q_{t|0}(\tilde{x}|x)}\big\|s_{\theta,t}(\tilde{x}) - \nabla_{\tilde{x}}\log q_{t|0}(\tilde{x}|x)\big\|^2}_{D_{\mathrm{DSM}_{q_{t|0}}}(q_0\|\tilde{p}_{\theta,t})}\Big].$$

- If $q_{t|0}(\tilde{x}|x) = \mathcal{N}(\tilde{x} \mid a_t x, \sigma_t^2 I)$, then $\lambda_t \propto 1/\mathbb{E}\big\|\nabla_{\tilde{x}}\log q_{t|0}(\tilde{x}|x)\big\|^2 \propto \sigma_t^2$, and

$$D_{\mathrm{DSM}}(\theta) \coloneqq \mathbb{E}_{\mathrm{U}(t|[0,1])}\left[\sigma_t^2 \mathbb{E}_{q_0(x)}\mathbb{E}_{p(\epsilon)}\left\|s_{\theta,t}(\tilde{x}) + \frac{\epsilon}{\sigma_t}\right\|^2\right] = \mathbb{E}_{\mathrm{U}(t|[0,1])}\left[\mathbb{E}_{q_0(x)}\mathbb{E}_{p(\epsilon)}\big\|\sigma_t s_{\theta,t}(\tilde{x}) + \epsilon\big\|^2\right].$$

➔ $-\sigma_t s_{\theta,t}(\tilde{x})$ predicts the "noise": $\epsilon_{\theta,t}(\tilde{x}) = -\sigma_t s_{\theta,t}(\tilde{x})$.

➔ $D_{\mathrm{DSM}}(\theta)$ weighs noise-predicting losses equally (sim. DDPM simple loss).

Relation between VP and VE

	VP		VE
SDE:	$\mathrm{d}x_t = -\frac{\beta_t}{2} x_t \,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}B_t$	$\Leftrightarrow$	$\mathrm{d}x_t = \sqrt{(\sigma_t^2)'}\,\mathrm{d}B_t$
$q(x_t \mid x_0)$:	$\mathcal{N}(x_t \mid \varsigma_t x_0, (1-\varsigma_t^2)I) = \mathcal{N}(x_t \mid \varsigma_t x_0, \varsigma_t^2 \upsilon_t^2 I),$ $\varsigma_t := e^{-\frac{1}{2}\int_0^t \beta_s \,\mathrm{d}s}, \upsilon_t^2 := \int_0^t \frac{\beta_s}{\varsigma_s^2}\,\mathrm{d}s.$	$\Leftrightarrow$	$\mathcal{N}(x_t \mid x_0, (\sigma_t^2 - \sigma_0^2)I)$
Relation:	$x_t^{\mathrm{VP}} = \frac{x_t^{\mathrm{VE}}}{\sqrt{\sigma_t^2 - \sigma_0^2}},$	$\Leftrightarrow$	$x_t^{\mathrm{VE}} = \frac{x_t^{\mathrm{VP}}}{\varsigma_t},$
	$\beta_t = \frac{(\sigma_t^2)'}{\sigma_t^2 - \sigma_0^2}.$	$\Leftrightarrow$	$\sigma_t^2 = \sigma_0^2 + \upsilon_t^2.$

DDPM

- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:

- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching

- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time improvements

- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood

- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Cont.-time techniques:

- **sub-VP SDE**
- **Reverse-process simulation**
- Classifier-guided generation
- Probability flow

New Diffusion Process: sub-VP [SSK+21]

sub-VP SDE: $\mathrm{d}x_t = -\frac{\beta_t}{2} x_t \,\mathrm{d}t + \sqrt{\beta_t(1-\varsigma_t^4)}\,\mathrm{d}B_t$. (recall $\varsigma_t := e^{-\frac{1}{2}\int_0^t \beta_s \,\mathrm{d}s}$)

- $\Sigma_{q_t}^{\text{sub-VP}} = (1-\varsigma_t^2)^2 I + \varsigma_t^2 \Sigma_{q_0}^{\text{sub-VP}}$.
  - $\Sigma_{q_t}^{\text{sub-VP}} \preceq \Sigma_{q_t}^{\text{VP}}$ if $\Sigma_{q_0}^{\text{sub-VP}} = \Sigma_{q_0}^{\text{VP}}$: hence the name.
  - $\lim_{t\to\infty} \Sigma_{q_t}^{\text{sub-VP}} = \lim_{t\to\infty} \Sigma_{q_t}^{\text{VP}} = I$ if $\lim_{t\to\infty} \int_0^t \beta_s \,\mathrm{d}s = \infty$, hence q_t converges to $\mathcal{N}(0, I)$.
- DSM training: $D_{\text{DSM}}(\theta) := \mathbb{E}_{\mathrm{U}(t|[0,1])}\left[\lambda_t \mathbb{E}_{q_0(x)} \mathbb{E}_{q_{t|0}(\tilde{x}|x)} \left\| s_{\theta,t}(\tilde{x}) - \nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x) \right\|^2\right]$.
  - The noising distribution $q_{t|0}(\tilde{x}|x) = \mathcal{N}(\tilde{x} \mid \varsigma_t x, (1-\varsigma_t^2)^2 I)$ is available and a Gaussian.

General SDE:

- DSM training: $D_{\text{DSM}}(\theta) := \mathbb{E}_{\mathrm{U}(t|[0,1])}\left[\lambda_t \mathbb{E}_{q_0(x)} \mathbb{E}_{q_{t|0}(\tilde{x}|x)} \left[\left\| s_{\theta,t}(\tilde{x}) \right\|^2 + 2\nabla_{\tilde{x}} \cdot s_{\theta,t}(\tilde{x})\right]\right] + \text{const}$.
  - No need of $q_{t|0}(\tilde{x}|x)$ density: Only need samples drawn by the forward process.
  - But drawing samples for $q_{t|0}(\tilde{x}|x)$ takes $O(t)$ cost.
  - And $\nabla_{\tilde{x}} \cdot s_{\theta,t}(\tilde{x})$ requires d backprops.
    - Sliced score matching $\nabla \cdot s = \mathbb{E}_{p(\epsilon)}[\epsilon^\top \nabla (s^\top \epsilon)]$ [SGSE19]: 1 backprop but noisy.

Reverse Process Simulation [SSK+21]

Forward SDE

$\mathrm{d}x_t = f_t(x_t)\,\mathrm{d}t + g_t\,\mathrm{d}B_t,$ $\Leftrightarrow$

Forward SDE discretization

$x_i = x_{i-1} + \Delta f_{i-1} + \Delta g_{i-1}\epsilon_{i-1},$ $\Leftrightarrow$

Reverse SDE

$\mathrm{d}x_t = [f(x_t) - g_t^2 \nabla \log q_t(x_t)]\,\mathrm{d}t + g_t\,\mathrm{d}\bar{B}_t.$

Reverse-diffusion sampler:

$x_{i-1} = x_i - \Delta f_i + \Delta {g_i}^2 s_{\theta,i}(x_i) + \Delta g_i \epsilon_i.$

- VP SDE discretization (DDPM):

  $x_i = \sqrt{1-\beta_{i-1}}x_{i-1} + \sqrt{\beta_{i-1}}\epsilon_{i-1},$ $\Leftrightarrow$ $x_{i-1} = \left(2 - \sqrt{1-\beta_i}\right)x_i + \beta_i s_{\theta,i}(x_i) + \sqrt{\beta_i}\epsilon_i.$

  Not DDPM reverse process: "Ancestral sampler" $x_{i-1} = \frac{1}{\sqrt{1-\beta_i}}\left(x_i + \beta_i s_{\theta,i}(x_i)\right) + \sqrt{\beta_i}\epsilon_i$ (differ by $O(\beta_i)$).

- VE SDE discretization (NSCN):

  $x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\epsilon_{i-1},$ $\Leftrightarrow$ $x_{i-1} = x_i + \left(\sigma_i^2 - \sigma_{i-1}^2\right)s_{\theta,i}(x_i) + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\epsilon_i.$

  Ancestral sampler: parameterization for ease of ELBO,

  $x_{i-1} = x_i + \left(\sigma_i^2 - \sigma_{i-1}^2\right)s_{\theta,i}(x_i) + \sqrt{\frac{\sigma_{i-1}^2\left(\sigma_i^2 - \sigma_{i-1}^2\right)}{\sigma_i^2}}\epsilon_i.$

  Not the NCSN sampler $x_{i-1} = x_i + \sigma_i^2 s_{\theta,i}(x_i) + \sqrt{2}\,\sigma_i \epsilon_i$: directly targets p_{i-1} instead of $p_{i-1|i}$.

Reverse Process Simulation [SSK+21]

Predictor-Corrector (PC) framework:

- **Predictor (P)**: Reverse SDE discretizer for $p_{i-1|i}$ (e.g., reverse-diffusion sampler, ancestral sampler).
- **Corrector (C)**: dynamics-based MCMC targeting p_{i-1} (e.g., Langevin dynamics):
  Enabled by the score model $s_{\theta,i-1}$ for p_{i-1}!
- Original NCSN: C only. Original DDPM: P only.

Algorithm 1 PC sampling (VE SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2 \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}_i' \leftarrow \mathbf{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2)\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_{i+1}, \sigma_{i+1})$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i' + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\,\mathbf{z}$
  for $j = 1$ **to** M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_i, \sigma_i) + \sqrt{2\epsilon_i}\,\mathbf{z}$
return $\mathbf{x}_0$

Algorithm 2 PC sampling (VP SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}_i' \leftarrow (2 - \sqrt{1 - \beta_{i+1}})\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_{i+1}, i+1)$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i' + \sqrt{\beta_{i+1}}\,\mathbf{z}$
  for $j = 1$ **to** M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}^*}(\mathbf{x}_i, i) + \sqrt{2\epsilon_i}\,\mathbf{z}$
return $\mathbf{x}_0$

Predictor

Corrector

P: reverse-diffusion sampler. C: Langevin dynamics. ($\cdot_i \to \cdot_t$, $z \to \epsilon$, $\epsilon \to h$)

DDPM
- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:
- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching
- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- **Classifier-guided generation**
- Probability flow

Cont.-time improvements
- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood
- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Classifier-Guided Generation [SSK+21]

If we additionally have a classifier $p_t(y|x)$, then we can do controlled generation:

	Target	Reverse process (data generation)
Unconditioned:	$q_t(x_t)$	$\begin{aligned} \mathrm{d}x_t &= \left[f_t(x_t) - g_t^2 \nabla_{x_t} \log q_t(x_t)\right]\mathrm{d}t + g_t\, \mathrm{d}\bar{B}_t \\ &\approx \left[f_t(x_t) - g_t^2 s_{\theta,t}(x_t)\right]\mathrm{d}t + g_t\, \mathrm{d}\bar{B}_t. \end{aligned}$
Conditioned:	$q_t(x_t\|y) \propto q_t(x_t)p_t(y\|x_t)$	$\begin{aligned} \mathrm{d}x_t &= \left[f_t(x_t) - g_t^2 \nabla_{x_t} \log q_t(x_t\|y)\right]\mathrm{d}t + g_t\, \mathrm{d}\bar{B}_t \\ &\approx \left[f_t(x_t) - g_t^2 \left(s_{\theta,t}(x_t) + \nabla_{x_t} \log p_t(y\|x_t)\right)\right]\mathrm{d}t + g_t\, \mathrm{d}\bar{B}_t. \end{aligned}$
Energy-Guided: [ZBLZ22]	$\tilde{q}_t(x_t) \propto q_t(x_t)e^{-E(x_t)}$	$\begin{aligned} \mathrm{d}x_t &= \left[f_t(x_t) - g_t^2 \nabla_{x_t} \log \tilde{q}_t(x_t)\right]\mathrm{d}t + g_t\, \mathrm{d}\bar{B}_t \\ &\approx \left[f_t(x_t) - g_t^2 \left(s_{\theta,t}(x_t) - \nabla E(x_t)\right)\right]\mathrm{d}t + g_t\, \mathrm{d}\bar{B}_t. \end{aligned}$

- Examples: class-conditional image generation, image imputation, image colorization.
- [DN21,LZB+22b]: more results.

