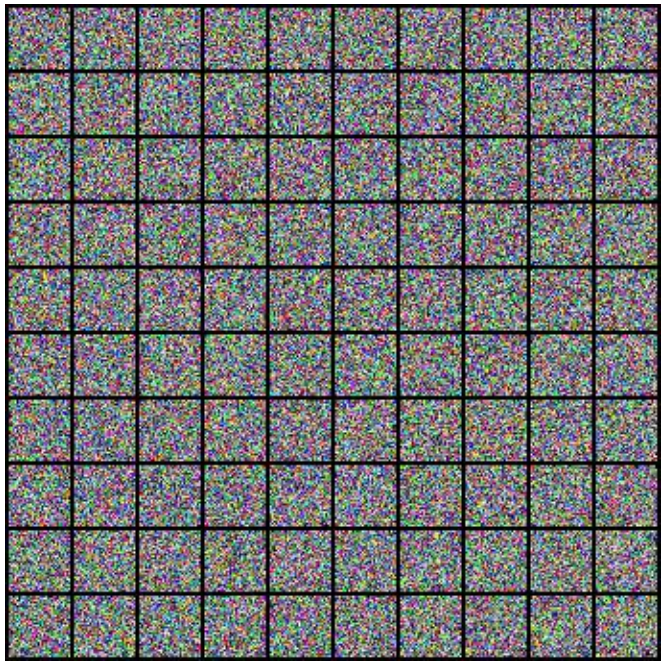
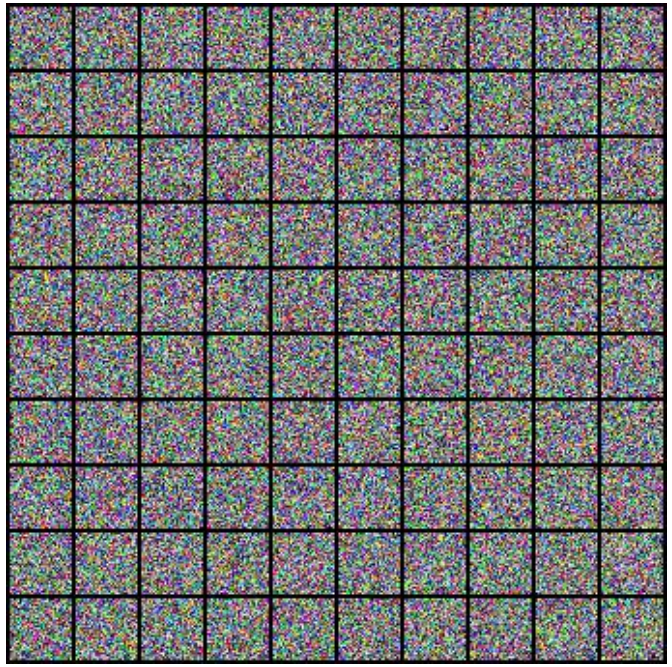
The Art of Diffusion Model

- Discrete Perspective

Bangyan Liao

liaobangyan@westlake.edu.cn





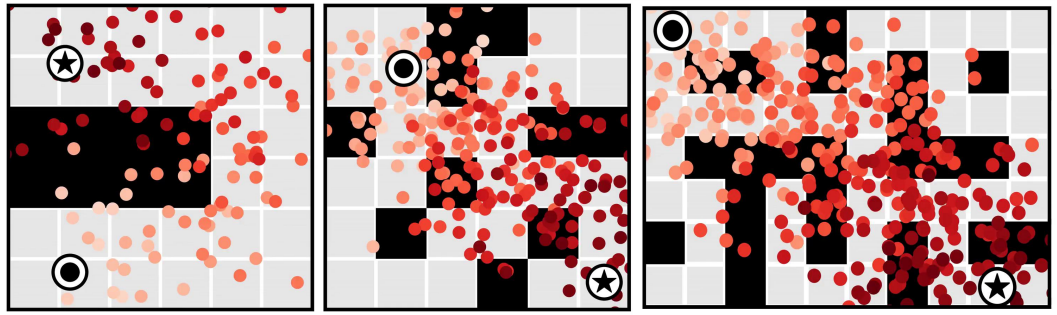
WESTLAKE UNIVERSITY

Motivation and Outline

Various Applications with Diffusion Model

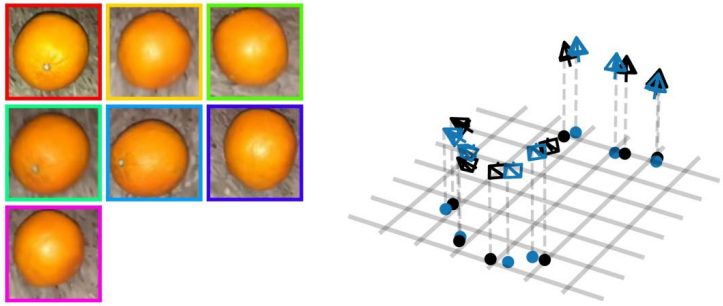
Motivation

Motion Planning[1]



3D Generation[3]

Camera Pose Estimation[5]



Reinforcement Learning[2]

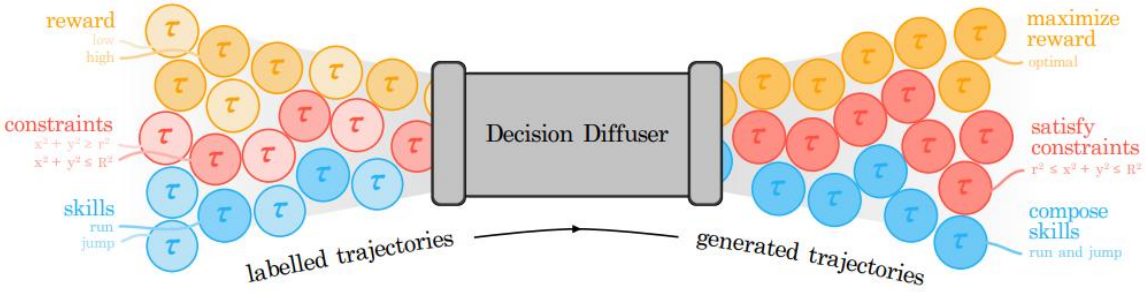
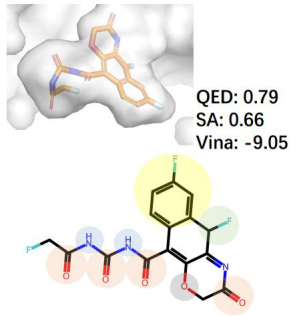


Image inpainting[4]

Protein Structure[6]



[1] Janner M, Du Y, Tenenbaum J B, et al. Planning with diffusion for flexible behavior synthesis[J]. arXiv preprint arXiv:2205.09991, 2022.

[2] Ajay A, Du Y, Gupta A, et al. Is conditional generative modeling all you need for decision-making?[J]. arXiv preprint arXiv:2211.15657, 2022.

[3] Shi Y, Wang P, Ye J, et al. Mvdream: Multi-view diffusion for 3d generation[J]. arXiv preprint arXiv:2308.16512, 2023.

[4] Lugmayr A, Danelljan M, Romero A, et al. Repaint: Inpainting using denoising diffusion probabilistic models[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 11461-11471.

[5] Zhang J Y, Lin A, Kumar M, et al. Cameras as rays: Pose estimation via ray diffusion[J]. arXiv preprint arXiv:2402.14817, 2024.

[6] Huang Z, Yang L, Zhou X, et al. Interaction-based Retrieval-augmented Diffusion Models for Protein-specific 3D Molecule Generation[C]//Forty-first International Conference on Machine Learning.

Three Perspectives on Discrete Version Diffusion Model

Motivation

Soce Matching
Noise Conditioned Soce Matching Network (NCSN)[1]

Hierarchical VAE
Denoising Diffusion Probabilistic Model (DDPM)[2]

General Signal-Noise-Ratio
Variational Diffusion Model (VDM)[3]

Motivation

Noise Conditioned Soce Matching Network (NCSN)[1]

- Revisiting Naive Score Matching
- Score Matching perspective of Diffusion Model
- Objective and Noise Schedule Improvement

Denoising Diffusion Probabilistic Model (DDPM)[2]

Variational Diffusion Model (VDM)[3]

- General Signal-Noise-Ratio perspective of Diffusion Model
- Neural Network Parameterized Signal-Noise-Ratio

Hierarchical VAE perspective of DDPM[2]

- Revisiting VAE and Evidence Lower BOund
- Hierarchical VAE perspective of Diffusion Model
- Reparameterization Triangle

Further Readings

Reference

Outline

Noise Conditioned Soce Matching Network (NCSN)[1]

- Revisiting Naive Score Matching
- Score Matching perspective of Diffusion Model
- Objective and Noise Schedule Improvement

Why score matching?

NCSN[1] - Revisiting Naive Score Matching

Score matching can be used to approximate the **intractable distribution q** with **tractable distribution p**.

$$\min_{p} F(p, q) = \frac{1}{2}\mathbb{E}_{q}[\|\nabla_x \log p(x) - \nabla_x \log q(x)\|_2^2] \quad \textbf{(score matching)}$$

Example of intractable distribution q

Energy-based Model	$q(x) = \frac{1}{Z}\exp[-E(x)]$	$\nabla_x \log q(x) = -\nabla_x E(x)$
Unnormalized Model	$q(x) = \frac{q^*(x)}{Z}$	$\nabla_x \log q(x) = \nabla_x \log q^*(x)$
Bayesian Posterior	$q(x\|D) = \frac{q(x,D)}{\int q(x,D)}$	$\nabla_x \log q(x) = \nabla_x \log q(x, D)$

Example of tractable distribution p

Gaussian Distribution	$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$	$\nabla_x \log p(x) = -\frac{(x-\mu)}{\sigma^2}$
Normaliaing Flow	$p(x) = MLP(x),\ x \sim \mathcal{N}(0,\ 1)$	$\nabla_x \log p(x) = \frac{\nabla_x MLP(x)}{MLP(x)}$

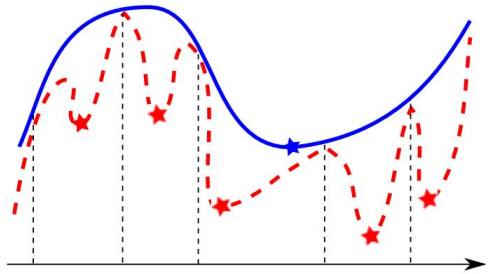
Why score matching?

