Deep Generative Models

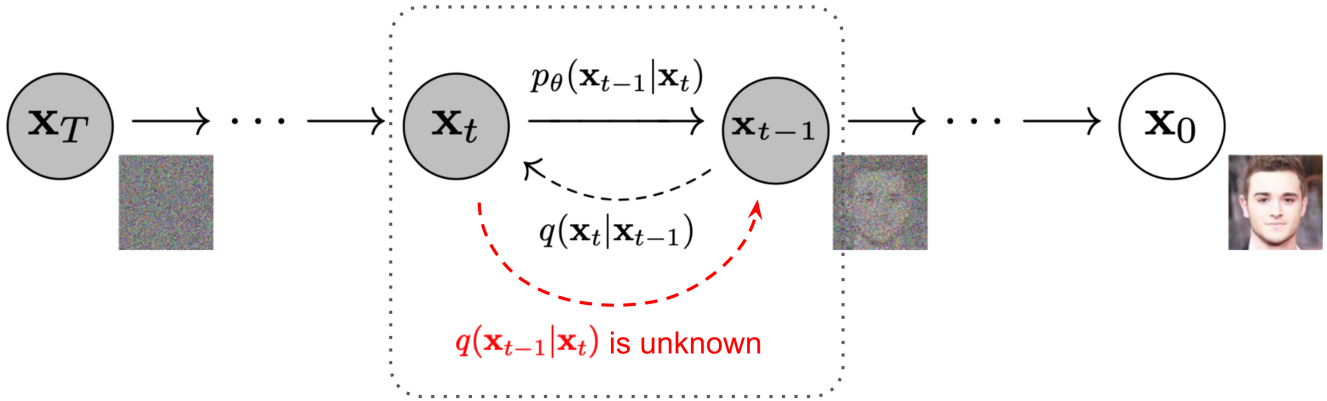
Lecture 12

Roman Isachenko

AI Masters

2024, Spring

Recap of previous lecture



Reverse process

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

Forward process

1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$.

Reverse process

1. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$;
2. $\mathbf{x}_{t-1} = \boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \cdot \boldsymbol{\epsilon} + \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$;
3. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;

Note: The forward process does not have any learnable parameters!

Weng L. What are Diffusion Models?, blog post, 2021

Recap of previous lecture

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

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

- 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

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

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

Das A. An introduction to Diffusion Probabilistic Models, blog post, 2021

Recap of previous lecture

ELBO of gaussian diffusion model

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

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

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

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Recap of previous lecture

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

Reparametrization

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

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

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

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

Simplified objective

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Outline

1. Denoising Diffusion Probabilistic Model (DDPM)

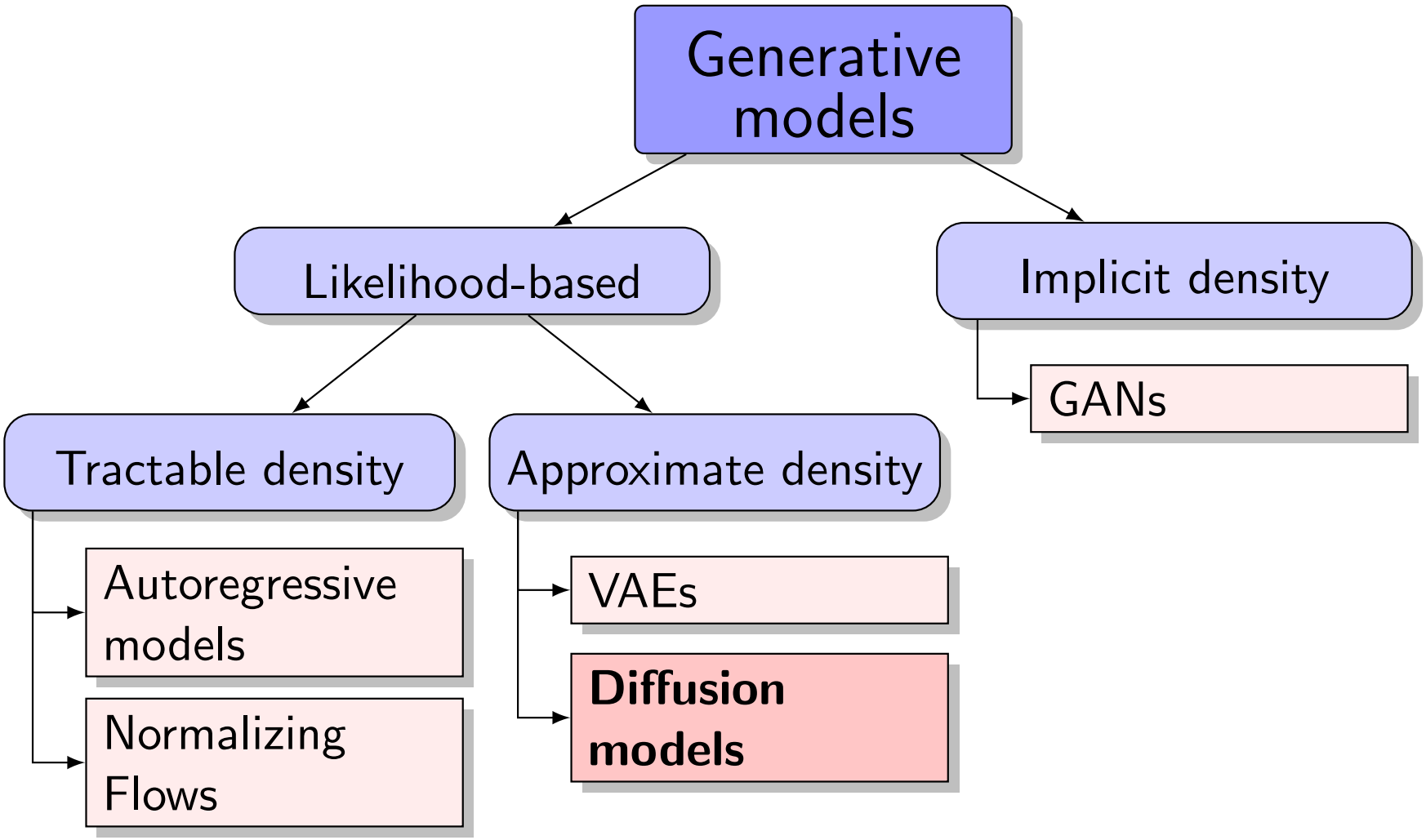
2. Denosing diffusion as score-based generative model