DDPM
- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:
- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching
- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- **Probability flow**

Cont.-time improvements
- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood
- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Probability Flow [SSK+21]

Diffusion process (SDE)

$$\mathrm{d}x_t = f_t(x)\,\mathrm{d}t + g_t\,\mathrm{d}B_t.$$

$\Leftrightarrow$

Equivalent flow (ODE): Probability Flow.

$$\mathrm{d}x_t = \left(f_t(x_t) - \frac{g_t^2}{2}\nabla \log q_t(x_t)\right)\mathrm{d}t.$$

$=: \tilde{f}_t(x_t)$

- Same marginal q_t, different joint $q_{0:t}$.
- **Point-to-point** process: deterministic and invertible.
  - x_T is now a **representation** of x_0 (for e.g., manipulated generation).
  - Unique identifiable encoding:
    the map $x_0 \to x_T$ is uniquely determined by data $q_0(x)$, regardless of model.
- **Likelihood/Density** evaluation: when $\mathrm{d}x_t = \tilde{f}_t(x_t)\,\mathrm{d}t$, FPE → $\frac{\mathrm{d}}{\mathrm{d}t}\log q_t(x_t) = -\nabla \cdot \tilde{f}_t(x_t)$ →

$$\log q_0(x_0) = \log p_T(x_T) + \int_0^T \nabla \cdot \tilde{f}_{\theta,t}(x_t)\,\mathrm{d}t.$$

- v.s. ODE flow / continuous normalizing flow (CNF) models:
  DSM training decomposes the loss into each step i, effective for deep models.

Probability Flow [SSK+21]

Diffusion process (SDE)

$$\mathrm{d}x_t = f_t(x)\,\mathrm{d}t + g_t\,\mathrm{d}B_t. \quad \Leftrightarrow$$

Equivalent flow (ODE): Probability Flow.

$$\mathrm{d}x_t = \underbrace{\left(f_t(x_t) - \frac{g_t^2}{2}\nabla \log q_t(x_t)\right)}_{=:\,\tilde{f}_t(x_t)}\mathrm{d}t.$$

- Reverse process (data generation).

Forward SDE discretization | Reverse ODE (prob. flow) discretization

$$x_i = x_{i-1} + \Delta f_{i-1} + \Delta g_{i-1}\epsilon_{i-1}, \quad \Leftrightarrow \quad x_{i-1} = x_i - \Delta f_i + \frac{\Delta {g_i}^2}{2} s_{\theta,i}(x_i).$$

- VP SDE:

$$x_i = \sqrt{1-\beta_{i-1}}\,x_{i-1} + \sqrt{\beta_{i-1}}\,\epsilon_{i-1}, \quad \Leftrightarrow \quad x_{i-1} = \left(2 - \sqrt{1-\beta_i}\right)x_i + \frac{1}{2}\beta_i s_{\theta,i}(x_i).$$

- VE SDE:

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\,\epsilon_{i-1}, \quad \Leftrightarrow \quad x_{i-1} = x_i + \frac{1}{2}\left(\sigma_i^2 - \sigma_{i-1}^2\right)s_{\theta,i}(x_i).$$

- v.s. Reverse SDE simulation: Determinacy allows using larger step size [LZB+22].

Diffusion Model as Diffusion Process [SSK+21]

Forward process:

- NCSN: $x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\epsilon_{i-1}, \ \ \epsilon_{i-1} \sim \mathcal{N}(0, I).$
- DDPM: $x_i = \sqrt{1-\beta_{i-1}} x_{i-1} + \sqrt{\beta_{i-1}}\epsilon_{i-1}, \epsilon_{i-1} \sim \mathcal{N}(0, I).$

Reverse process:

- NCSN:

(rev. diff.) $x_{i-1} = x_i + \left(\sigma_i^2 - \sigma_{i-1}^2\right) s_{\theta,i}(x_i) + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\epsilon_i.$

(ances.) $x_{i-1} = x_i + \left(\sigma_i^2 - \sigma_{i-1}^2\right) s_{\theta,i}(x_i) + \sqrt{\frac{\sigma_{i-1}^2\left(\sigma_i^2 - \sigma_{i-1}^2\right)}{\sigma_i^2}}\epsilon_i.$

(prob. flow) $x_{i-1} = x_i + \frac{1}{2}\left(\sigma_i^2 - \sigma_{i-1}^2\right) s_{\theta,i}(x_i).$

- DDPM($\gamma_i^2 = \beta_i$):

(rev. diff.) $x_{i-1} = \left(2 - \sqrt{1-\beta_i}\right) x_i + \beta_i s_{\theta,i}(x_i) + \sqrt{\beta_i}\epsilon_i.$

(ances.) $x_{i-1} = \frac{1}{\sqrt{1-\beta_i}}\left(x_i + \beta_i s_{\theta,i}(x_i)\right) + \sqrt{\beta_i}\epsilon_i.$

(prob. flow) $x_{i-1} = \left(2 - \sqrt{1-\beta_i}\right) x_i + \frac{1}{2}\beta_i s_{\theta,i}(x_i).$

Loss:

- NCSN loss, DDPM simple loss

SDE/ODE:

$\Leftrightarrow \mathrm{d}x_t = \sqrt{(\sigma_t^2)'}\,\mathrm{d}B_t, \qquad t \in (0, T].$

$\Leftrightarrow \mathrm{d}x_t = -\frac{1}{2}\beta_t x_t\,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}B_t, \qquad t \in (0, T].$

$\Leftrightarrow \mathrm{d}x_t = -(\sigma_t^2)'\nabla \log q_t\,\mathrm{d}t + \sqrt{(\sigma_t^2)'}\,\mathrm{d}\bar{B}_t.$

$\Leftrightarrow \mathrm{d}x_t = -\frac{1}{2}(\sigma_t^2)'\nabla \log q_t.$

$\Leftrightarrow \mathrm{d}x_t = -\beta_t\left(\frac{x_t}{2} + \nabla \log q_t\right)\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}\bar{B}_t.$

$\Leftrightarrow \mathrm{d}x_t = -\frac{\beta_t}{2}(x_t + \nabla \log q_t)\mathrm{d}t.$

$\Leftrightarrow$ DSM $\mathbb{E}_t \lambda_t \mathbb{E}_{q_0(x) q_{t|0}(\tilde{x}|x)} \left\| s_{\theta,t}(\tilde{x}) - \nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x) \right\|^2.$

Diffusion Model as Diffusion Process

- Quantitative convergence result [DTHD21]:

Let the forward process be $\mathrm{d}x_t = -\alpha x_t \,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t$, $\alpha \geq 0$, *and discretization step size be* γ_t.

Theorem 1. *Assume that there exists* $\mathtt{M} \geq 0$ *such that for any* $t \in [0, T]$ *and* $x \in \mathbb{R}^d$

$$\| s_{\theta^\star}(t, x) - \nabla \log p_t(x) \| \leq \mathtt{M}, \tag{8}$$

with $s_{\theta^\star} \in \mathrm{C}([0,T] \times \mathbb{R}^d, \mathbb{R}^d)$. *Assume that* $p_{\text{data}} \in \mathrm{C}^3(\mathbb{R}^d, (0, +\infty))$ *is bounded and that there exist* $d_1, A_1, A_2, A_3 \geq 0$, $\beta_1, \beta_2, \beta_3 \in \mathbb{N}$ *and* $\mathtt{m}_1 > 0$ *such that for any* $x \in \mathbb{R}^d$ *and* $i \in \{1, 2, 3\}$

$$\|\nabla^i \log p_{\text{data}}(x)\| \leq A_i(1 + \|x\|^{\beta_i}), \quad \langle \nabla \log p_{\text{data}}(x), x \rangle \leq -\mathtt{m}_1 \|x\|^2 + d_1 \|x\|,$$

with $\beta_1 = 1$. *Then for any* $\alpha \geq 0$, *there exist* $B_\alpha, C_\alpha, D_\alpha \geq 0$ *such that for any* $N \in \mathbb{N}$ *and* $\{\gamma_k\}_{k=1}^N$ *with* $\gamma_k > 0$ *for any* $k \in \{1, \ldots, N\}$, *the following hold:*

(a) if $\alpha > 0$, *we have* $\|\mathcal{L}(X_0) - p_{\text{data}}\|_{\text{TV}} \leq \boxed{B_\alpha \exp[-\alpha^{1/2} T]} + \boxed{C_\alpha(\mathtt{M} + \bar{\gamma}^{1/2}) \exp[D_\alpha T]}$;

(b) if $\alpha = 0$, *we have* $\|\mathcal{L}(X_0) - p_{\text{data}}\|_{\text{TV}} \leq \boxed{B_0(T^{-1} + T^{-1/2})} + \boxed{C_0(\mathtt{M} + \bar{\gamma}^{1/2}) \exp[D_0 T]}$;

Due to the error between p_T and p_{prior}.

Due to discretization error.

where $T = \sum_{k=1}^N \gamma_k$, $\bar{\gamma} = \sup_{k \in \{1,\ldots,N\}} \gamma_k$ *and* $p(x_0)$ *is the distr. of* x_0 *from the discretized reverse process from* $p_{\text{prior}}(x_T)$.