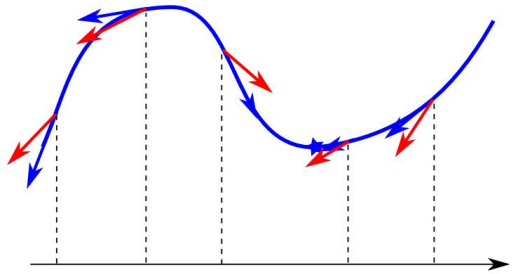
NCSN[1] - Revisiting Naive Score Matching

Score matching can be used to approximate the **intractable distribution q** with **tractable distribution p**.

$$\min_{p} F(p, q) = \frac{1}{2}\mathbb{E}_{q}[\|\nabla_x \log p(x) - \nabla_x \log q(x)\|_2^2] \quad \textbf{(score matching)}$$

Connection with VI (Variational Inference)





$$\min_{p} KL\,[q \mid p] = \mathbb{E}_{q}[\log q(x) - \log p(x)]$$

Variational Inference **Score Matching**

Compared to VI, score matching leads to more stable score estimation and can get rid of the intractable normalization constant.

Score matching with data is not trival

NCSN[1] - Revisiting Naive Score Matching

Score matching can be used to approximate the **intractable target data distribution q_{data}** with **tractable neural network p_θ**.

$$\min_\theta F(p_\theta, q_{data}) = \frac{1}{2}\mathbb{E}_{q_{data}}[\|\nabla_x \log p_\theta(x) - \nabla_x \log q_{data}(x)\|_2^2]$$

the exact score function is unknown

samples are available in this data distribution

Two possible choices for learning the target data score function

- Integration by parts trick
- Denoising Score Matching (DSM)

One choice for sample according to the learned score function

- Langevin Dynamics

Integration by parts trick

NCSN[1] - Revisiting Naive Score Matching

Integration by parts trick

$$\frac{1}{2}\mathbb{E}_{\boldsymbol{q_{data}}}[\|\nabla_x \log \boldsymbol{p_\theta}(x) - \nabla_x \log \boldsymbol{q_{data}}(x)\|_2^2]$$
$$= \frac{1}{2}\mathbb{E}_{\boldsymbol{q_{data}}}[\frac{1}{2}(\nabla_x \log \boldsymbol{p_\theta}(x))^2 + tr(\nabla_x^2 \log \boldsymbol{p_\theta}(x))]$$

Proof.

$$\min_\theta F(\boldsymbol{q_{data}}, \boldsymbol{p_\theta})$$
$$= \frac{1}{2}\mathbb{E}_{\boldsymbol{q_{data}}}[\|\nabla_x \log \boldsymbol{q_{data}}(x) - \nabla_x \log \boldsymbol{p_\theta}(x)\|_2^2]$$
$$= \frac{1}{2}\int \boldsymbol{q_{data}}(x)\,(\nabla_x \log \boldsymbol{q_{data}}(x) - \nabla_x \log \boldsymbol{p_\theta}(x))^2 dx$$
$$= \underbrace{\frac{1}{2}\int \boldsymbol{q_{data}}(x)\,(\nabla_x \log \boldsymbol{q_{data}}(x))^2 dx}_{\text{const}} + \frac{1}{2}\int \boldsymbol{q_{data}}(x)\,(\nabla_x \log \boldsymbol{p_\theta}(x))^2 dx - \int \underbrace{\boldsymbol{q_{data}}(x)\,\nabla_x \log \boldsymbol{q_{data}}(x)}_{\boldsymbol{q_{data}}(x)\nabla_x \log \boldsymbol{q_{data}}(x) = \nabla_x \boldsymbol{q_{data}}(x)} \nabla_x \log \boldsymbol{p_\theta}(x) dx$$
$$= \frac{1}{2}\int \boldsymbol{q_{data}}(x)\,(\nabla_x \log \boldsymbol{p_\theta}(x))^2 dx \underbrace{- \boldsymbol{q_{data}}(x)\nabla_x \log \boldsymbol{p_\theta}(x)|_{-\infty}^{+\infty}}_{q_{data}(x) \to 0\ when\ |x| \to \infty} + \int \boldsymbol{q_{data}}(x)\ tr(\nabla_x^2 \log \boldsymbol{p_\theta}(x)) dx$$
$$= \frac{1}{2}\int \boldsymbol{q_{data}}(x)\,(\nabla_x \log \boldsymbol{p_\theta}(x))^2 dx + \int \boldsymbol{q_{data}}(x)\ tr(\nabla_x^2 \log \boldsymbol{p_\theta}(x)) dx$$
$$= \frac{1}{2}\mathbb{E}_{\boldsymbol{q_{data}}}[\frac{1}{2}(\nabla_x \log \boldsymbol{p_\theta}(x))^2 + tr(\nabla_x^2 \log \boldsymbol{p_\theta}(x))]$$

■

Denoising Score Matching (DSM)

NCSN[1] - Revisiting Naive Score Matching

Denoising Score Matching

$$\frac{1}{2}\mathbb{E}_{q_{data}}[\|\nabla_x \log p_\theta(x) - \nabla_x \log q_{data}(x)\|_2^2] \quad \textbf{(score matching)}$$

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{x}|x)q_{data}}[\|\nabla_x \log p_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x)\|_2^2] \quad \textbf{(denoising score matching)}$$

$q_\sigma(\tilde{x}) = \int q_\sigma(\tilde{x}|x) q_{data}(x)\, dx$ represents the noised data

$q_\sigma(\tilde{x}) \approx q_{data}(x)$ when the noise is small enough

Denoising Score Matching for Gaussian Noise

Instead of approximate $q_{data}(x)$ directly, noised $q_\sigma(\tilde{x}|x)$ can be specifically defined (Gaussian Noise)

$$q_\sigma(\tilde{x}|x) = \mathcal{N}(\tilde{x};\ x,\ \sigma^2 \mathbf{I})$$

$$\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x) = \frac{\tilde{x} - x}{\sigma^2}$$

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{x}|x)q_{data}}\left[\left\|\nabla_x \log p_\theta(\tilde{x}) - \frac{\tilde{x} - x}{\sigma^2}\right\|_2^2\right] \quad \textbf{(denoising score matching)}$$

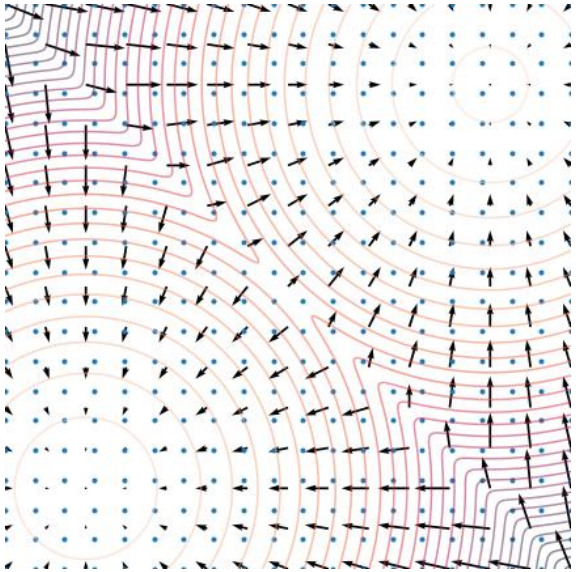
Recall the Langevin Dynamics

NCSN[1] - Revisiting Naive Score Matching

Langevin Dynamics

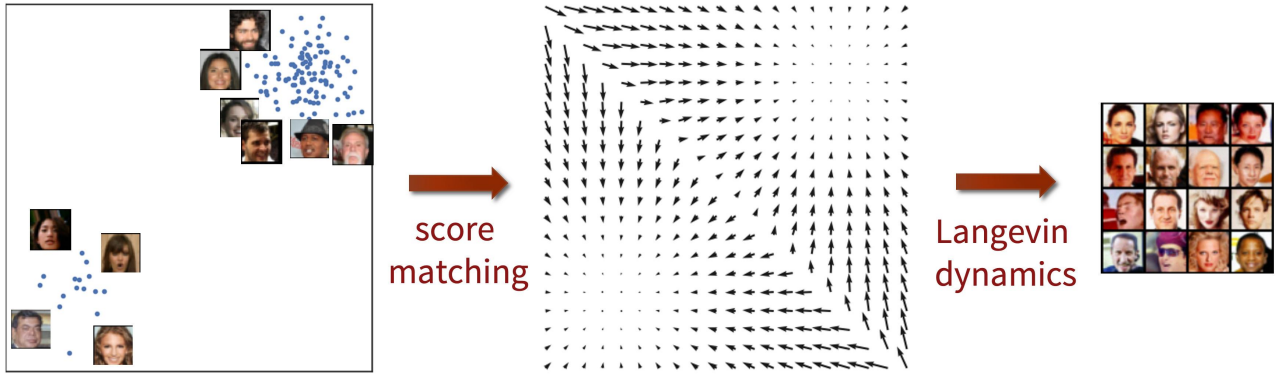
$$x_t = x_{t-1} + \frac{\sigma}{2}\nabla_x \log \boldsymbol{p_\theta}(x_{t-1}) + \sqrt{\sigma}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0,\, \mathbf{I})$$

x_0 follows some prior distribution. The distribution of x_T converges to $\boldsymbol{p_\theta}(x)$ when $T \to \infty$



