

Deep Generative Models

Lecture 10

Roman Isachenko



2025, Spring

Recap of previous lecture

Forward process goes from any distribution $\pi(\mathbf{x})$ to $\mathcal{N}(0, \mathbf{I})$ via noise injection.

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\sqrt{1 - \beta_t} \cdot \mathbf{x}_{t-1}, \beta_t \cdot \mathbf{I});$$
$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0, (1 - \bar{\alpha}_t) \cdot \mathbf{I}).$$

Reverse process is Intractable distribution that is able to be approximated by Normal (with unknown parameters) for small β_t .

$$q(\mathbf{x}_{t-1} | \mathbf{x}_t) = \frac{q(\mathbf{x}_t | \mathbf{x}_{t-1}) q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t)} \approx \mathcal{N}(\boldsymbol{\mu}_{\theta,t}(\mathbf{x}_t), \boldsymbol{\sigma}_{\theta,t}^2(\mathbf{x}_t))$$

Conditioned reverse process is Normal with the known parameters, which defines how to denoise a noisy image \mathbf{x}_t with access to what the final, completely denoised image \mathbf{x}_0 should be.

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

Recap of previous lecture

- ▶ $\mathbf{z} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$ is a latent variable.
- ▶ Variational posterior distribution

$$q(\mathbf{z}|\mathbf{x}) = q(\mathbf{x}_1, \dots, \mathbf{x}_T|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}).$$

- ▶ Generative distribution and prior

$$p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta}) = p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}); \quad p(\mathbf{z}|\boldsymbol{\theta}) = \prod_{t=2}^T p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) \cdot p(\mathbf{x}_T)$$

ELBO

$$\log p(\mathbf{x}|\boldsymbol{\theta}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log \frac{p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta})}{q(\mathbf{z}|\mathbf{x})} = \mathcal{L}_{\phi, \boldsymbol{\theta}}(\mathbf{x}) \rightarrow \max_{q, \boldsymbol{\theta}}$$

$$\begin{aligned} \mathcal{L}_{\phi, \boldsymbol{\theta}}(\mathbf{x}) &= \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta}) - KL(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)) - \\ &\quad - \sum_{t=2}^T \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} KL(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}))}_{\mathcal{L}_t} \end{aligned}$$

Recap of previous lecture

ELBO of Gaussian diffusion model

$$\begin{aligned}\mathcal{L}_{\phi, \theta}(\mathbf{x}) = & \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p(\mathbf{x}_0|\mathbf{x}_1, \theta) - KL(q(\mathbf{x}_T|\mathbf{x}_0)||p(\mathbf{x}_T)) - \\ & - \sum_{t=2}^T \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} KL(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)||p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta))}_{\mathcal{L}_t}\end{aligned}$$

$$\begin{aligned}q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= \mathcal{N}(\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}), \\ p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta) &= \mathcal{N}(\boldsymbol{\mu}_{\theta, t}(\mathbf{x}_t), \sigma_{\theta, t}^2(\mathbf{x}_t))\end{aligned}$$

Our assumption: $\sigma_{\theta, t}^2(\mathbf{x}_t) = \tilde{\beta}_t \mathbf{I}$.

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[\frac{1}{2\tilde{\beta}_t} \|\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) - \boldsymbol{\mu}_{\theta, t}(\mathbf{x}_t)\|^2 \right]$$

Recap of previous lecture

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{x}_t | \mathbf{x}_0)} \left[\frac{1}{2\tilde{\beta}_t} \|\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_{\theta,t}(\mathbf{x}_t)\|^2 \right]$$

Reparametrization

$$\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} \cdot \epsilon$$

$$\mu_{\theta,t}(\mathbf{x}_t) = \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} \cdot \epsilon_{\theta,t}(\textcolor{teal}{x}_t)$$

$$\mathcal{L}_t = \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} \left[\frac{(1 - \alpha_t)^2}{2\tilde{\beta}_t \alpha_t (1 - \bar{\alpha}_t)} \left\| \epsilon - \epsilon_{\theta,t}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon) \right\|^2 \right]$$

At each step of reverse diffusion process we try to predict the noise ϵ that we used in the forward diffusion process!