DDPM
- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:
- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching
- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Cont.-time improvements
- **DPM-Solver**
- Elucidating the design of diffusion model

Cont.-time likelihood
- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Fast Reverse-Process Simulation (DPM-Solver) [LZB+22b]

- For fast simulation:
  - **Prob. ODE** is preferred: deterministic dynamics allows larger step size.
  - **Reverse SDE**: more robust to model error but the step size is limited by the randomness.
- Formulation:
  - For semi-linear ODE $\mathrm{d}x_t = a_t x_t \,\mathrm{d}t + h_t(x_t)\,\mathrm{d}t$,

    ➔ "Variation of constants" formula: $x_t = \varsigma_{t|s} x_s + \int_s^t \varsigma_{t|\tau} h_\tau(x_\tau)\,\mathrm{d}\tau$, where $\varsigma_{t|s} \coloneqq e^{\int_s^t a_\tau \,\mathrm{d}\tau}$.
  - Forward process: $q(x_i|x_0) = \mathcal{N}\left(x_i | \sqrt{\alpha_i} x_0, \sigma_i^2 I\right)$, and SNR $\xi_i \coloneqq \alpha_i / \sigma_i^2$ decreases in i [KSPH21].

    ➔ Forward SDE: $\mathrm{d}x_t = \frac{1}{2}(\log \alpha_t)' x_t \,\mathrm{d}t + \sigma_t \sqrt{-2\lambda_t'}\,\mathrm{d}B_t$, where $\lambda_t \coloneqq \frac{1}{2}\log \xi_t$.

    ➔ Prob. flow ODE: $\mathrm{d}x_t = \frac{1}{2}(\log \alpha_t)' x_t \,\mathrm{d}t - \sigma_t \lambda_t' \epsilon_t(x_t)\,\mathrm{d}t$.

    ➔ VoC formula: $x_t = \sqrt{\frac{\alpha_t}{\alpha_s}} x_s - \sqrt{\alpha_t} \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \epsilon_\lambda(x_\lambda)\,\mathrm{d}\lambda$.
    - Integrate w.r.t t → integrate w.r.t λ.
    - Exponentially weighted integral of ϵ_λ: kind of exponential integrators in ODE solvers.

Fast Reverse-Process Simulation (DPM-Solver) [LZB+22b]

- Implementation using VoC formula: $x_t = \sqrt{\frac{\alpha_t}{\alpha_s}} x_s - \sqrt{\alpha_t} \int_{\lambda_s}^{\lambda_t} e^{-\lambda} \epsilon_\lambda(x_\lambda)\, \mathrm{d}\lambda.$
  - Taylor-expand $\hat{\boldsymbol{\epsilon}}_\theta(\hat{\boldsymbol{x}}_\lambda, \lambda) = \sum_{n=0}^{k-1} \frac{(\lambda - \lambda_{t_{i-1}})^n}{n!} \hat{\boldsymbol{\epsilon}}_\theta^{(n)}(\hat{\boldsymbol{x}}_{\lambda_{t_{i-1}}}, \lambda_{t_{i-1}}) + \mathcal{O}((\lambda - \lambda_{t_{i-1}})^k),$

  and the integral becomes $\sum_{n=0}^{k-1} \hat{\boldsymbol{\epsilon}}_\theta^{(n)}(\hat{\boldsymbol{x}}_{\lambda_{t_{i-1}}}, \lambda_{t_{i-1}}) \underbrace{\int_{\lambda_{t_{i-1}}}^{\lambda_{t_i}} e^{-\lambda} \frac{(\lambda - \lambda_{t_{i-1}})^n}{n!} \mathrm{d}\lambda}_{\text{Analytically available!}} + \mathcal{O}(h_i^{k+1})$

Does not actually depend on $\hat{\boldsymbol{\epsilon}}_\theta^{(n)}$:

DPM-Solver-1. Recovers DDIM!

$$\tilde{\boldsymbol{x}}_{t_i} = \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{\boldsymbol{x}}_{t_{i-1}} - \sigma_{t_i}(e^{h_i} - 1)\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t_{i-1}}, t_{i-1}), \quad \text{where } h_i = \lambda_{t_i} - \lambda_{t_{i-1}}.$$

Algorithm 1 DPM-Solver-2.

Require: initial value $\boldsymbol{x}_T$, time steps $\{t_i\}_{i=0}^M$, model $\boldsymbol{\epsilon}_\theta$

$\tilde{\boldsymbol{x}}_{t_0} \leftarrow \boldsymbol{x}_T$

for $i \leftarrow 1$ to M **do**

$\quad s_i \leftarrow t_\lambda\left(\frac{\lambda_{t_{i-1}} + \lambda_{t_i}}{2}\right)$

$\quad \boldsymbol{u}_i \leftarrow \frac{\alpha_{s_i}}{\alpha_{t_{i-1}}} \tilde{\boldsymbol{x}}_{t_{i-1}} - \sigma_{s_i}\left(e^{\frac{h_i}{2}} - 1\right)\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t_{i-1}}, t_{i-1})$

$\quad \tilde{\boldsymbol{x}}_{t_i} \leftarrow \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{\boldsymbol{x}}_{t_{i-1}} - \sigma_{t_i}\left(e^{h_i} - 1\right)\boldsymbol{\epsilon}_\theta(\boldsymbol{u}_i, s_i)$

end for

return $\tilde{\boldsymbol{x}}_{t_M}$

Algorithm 2 DPM-Solver-3.

Require: initial value $\boldsymbol{x}_T$, time steps $\{t_i\}_{i=0}^M$, model $\boldsymbol{\epsilon}_\theta$

$\tilde{\boldsymbol{x}}_{t_0} \leftarrow \boldsymbol{x}_T, r_1 \leftarrow \frac{1}{3}, r_2 \leftarrow \frac{2}{3}$

for $i \leftarrow 1$ to M **do**

$\quad s_{2i-1} \leftarrow t_\lambda\left(\lambda_{t_{i-1}} + r_1 h_i\right), \quad s_{2i} \leftarrow t_\lambda\left(\lambda_{t_{i-1}} + r_2 h_i\right)$

$\quad \boldsymbol{u}_{2i-1} \leftarrow \frac{\alpha_{s_{2i-1}}}{\alpha_{t_{i-1}}} \tilde{\boldsymbol{x}}_{t_{i-1}} - \sigma_{s_{2i-1}}\left(e^{r_1 h_i} - 1\right)\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t_{i-1}}, t_{i-1})$

$\quad \boldsymbol{D}_{2i-1} \leftarrow \boldsymbol{\epsilon}_\theta(\boldsymbol{u}_{2i-1}, s_{2i-1}) - \boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t_{i-1}}, t_{i-1})$

$\quad \boldsymbol{u}_{2i} \leftarrow \frac{\alpha_{s_{2i}}}{\alpha_{t_{i-1}}} \tilde{\boldsymbol{x}}_{t_{i-1}} - \sigma_{s_{2i}}\left(e^{r_2 h_i} - 1\right)\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t_{i-1}}, t_{i-1}) - \frac{\sigma_{s_{2i}} r_2}{r_1}\left(\frac{e^{r_2 h_i} - 1}{r_2 h_i} - 1\right)\boldsymbol{D}_{2i-1}$

$\quad \boldsymbol{D}_{2i} \leftarrow \boldsymbol{\epsilon}_\theta(\boldsymbol{u}_{2i}, s_{2i}) - \boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t_{i-1}}, t_{i-1})$

$\quad \tilde{\boldsymbol{x}}_{t_i} \leftarrow \frac{\alpha_{t_i}}{\alpha_{t_{i-1}}} \tilde{\boldsymbol{x}}_{t_{i-1}} - \sigma_{t_i}\left(e^{h_i} - 1\right)\boldsymbol{\epsilon}_\theta(\tilde{\boldsymbol{x}}_{t_{i-1}}, t_{i-1}) - \frac{\sigma_{t_i}}{r_2}\left(\frac{e^{h_i} - 1}{h} - 1\right)\boldsymbol{D}_{2i}$

end for

return $\tilde{\boldsymbol{x}}_{t_M}$

Saliently better (in FID) than RK (same order, same step size). Error of the linear part ODE may increase exponentially.

DDPM
- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:
- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching
- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Cont.-time improvements
- DPM-Solver
- **Elucidating the design of diffusion model**

Cont.-time likelihood
- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Relation between VP and VE

	VP		VE
	VP	$\Leftrightarrow$	VE
SDE:	$\mathrm{d}x_t = -\frac{\beta_t}{2} x_t \,\mathrm{d}t + \sqrt{\beta_t}\,\mathrm{d}B_t$	$\Leftrightarrow$	$\mathrm{d}x_t = \sqrt{(\sigma_t^2)'}\,\mathrm{d}B_t$
$q(x_t \vert x_0)$:	$\mathcal{N}(x_t \vert \varsigma_t x_0, (1-\varsigma_t^2)I) = \mathcal{N}(x_t \vert \varsigma_t x_0, \varsigma_t^2 \upsilon_t^2 I),$	$\Leftrightarrow$	$\mathcal{N}(x_t \vert x_0, (\sigma_t^2 - \sigma_0^2)I)$
	$\varsigma_t := e^{-\frac{1}{2}\int_0^t \beta_s \,\mathrm{d}s}, \upsilon_t^2 := \int_0^t \frac{\beta_s}{\varsigma_s^2}\,\mathrm{d}s.$		
Relation:	$x_t^{\mathrm{VP}} = \frac{x_t^{\mathrm{VE}}}{\sqrt{\sigma_t^2 - \sigma_0^2}},$	$\Leftrightarrow$	$x_t^{\mathrm{VE}} = \frac{x_t^{\mathrm{VP}}}{\varsigma_t},$
	$\beta_t = \frac{(\sigma_t^2)'}{\sigma_t^2 - \sigma_0^2}.$	$\Leftrightarrow$	$\sigma_t^2 = \sigma_0^2 + \upsilon_t^2.$

Elucidating the Design of Diffusion Model [KAAL22]

	General affine-drift diffusion		**Time-dilated Brownian motion**
SDE:	$\mathrm{d}x_t = a_t x_t \,\mathrm{d}t + g_t \,\mathrm{d}B_t$	$\Leftrightarrow$	$\mathrm{d}\hat{x}_t = \sqrt{(v_t^2)'}\,\mathrm{d}B_t$
$q(x_t \vert x_0)$:	$\mathcal{N}(x_t \vert \varsigma_t x_0, \varsigma_t^2 v_t^2 I)$,	$\Leftrightarrow$	$\mathcal{N}(\hat{x}_t \vert \hat{x}_0, v_t^2 I)$
	$\varsigma_t := e^{\int_0^t a_s \,\mathrm{d}s}$, $v_t^2 := \int_0^t \frac{g_s^2}{\varsigma_s^2}\,\mathrm{d}s$.		
$q(x_t)$:	$q_t(\tilde{x}) = \varsigma_t^{-d} q_{v_t}(\tilde{x}/\varsigma_t)$,	$\Leftrightarrow$	$\hat{q}_t(\hat{x}_t) = q_{v_t}(\hat{x}_t)$
	$q_v(x) := \big(q_0 * \mathcal{N}(0, v^2 I)\big)(x)$.		
Relation:		$\Leftrightarrow$	$\hat{x}_t := \frac{x_t}{\varsigma_t}$.