Example of Langevin Dynamics

The full pipeline of score matching



Data samples

$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$

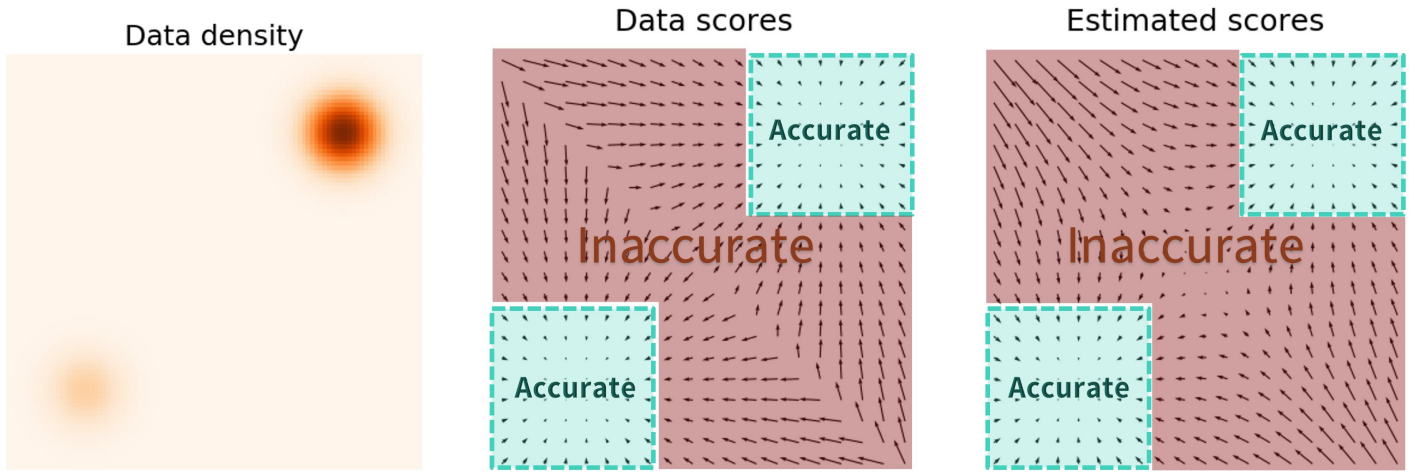
Scores

$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$

New samples

Challenges of score-based generative modeling

NCSN[1] - Score Matching perspective of Diffusion Model



Example of inaccurate region of learned score

The manifold hypothesis

The manifold hypothesis states that data in the real world tend to concentrate on low dimensional manifolds embedded in a high dimensional space

Low data density regions

The scarcity of data in low density regions can cause difficulties for both score estimation with score matching and MCMC sampling with Langevin dynamics.

Learning score with multi-level noise

NCSN[1] - Score Matching perspective of Diffusion Model

Noise Conditioned Score Matching (NCSN)

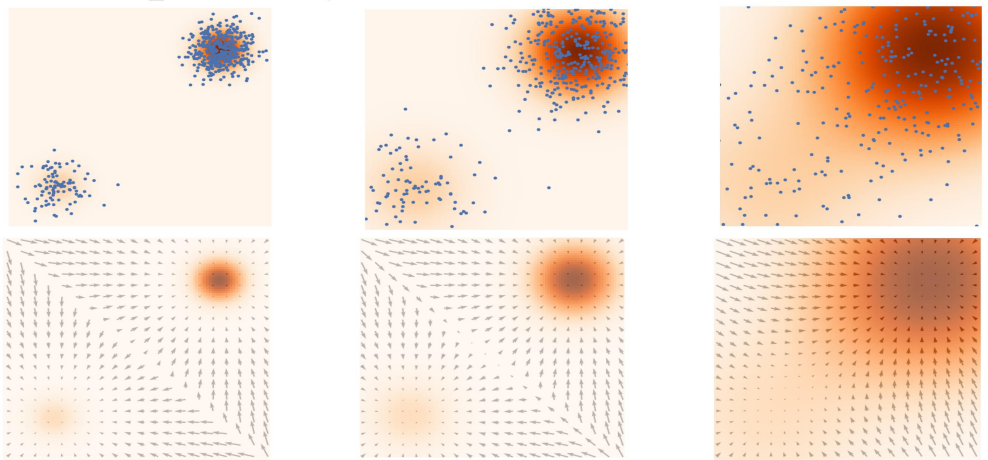
$$\mathcal{L}(\theta, \{\sigma_i\}_{i=1}^{L}) = \sum_{i=1}^{L} \lambda(\sigma_i)\, \ell(\theta, \sigma_i)$$

$$\ell(\theta, \sigma) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim \boldsymbol{q_{data}}(\boldsymbol{x})}\mathbb{E}_{\tilde{\boldsymbol{x}}\sim \boldsymbol{q_\sigma}(\tilde{\boldsymbol{x}}|\boldsymbol{x})}[\|\nabla_x \log \boldsymbol{p_\theta}(\tilde{x}) - \nabla_{\tilde{x}} \log \boldsymbol{q_\sigma}(\tilde{\boldsymbol{x}}|\boldsymbol{x})\|_2^2]$$

$$= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim \boldsymbol{q_{data}}(\boldsymbol{x})}\mathbb{E}_{\tilde{\boldsymbol{x}}\sim \mathcal{N}(x, \sigma^2 \mathbf{I})}\left[\left\| \boldsymbol{s_\theta}(\tilde{x}, \sigma) - \frac{\tilde{x} - x}{\sigma^2}\right\|_2^2\right]$$

Let $\{\sigma_i\}_{i=1}^{L}$ be a positive geometric sequence that satisfies $\boldsymbol{\epsilon} = \frac{\sigma_1}{\sigma_2} = \ldots = \frac{\sigma_{L-1}}{\sigma_L} > 1$

Example of Noise Conditioned Score Matching



$\sigma_3 \quad < \quad \sigma_2 \quad < \quad \sigma_1$

Different noise level of data density and samples.

Different noise level of learned score results.

Annealed Langevin Sampling

NCSN[1] - Score Matching perspective of Diffusion Model

Annealed Langevin Sampling

Algorithm 1 Annealed Langevin dynamics.

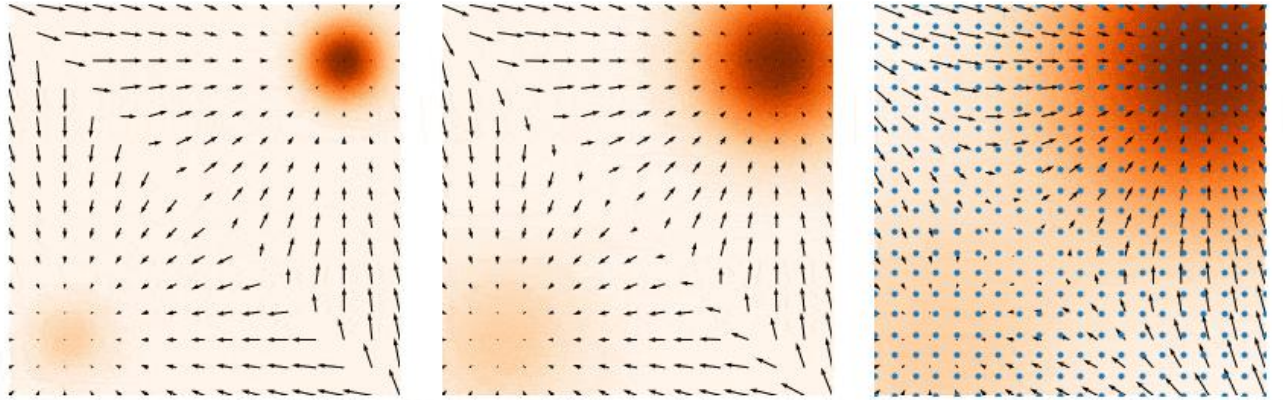
Require: $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$.

1: Initialize $\tilde{\mathbf{x}}_0$
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad \alpha_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$ ▷ α_i is the step size.
4: $\quad$ **for** $t \leftarrow 1$ to T **do**
5: $\quad\quad$ Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
6: $\quad\quad \tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$
7: $\quad$ **end for**
8: $\quad \tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
9: **end for**
return $\tilde{\mathbf{x}}_T$

Annealed Denoised Iteration

Langevin Dynamics Sampling

Example of Annealed Langevin Sampling



σ_3 σ_2 σ_1

Objective Improvement

NCSN[1] - Objective and Noise Schedule Improvement

Score-based Objective

$$\mathcal{L}(\theta, \{\sigma_i\}_{i=1}^{L}) = \sum_{i=1}^{L} \lambda(\sigma_i)\, \ell(\theta, \sigma_i)$$

$$\ell(\theta, \sigma) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim q_{data}(\boldsymbol{x})}\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(x,\, \sigma^2\mathbf{I})}\left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\tilde{x}, \sigma) - \frac{\tilde{x} - x}{\sigma^2}\right\|_2^2\right]$$

This score neural network $\boldsymbol{s}_{\boldsymbol{\theta}}(\tilde{x}, \sigma)$ has unnormalized output score

Residual-based Objective

$$\ell(\theta, \sigma) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim q_{data}(\boldsymbol{x})}\mathbb{E}_{\tilde{\boldsymbol{x}}\sim\mathcal{N}(x,\, \sigma^2\mathbf{I})}\left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\tilde{x}, \sigma) - \frac{\tilde{x} - x}{\sigma^2}\right\|_2^2\right]$$

$$= \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim q_{data}(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(0,\, \sigma^2\mathbf{I})}\left[\frac{1}{\sigma^2}\left\| \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\tilde{x}, \sigma) - \boldsymbol{\varepsilon}\right\|_2^2\right]$$

This residual neural network $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\tilde{x}, \sigma)$ has normalized output noise

Proof.

$$q(\tilde{x}|x) = \mathcal{N}(x, \sigma^2\mathbf{I})$$
$$\tilde{x} = x + \sigma\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\, \mathbf{I})$$

■

Noise Schedule Improvement

NCSN[1] - Objective and Noise Schedule Improvement

Variance-Exploding Noise Schedule

$$\mathcal{L}(\theta, \{\sigma_i\}_{i=1}^{L}) = \sum_{i=1}^{L} \lambda(\sigma_i)\, \ell(\theta, \sigma_i)$$

$$\ell(\theta, \sigma) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim \boldsymbol{q_{data}}(\boldsymbol{x})}\mathbb{E}_{\tilde{x}\sim\mathcal{N}(x,\,\sigma^2\mathbf{I})}\left[\left\| \boldsymbol{s_\theta}(\tilde{x}, \sigma) - \frac{\tilde{x} - x}{\sigma^2}\right\|_2^2\right]$$