Simplified objective

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t \sim U\{2, T\}} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} \left\| \epsilon - \epsilon_{\theta,t}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon) \right\|^2$$

Recap of previous lecture

Training of DDPM

1. Get the sample $\mathbf{x}_0 \sim \pi(\mathbf{x})$.
2. Sample timestamp $t \sim U\{1, T\}$ and the noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.
3. Get noisy image $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \epsilon$.
4. Compute loss $\mathcal{L}_{\text{simple}} = \|\epsilon - \epsilon_{\theta, t}(\mathbf{x}_t)\|^2$.

Sampling of DDPM

1. Sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.
2. Compute mean of $p(\mathbf{x}_{t-1} | \mathbf{x}_t, \theta) = \mathcal{N}(\mu_{\theta, t}(\mathbf{x}_t), \sigma_t^2 \cdot \mathbf{I})$:

$$\mu_{\theta, t}(\mathbf{x}_t) = \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} \cdot \epsilon_{\theta, t}(\mathbf{x}_t)$$

3. Get denoised image $\mathbf{x}_{t-1} = \mu_{\theta, t}(\mathbf{x}_t) + \sigma_t \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

Outline

1. DDPM as score-based generative model
2. Guidance
 - Classifier guidance
 - Classifier-free guidance
3. Continuous-in-time normalizing flows

Outline

1. DDPM as score-based generative model

2. Guidance

Classifier guidance

Classifier-free guidance

3. Continuous-in-time normalizing flows

Denoising diffusion as score-based generative model

DDPM objective

$$\begin{aligned}\mathcal{L}_t &= \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} \left[\frac{(1 - \alpha_t)^2}{2\tilde{\beta}_t \alpha_t (1 - \bar{\alpha}_t)} \|\epsilon_{\theta, t}(\mathbf{x}_t) - \epsilon\|_2^2 \right] \\ &= \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \mathbf{I})} \left[\frac{(1 - \alpha_t)^2}{2\tilde{\beta}_t \alpha_t} \left\| \frac{\epsilon_{\theta, t}(\mathbf{x}_t)}{\sqrt{1 - \bar{\alpha}_t}} - \frac{\epsilon}{\sqrt{1 - \bar{\alpha}_t}} \right\|_2^2 \right]\end{aligned}$$

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

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t | \mathbf{x}_0) = -\frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0}{1 - \bar{\alpha}_t} = -\frac{\epsilon}{\sqrt{1 - \bar{\alpha}_t}}.$$

Let reparametrize our model:

$$\mathbf{s}_{\theta, t}(\mathbf{x}_t) = \frac{\epsilon_{\theta, t}(\mathbf{x}_t)}{\sqrt{1 - \bar{\alpha}_t}} = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \theta).$$

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{x}_t | \mathbf{x}_0)} \left[\frac{(1 - \alpha_t)^2}{2\tilde{\beta}_t \alpha_t} \left\| \mathbf{s}_{\theta, t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t | \mathbf{x}_0) \right\|_2^2 \right]$$

DDPM vs NCSN: objectives

DDPM objective

$$\mathbb{E}_{\pi(\mathbf{x}_0)} \mathbb{E}_{t \sim U\{1, T\}} \mathbb{E}_{q(\mathbf{x}_t | \mathbf{x}_0)} \left[\frac{(1 - \alpha_t)^2}{2\tilde{\beta}_t \alpha_t} \left\| \mathbf{s}_{\theta, t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t | \mathbf{x}_0) \right\|_2^2 \right]$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \boldsymbol{\epsilon}$$

In practice the coefficient is omitted.

NCSN objective

$$\mathbb{E}_{\pi(\mathbf{x}_0)} \mathbb{E}_{t \sim U\{1, T\}} \mathbb{E}_{q(\mathbf{x}_t | \mathbf{x}_0)} \left\| \mathbf{s}_{\theta, \sigma_t}(\mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t | \mathbf{x}_0) \right\|_2^2$$

$$\mathbf{x}_t = \mathbf{x}_0 + \sigma_t \cdot \boldsymbol{\epsilon}$$

ELBO maximization approach gives the same objective as denoising score-matching approach!