Probabilistic flow: $\mathrm{d}\hat{x}_t = -\frac{(v_t^2)'}{2}\nabla_{\hat{x}_t} \log \hat{q}_t(\hat{x}_t)\,\mathrm{d}t = -\frac{1}{2}\nabla_{\hat{x}_t} \log \hat{q}_t(\hat{x}_t)\,\mathrm{d}v_t^2$.

➔ Every realization of prob. flow is a **reparam of the same ODE**! v_t reparams t, ς_t reparams x.

➔ So the generation process is largely **independent** of the model structure and training details.

➔ Design diffusion process by (ς_t, v_t) schedule in stead of (a_t, g_t).

		VP [42]	VE [42]	iDDPM [33] + DDIM [40]	Ours
Sampling (Section 3)					
Schedule	v_t	$\sqrt{e^{\frac{1}{2}\beta_\mathrm{d} t^2 + \beta_\mathrm{min} t} - 1}$	$\sqrt{t}$	t	t
Scaling	ς_t	$1/\sqrt{e^{\frac{1}{2}\beta_\mathrm{d} t^2 + \beta_\mathrm{min} t}}$	1	1	1

Elucidating the Design of Diffusion Model [KAAL22]

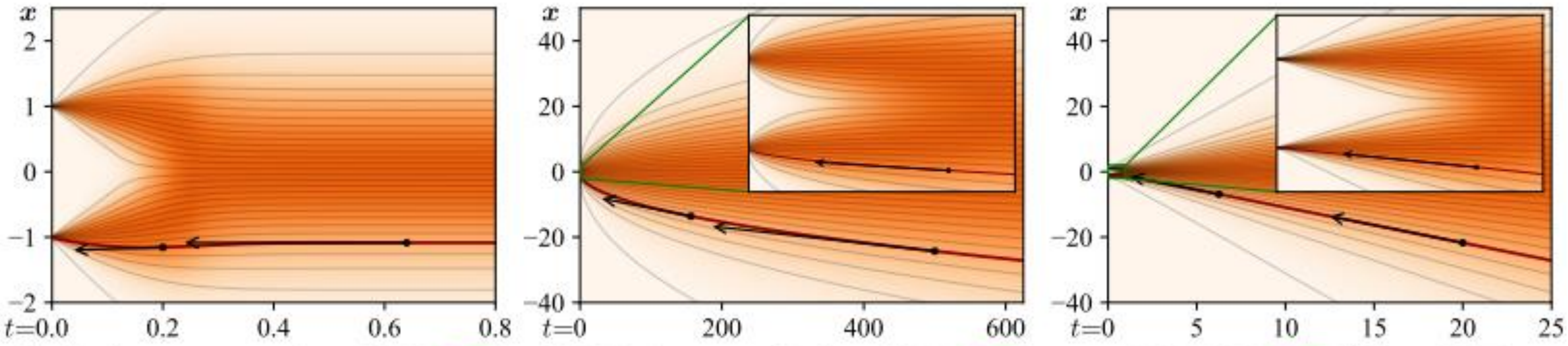
- Deterministic Sampling (Data Generation)
  - Model: Denoising Auto-Encoder framework: $\nabla \log q_{\upsilon_t}(x) \approx \frac{D_\theta(x;\upsilon_t)-x}{\upsilon_t^2}$.
  - Prob. flow: $\frac{\mathrm{d}x_t}{\mathrm{d}t} = \left(\frac{\varsigma_t'}{\varsigma_t} + \frac{\upsilon_t'}{\upsilon_t}\right) x_t - \varsigma_t \frac{\upsilon_t'}{\upsilon_t} D_\theta\left(\frac{x_t}{\varsigma_t}; \upsilon_t\right)$.
  - $(\varsigma_t, \upsilon_t)$ schedule: $\varsigma_t \equiv 1, \upsilon_t = t$.

  ➔ s.t. Prob. flow is $\frac{\mathrm{d}x_t}{\mathrm{d}t} = \frac{x_t - D_\theta(x_t;t)}{t}$:

  A single Euler step to $t = 0$, $x_0 = x_t - t\,\frac{x_t - D_\theta(x_t;t)}{t} = D_\theta(x_t;t)$, is the denoised image.

  ➔ Recovers DDIM [SME21].

$q_t(x)$ plot,
with $q_0 = \frac{1}{2}(\delta_1 + \delta_{-1})$



(a) Variance preserving ODE [42]
Local grad does not point to data.

(b) Variance exploding ODE [42]
Extreme curvature near data (large integrator error).

(c) DDIM [40] / Our ODE
Solution trajectories are lines pointing to the mean of data.

Elucidating the Design of Diffusion Model [KAAL22]

- Deterministic Sampling (Data Generation)
  - Simulating Prob. flow: $\frac{\mathrm{d}x_t}{\mathrm{d}t} = \left(\frac{\varsigma_t'}{\varsigma_t} + \frac{\upsilon_t'}{\upsilon_t}\right) x_t - \varsigma_t \frac{\upsilon_t'}{\upsilon_t} D_\theta\left(\frac{x_t}{\varsigma_t}; \upsilon_t\right)$:
    - RK45 not suitable: multiple D_θ evaluations outweighs its better order.
    - Leverage higher-order solver: Heun's 2nd-order ($O(\Delta t^3)$ local error) integrator.
    - Time steps: $|t_{i+1} - t_i|$ should decrease monotonically with decreasing υ_{t_i} (std of blurring Gaussian). E.g., choose t_i s.t. $\upsilon_{t_i} = \left(\upsilon_{\max}^{1/\rho} + \frac{i}{N-1}\left(\upsilon_{\min}^{1/\rho} - \upsilon_{\max}^{1/\rho}\right)\right)^\rho \mathbb{I}_{i<N} + 0\mathbb{I}_{i=N}$ (best $\rho = 7$).

Algorithm 1 Deterministic sampling using Heun's 2nd order method with arbitrary $\sigma(t)$ and $s(t)$.

1: **procedure** HeunSampler($D_\theta(\boldsymbol{x};\sigma)$, $\sigma(t)$, $s(t)$, $t_{i\in\{0,\dots,N\}}$)

2: **sample** $\boldsymbol{x}_0 \sim \mathcal{N}\left(\mathbf{0},\ \sigma^2(t_0)\, s^2(t_0)\, \mathbf{I}\right)$ ▷ Generate initial sample at t_0

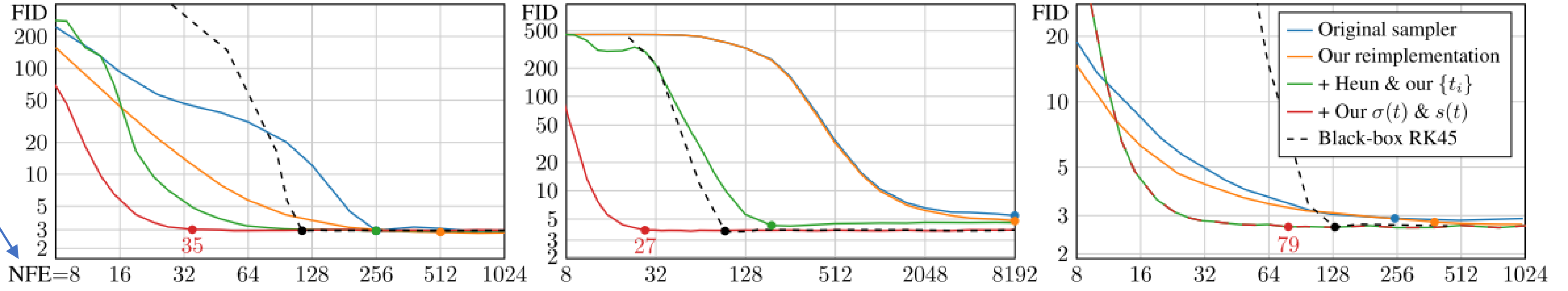
υ_t ς_t

i is reverted:

$t_0 = T$ is the

$t_N = 0$ is the

#{neural function eval.}



(a) Uncond. CIFAR-10, VP ODE (b) Uncond. CIFAR-10, VE ODE (c) Class-cond. ImageNet-64, DDIM

Elucidating the Design of Diffusion Model [KAAL22]

- Stochastic Sampling (Data Generation)
  - Generalized SDE "[19, 51]":

$$\mathrm{d}\boldsymbol{x}_{\pm} = \underbrace{-\,(v_t^2)'/2\ \nabla_{\boldsymbol{x}} \log\ q_{v_t}(x)\quad \mathrm{d}t}_{\text{probability flow ODE (Eq. 1) (take } \varsigma_t \equiv 1)} \pm \underbrace{\underbrace{\beta(t)\ v_t^2\quad \nabla_{\boldsymbol{x}} \log\ q_{v_t}(x)\quad \mathrm{d}t}_{\text{deterministic noise decay}} + \underbrace{\sqrt{2\beta(t)}\ v_t\ \mathrm{d}B_t}_{\text{noise injection}}}_{\text{Langevin diffusion SDE}}$$

+: forward.

-: reverse.

Predictor

Corrector

  - $\beta_t = v_t'/v_t$ ➔ Forward & reverse VE SDEs [SSK+21].
  - Oversaturated colors: score model $(D_\theta(x; v_t) - x)/v_t^2$ is **non-conservative**.

Algorithm 2 Our stochastic sampler with $v_t = t$ and $\varsigma_t = 1$ and $\beta_t = v_t'/v_t$ ➔ $\mathrm{d}x_t = 2\big(x_t - D_\theta(x_t; t)\big)/t\ \mathrm{d}t + \sqrt{2t}\ \mathrm{d}B_t$.

procedure StochasticSampler($D_\theta(\boldsymbol{x}; v)$, $t_{i\in\{0,\dots,N\}}$, $\gamma_{i\in\{0,\dots,N-1\}}$, S_{noise})
  sample $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0},\ t_0^2\ \mathbf{I})$
  for $i \in \{0, \dots, N-1\}$ **do** ▷ $\gamma_i = \begin{cases} \min\left(\frac{S_{\text{churn}}}{N}, \sqrt{2}-1\right) & \text{if } t_i \in [S_{\text{tmin}}, S_{\text{tmax}}] \\ 0 & \text{otherwise} \end{cases}$
    sample $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0},\ S_{\text{noise}}^2\ \mathbf{I})$
    $\hat{t}_i \leftarrow t_i + \gamma_i t_i$ ▷ Select temporarily increased noise level $\hat{t}_i$
    $\hat{\boldsymbol{x}}_i \leftarrow \boldsymbol{x}_i + \sqrt{\hat{t}_i^2 - t_i^2}\ \boldsymbol{\epsilon}_i$ ▷ Add new noise to move from t_i to $\hat{t}_i$
    $\boldsymbol{d}_i \leftarrow \big(\hat{\boldsymbol{x}}_i - D_\theta(\hat{\boldsymbol{x}}_i; \hat{t}_i)\big)/\hat{t}_i$ ▷ Evaluate $\mathrm{d}\boldsymbol{x}/\mathrm{d}t$ at $\hat{t}_i$
    $\boldsymbol{x}_{i+1} \leftarrow \hat{\boldsymbol{x}}_i + (t_{i+1} - \hat{t}_i)\boldsymbol{d}_i$ ▷ Take Euler step from $\hat{t}_i$ to t_{i+1}
    if $t_{i+1} \neq 0$ **then**
      $\boldsymbol{d}_i' \leftarrow \big(\boldsymbol{x}_{i+1} - D_\theta(\boldsymbol{x}_{i+1}; t_{i+1})\big)/t_{i+1}$ ▷ Apply 2nd order correction
      $\boldsymbol{x}_{i+1} \leftarrow \hat{\boldsymbol{x}}_i + (t_{i+1} - \hat{t}_i)\left(\tfrac{1}{2}\boldsymbol{d}_i + \tfrac{1}{2}\boldsymbol{d}_i'\right)$
  return $\boldsymbol{x}_N$

Only enable stochasticity within a range of noise level.