This can be viewed as the Variance-Exploding Noise Schedule as

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \boldsymbol{\epsilon}^{\,t}\sigma_0\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\,\mathbf{I})$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\, \boldsymbol{x}_0, \boldsymbol{\epsilon}^{\,t}\sigma_0\mathbf{I})$$

and this variance grows exponentially

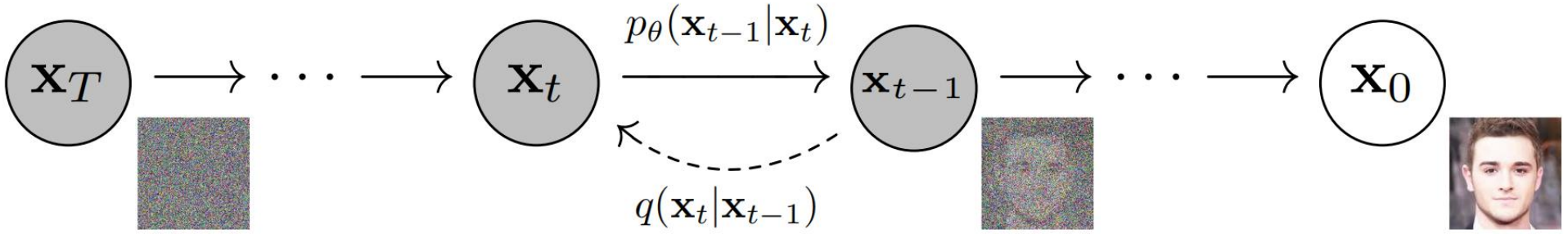
Variance-Preserving Noise Schedule

$$\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\,\mathbf{I})$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\, \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\mathbf{I})$$

This variance preserving noise schedule leads more stable learning process

Denoising Diffusion Probabilistic Model (DDPM)[2]

$$\mathcal{L}(\theta) = \frac{1}{2}\mathbb{E}_{t\sim[0,1]}\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{q_{data}}(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(0,\,\sigma^2\mathbf{I})}\left[\left\| \boldsymbol{\varepsilon_\theta}(\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}) - \boldsymbol{\varepsilon}\right\|_2^2\right]$$



Denoising Diffusion Probabilistic Model (DDPM)[2]

VARIATIONAL BOUND FOR TRAINING

DDPM[2]

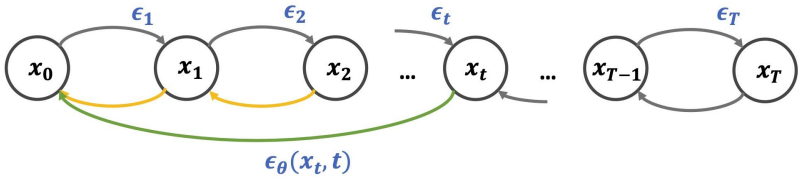
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

$$\mathbb{E}\left[-\log p_\theta(\mathbf{x}_0)\right] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\mathbf{x}_T) - \sum_{t\geq 1}\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right] =: L$$

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}) \qquad \alpha_t := 1-\beta_t \qquad \bar{\alpha}_t := \prod_{s=1}^{t}\alpha_s$$

Efficient training:

$$\mathbb{E}_q\Big[\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0)\parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1}\underbrace{D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} \underbrace{-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0}\Big]$$



$$q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0), \tilde{\beta}_t\mathbf{I}),$$

$$\text{where} \quad \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t,\mathbf{x}_0) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \quad \text{and} \quad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

VARIATIONAL BOUND FOR TRAINING

DDPM[2]

Gaussian stochastic process:

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

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

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

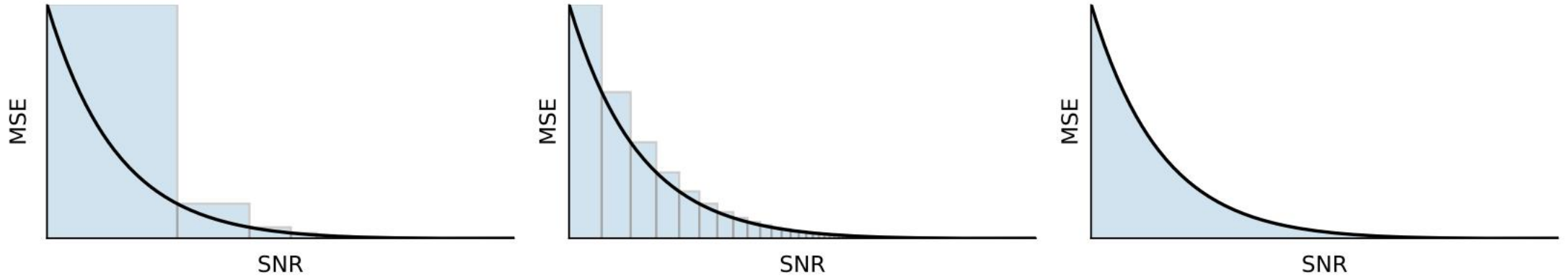
$$\mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)} \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}, t) \right\|^2 \right]$$

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$



Variational Diffusion Model (VDM)[3]

- General Signal-Noise-Ratio perspective of Diffusion Model
- Neural Network Parameterized Signal-Noise-Ratio

General Signal-Noise-Ratio

VDM[3] - General Signal-Noise-Ratio perspective of Diffusion Model

Variance-Exploding Noise Schedule (NCSN[1])

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \boldsymbol{\epsilon}^{\,t}\sigma_0\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\,\mathbf{I})$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\,\boldsymbol{x}_0, \boldsymbol{\epsilon}^{\,t}\sigma_0\mathbf{I})$$

Variance-Preserving Noise Schedule (DDPM[2])

$$\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\,\mathbf{I})$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\,\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\mathbf{I})$$

Variance-Nueral Noise Schedule (VDM[3])

$$\boldsymbol{x}_t = NN(t)\,\boldsymbol{x}_0 + NN'(t)\,\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\,\mathbf{I})$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\,NN(t)\,\boldsymbol{x}_0, [NN'(t)]^2\mathbf{I})$$

This will be introduced in this section.

Neural Network Parameterized Signal-Noise-Ratio

VDM[3] - Neural Network Parameterized Signal-Noise-Ratio

Variance-Exploding Noise Schedule (NCSN[1])

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \boldsymbol{\epsilon}^{\,t}\sigma_0\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\,\mathbf{I})$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\,\boldsymbol{x}_0, \boldsymbol{\epsilon}^{\,t}\sigma_0\mathbf{I})$$

Variance-Preserving Noise Schedule (DDPM[2])

$$\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\,\mathbf{I})$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\,\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\mathbf{I})$$

Signal-Noise-Ratio must be non-negative and monotonically decreasing.

$$\mathrm{SNR}(t) = \frac{\alpha_t^2}{\sigma_t^2} = exp(-\gamma_\eta(t))$$
$$\sigma_t^2 = sigmoid(\gamma_\eta(t))$$
$$\alpha_t^2 = sigmoid(-\gamma_\eta(t))$$

Neural Network Parameterized Noise Schedule (VDM[3])

$$\boldsymbol{x}_t = \sqrt{sigmoid(-\gamma_\eta(t))}\,\boldsymbol{x}_0 + \sqrt{sigmoid(\gamma_\eta(t))}\,\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}\sim\mathcal{N}(0,\,\mathbf{I})$$
$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\,\sqrt{sigmoid(-\gamma_\eta(t))}\,\boldsymbol{x}_0, sigmoid(\gamma_\eta(t))\mathbf{I})$$

General Signal-Noise-Ratio

VDM[3] - General Signal-Noise-Ratio perspective of Diffusion Model

Markov Process in DDPM[2]

Forward Diffusion Process

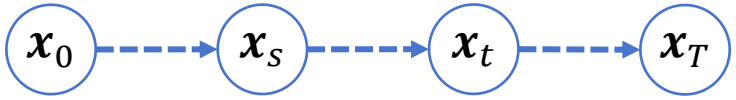
$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t;\ \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

Backward Process

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \propto \mathcal{N}(\boldsymbol{x}_{t-1};\ \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x_0}}{1-\overline{\alpha}_t}, \frac{(1-\alpha_t)(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\mathbf{I})$$

General Markov Process

Forward Diffusion Process



$$q(\boldsymbol{x}_t|\boldsymbol{x}_s) = \mathcal{N}(\boldsymbol{x}_t;\ \alpha_{t|s}\,\boldsymbol{x}_s, \sigma_{t|s}^2\,\mathbf{I})$$

$$\alpha_{t|s} = \alpha_t\ /\ \alpha_s \qquad \sigma_{t|s}^2 = \sigma_t^2 - \alpha_{t|s}^2\sigma_s^2$$

Backward Process

$$q(\boldsymbol{x}_s|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_s;\ \mu(s,t),\ \sigma^2(s,t)\ \mathbf{I})$$

ground truth	*neural approxiamation*
$\mu(s,t) = \frac{\alpha_{t\|s}\sigma_s^2}{\sigma_t^2}\boldsymbol{x}_t + \frac{\alpha_s\sigma_{t\|s}^2}{\sigma_t^2}\boldsymbol{x}_0$	$\mu_\theta(s,t) = \frac{\alpha_{t\|s}\sigma_s^2}{\sigma_t^2}\boldsymbol{x}_t + \frac{\alpha_s\sigma_{t\|s}^2}{\sigma_t^2}\boldsymbol{x}_\theta(\boldsymbol{x}_t, t)$
$\sigma^2(s,t) = \frac{\sigma_{t\|s}^2\sigma_s^2}{\sigma_t^2}$	$\sigma^2(s,t) = \frac{\sigma_{t\|s}^2\sigma_s^2}{\sigma_t^2}$

Objective function in VDM

VDM[3] - General Signal-Noise-Ratio perspective of Diffusion Model

consistency/diffusion term

$$\sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \mid p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]$$