DDPM vs NCSN: sampling

DDPM sampling (ancestral sampling)

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

$$\begin{aligned}\mathbf{x}_{t-1} &= \mu_{\theta,t}(\mathbf{x}_t) + \sigma_t \cdot \epsilon \\&= \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} \cdot \epsilon_{\theta,t}(\mathbf{x}_t) + \sigma_t \cdot \epsilon \\&= \frac{1}{\sqrt{1 - \beta_t}} \cdot \mathbf{x}_t + \frac{\beta_t}{\sqrt{1 - \beta_t}} \cdot \mathbf{s}_{\theta,t}(\mathbf{x}_t) + \sigma_t \cdot \epsilon\end{aligned}$$

NCSN sampling (annealed Langevin dynamics)

- ▶ Sample $\mathbf{x}_T^0 \sim \mathcal{N}(0, \sigma_T^2 \cdot \mathbf{I}) \approx q(\mathbf{x}_T)$.
- ▶ Apply L steps of Langevin dynamic

$$\mathbf{x}_t^l = \mathbf{x}_t^{l-1} + \frac{\eta_t}{2} \cdot \mathbf{s}_{\theta,\sigma_t}(\mathbf{x}_t^{l-1}) + \sqrt{\eta_t} \cdot \epsilon_t^l.$$

- ▶ Update $\mathbf{x}_{t-1}^0 = \mathbf{x}_t^L$ and choose the next σ_t .

DDPM vs NCSN

Summary

- ▶ Different Markov chains:
 - ▶ DDPM: $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \cdot \boldsymbol{\epsilon};$
 - ▶ NCSN: $\mathbf{x}_t = \mathbf{x}_0 + \sigma_t \cdot \boldsymbol{\epsilon}.$
 - ▶ It is possible to consider the more general framework
 $q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}(\alpha_t \cdot \mathbf{x}_0, \sigma_t^2 \cdot \mathbf{I})$
- ▶ Identical objectives: ELBO \equiv score-matching.
- ▶ Different sampling schemes:
 - ▶ ancestral sampling for DDPM;
 - ▶ annealed Langevin dynamics for NCSN;
 - ▶ there is a combined approach with alternating updates of DDPM and NCSN.

Kingma D. et al. *Variational Diffusion Models*, 2021

Song Y. et al. *Score-Based Generative Modeling through Stochastic Differential Equations*, 2020

Outline

1. DDPM as score-based generative model

2. Guidance

Classifier guidance

Classifier-free guidance

3. Continuous-in-time normalizing flows

Guidance

- ▶ Throughout the whole course we have discussed unconditional generative models $p(\mathbf{x}|\theta)$.
- ▶ In practice the majority of the generative models are **conditional**: $p(\mathbf{x}|\mathbf{y}, \theta)$.
- ▶ Here \mathbf{y} could be the class label or **text** (for text-to-image models).



Кот ныряет в бассейн, как ребенок на обложке альбома Nevermind, реалистично



рука человека с пятью пальцами, ни четырьмя, ни шестью, а с 5 (пять) пальцами

Taxonomy of conditional tasks

In practice the popular task is to create a conditional model $\pi(x|y)$.

- ▶ y – class label, x – image \Rightarrow image conditional model.
- ▶ y – text prompt, x – image \Rightarrow text-to-image model.
- ▶ y – image, x – image \Rightarrow image-to-image model.
- ▶ y – image, x – text \Rightarrow image-to-text model (image captioning).
- ▶ y – sound, x – text \Rightarrow speech-to-text model (automatic speech recognition).
- ▶ y – English text, x – Russian text \Rightarrow sequence-to-sequence model (machine translation).
- ▶ $y = \emptyset$, x – image \Rightarrow image unconditional model.

Label guidance

Label: Ostrich (10th ImageNet class)



VQ-VAE (Proposed)

BigGAN deep

Text guidance

Prompt: a stained glass window of a panda eating bamboo

Left: $\gamma = 1$, Right: $\gamma = 3$.