High-order discretization: Langevin churn γ_i for looking gradient ahead.

Elucidating the Design of Diffusion Model [KAAL22]

- Training
  - Score function model F_θ has a target with a less variant noise level than $D_\theta(x;\upsilon) = x - \upsilon F_\theta(x;\upsilon)$, but error occurs for large υ.
  - New formulation: $D_\theta(\boldsymbol{x};\upsilon) = c_{\text{skip}}(\upsilon)\,\boldsymbol{x} + c_{\text{out}}(\upsilon)\,F_\theta\big(c_{\text{in}}(\upsilon)\,\boldsymbol{x};\; c_{\text{noise}}(\upsilon)\big)$

  ➔ Training Loss = $\mathbb{E}_{\upsilon,\boldsymbol{y},\boldsymbol{n}}\Big[\underbrace{\lambda(\upsilon)\,c_{\text{out}}(\upsilon)^2}_{\text{effective weight}}\,\big\|\underbrace{F_\theta\big(c_{\text{in}}(\upsilon)\cdot(\boldsymbol{y}+\boldsymbol{n});c_{\text{noise}}(\upsilon)\big)}_{\text{network output}} - \underbrace{\tfrac{1}{c_{\text{out}}(\upsilon)}\big(\boldsymbol{y} - c_{\text{skip}}(\upsilon)\cdot(\boldsymbol{y}+\boldsymbol{n})\big)}_{\text{effective training target}}\big\|_2^2\Big].$

    - $c_{\text{in}}(\upsilon)$, $c_{\text{out}}(\upsilon)$: make F_θ's input & output have unit variance.
    - $c_{\text{skip}}(\upsilon)$: amplifying errors in F_θ as little as possible.
    - $\lambda(\upsilon)$=$1/c_{\text{out}}(\upsilon)^2$.
    - $\upsilon \sim p_{\text{train}}(\upsilon)$: log normal distribution.

Elucidating the Design of Diffusion Model [KAAL22]

- Summary

$D_\theta(\boldsymbol{x};\sigma) = c_{\text{skip}}(\sigma)\boldsymbol{x} + c_{\text{out}}(\sigma)F_\theta\big(c_{\text{in}}(\sigma)\boldsymbol{x}; c_{\text{noise}}(\sigma)\big)$; F_θ represents the raw neural network layers.

i is reverted:
$t_0 = T$ is the prior step,
$t_N = 0$ is the data step.

υ_t

ς_t

		VP [42]	VE [42]	iDDPM [33] + DDIM [40]	Ours
Sampling (Section 3)					
ODE solver		Euler	Euler	Euler	2nd order Heun
Time steps	$t_{i<N}$	$1+\frac{i}{N-1}(\epsilon_s-1)$	$\sigma_{\max}^2\left(\sigma_{\min}^2/\sigma_{\max}^2\right)^{\frac{i}{N-1}}$	$u_{\lfloor j_0+\frac{M-1-j_0}{N-1}i+\frac{1}{2}\rfloor}$, where $u_M=0$, $u_{j-1}=\sqrt{\frac{u_j^2+1}{\max(\bar{\alpha}_{j-1}/\bar{\alpha}_j, C_1)}-1}$	$\left(\sigma_{\max}{}^{\frac{1}{\rho}}+\frac{i}{N-1}(\sigma_{\min}{}^{\frac{1}{\rho}}-\sigma_{\max}{}^{\frac{1}{\rho}})\right)^\rho$
Schedule	$\sigma(t)$	$\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}-1}$	$\sqrt{t}$	t	t
Scaling	$s(t)$	$1/\sqrt{e^{\frac{1}{2}\beta_d t^2+\beta_{\min}t}}$	1	1	1
Network and preconditioning (Section 5)					
Architecture of F_θ		DDPM++	NCSN++	DDPM	(any)
Skip scaling	$c_{\text{skip}}(\sigma)$	1	1	1	$\sigma_{\text{data}}^2/\left(\sigma^2+\sigma_{\text{data}}^2\right)$
Output scaling	$c_{\text{out}}(\sigma)$	$-\sigma$	σ	$-\sigma$	$\sigma\cdot\sigma_{\text{data}}/\sqrt{\sigma_{\text{data}}^2+\sigma^2}$
Input scaling	$c_{\text{in}}(\sigma)$	$1/\sqrt{\sigma^2+1}$	1	$1/\sqrt{\sigma^2+1}$	$1/\sqrt{\sigma^2+\sigma_{\text{data}}^2}$
Noise cond.	$c_{\text{noise}}(\sigma)$	$(M-1)\,\sigma^{-1}(\sigma)$	$\ln(\frac{1}{2}\sigma)$	$M-1-\arg\min_j \lvert u_j-\sigma\rvert$	$\frac{1}{4}\ln(\sigma)$
Training (Section 5)					
Noise distribution		$\sigma^{-1}(\sigma)\sim\mathcal{U}(\epsilon_t,1)$	$\ln(\sigma)\sim\mathcal{U}(\ln(\sigma_{\min}), \ln(\sigma_{\max}))$	$\sigma=u_j,\ \ j\sim\mathcal{U}\{0,M-1\}$	$\ln(\sigma)\sim\mathcal{N}(P_{\text{mean}},P_{\text{std}}^2)$
Loss weighting	$\lambda(\sigma)$	$1/\sigma^2$	$1/\sigma^2$	$1/\sigma^2$ (note: *)	$\left(\sigma^2+\sigma_{\text{data}}^2\right)/(\sigma\cdot\sigma_{\text{data}})^2$
Parameters		$\beta_d=19.9, \beta_{\min}=0.1$ $\epsilon_s=10^{-3}, \epsilon_t=10^{-5}$ $M=1000$	$\sigma_{\min}=0.02$ $\sigma_{\max}=100$	$\bar{\alpha}_j=\sin^2(\frac{\pi}{2}\frac{j}{M(C_2+1)})$ $C_1=0.001, C_2=0.008$ $M=1000, j_0=8^\dagger$	$\sigma_{\min}=0.002, \sigma_{\max}=80$ $\sigma_{\text{data}}=0.5, \rho=7$ $P_{\text{mean}}=-1.2, P_{\text{std}}=1.2$

* iDDPM also employs a second loss term L_{vlb} † In our tests, $j_0=8$ yielded better FID than $j_0=0$ used by iDDPM

DDPM
- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:
- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

- VE SDE: cont.-time NCSN

Interlude: Score Matching
- Denoising score-matching
- NCSN

Cont.-time improvements
- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood
- $\boldsymbol{p_{\theta,t}^{\mathrm{SDE}}}$ **bound.**
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Diffusion Process and Data Likelihood [SDME21]

- Question:
  - DDPM simple loss ⇔ Weighted Denoising Score Matching,
    $D_{\mathrm{DSM}}(\theta; \lambda_{(\cdot)}) := \mathbb{E}_t \Big[\lambda_t \underbrace{\mathbb{E}_{q_0(x) q_{t|0}(\tilde{x}|x)} \big\| s_{\theta,t}(\tilde{x}) - \nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x) \big\|^2}_{D_{\mathrm{DSM}_{q_{t|0}}}(q_0 \| \tilde{p}_{\theta,t})}\Big]$, with $\lambda_t \propto 1/\mathbb{E}\big\|\nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x)\big\|^2$.
  - DDPM loss (ELBO) ⇔ ?
- Setup

Effective drift $\tilde{f}_t(x_t)$:

  - q_t: Forward SDE $\mathrm{d}x_t = f_t(x_t)\,\mathrm{d}t + g_t\,\mathrm{d}B_t,\ x_0 \sim q_0$ ➔ $f_t - \frac{g_t^2}{2} \nabla \log q_t$.
  - $p_{\theta,t}^{\mathrm{SDE}}$: Reverse SDE $\mathrm{d}x_t = f_t(x_t)\,\mathrm{d}t - g_t^2 s_{\theta,t}(x_t)\,\mathrm{d}t + g_t\,\mathrm{d}\bar{B}_t,\ x_T \sim p_T$ ➔ $f_t - g_t^2 s_{\theta,t} + \frac{g_t^2}{2} \nabla \log p_{\theta,t}^{\mathrm{SDE}}$.
  - $p_{\theta,t}^{\mathrm{ODE}}$: Reverse ODE $\mathrm{d}x_t = f_t(x_t)\,\mathrm{d}t - \frac{g_t^2}{2} s_{\theta,t}(x_t)\,\mathrm{d}t,\ x_T \sim p_T$ ➔ $f_t - \frac{g_t^2}{2} s_{\theta,t}$.
    - $\log p_{\theta,0}^{\mathrm{ODE}}(x_0) = \log p_T(x_T) + \int_0^T \nabla \cdot \tilde{f}_{\theta,t}(x_t)\,\mathrm{d}t$: too costly for optimization (no step-by-step loss).