3. Guidance
   - Classifier guidance
   - Classifier-free guidance

Outline

1. Denoising Diffusion Probabilistic Model (DDPM)
2. Denosing diffusion as score-based generative model
3. Guidance
   - Classifier guidance
   - Classifier-free guidance

Generative models zoo



Denoising diffusion probabilistic model (DDPM)

DDPM is a VAE model

- Encoder is a fixed Gaussian Markov chain $q(\mathbf{x}_1, \ldots, \mathbf{x}_T|\mathbf{x}_0)$.
- Latent variable is a hierarchical (in each step the dim. of the latent equals to the dim of the input).
- Decoder is a simple Gaussian model $p(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\theta})$.
- Prior distribution is given by parametric Gaussian Makov chain $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})$.

Forward process

1. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;
2. $\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}$;
3. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$.

Reverse process

1. $\mathbf{x}_T \sim p_\infty(\mathbf{x}) = \mathcal{N}(0, \mathbf{I})$;
2. $\mathbf{x}_{t-1} = \boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \cdot \boldsymbol{\epsilon} + \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$;
3. $\mathbf{x}_0 = \mathbf{x} \sim \pi(\mathbf{x})$;

Ho J. Denoising Diffusion Probabilistic Models, 2020

Denoising diffusion probabilistic model (DDPM)

Training

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

Sampling

1. Sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.
2. Compute mean of $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \tilde{\beta}_t \mathbf{I})$:

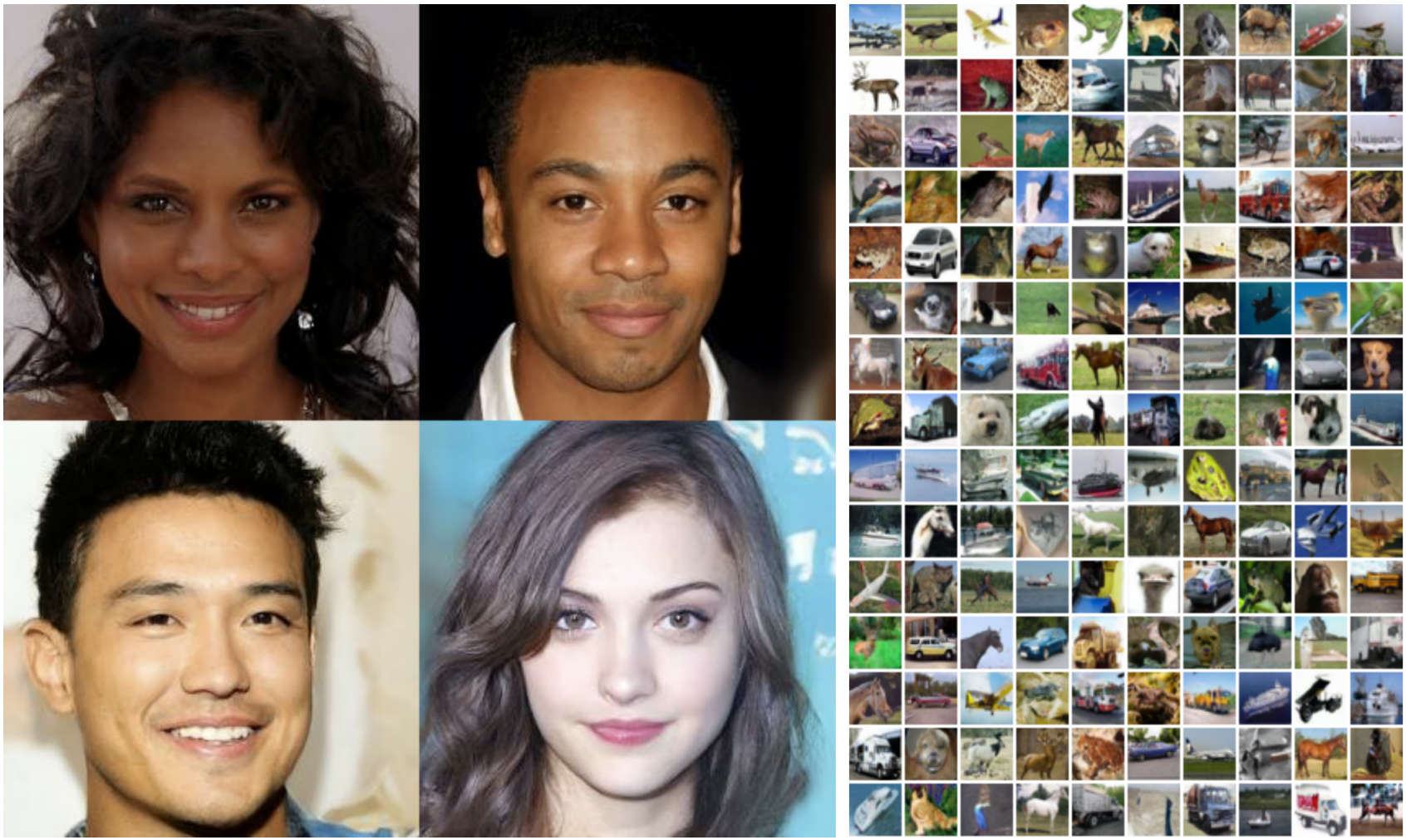
$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \cdot \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{\alpha_t(1 - \bar{\alpha}_t)}} \cdot \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$$

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Denoising diffusion probabilistic model (DDPM)

Samples



Ho J. Denoising Diffusion Probabilistic Models, 2020

Outline

1. Denoising Diffusion Probabilistic Model (DDPM)

2. Denosing diffusion as score-based generative model

3. Guidance
   - Classifier guidance
   - Classifier-free guidance

Denosing diffusion as score-based generative model

DDPM objective

$$\mathcal{L}_t = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left[\frac{(1-\alpha_t)^2}{2\tilde{\beta}_t\alpha_t(1-\bar{\alpha}_t)}\left\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},t\right)-\boldsymbol{\epsilon}\right\|_2^2\right]$$

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

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

Let reparametrize our model:

$$\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)=-\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)}{\sqrt{1-\bar{\alpha}_t}}.$$

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

Ho J. Denoising Diffusion Probabilistic Models, 2020

Denosing diffusion as score-based generative model

DDPM objective

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

In practice the coefficient is omitted.

NCSN objective

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

Note: The objective of DDPM and NCSN is almost identical. But the difference in sampling scheme:

- NCSN uses annealed Langevin dynamics;
- DDPM uses ancestral sampling.