Guidance

- ▶ If we have **supervised** data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ we could treat \mathbf{y} as additional model input:
 - ▶ $p(x_j | \mathbf{x}_{1:j-1}, \mathbf{y}, \theta)$ for AR;
 - ▶ Encoder $q(\mathbf{z} | \mathbf{x}, \mathbf{y}, \phi)$ and decoder $p(\mathbf{x} | \mathbf{z}, \mathbf{y}, \theta)$ for VAE;
 - ▶ $G_\theta(\mathbf{z}, \mathbf{y})$ for NF and GAN;
 - ▶ $p(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{y}, \theta)$ for DDPM.
- ▶ If we have **unsupervised** data $\{\mathbf{x}_i\}_{i=1}^n$ we need to create the way to convert unconditional model $p(\mathbf{x} | \theta)$ to the conditional.
- ▶ It is really helpful to have the way to control the power of guidance.

Guidance types

- ▶ **Classifier guidance:**
 - ▶ suitable for unsupervised data;
 - ▶ uses the additional classifier model (we need supervised data to train the classifier).
- ▶ **Classifier-free guidance:**
 - ▶ suitable for supervised data;
 - ▶ get rid of the additional classifier model.

Outline

1. DDPM as score-based generative model

2. Guidance

Classifier guidance

Classifier-free guidance

3. Continuous-in-time normalizing flows

Classifier guidance

DDPM sampling

1. Sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.
2. Get denoised image (unconditional generation)

$$\begin{aligned}\mathbf{x}_{t-1} &= \frac{1}{\sqrt{1 - \beta_t}} \cdot \mathbf{x}_t + \frac{\beta_t}{\sqrt{1 - \beta_t}} \cdot \mathbf{s}_{\theta, t}(\mathbf{x}_t) + \sigma_t \cdot \epsilon \\ &= \frac{1}{\sqrt{1 - \beta_t}} \cdot \mathbf{x}_t + \frac{\beta_t}{\sqrt{1 - \beta_t}} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \theta) + \sigma_t \cdot \epsilon\end{aligned}$$

Conditional generation

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{1 - \beta_t}} \cdot \mathbf{x}_t + \frac{\beta_t}{\sqrt{1 - \beta_t}} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{y}, \theta) + \sigma_t \cdot \epsilon$$

- ▶ Assume for simplicity \mathbf{y} is a class labels.
- ▶ Suppose that we have the distribution $p(\mathbf{y}|\mathbf{x}_t)$ – classifier on noisy samples.

Classifier guidance

Conditional generation

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{1 - \beta_t}} \cdot \mathbf{x}_t + \frac{\beta_t}{\sqrt{1 - \beta_t}} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{y}, \theta) + \sigma_t \cdot \epsilon$$

Conditional distribution

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{y}, \theta) &= \nabla_{\mathbf{x}_t} \log \left(\frac{p(\mathbf{y} | \mathbf{x}_t) p(\mathbf{x}_t | \theta)}{p(\mathbf{y})} \right) \\ &= \nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \theta) \\ &= \nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t) - \frac{\epsilon_{\theta, t}(\mathbf{x}_t)}{\sqrt{1 - \bar{\alpha}_t}}\end{aligned}$$

Let parametrize $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{y}, \theta) = -\frac{\epsilon_{\theta, t}(\mathbf{x}_t, \mathbf{y})}{\sqrt{1 - \bar{\alpha}_t}}$.

Classifier-corrected noise prediction

$$\epsilon_{\theta, t}(\mathbf{x}_t, \mathbf{y}) = \epsilon_{\theta, t}(\mathbf{x}_t) - \sqrt{1 - \bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y} | \mathbf{x}_t)$$

Classifier guidance

Classifier-corrected noise prediction

$$\epsilon_{\theta,t}(\mathbf{x}_t, \mathbf{y}) = \epsilon_{\theta,t}(\mathbf{x}_t) - \sqrt{1 - \bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

Guidance scale

$$\epsilon_{\theta,t}(\mathbf{x}_t, \mathbf{y}) = \epsilon_{\theta,t}(\mathbf{x}_t) - \gamma \cdot \sqrt{1 - \bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

Here we introduce **guidance scale γ** that controls the magnitude of the classifier guidance.

Training

- ▶ Train DDPM as usual.
- ▶ Train the additional classifier $p(\mathbf{y}|\mathbf{x}_t)$ on the noisy samples \mathbf{x}_t .