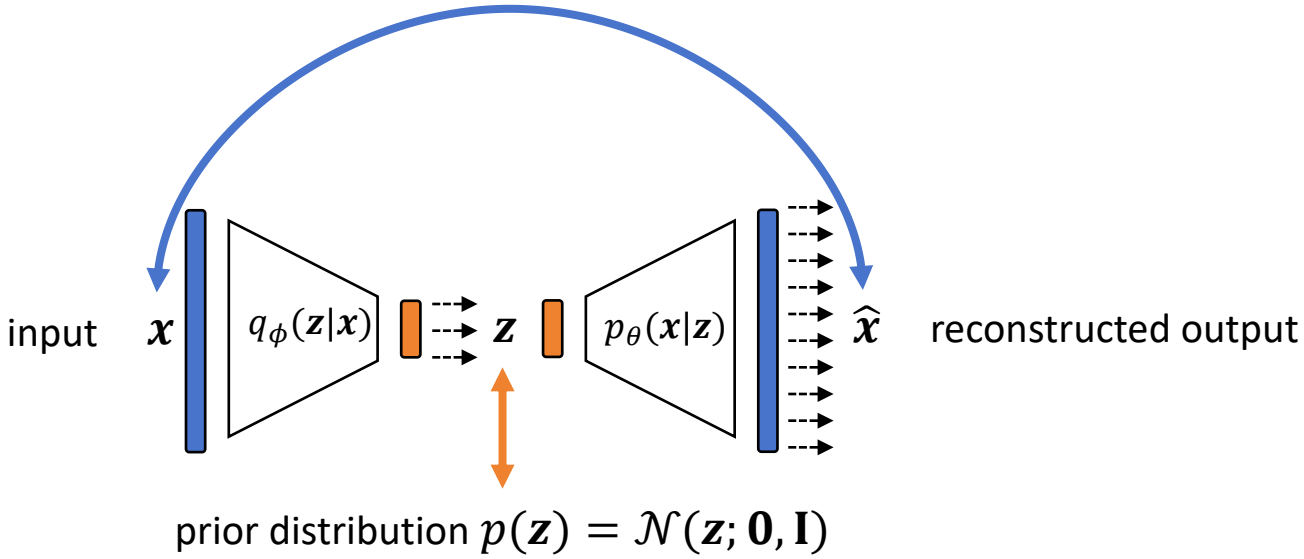
$$\begin{aligned} & D_{KL}(q(\boldsymbol{x}_s|\boldsymbol{x}_t, \boldsymbol{x}_0) \mid p_\theta(\boldsymbol{x}_s|\boldsymbol{x}_t)) \\ &= \frac{1}{2\,\sigma^2(s,t)} \| \mu(s,t) - \mu_\theta(s,t)\|_2^2 \\ &= \frac{1}{2}\left(\frac{\alpha_s^2}{\sigma_s^2} - \frac{\alpha_t^2}{\sigma_t^2}\right) \| \boldsymbol{x}_0 - \boldsymbol{x}_\theta\|_2^2 \\ &= \frac{1}{2}(\mathrm{SNR}(s) - \mathrm{SNR}(t)) \| \boldsymbol{x}_0 - \boldsymbol{x}_\theta(\boldsymbol{x}_t, t)\|_2^2 \end{aligned}$$

$$\frac{1}{2}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I}),\, t\sim\mathcal{U}(0,T)}[(\mathrm{SNR}(t-1) - \mathrm{SNR}(t)) \| \boldsymbol{x}_0 - \boldsymbol{x}_\theta(\alpha_t \boldsymbol{x}_0 + \sigma_t \epsilon, t)\|_2^2]$$

reparameterization trick

$$\boldsymbol{x}_\theta(\boldsymbol{x}_t, t) = (\boldsymbol{x}_t - \sigma_t \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t))/\alpha_t$$

$$\frac{1}{2}\mathbb{E}_{\epsilon\sim\mathcal{N}(0,\mathbf{I}),\, t\sim\mathcal{U}(0,T)}[(\mathrm{SNR}(t-1) - \mathrm{SNR}(t))\,(\sigma_t/\alpha_t) \| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}, t)\|_2^2]$$



Hierarchical VAE perspective of DDPM[2]

- Revisiting VAE and Evidence Lower BOund
- Hierarchical VAE perspective of Diffusion Model
- Reparameterization Triangle

Evidence Lower BOund of VI and VAE

DDPM[2] - Revisiting VAE and Evidence Lower BOund

Evidence Lower BOund of VI

$$\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z})}\right]$$

Proof.

$$\log p(\boldsymbol{x};\theta) = \int q_\phi(\boldsymbol{z}) \log p(\boldsymbol{x};\theta) dz$$

$$= \int q_\phi(\boldsymbol{z})(\log p(\boldsymbol{x},\boldsymbol{z};\theta) - \log p(\boldsymbol{z}|\boldsymbol{x};\theta))\, dz$$

$$= \int q_\phi(\boldsymbol{z}) \log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z})} dz - \int q_\phi(\boldsymbol{z}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x};\theta)}{q_\phi(\boldsymbol{z})} dz$$

$$= \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z})}\right] + D_{KL}(q(\boldsymbol{z}) \parallel p(\boldsymbol{z}|\boldsymbol{x};\theta))$$

$$\geq \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z})}\right]$$

■

Evidence Lower BOund of VAE

$$\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right]$$

Proof.

$$\log p(\boldsymbol{x};\theta) = \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log p(\boldsymbol{x};\theta) dz$$

$$= \int q_\phi(\boldsymbol{z}|\boldsymbol{x})(\log p(\boldsymbol{x},\boldsymbol{z};\theta) - \log p(\boldsymbol{z}|\boldsymbol{x};\theta))\, dz$$

$$= \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} dz - \int q_\phi(\boldsymbol{z}|\boldsymbol{x}) \log \frac{p(\boldsymbol{z}|\boldsymbol{x};\theta)}{q_\phi(\boldsymbol{z}|\boldsymbol{x})} dz$$

$$= \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] + D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z}|\boldsymbol{x};\theta))$$

$$\geq \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] = \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})\, p(\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right]$$

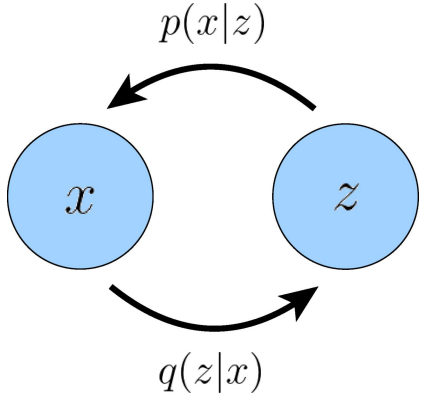
$$= \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \mid p(\boldsymbol{z}))$$

■

Variational Auto Encoder

DDPM[2] - Revisiting VAE and Evidence Lower BOund

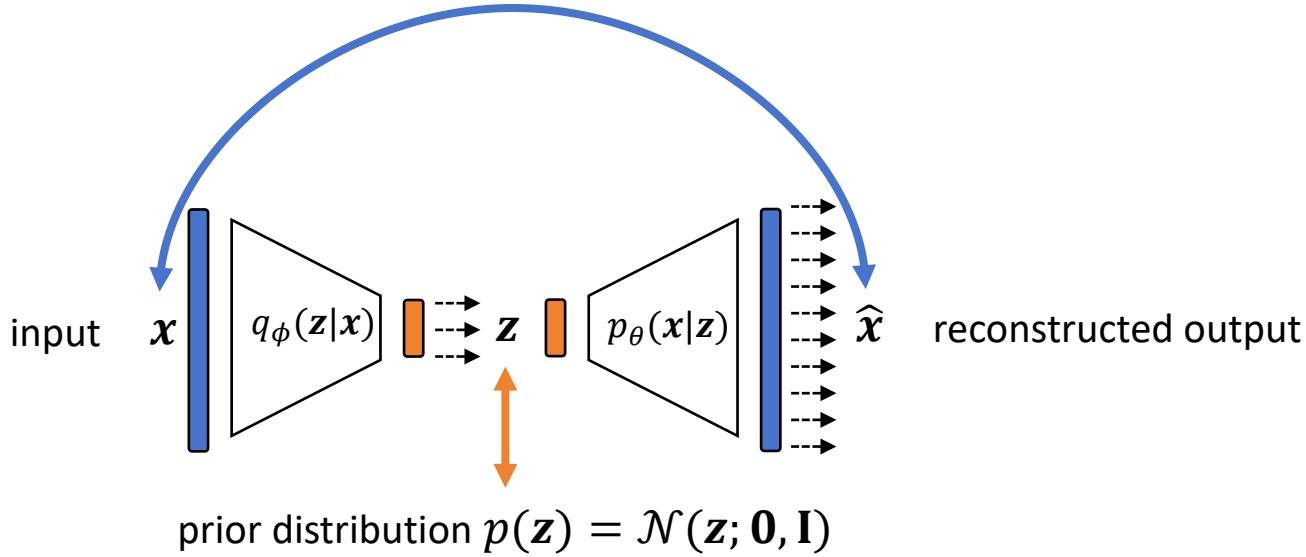
Graph Structure



ELBO Objective of VAE

$$\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] = \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \mid p(\boldsymbol{z}))$$

reconstruction term *prior matching term*



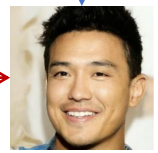
From VAE to Hierarchical VAE

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

VAE

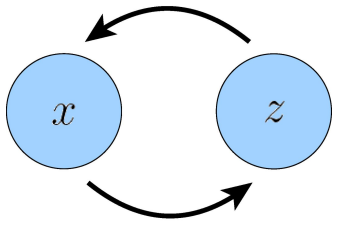
reconstruct

input →



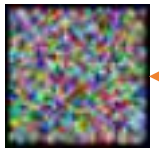
Backward (Decode)

$p(x|z)$



$q(z|x)$

Forward (Encode)