Reverse SDE and Data Likelihood [SDME21]

- Results
  - Thm. 1. $\mathrm{KL}\left(q_0 \| p_{\theta,0}^{\mathrm{SDE}}\right) \leq D_{\mathrm{Fisher}}\left(\theta; \lambda_{(\cdot)} = g_{(\cdot)}^2/2\right) + \mathrm{KL}(q_T \| p_T)$ (under some regularity),

    "likelihood weighting"

    where $D_{\mathrm{Fisher}}\left(\theta; \lambda_{(\cdot)}\right) \coloneqq \mathbb{E}_t\left[\lambda_t \underbrace{\mathbb{E}_{q_t(\tilde{x})}\left\| s_{\theta,t}(\tilde{x}) - \nabla \log q_t(\tilde{x}) \right\|^2}_{D_{\mathrm{Fisher}}\left(q_t \| \tilde{p}_{\theta,t}\right) = D_{\mathrm{DSM}_{q_{t|0}}}\left(q_0 \| \tilde{p}_{\theta,t}\right) + \mathrm{C.}}\right]$.
  - Cor. 1. $-\mathbb{E}_{q_0}\left[\log p_{\theta,0}^{\mathrm{SDE}}\right] \leq D_{\mathrm{Fisher}}\left(\theta; g_{(\cdot)}^2/2\right) + \mathrm{C.} = D_{\mathrm{DSM}}\left(\theta; g_{(\cdot)}^2/2\right) + \mathrm{C.}$
  - Thm. 2. Assume $\exists \{r_t\}_t$ be led by the forward process from some r_0 s.t. $r_T = p_T$ and $s_{\theta,t} \equiv \nabla \log r_t$. Then $p_{\theta,0}^{\mathrm{SDE}} = p_{\theta,0}^{\mathrm{ODE}} = r_0$, and the equality holds: $\mathrm{KL}\left(q_0 \| p_{\theta,0}^{\mathrm{SDE}}\right) = D_{\mathrm{Fisher}}\left(\theta; g_{(\cdot)}^2/2\right) + \mathrm{KL}(q_T \| p_T)$.
    - Understand the condition: "self-consistency".
      $s_{\theta,t} = \nabla \log p_{\theta,t}^{\mathrm{SDE}} \iff s_{\theta,t} = \nabla \log p_{\theta,t}^{\mathrm{ODE}} \iff p_{\theta,t}^{\mathrm{SDE}} = p_{\theta,t}^{\mathrm{ODE}}$.

Reverse SDE and Data Likelihood [SDME21]

- Results
  - Thm. 3. $-\log p_{\theta,0}^{\mathrm{SDE}}(x) \leq \mathcal{L}_\theta^{\mathrm{Fisher}}(x) = \mathcal{L}_\theta^{\mathrm{DSM}}(x)$, where:
    $$\mathcal{L}_\theta^{\mathrm{Fisher}}(x) := -\mathbb{E}_{q_{T|0}(\tilde{x}|x)}[\log p_T(\tilde{x})] + \mathbb{E}_t \mathbb{E}_{q_{t|0}(\tilde{x}|x)} \left[\frac{g_t^2}{2} \left\| s_{\theta,t}(\tilde{x}) \right\|^2 + g_t^2 \nabla \cdot s_{\theta,t}(\tilde{x}) - \nabla \cdot f_t(\tilde{x})\right],$$
    $$\mathcal{L}_\theta^{\mathrm{DSM}}(x) := -\mathbb{E}_{q_{T|0}(\tilde{x}|x)}[\log p_T(\tilde{x})] + \mathbb{E}_t \left[\frac{g_t^2}{2} \mathbb{E}_{q_{t|0}(\tilde{x}|x)} \left\| s_{\theta,t}(\tilde{x}) - \nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x) \right\|^2\right]$$
    $$-\mathbb{E}_t \mathbb{E}_{q_{t|0}(\tilde{x}|x)} \left[\frac{g_t^2}{2} \left\| \nabla_{\tilde{x}} \log q_{t|0}(\tilde{x}|x) \right\|^2 + \nabla \cdot f_t(\tilde{x})\right].$$
    - Point-wise bound. **Allow estimating likelihood/density** for $p_{\theta,0}^{\mathrm{SDE}}$ (constant known).
    - **Continuous-time version of the DDPM loss (ELBO)!**
      - The weight of score-loss term $\frac{g_t^2}{2} \rightarrow \frac{\beta_i}{2}$ matches the DDPM loss weight $\frac{\beta_i^2}{2\sigma_i^2(1-\beta_i)}$ if adopting the analytic optimal reverse variance $\sigma_i^{*2} = \frac{\beta_i}{1-\beta_i}\left(1 - \frac{\beta_i}{d} \mathbb{E}_{q_{\tilde{\sigma}}(x_i)} \|\nabla \log q_{\tilde{\sigma}}(x_i)\|^2\right) \leq \frac{\beta_i}{1-\beta_i}$.

DDPM
- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:
- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching
- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time techniques:
- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Cont.-time improvements
- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood
- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $\boldsymbol{p_{\theta,t}^{ODE}}$ **bound.**

Schrödinger Bridge

Reverse Prob. Flow ODE and Data Likelihood [LZB+22a]

Taking $\lambda_{(\cdot)} = g_{(\cdot)}^2/2$

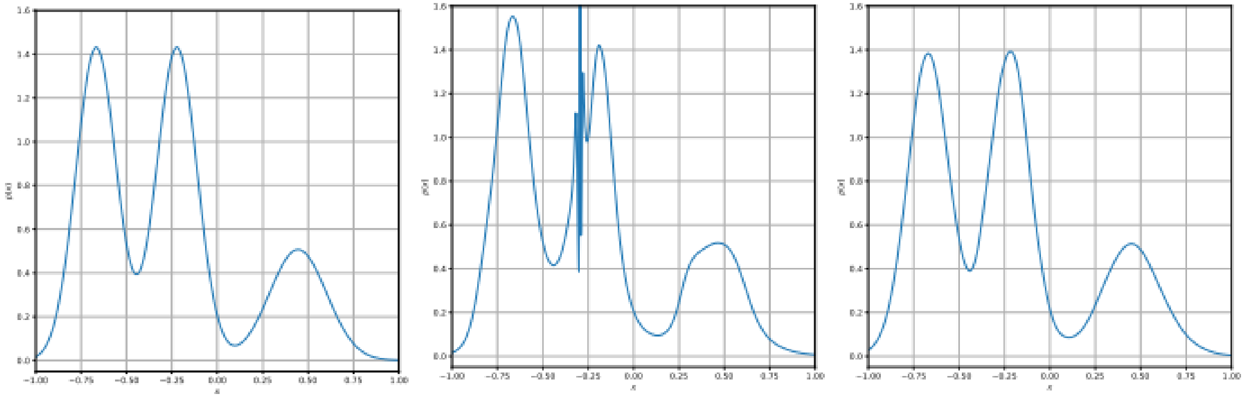
- Thm. 1. $\mathrm{KL}\big(q_0 \| p_{\theta,0}^{\mathrm{ODE}}\big) = D_{\mathrm{Fisher}}(\theta) + \mathrm{KL}(q_T \| p_T) + D_{\mathrm{diff}}(\theta) = D_{\mathrm{ODE}}(\theta) + \mathrm{KL}(q_T \| p_T)$,

  where $D_{\mathrm{diff}}(\theta) \coloneqq \mathbb{E}_t \frac{g_t^2}{2} \mathbb{E}_{q_t(x_t)} \left[\big(s_{\theta,t}(x_t) - \nabla \log q_t(x_t)\big)^\top \big(\nabla \log p_{\theta,t}^{\mathrm{ODE}}(x_t) - s_{\theta,t}(x_t)\big) \right]$,

  and $D_{\mathrm{ODE}}(\theta) \coloneqq \mathbb{E}_t \frac{g_t^2}{2} \mathbb{E}_{q_t(x_t)} \left[\big(s_{\theta,t}(x_t) - \nabla \log q_t(x_t)\big)^\top \big(\nabla \log p_{\theta,t}^{\mathrm{ODE}}(x_t) - \nabla \log q_t(x_t)\big) \right]$.

  - Minimizing $D_{\mathrm{Fisher}}(\theta)$ does not guarantee a good $p_{\theta,0}^{\mathrm{ODE}}$:

Needs ODE solver: costly.



(a) Data　(b) ScoreODE (first-order SM)　(c) ScoreODE (third-order SM)

Reverse Prob. Flow ODE and Data Likelihood [LZB+22a]

- Let $D^{\mathrm{ODE}}_{\mathrm{Fisher}}(\theta) := \mathbb{E}_t\left[\frac{g_t^2}{2}\underbrace{\mathbb{E}_{q_t(\tilde{x})}\left\|\nabla \log p^{\mathrm{ODE}}_{\theta,t}(x_t) - \nabla \log q_t(\tilde{x})\right\|^2}_{D_{\mathrm{Fisher}}\left(q_t \| p^{\mathrm{ODE}}_{\theta,t}\right)}\right]$.

  Cauchy-Schwarz ➔ $D_{\mathrm{ODE}}(\theta) \le \sqrt{D_{\mathrm{Fisher}}(\theta)}\sqrt{D^{\mathrm{ODE}}_{\mathrm{Fisher}}(\theta)}$:

  ➔ To learn $p^{\mathrm{ODE}}_{\theta,0}$, min. both $D_{\mathrm{Fisher}}(\theta)$ and $D^{\mathrm{ODE}}_{\mathrm{Fisher}}(\theta)$ ➔ Hard to estimate $D_{\mathrm{Fisher}}\left(q_t \| p^{\mathrm{ODE}}_{\theta,t}\right)$.
- Thm. 2.

$$\begin{cases} \left\|\nabla\nabla^\top \log p^{\mathrm{ODE}}_{\theta,t}(x_t)\right\|_2 \le C, \\ \left\|s_{\theta,t}(x_t) - \nabla \log q_t(x_t)\right\|_2 \le \delta_1, \\ \left\|\nabla s^\top_{\theta,t}(x_t) - \nabla\nabla^\top \log q_t(x_t)\right\|_F \le \delta_2, \\ \left\|\nabla \mathrm{tr}\left(\nabla s^\top_{\theta,t}(x_t)\right) - \nabla \mathrm{tr}(\nabla\nabla^\top \log q_t(x_t))\right\|_2 \le \delta_3, \end{cases} \quad \forall t, x_t \Longrightarrow D_{\mathrm{Fisher}}\left(q_t \| p^{\mathrm{ODE}}_{\theta,t}\right) \le U(t; \delta_1, \delta_2, \delta_3, C, q).$$

  U is strictly increasing with $\delta_1, \delta_2, \delta_3$ if $g_t \neq 0$.
  - $D_{\mathrm{Fisher}}(\theta)$ can also be bounded by δ_1:

    It suffices to match 1st – 3rd-order score functions to learn $p^{\mathrm{ODE}}_{\theta,0}$!

Reverse Prob. Flow ODE and Data Likelihood [LZB+22a]

- High-order denoising score matching:

  Iteratively leverage the known $q(x_t|x_0) = \mathcal{N}(x_t|\sqrt{\alpha_t}x_0, \sigma_t^2 I)$ as a noising distribution.