Guided sampling

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} \cdot \epsilon_{\theta,t}(\mathbf{x}_t, \mathbf{y}) + \sigma_t \cdot \epsilon$$

Classifier guidance

Classifier-corrected noise prediction

$$\epsilon_{\theta,t}(\mathbf{x}_t, \mathbf{y}) = \epsilon_{\theta,t}(\mathbf{x}_t) - \gamma \cdot \sqrt{1 - \bar{\alpha}_t} \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

Guidance-scaled conditional distribution

$$\frac{\epsilon_{\theta,t}(\mathbf{x}_t, \mathbf{y})}{\sqrt{1 - \bar{\alpha}_t}} = \frac{\epsilon_{\theta,t}(\mathbf{x}_t)}{\sqrt{1 - \bar{\alpha}_t}} - \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

$$\begin{aligned}\nabla_{\mathbf{x}_t}^{\gamma} \log p(\mathbf{x}_t|\mathbf{y}, \theta) &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\theta) + \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \\ &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\theta) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)^{\gamma} \\ &= \nabla_{\mathbf{x}_t} \log \left(\frac{p(\mathbf{y}|\mathbf{x}_t)^{\gamma} p(\mathbf{x}_t|\theta)}{Z} \right)\end{aligned}$$

Note: Guidance scale γ tries to sharpen the distribution $p(\mathbf{y}|\mathbf{x}_t)$ (in this case Z should not depend on \mathbf{x}_t).

Outline

1. DDPM as score-based generative model

2. Guidance

Classifier guidance

Classifier-free guidance

3. Continuous-in-time normalizing flows

Classifier-free guidance

- ▶ Previous method requires training the additional classifier model $p(\mathbf{y}|\mathbf{x}_t)$ on the noisy data.
- ▶ Let try to avoid this requirement.

$$\nabla_{\mathbf{x}_t}^{\gamma} \log p(\mathbf{x}_t|\mathbf{y}, \theta) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\theta) + \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$$

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) &= \nabla_{\mathbf{x}_t} \log \left(\frac{p(\mathbf{x}_t|\mathbf{y}, \theta)p(\mathbf{y})}{p(\mathbf{x}_t|\theta)} \right) \\ &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \theta) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\theta)\end{aligned}$$

$$\begin{aligned}\nabla_{\mathbf{x}_t}^{\gamma} \log p(\mathbf{x}_t|\mathbf{y}, \theta) &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\theta) + \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \\ &= \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\theta) + \gamma \cdot (\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \theta) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\theta)) = \\ &= (1 - \gamma) \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\theta) + \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \theta)\end{aligned}$$

Note: In the case of $\gamma = 1$ we will get the identity statement.

Classifier-free guidance

$$\nabla_{\mathbf{x}_t}^\gamma \log p(\mathbf{x}_t | \mathbf{y}, \theta) = (1 - \gamma) \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \theta) + \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t | \mathbf{y}, \theta)$$

$$\frac{\hat{\epsilon}_{\theta,t}(\mathbf{x}_t, \mathbf{y})}{\sqrt{1 - \bar{\alpha}_t}} = (1 - \gamma) \cdot \frac{\epsilon_{\theta,t}(\mathbf{x}_t)}{\sqrt{1 - \bar{\alpha}_t}} + \gamma \cdot \frac{\epsilon_{\theta,t}(\mathbf{x}_t, \mathbf{y})}{\sqrt{1 - \bar{\alpha}_t}}$$

Classifier-free-corrected noise prediction

$$\hat{\epsilon}_{\theta,t}(\mathbf{x}_t, \mathbf{y}) = \gamma \cdot \epsilon_{\theta,t}(\mathbf{x}_t, \mathbf{y}) + (1 - \gamma) \cdot \epsilon_{\theta,t}(\mathbf{x}_t)$$

- ▶ Train the single model $\epsilon_{\theta,t}(\mathbf{x}_t, \mathbf{y})$ on **supervised** data alternating with real conditioning \mathbf{y} and empty conditioning $\mathbf{y} = \emptyset$.
- ▶ Apply the model twice during inference.