↔ prior distribution

consistency term

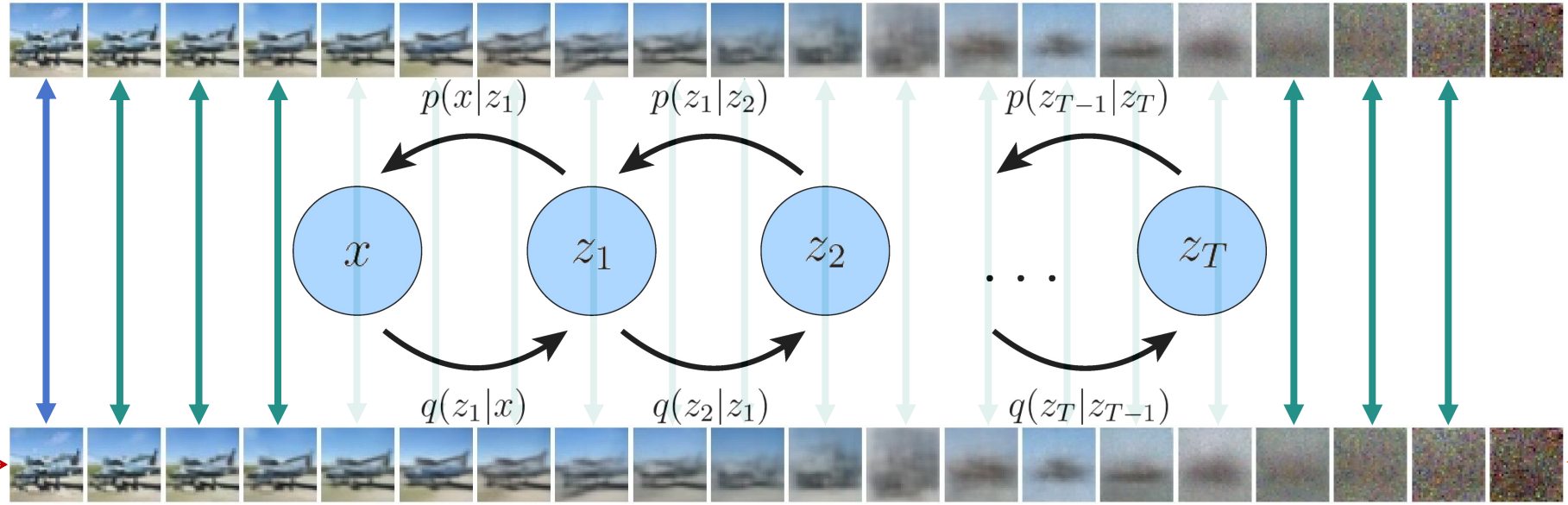
prior matching term

reconstruction term

Add hierarchical structure

VAE (Hierarchical)

Backward Process



input →

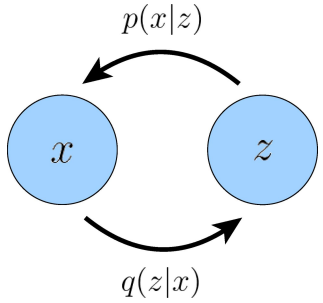
↔ prior distribution

Forward Process

From VAE to Hierarchical VAE

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

Graph Structure



ELBO Objective of VAE

$$\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] = \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}) \mid p(\boldsymbol{z}))$$

reconstruction term *prior matching term*

ELBO Objective of Hierarchical VAE

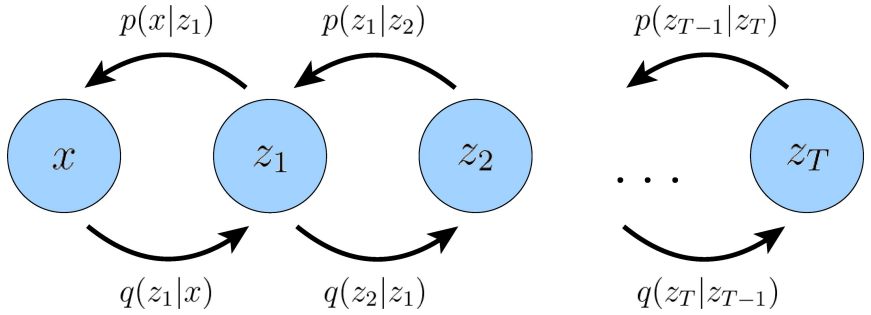
$$\mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z}_{1:T};\theta)}{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\right]$$

$$= \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}_1) + \log \frac{p(\boldsymbol{z}_T)}{q_\phi(\boldsymbol{z}_T|\boldsymbol{z}_{T-1})} + \log \frac{\prod_{t=1}^{T-1} p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t+1})}{\prod_{t=1}^{T-1} q_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1})}\right]$$

$$= \mathbb{E}_{q_\phi(\boldsymbol{z}_1|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z}_1)] - \mathbb{E}_{q_\phi(\boldsymbol{z}_{T-1}|\boldsymbol{x})}[D_{KL}(q_\phi(\boldsymbol{z}_T|\boldsymbol{z}_{T-1}) \mid p(\boldsymbol{z}_T))] - \sum_{t=1}^{T-1} \mathbb{E}_{q_\phi(\boldsymbol{z}_{t-1},\boldsymbol{z}_{t+1}|\boldsymbol{x})}[D_{KL}(q_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) \mid p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t+1}))]$$

reconstruction term *prior matching term* *consistency/diffusion term*

Graph Structure



From Hierarchical VAE to Diffusion Model

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

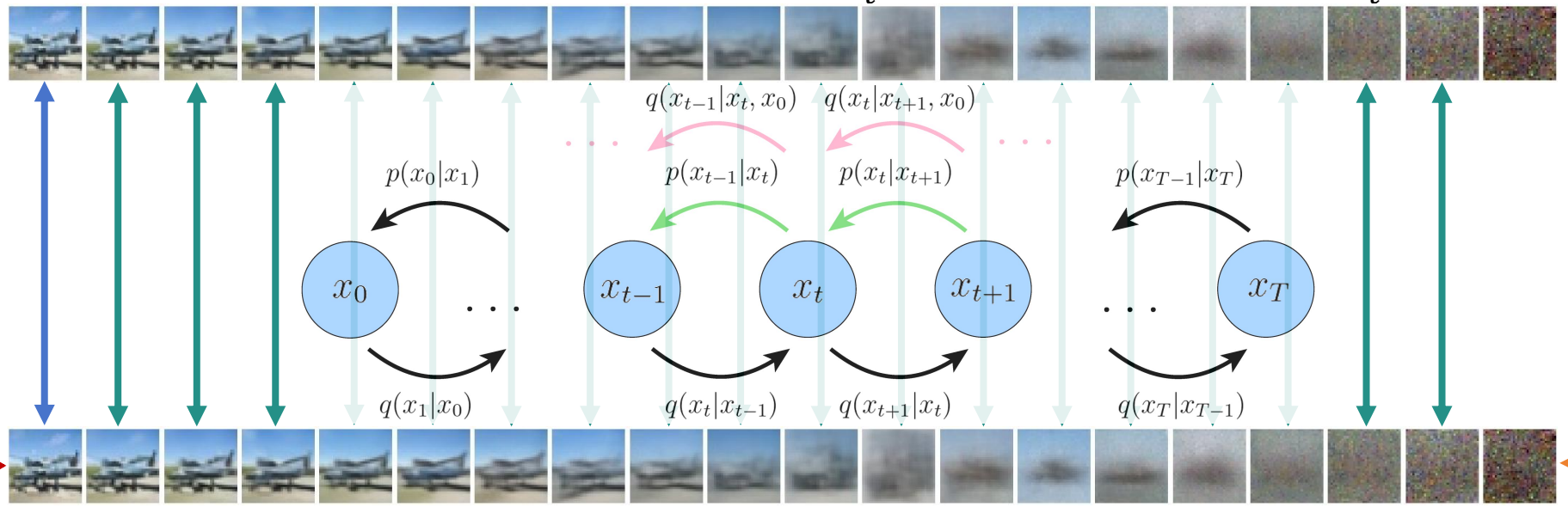
consistency term

prior matching term

reconstruction term

Backward Process (Denoising)

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \propto \mathcal{N}(\boldsymbol{x}_{t-1};\ \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\overline{\alpha}_t}, \frac{(1-\alpha_t)(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\mathbf{I})$$



input →

prior distribution

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t;\ \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

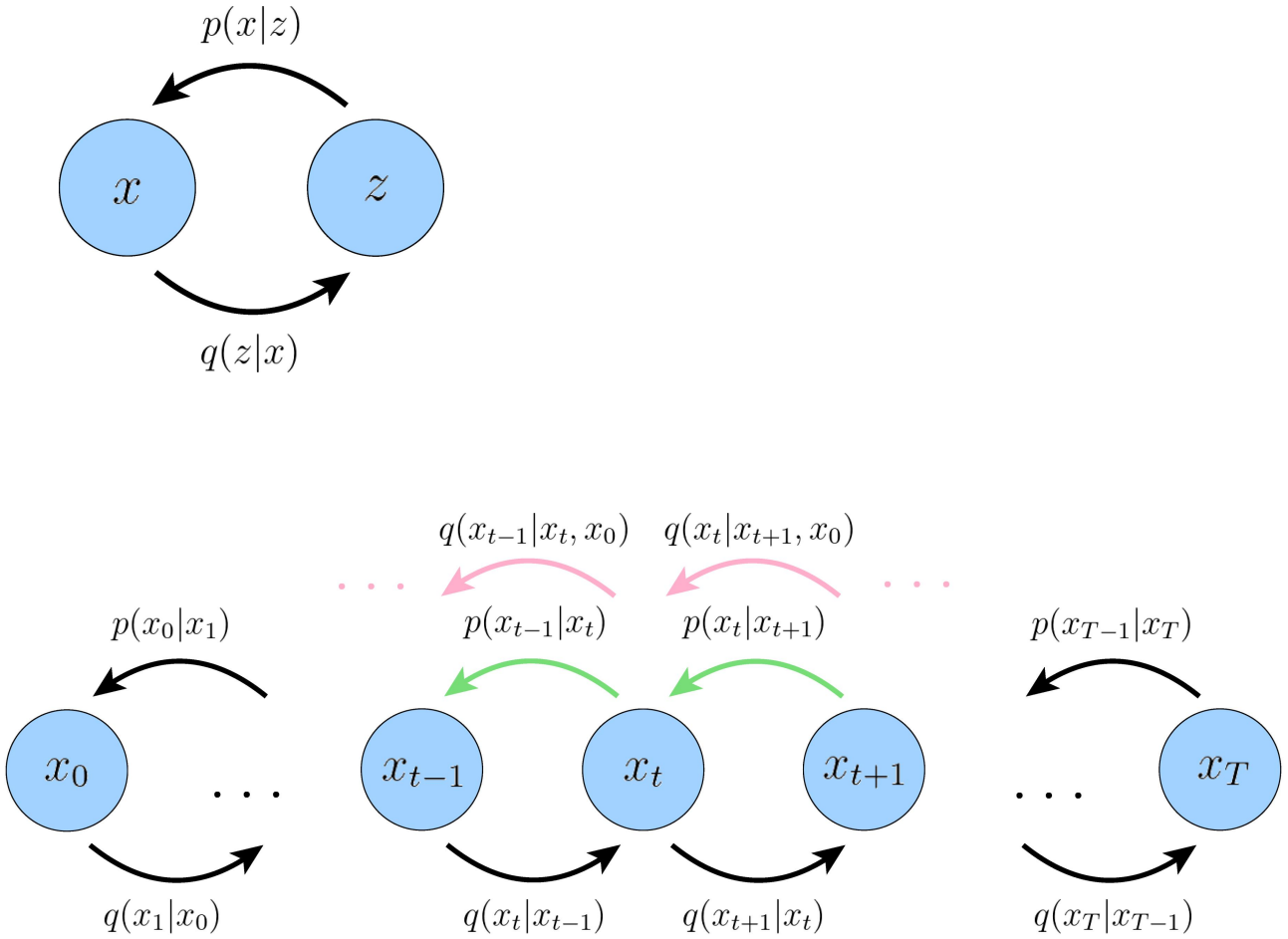
Forward Process (add noise)

Benifits.
The forward process is "deterministic" without neural network. Instead of minimizing the $D_{KL}(q_\phi(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}) \mid p_\theta(\boldsymbol{z}_t|\boldsymbol{z}_{t+1}))$, diffision model minimizes $D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \mid p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))$ which have more supervised signal (more stable to train).