  - First-order: $\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\left\|\boldsymbol{s}_1(\boldsymbol{x}_t,t;\theta) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\right\|_2^2\right] \Rightarrow \theta^* = \operatorname*{argmin}_\theta \underbrace{\mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{\sigma_t^2}\left\|\sigma_t \boldsymbol{s}_1(\boldsymbol{x}_t,t;\theta) + \boldsymbol{\epsilon}\right\|_2^2\right]}_{D_{\mathrm{DSM}_{q_{t|0}}}(q_0 \| \tilde{p}_{\theta,t})}$

  - Second-order: with a good first-order model $\hat{s}_1$,

$$\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\left\|\boldsymbol{s}_2(\boldsymbol{x}_t,t;\theta) - \nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right\|_F^2\right] \Rightarrow \theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{\sigma_t^4}\left\|\sigma_t^2 \boldsymbol{s}_2(\boldsymbol{x}_t,t;\theta) + \boldsymbol{I} - \boldsymbol{\ell}_1\boldsymbol{\ell}_1^\top\right\|_F^2\right],$$

$$\boldsymbol{\ell}_1(\boldsymbol{\epsilon},\boldsymbol{x}_0,t) := \sigma_t \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t,t) + \boldsymbol{\epsilon}$$

  Effectiveness: $\left\|\boldsymbol{s}_2(\boldsymbol{x}_t,t;\theta) - \nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right\|_F$

$$\le \left\|\boldsymbol{s}_2(\boldsymbol{x}_t,t,\theta) - \boldsymbol{s}_2(\boldsymbol{x}_t,t;\theta^*)\right\|_F + \delta_1^2(\boldsymbol{x}_t,t). \quad \delta_1(\boldsymbol{x}_t,t) := \|\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t,t) - \nabla_{\boldsymbol{x}} \log q_t(\boldsymbol{x}_t)\|_2,$$

  - Laplacian (trace) version:

$$\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\left|\boldsymbol{s}_2^{trace}(\boldsymbol{x}_t,t;\theta) - \operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)\right|^2\right] \Rightarrow \theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0,\boldsymbol{\epsilon}}\left[\frac{1}{\sigma_t^4}\left|\sigma_t^2 \boldsymbol{s}_2^{trace}(\boldsymbol{x}_t,t;\theta) + d - \|\boldsymbol{\ell}_1\|_2^2\right|^2\right]$$

Reverse Prob. Flow ODE and Data Likelihood [LZB+22a]

- High-order denoising score matching:

  Iteratively leverage the known $q(x_t|x_0) = \mathcal{N}\left(x_t|\sqrt{\alpha_t}x_0, \sigma_t^2 I\right)$ as a noising distribution.

  - Third-order: with good first & second-order models $\hat{s}_1$ & $\hat{s}_2$,

$$\mathbb{E}_{q_t(\boldsymbol{x}_t)}\left[\left\|\boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta) - \nabla_{\boldsymbol{x}} \operatorname{tr}\left(\nabla_{\boldsymbol{x}}^2 \log q_t(\boldsymbol{x}_t)\right)\right\|_2^2\right] \Longrightarrow \theta^* = \operatorname*{argmin}_\theta \mathbb{E}_{\boldsymbol{x}_0, \boldsymbol{\epsilon}}\left[\frac{1}{\sigma_t^6}\left\|\sigma_t^3 \boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta) + \boldsymbol{\ell}_3\right\|_2^2\right]$$

$$\boldsymbol{\ell}_1(\boldsymbol{\epsilon}, \boldsymbol{x}_0, t) := \sigma_t \hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t) + \boldsymbol{\epsilon},$$

$$\boldsymbol{\ell}_2(\boldsymbol{\epsilon}, \boldsymbol{x}_0, t) := \sigma_t^2 \hat{\boldsymbol{s}}_2(\boldsymbol{x}_t, t) + \boldsymbol{I},$$

$$\boldsymbol{\ell}_3(\boldsymbol{\epsilon}, \boldsymbol{x}_0, t) := \left(\|\boldsymbol{\ell}_1\|_2^2 \boldsymbol{I} - \operatorname{tr}(\boldsymbol{\ell}_2)\boldsymbol{I} - 2\boldsymbol{\ell}_2\right)\boldsymbol{\ell}_1$$

  In practice:

  - Ignore the $\sigma_t^2, \sigma_t^4, \sigma_t^6$ weights in the objectives to reduce variance.
  - Optimize the same model: Let $\hat{\boldsymbol{s}}_1(\boldsymbol{x}_t, t) := \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ and $\hat{\boldsymbol{s}}_2(\boldsymbol{x}_t, t) := \nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)$ and $\boldsymbol{s}_3(\boldsymbol{x}_t, t; \theta) = \nabla_{\boldsymbol{x}} \operatorname{tr}(\nabla_{\boldsymbol{x}} \boldsymbol{s}_\theta(\boldsymbol{x}_t, t))$
  - Stop-gradient of $\hat{s}_1$ and $\hat{s}_2$ w.r.t θ in second and third-order score matching.
  - vs. Directly maximizing $\log p_{\theta,0}^{\mathrm{ODE}}$: Step-by-step training (and $O(1)$ cost in each step) is more efficient.

Reverse Prob. Flow ODE and Data Likelihood [LZB+22a]

- Variational gap of [SDME21, Thm.1]:

$$\mathrm{KL}\left(q_0 \| p^{\mathrm{SDE}}_{\theta,0}\right) = D_{\mathrm{Fisher}}\left(\theta; \lambda_{(\cdot)} = \frac{g^2_{(\cdot)}}{2}\right) + \mathrm{KL}(q_T \| p_T) - \int_0^T \frac{g_t^2}{2} \mathbb{E}_{q_t(\tilde{x})} \left\| s_{\theta,t}(\tilde{x}) - \nabla \log p_t^{\mathrm{SDE}}(\tilde{x}) \right\|^2 \mathrm{d}t$$

$$= \int_0^T \frac{g_t^2}{2} \mathbb{E}_{q_t(\tilde{x})} \left[\left\| s_{\theta,t}(\tilde{x}) - \nabla \log q_t(\tilde{x}) \right\|^2 - \left\| s_{\theta,t}(\tilde{x}) - \nabla \log p_t^{\mathrm{SDE}}(\tilde{x}) \right\|^2 \right] \mathrm{d}t + \mathrm{KL}(q_T \| p_T).$$

  - **Self-consistency** $s_{\theta,t}(\tilde{x}) = \nabla \log p_t^{\mathrm{SDE}}(\tilde{x})$ indeed closes the gap.
  - But for $f_t(x_t) = a_t x_t$ $(a_t < 0)$ and a finite T, when **self-consistent**, p_t^{SDE} (incl. p_0^{SDE}) **is doomed a Gaussian**:
    - Reverse SDE $\mathrm{d}x_t = \left(a_t x_t - \frac{g_t^2}{2} \nabla \log p_t^{\mathrm{SDE}}(x_t)\right) \mathrm{d}t + \frac{g_t^2}{2} \mathrm{d}\bar{B}_t$, $\quad p_T^{\mathrm{SDE}}(x_T) = \mathcal{N}(x_T | 0, I)$.

      ⇔ Forward SDE $\mathrm{d}x_t = a_t x_t \, \mathrm{d}t + \frac{g_t^2}{2} \mathrm{d}B_t$, $\quad p_T^{\mathrm{SDE}}(x_T) = \mathcal{N}(x_T | 0, I)$.

      ➔ $p_{T|0}^{\mathrm{SDE}}(x_T | x_0) = \mathcal{N}(x_T | \varsigma_T x_0, \varsigma_T^2 \upsilon_T^2 I)$, $\quad p_T^{\mathrm{SDE}}(x_T) = \mathcal{N}(x_T | 0, I)$.

      ➔ When T is finite, $\varsigma_T \neq 0$, so $p_0^{\mathrm{SDE}}(x_0)$ is also a Gaussian.
  - For a finite T, **the nongaussianity of** $p_{\theta,0}$ **is encoded in the non-self-consistency** $s_{\theta,t} - \nabla \log p_t^{\mathrm{SDE}}$.
    - For a finite T, $\mathrm{KL}(q_T \| p_T) > 0$ and constant, so **non-self-consistency helps minimizing** $\mathrm{KL}\left(q_0 \| p_{\theta,0}^{\mathrm{SDE}}\right)$.
    - Does not conflict the reverse-SDE perspective when $T \to \infty$: $\varsigma_\infty = 0$.

DDPM

- Evidence Lower BOund
- DDPM simple loss
- DDPM variants

Cont.-time view:

- Diffusion process
- VP SDE: Cont.-time DDPM
- Training

Interlude: Score Matching

- Denoising score-matching
- NCSN

- VE SDE: cont.-time NCSN

Cont.-time techniques:

- sub-VP SDE
- Reverse-process simulation
- Classifier-guided generation
- Probability flow

Cont.-time improvements

- DPM-Solver
- Elucidating the design of diffusion model

Cont.-time likelihood

- $p_{\theta,t}^{\mathrm{SDE}}$ bound.
- $p_{\theta,t}^{\mathrm{ODE}}$ bound.

Schrödinger Bridge

Schrödinger Bridge

- The undilated dynamics converges to p_{prior} only asymptotically:

  Trade-off between #layers N and $|p_N - p_{\text{prior}}|$ error.
  - Discretization with time dilation/inhomogeneity transfers the error to discretization error.
- Schrödinger Bridge: Exactly connects the two distributions.

$$\pi^\star = \arg\min \left\{ \mathrm{KL}(\pi | p) \ : \ \pi \in \mathscr{P}_{N+1}, \ \pi_0 = p_{\text{data}}, \ \pi_N = p_{\text{prior}} \right\}.$$

  - $\mathcal{P}_{N+1}$: space of distributions on $\mathcal{X}^{N+1}$.
  - $p = p_{0:N}$: a reference defined by a forward process.

Schrödinger Bridge

Schrödinger Bridge: Background

$$\pi^\star = \arg\min \left\{ \mathrm{KL}(\pi | p) \ : \ \pi \in \mathscr{P}_{N+1}, \ \pi_0 = p_{\mathrm{data}}, \ \pi_N = p_{\mathrm{prior}} \right\}.$$

- Static Schrödinger Bridge:

  $\mathrm{KL}(\pi|p) = \mathrm{KL}(\pi_{0,N}|p_{0,N}) + \mathbb{E}_{\pi_{0,N}}[\mathrm{KL}(\pi_{|0,N}|p_{|0,N})]$ ➔ $\pi^\star(x_{0:N}) = \pi^{\mathrm{s},\star}(x_0, x_N) p_{|0,N}(x_{1:N-1}|x_0, x_N)$

  where $\pi^{\mathrm{s},\star} = \arg\min \left\{ \mathrm{KL}(\pi^{\mathrm{s}} | p_{0,N}) \ : \ \pi^{\mathrm{s}} \in \mathscr{P}_2, \ \pi_0^{\mathrm{s}} = p_{\mathrm{data}}, \ \pi_N^{\mathrm{s}} = p_{\mathrm{prior}} \right\}.$

- Entropy-Regularized optimal transport formulation:

$$\pi^{\mathrm{s},\star} = \arg\min \left\{ -\mathbb{E}_{\pi^{\mathrm{s}}}[\log p_{N|0}(X_N|X_0)] - \mathrm{H}(\pi^{\mathrm{s}}) \ : \ \pi^{\mathrm{s}} \in \mathscr{P}_2, \ \pi_0^{\mathrm{s}} = p_{\mathrm{data}}, \ \pi_N^{\mathrm{s}} = p_{\mathrm{prior}} \right\}.$$

  For VE SDE (NCSN): $p_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k, \sigma_{k+1}^2)$ ➔ $p_{N|0}(x_N|x_0) = \mathcal{N}(x_N; x_0, \sigma^2)$ with $\sigma^2 = \sum_{k=1}^N \sigma_k^2$

  ➔ $\pi^{\mathrm{s},\star} = \arg\min \left\{ \mathbb{E}_{\pi^{\mathrm{s}}}[||X_0 - X_N||^2] - 2\sigma^2 \mathrm{H}(\pi^{\mathrm{s}}) \ : \ \pi^{\mathrm{s}} \in \mathscr{P}_2, \ \pi_0^{\mathrm{s}} = p_{\mathrm{data}}, \ \pi_N^{\mathrm{s}} = p_{\mathrm{prior}} \right\}$

- Practical algorithm: Iterative Proportional Fitting (IPF).