Guided sampling

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} \cdot \hat{\epsilon}_{\theta,t}(\mathbf{x}_t, \mathbf{y}) + \sigma_t \cdot \epsilon$$

Outline

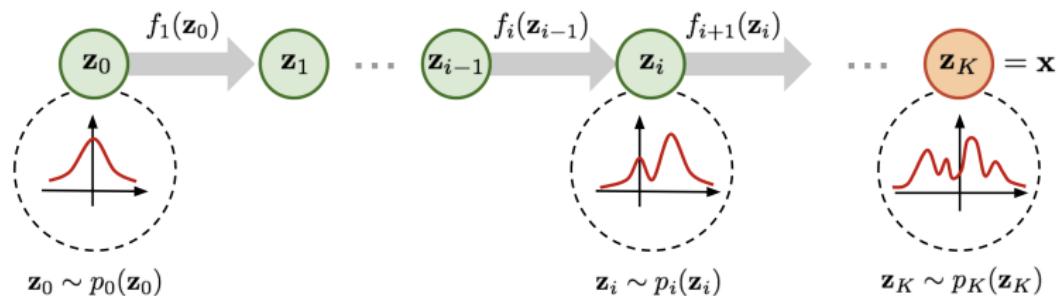
1. DDPM as score-based generative model
2. Guidance
 - Classifier guidance
 - Classifier-free guidance
3. Continuous-in-time normalizing flows

Discrete-in-time NF

Change of variable theorem (CoV)

Let \mathbf{x} be a random variable with density function $p(\mathbf{x})$ and $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ is a differentiable, **invertible** function. If $\mathbf{z} = \mathbf{f}(\mathbf{x})$, $\mathbf{x} = \mathbf{f}^{-1}(\mathbf{z}) = \mathbf{g}(\mathbf{z})$, then

$$p(\mathbf{x}) = p(\mathbf{z}) |\det(\mathbf{J}_\mathbf{f})| = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(\mathbf{f}(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$



$$\log p(\mathbf{x}|\theta) = \log p(\mathbf{f}_K \circ \dots \circ \mathbf{f}_1(\mathbf{x})) + \sum_{k=1}^K \log \left| \det \left(\frac{\partial \mathbf{f}_k}{\partial \mathbf{f}_{k-1}} \right) \right|.$$

Discrete-in-time NF

- ▶ Previously we assumed that the time axis is discrete:

$$\mathbf{x}_{t+1} = \mathbf{f}_\theta(\mathbf{x}_t, t); \quad \log p(\mathbf{x}_{t+1}) = \log p(\mathbf{x}_t) - \log \left| \det \frac{\partial \mathbf{f}_\theta(\mathbf{x}_t)}{\partial \mathbf{x}_t} \right|.$$

- ▶ Let consider the more general case of continuous time. It means that we will have the function $\mathbf{x}(t) : \mathbb{R} \rightarrow \mathbb{R}^m$ of continuous dynamic.

Continuous-in-time dynamics

Consider Ordinary Differential Equation (ODE)

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}_\theta(\mathbf{x}(t), t); \quad \text{with initial condition } \mathbf{x}(t_0) = \mathbf{x}_0.$$

$$\mathbf{x}(t_1) = \int_{t_0}^{t_1} \mathbf{f}_\theta(\mathbf{x}(t), t) dt + \mathbf{x}_0$$

Here $\mathbf{f}_\theta : \mathbb{R}^m \times [t_0, t_1] \rightarrow \mathbb{R}^m$ is a vector field.

Numerical solution of ODE

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}_\theta(\mathbf{x}(t), t); \quad \text{with initial condition } \mathbf{x}(t_0) = \mathbf{x}_0.$$

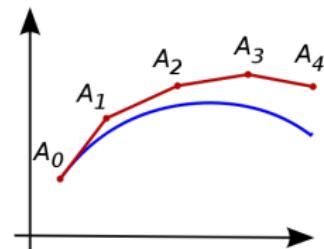
$$\mathbf{x}(t_1) = \int_{t_0}^{t_1} \mathbf{f}_\theta(\mathbf{x}(t), t) dt + \mathbf{x}_0 \approx \text{ODESolve}_f(\mathbf{x}_0, \theta, t_0, t_1).$$

Here we need to define the computational procedure
 $\text{ODESolve}_f(\mathbf{x}_0, \theta, t_0, t_1)$.