Let us derive the objective function

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

Graph Structure



ELBO Objective of VAE

$$\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z};\theta)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\right]$$

$$\boldsymbol{z} \rightarrow \boldsymbol{x}_{1:T}$$

1. Replace single variable distribution with the hierarchical joint distribution

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

2. Decomposite the joint distribution

Let us derive the objective function

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

Joint Distribution Decomposite

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

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} \frac{\prod_{t=2}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\prod_{t=2}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{x}_0)}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} \frac{\prod_{t=2}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\prod_{t=2}^{T} \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)q(\boldsymbol{x}_t|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)}}\right]$$

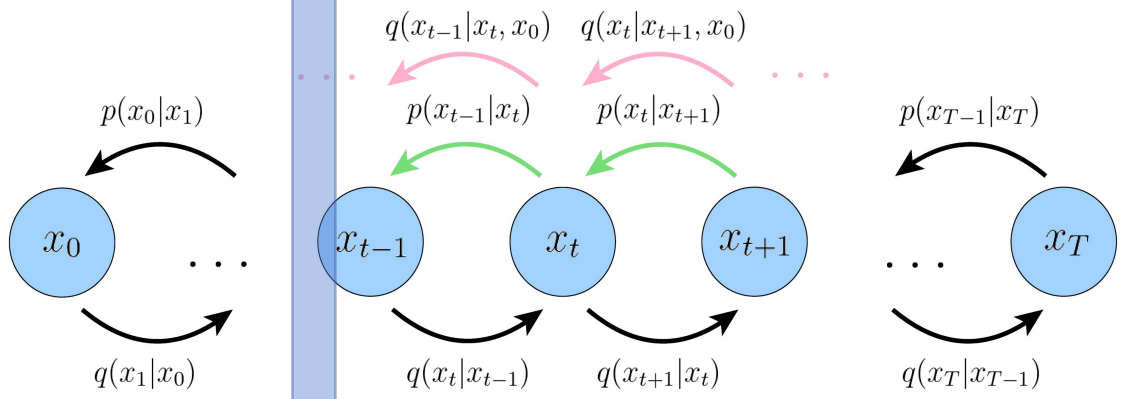
$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_1|\boldsymbol{x}_0)} + \log \frac{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)} + \log \prod_{t=2}^{T} \frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{\prod_{t=2}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\prod_{t=2}^{T} q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)] + \mathbb{E}_{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_0)}\right] + \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)}\left[\log \frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)}\right]$$

$$= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)] - D_{KL}(q(\boldsymbol{x}_T|\boldsymbol{x}_0) \mid p(\boldsymbol{x}_T)) - \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0) \mid p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]$$

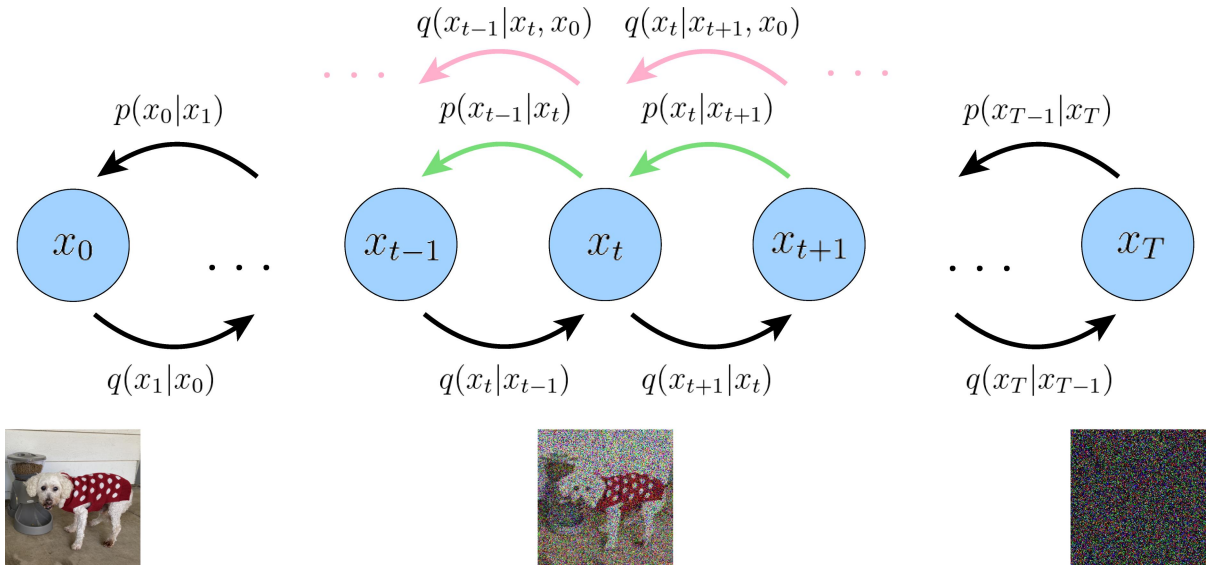
Graph Structure



Let us derive the objective function

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

Graph Structure



$$\mathbb{E}_{q(\boldsymbol{x_1}|\boldsymbol{x_0})}[\log p_\theta(\boldsymbol{x_0}|\boldsymbol{x_1})] - D_{KL}(q(\boldsymbol{x_T}|\boldsymbol{x_0}) \mid p(\boldsymbol{x_T})) - \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x_t}|\boldsymbol{x_0})}[D_{KL}(q(\boldsymbol{x_{t-1}}|\boldsymbol{x_t}, \boldsymbol{x_0}) \mid p_\theta(\boldsymbol{x_{t-1}}|\boldsymbol{x_t}))]$$

Can be ignored

reconstruction term *prior matching term* *consistency/diffusion term*

3. Combine with the deterministic noise schedule.

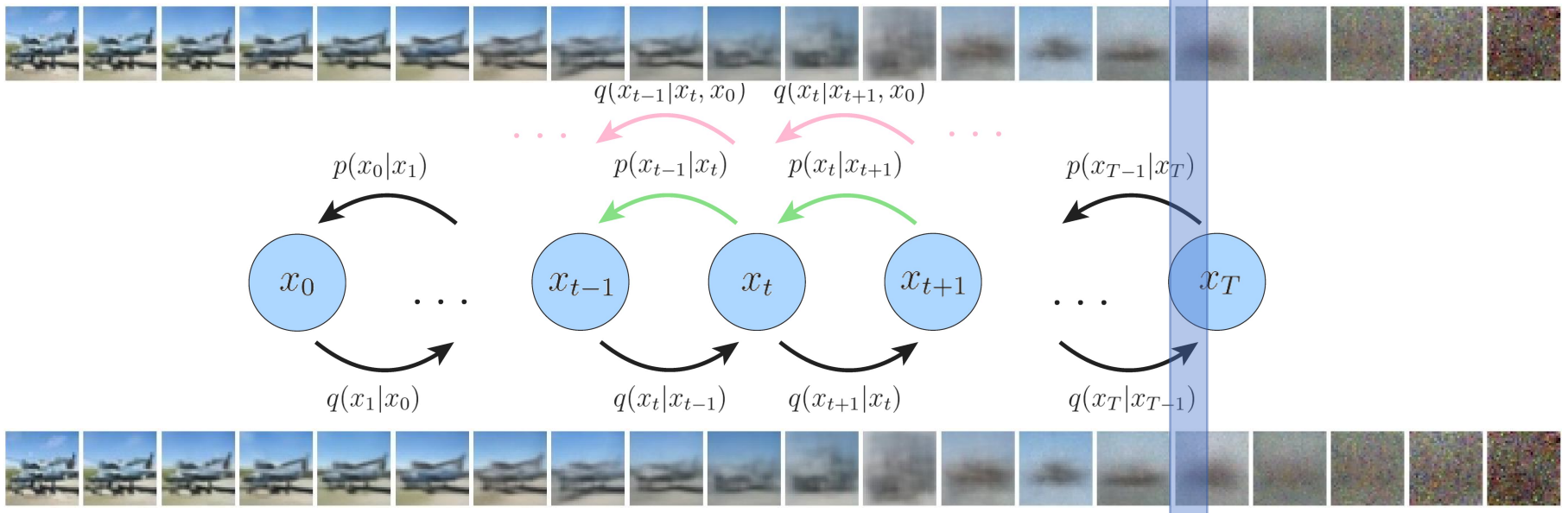
Let us derive the objective function

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

Deterministic Noise Schedule

Backward Process (Denoising)

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \propto \mathcal{N}(\boldsymbol{x}_{t-1};\ \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x}_0}{1-\overline{\alpha}_t}, \frac{(1-\alpha_t)(1-\overline{\alpha}_{t-1})}{1-\overline{\alpha}_t}\mathbf{I})$$



$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t;\ \sqrt{\alpha_t}\boldsymbol{x}_{t-1}, (1-\alpha_t)\mathbf{I})$$

Forward Process (add noise)