Ho J. Denoising Diffusion Probabilistic Models, 2020

Outline

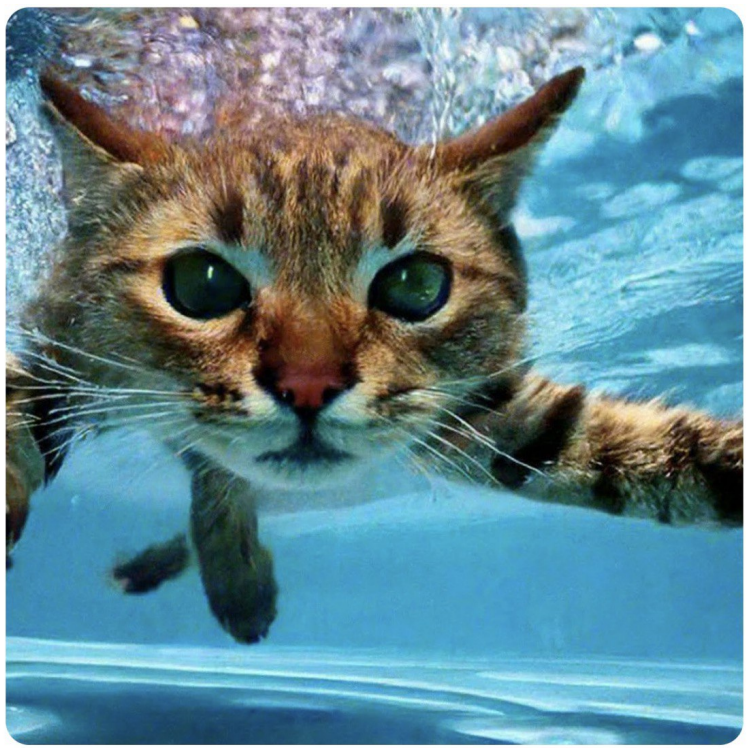
1. Denoising Diffusion Probabilistic Model (DDPM)

2. Denosing diffusion as score-based generative model

3. Guidance
   - Classifier guidance
   - Classifier-free guidance

Guidance

- Throughout the whole course we have discussed unconditional generative models $p(\mathbf{x}|\boldsymbol{\theta})$.
- In practice the majority of the generative models are **conditional**: $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$.
- 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 четыремя, ни шестью, а с 5 (пять) пальцами

Guidance

How to make conditional model $p(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$?

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

DDPM **unsupervised** guidance

- Let imagine we are given the distribution $q(\mathbf{y}|\mathbf{x}_0)$.
- Since we have already defined Markov chain, we have $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y}) = q(\mathbf{x}_t|\mathbf{x}_{t-1})$.
- Let try to find reverse $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y})$.

Label guidance

Label: Ostrich (10th ImageNet class)

VQ-VAE (Proposed) **BigGAN deep**

Razavi A., Oord A., et al. Generating Diverse High-Fidelity Images with VQ-VAE-2, 2019

Text guidance

Prompt: a stained glass window of a panda eating bamboo
Left: $\gamma = 1$, Right: $\gamma = 3$.

Nichol A., et al. GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models, 2022

Outline

1. Denoising Diffusion Probabilistic Model (DDPM)

2. Denosing diffusion as score-based generative model

3. Guidance
   - Classifier guidance
   - Classifier-free guidance

Classifier guidance

$$q(\mathbf{y}|\mathbf{x}_{t-1}, \mathbf{x}_t) = \frac{q(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{y})}{q(\mathbf{x}_{t-1}, \mathbf{x}_t)} =$$

$$= \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{y})q(\mathbf{y}|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1})}{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1})} = q(\mathbf{y}|\mathbf{x}_{t-1}).$$

Conditional distribution

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}) = \frac{q(\mathbf{x}_{t-1}, \mathbf{x}_t, \mathbf{y})}{q(\mathbf{x}_t, \mathbf{y})} =$$

$$= \frac{q(\mathbf{y}|\mathbf{x}_{t-1}, \mathbf{x}_t)q(\mathbf{x}_{t-1}|\mathbf{x}_t)q(\mathbf{x}_t)}{q(\mathbf{y}|\mathbf{x}_t)q(\mathbf{x}_t)} =$$

$$= q(\mathbf{y}|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1}|\mathbf{x}_t) \cdot \text{const}(\mathbf{x}_{t-1}).$$

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi})p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) \cdot \text{const}(\mathbf{x}_{t-1}).$$

- $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta})$ - our unsupervised diffusion model.
- $p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi})$ - classifier for noised samples $\mathbf{x}_{t-1}$

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

Conditional distribution

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi}) \cdot p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) \cdot \text{const}(\mathbf{x}_{t-1})$$

$$\log p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \log p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi}) + \log p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) + \text{const}$$

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

$$\log p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = -\frac{\|\mathbf{x}_{t-1} - \boldsymbol{\mu}\|^2}{2\boldsymbol{\sigma}^2} + \text{const}(\mathbf{x}_{t-1})$$

Taylor expansion

$$\log p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi}) \approx \log p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi})|_{\mathbf{x}_{t-1}=\boldsymbol{\mu}} +$$
$$+ (\mathbf{x}_{t-1} - \boldsymbol{\mu}) \cdot \nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi})|_{\mathbf{x}_{t-1}=\boldsymbol{\mu}} =$$
$$= (\mathbf{x}_{t-1} - \boldsymbol{\mu}) \cdot \mathbf{g} + \text{const}(\mathbf{x}_{t-1}),$$

where $\mathbf{g} = \nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi})|_{\mathbf{x}_{t-1}=\boldsymbol{\mu}}$.