Euler update step

$$\frac{\mathbf{x}(t + \Delta t) - \mathbf{x}(t)}{\Delta t} = \mathbf{f}_\theta(\mathbf{x}(t), t)$$

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{f}_\theta(\mathbf{x}(t), t)$$



Note: Euler method is the simplest version of the ODESolve that is unstable in practice. It is possible to use more sophisticated numerical methods instead if Euler (e.g. Runge-Kutta methods).

Continuous-in-time NF

Neural ODE

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}_\theta(\mathbf{x}(t), t); \quad \text{with initial condition } \mathbf{x}(t_0) = \mathbf{x}_0$$

Euler ODESolve

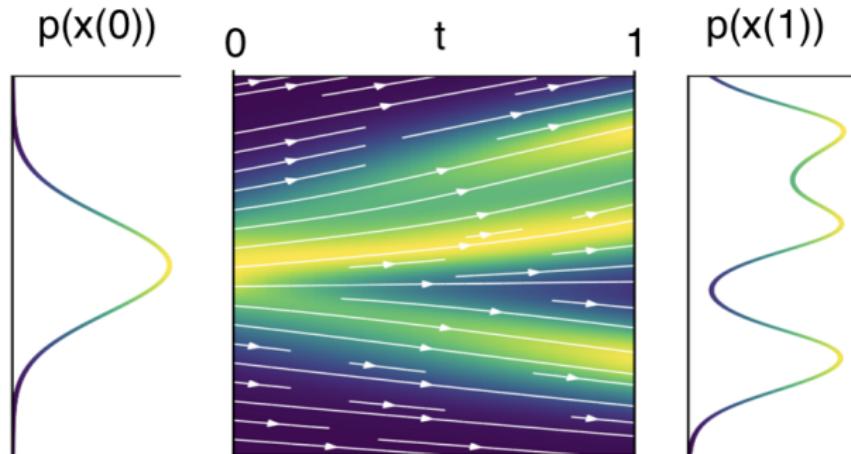
$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \cdot \mathbf{f}_\theta(\mathbf{x}(t), t)$$

- ▶ Let consider time interval $[t_0, t_1] = [0, 1]$ without loss of generality.
- ▶ Assume that $\mathbf{x}(0)$ is a random variable with the density function $p_0(\mathbf{x})$.
- ▶ Then $\mathbf{x}(t)$ is a random variable with the density function $p_t(\mathbf{x})$.

Continuous-in-time NF

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}_\theta(\mathbf{x}(t), t); \quad \text{with initial condition } \mathbf{x}(t_0) = \mathbf{x}_0$$

- ▶ $p_t(\mathbf{x}) = p(\mathbf{x}, t)$ is the **probability path** between $p_0(\mathbf{x})$ and $p_1(\mathbf{x})$.
- ▶ What is the difference between $p_t(\mathbf{x}(t))$ and $p_t(\mathbf{x})$?



Continuous-in-time NF

Theorem (Picard)

If \mathbf{f} is uniformly Lipschitz continuous in \mathbf{x} and continuous in t , then the ODE has a **unique** solution.

It means that we are able **uniquely revert** our ODE.

$$\mathbf{x}(1) = \mathbf{x}(0) + \int_0^1 \mathbf{f}_\theta(\mathbf{x}(t), t) dt$$

$$\mathbf{x}(0) = \mathbf{x}(1) - \int_1^0 \mathbf{f}_\theta(\mathbf{x}(t), t) dt$$

Note: Unlike discrete-in-time NF, \mathbf{f} does not need to be invertible (uniqueness guarantees bijectivity).

How to compute $p_t(\mathbf{x})$ at any moment t ?

Summary

- ▶ DDPM and NCSN are closely related in terms of objectives.
- ▶ Classifier guidance is the way to turn the unconditional model to the conditional one via the training additional classifier on the noisy data.
- ▶ Classifier-free guidance allows to avoid the training additional classifier to get the conditional model. It is widely used in practice.
- ▶ Continuous-in-time NF uses neural ODE to define continuous dynamic $\mathbf{x}(t)$. It has less functional restrictions.
- ▶ If we assume that \mathbf{x}_0 is a random variable then we get the probability path $p_t(\mathbf{x})$ across the time.