Let us derive the objective function

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

ground-truth

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x_0}}{1-\overline{\alpha}_t}$$

neural approxiamation

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) \quad = \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)\widehat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$$

consistency/diffusion term

$$\arg\min_{\boldsymbol{\theta}} \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \mid p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]$$

$$= \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\frac{1}{2\sigma_q^2(\mathrm{t})}\left[\|\boldsymbol{\mu}_q - \boldsymbol{\mu}_{\boldsymbol{\theta}}\|_2^2\right]]$$

$$= \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[\frac{1}{2\sigma_q^2(\mathrm{t})} \frac{\overline{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\overline{\alpha}_t)^2}[\|\boldsymbol{x_0} - \widehat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2]]$$

4. Rewrite the objective with ε - prediction.

Let us derive the objective function

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

reparameterization trick

$$\boldsymbol{x}_t = \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon_0$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t;\ \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\mathbf{I})$$

$$\boldsymbol{x_0} = \frac{\boldsymbol{x}_t - \sqrt{1-\overline{\alpha}_t}\epsilon_0}{\sqrt{\overline{\alpha}_t}}$$

ground-truth

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x_0}}{1-\overline{\alpha}_t}$$

$$= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}\sqrt{\alpha_t}}\epsilon_0$$

neural approxiamation

$$\boldsymbol{\mu_\theta}(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$$

$$= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}\sqrt{\alpha_t}}\hat{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

Let us derive the objective function

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

ground-truth

$$\boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)\boldsymbol{x_0}}{1-\overline{\alpha}_t}$$
$$= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}\sqrt{\alpha_t}}\epsilon_0$$

neural approxiamation

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\overline{\alpha}_{t-1})\boldsymbol{x}_t + \sqrt{\overline{\alpha}_{t-1}}(1-\alpha_t)\hat{\boldsymbol{x}}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)}{1-\overline{\alpha}_t}$$
$$= \frac{1}{\sqrt{\alpha_t}}\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}\sqrt{\alpha_t}}\hat{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)$$

See proof in [4]

consistency/diffusion term

$$\arg\min_{\boldsymbol{\theta}} \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \mid p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\frac{1}{2\sigma_q^2(\mathrm{t})} \left[\left\| \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}\sqrt{\alpha_t}} (\epsilon_0 - \hat{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)) \right\|_2^2 \right] \right]$$

$$= \arg\min_{\boldsymbol{\theta}} \sum_{t=2}^{T} \mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\frac{1}{2\sigma_q^2(t)} \frac{(1-\alpha_t)^2}{(1-\overline{\alpha}_t)\alpha_t} [\|\epsilon_0 - \hat{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|_2^2] \right]$$

Can be ignored

Denoising Diffusion Probabilistic Model (DDPM)[2]

$$\mathcal{L}(\theta) = \frac{1}{2}\mathbb{E}_{t\sim[0,1]}\mathbb{E}_{\boldsymbol{x}\sim\boldsymbol{q_{data}}(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{\varepsilon}\sim\mathcal{N}(0,\sigma^2\mathbf{I})}\left[\left\| \boldsymbol{\varepsilon}_{\boldsymbol{\theta}}(\sqrt{\overline{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\overline{\alpha}_t}\boldsymbol{\varepsilon}) - \boldsymbol{\varepsilon}\right\|_2^2\right]$$

Training and sampling in DDPM

DDPM[2] - Hierarchical VAE perspective of Diffusion Model

Algorithm Definition

Algorithm 1 Training

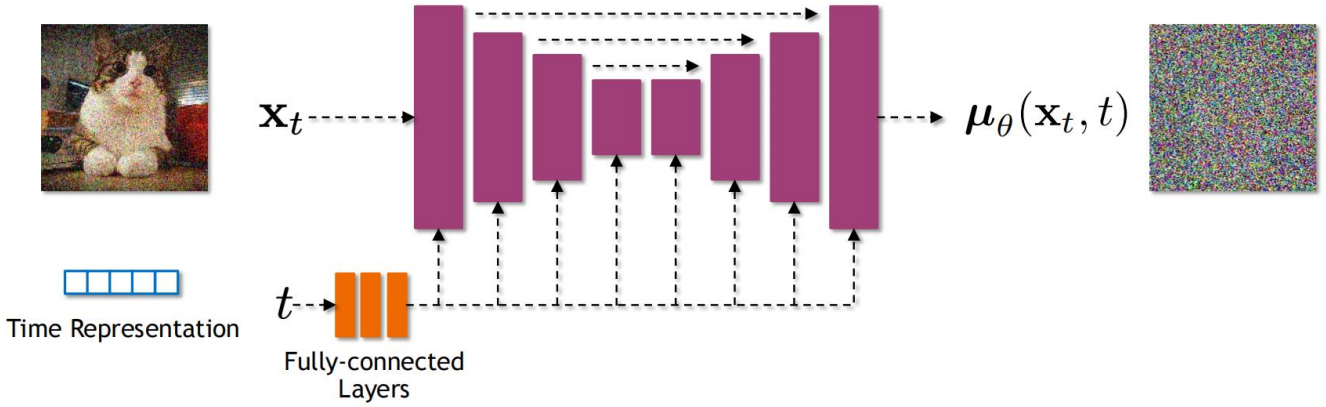
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Nueral Network Structure

Diffusion models often use U-Net architectures with ResNet blocks, skip connections, and self-attention layers.



Time representation: sinusoidal positional embeddings or random Fourier features.

Time features are fed to residual blocks using either simple spatial addition or adaptive group normalization layers.

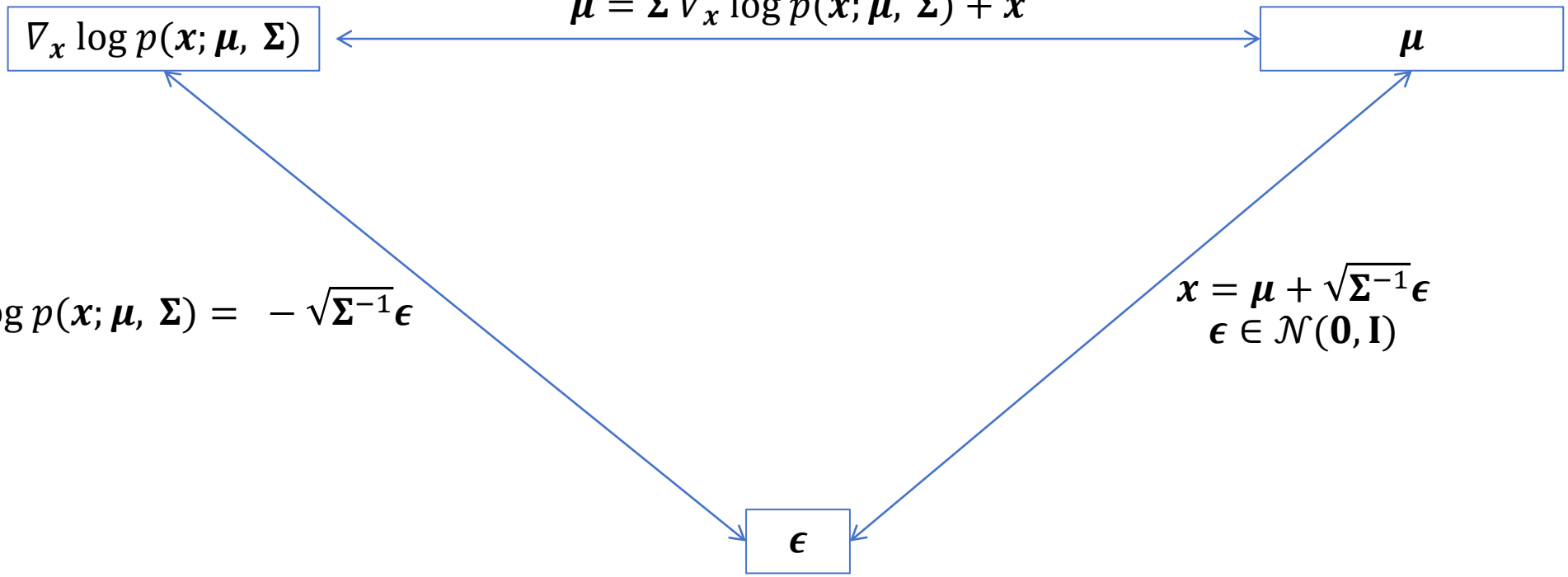
Reparameterization Triangle

DDPM[2] - Reparameterization Triangle

$$p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^N \det \boldsymbol{\Sigma}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \nabla_{\boldsymbol{x}}\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right) = -\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$$

$$\boldsymbol{\mu} = \boldsymbol{\Sigma}\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) + \boldsymbol{x}$$



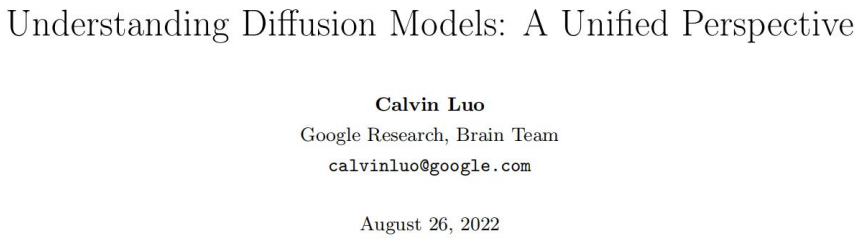
$$\boldsymbol{\Sigma}\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = -\sqrt{\boldsymbol{\Sigma}^{-1}}\boldsymbol{\epsilon}$$

$$\boldsymbol{x} = \boldsymbol{\mu} + \sqrt{\boldsymbol{\Sigma}^{-1}}\boldsymbol{\epsilon}$$
$$\boldsymbol{\epsilon} \in \mathcal{N}(\mathbf{0},\mathbf{I})$$

Further Readings

https://yang-song.net/blog/2021/score/

https://slideslive.com/39006359/



https://arxiv.org/abs/2208.11970

Reference

[1] Song Y, Ermon S. Generative modeling by estimating gradients of the data distribution[J]. Advances in neural information processing systems, 2019, 32.
[2] Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in neural information processing systems, 2020, 33: 6840-6851.
[3] Kingma D, Salimans T, Poole B, et al. Variational diffusion models[J]. Advances in neural information processing systems, 2021, 34: 21696-21707.
[4] Luo C. Understanding diffusion models: A unified perspective[J]. arXiv preprint arXiv:2208.11970, 2022.