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

$$\log p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = (\mathbf{x}_{t-1} - \boldsymbol{\mu}) \cdot \mathbf{g} - \frac{\|\mathbf{x}_{t-1} - \boldsymbol{\mu}\|^2}{2\boldsymbol{\sigma}^2} + \text{const}(\mathbf{x}_{t-1})$$
$$= -\frac{\|\mathbf{x}_{t-1} - \boldsymbol{\mu} - \boldsymbol{\sigma} \odot \mathbf{g}\|^2}{2\boldsymbol{\sigma}^2} + \text{const}(\mathbf{x}_{t-1})$$
$$= \log \mathcal{N}(\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \mathbf{g}, \boldsymbol{\sigma}^2) + \text{const}(\mathbf{x}_{t-1})$$

Guided sampling

1. Sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.
2. Compute mean of $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \tilde{\beta}_t \mathbf{I})$:

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

3. Compute $\mathbf{g} = \nabla_{\mathbf{x}_{t-1}} \log p(\mathbf{y}|\mathbf{x}_{t-1}, \boldsymbol{\phi})|_{\mathbf{x}_{t-1}=\boldsymbol{\mu}}$.
4. Get denoised image $\mathbf{x}_{t-1} = (\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) + \sqrt{\tilde{\beta}_t} \cdot \mathbf{g}) + \sqrt{\tilde{\beta}_t} \cdot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$.

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

Theorem (denoising score matching)

$$\mathbb{E}_{q(\mathbf{x}_t)} \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) \right\|_2^2 =$$

$$= \mathbb{E}_{\pi(\mathbf{x}_0)} \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0) \right\|_2^2 + \text{const}(\boldsymbol{\theta})$$

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

Conditional distribution

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) = \nabla_{\mathbf{x}_t} \log \left(\frac{p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\phi}) p(\mathbf{x}_t|\boldsymbol{\theta})}{p(\mathbf{y}|\boldsymbol{\phi})} \right)$$

$$= \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\phi}) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta})$$

$$= \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\phi}) - \frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}$$

Classifier-corrected noise prediction

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

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

Classifier-corrected noise prediction

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

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

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

Conditional distribution

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

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

$$p(\mathbf{x}_t|\mathbf{y}, \theta, \phi) = \frac{p(\mathbf{y}|\mathbf{x}_t, \phi)^\gamma p(\mathbf{x}_t|\theta)}{Z}$$

Check that $\nabla_{\mathbf{x}_t} \log Z \neq 0$.

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Classifier guidance

Guided sampling

1. Sample $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$.
2. Compute "corrected" $\hat{\epsilon}_{\theta}(\mathbf{x}_t, t)$:

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

3. Compute mean of $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \theta) = \mathcal{N}(\mu_{\theta}(\mathbf{x}_t, t), \tilde{\beta}_t \mathbf{I})$:

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

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

Dhariwal P., Nichol A. Diffusion Models Beat GANs on Image Synthesis, 2021

Outline

1. Denoising Diffusion Probabilistic Model (DDPM)

2. Denosing diffusion as score-based generative model

3. Guidance
   - Classifier guidance
   - Classifier-free guidance

Classifier-free guidance

Previous methods require training the additional classifier model $p(\mathbf{y}|\mathbf{x}_t, \boldsymbol{\theta})$ on the noisy data. Let try to avoid this requirement.

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

$$= \gamma \cdot \nabla_{\mathbf{x}_t} \log \left(\frac{p(\mathbf{x}_t|\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\phi}) p(\mathbf{y}|\boldsymbol{\phi})}{p(\mathbf{x}_t|\boldsymbol{\theta})} \right) + \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\boldsymbol{\theta}) =$$

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

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

What will we get if $\gamma = 1$?

Classifier-free-corrected noise prediction

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

In practice we could train the single model $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{y}, t)$ on **supervised** data alternating with real conditioning $\mathbf{y}$ and empty conditioning $\mathbf{y} = \emptyset$.

Ho J., Salimans T. Classifier-Free Diffusion Guidance, 2022

Summary

- DDPM is a VAE model that tries to invert forward diffusion process using variational inference. DDPM is really slow, because we have to apply the model T times.
- Objective of DDPM is closely related to the noise conditioned score network and score matching.
- Conditional models use labels $\mathbf{y}$ as the additional input. Majority of the modern generative models are conditional.
- 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.