  $\pi^{2n+1} = \arg\min \left\{ \mathrm{KL}(\pi | \pi^{2n}) \ : \ \pi \in \mathscr{P}_{N+1}, \ \pi_N = p_{\mathrm{prior}} \right\},$ ⟶ Reverse process

  $\pi^{2n+2} = \arg\min \left\{ \mathrm{KL}(\pi | \pi^{2n+1}) \ : \ \pi \in \mathscr{P}_{N+1}, \ \pi_0 = p_{\mathrm{data}} \right\}.$ ⟶ Forward process

  Starts with $\pi^0 = p$. ⟶ Forward process

Schrödinger Bridge

- Representation of IPF iteration [DTHD21]:

$$\pi^{2n+1} = \arg\min \{\mathrm{KL}(\pi|\pi^{2n}) \; : \; \pi \in \mathscr{P}_{N+1}, \; \pi_N = p_{\mathrm{prior}}\}, \longrightarrow =: q^n, \text{ reverse process}$$

$$\pi^{2n+2} = \arg\min \{\mathrm{KL}(\pi|\pi^{2n+1}) \; : \; \pi \in \mathscr{P}_{N+1}, \; \pi_0 = p_{\mathrm{data}}\}. \longrightarrow =: p^n, \text{ forward process}$$

$$\Rightarrow q^n(x_{0:N}) = p_{\mathrm{prior}}(x_N) \prod_{k=0}^{N-1} \underbrace{p^n_{k|k+1}(x_k|x_{k+1})}_{= \frac{p^n_{k+1|k}(x_{k+1}|x_k) p^n_k(x_k)}{p^n_{k+1}(x_{k+1})}}, \; p^{n+1}(x_{0:N}) = p_{\mathrm{data}}(x_0) \prod_{k=0}^{N-1} \underbrace{q^n_{k+1|k}(x_{k+1}|x_k)}_{= \frac{q^n_{k|k+1}(x_k|x_{k+1}) q^n_{k+1}(x_{k+1})}{q^n_k(x_k)}}.$$

reverse conditional of the forward process

forward conditional of the reverse process

- Iterative Mean-Matching Proportional Fitting:

If $q^n_{k|k+1}(x_k|x_{k+1}) = \mathcal{N}(x_k; B^n_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I}), \; p^n_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; F^n_k(x_k), 2\gamma_{k+1}\mathbf{I}),$

then

$$B^n_{k+1} = \arg\min_{\mathrm{B} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p^n_{k,k+1}} [\|\mathrm{B}(X_{k+1}) - (X_{k+1} + F^n_k(X_k) - F^n_k(X_{k+1}))\|^2],$$

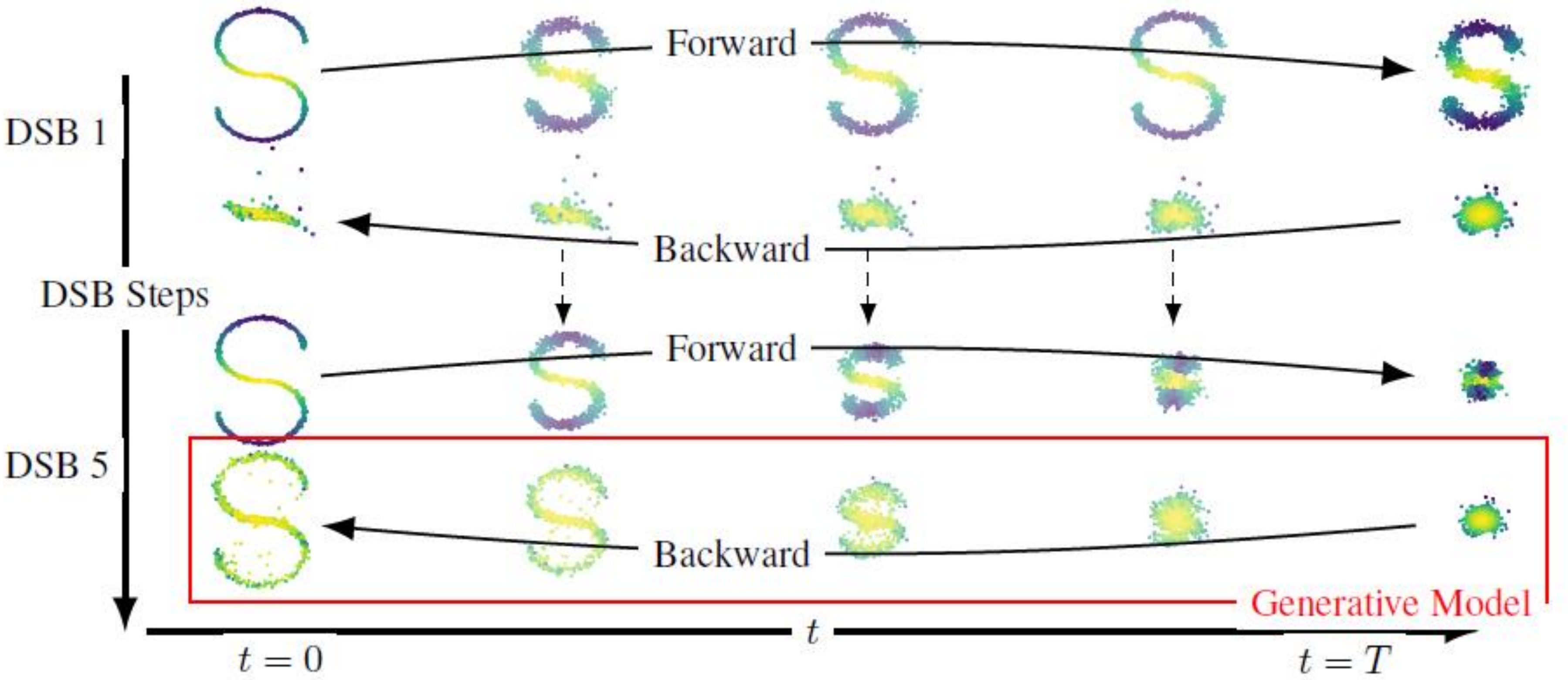
$$F^{n+1}_k = \arg\min_{\mathrm{F} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{q^n_{k,k+1}} [\|\mathrm{F}(X_k) - (X_k + B^n_{k+1}(X_{k+1}) - B^n_{k+1}(X_k))\|^2].$$

- Diffusion Schrödinger Bridge:

Learn step-conditioned models: $B_{\beta^n}(k, x) \approx B^n_k(x)$ and $F_{\alpha^n}(k, x) \approx F^n_k(x)$.

Schrödinger Bridge

- Diffusion Schrödinger Bridge [DTHD21]:



Microsoft

Thanks!

References

References

- Probabilistic Graphical Models
  - Diffusion-based models

    - [SWMG15] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., & Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning* (pp. 2256-2265).
    - [HJA20] Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*.
    - [SSK+21] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., & Poole, B. (2021). Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.
    - [SME21] Song, J., Meng, C., & Ermon, S. (2021). Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.
    - [ND21] Nichol, A. Q., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning* (pp. 8162-8171). PMLR.
    - [SDME21] Song, Y., Durkan, C., Murray, I., & Ermon, S. (2021). Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems*, 34, 1415-1428.
    - [KSPH21] Kingma, D., Salimans, T., Poole, B., & Ho, J. (2021). Variational diffusion models. In *Advances in neural information processing systems*, 34, 21696-21707.
    - [DN21] Dhariwal, P., & Nichol, A. (2021). Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 34, 8780-8794.

References

- Probabilistic Graphical Models
  - Diffusion-based models

    - [DTHD21] De Bortoli, V., Thornton, J., Heng, J., & Doucet, A. (2021). Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, 34, 17695-17709.
    - [DVK22] Dockhorn, T., Vahdat, A., & Kreis, K. (2022). Score-Based Generative Modeling with Critically-Damped Langevin Diffusion. In *International Conference on Learning Representations*.
    - [BLZZ22] Bao, F., Li, C., Zhu, J., & Zhang, B. (2022). Analytic-DPM: an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models. In *International Conference on Learning Representations*.
    - [BLS+22] Bao, F., Li, C., Sun, J., Zhu, J., & Zhang, B. (2022). Estimating the Optimal Covariance with Imperfect Mean in Diffusion Probabilistic Models. In *International Conference on Machine Learning*.
    - [LZB+22a] Lu, C., Zheng, K., Bao, F., Chen, J., Li, C., & Zhu, J. (2022). Maximum Likelihood Training for Score-Based Diffusion ODEs by High-Order Denoising Score Matching. In *International Conference on Machine Learning*.
    - [LZB+22b] Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., & Zhu, J. (2022). DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. In *Advances in Neural Information Processing Systems*.
    - [KAAL22] Karras, T., Aittala, M., Aila, T., & Laine, S. (2022). Elucidating the Design Space of Diffusion-Based Generative Models. In *Advances in Neural Information Processing Systems*.
    - [ZBLZ22] Zhao, M., Bao, F., Li, C., & Zhu, J. (2022). EGSDE: Unpaired Image-to-Image Translation via Energy-Guided Stochastic Differential Equations. In *Advances in Neural Information Processing Systems*.

References

- Probabilistic Graphical Models
  - Related
    - Sliced score matching: [SGSE19] Song, Y., Garg, S., Shi, J., & Ermon, S. (2019). Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence* (pp. 574-584). PMLR.
    - Optimization: [WWJ16] Wibisono, A., Wilson, A. C., & Jordan, M. I. (2016). A variational perspective on accelerated methods in optimization. In *Proceedings of the National Academy of Sciences*, 113(47), E7351-E7358.