Generative Models II

Nikita Kiselev

Deep Learning, Intelligent Systems

November 25, 2025

Recap

- What are Generative Models?
- Taxonomy
- Autoregressive Models
- Autoencoders
- VAEs
- GANs

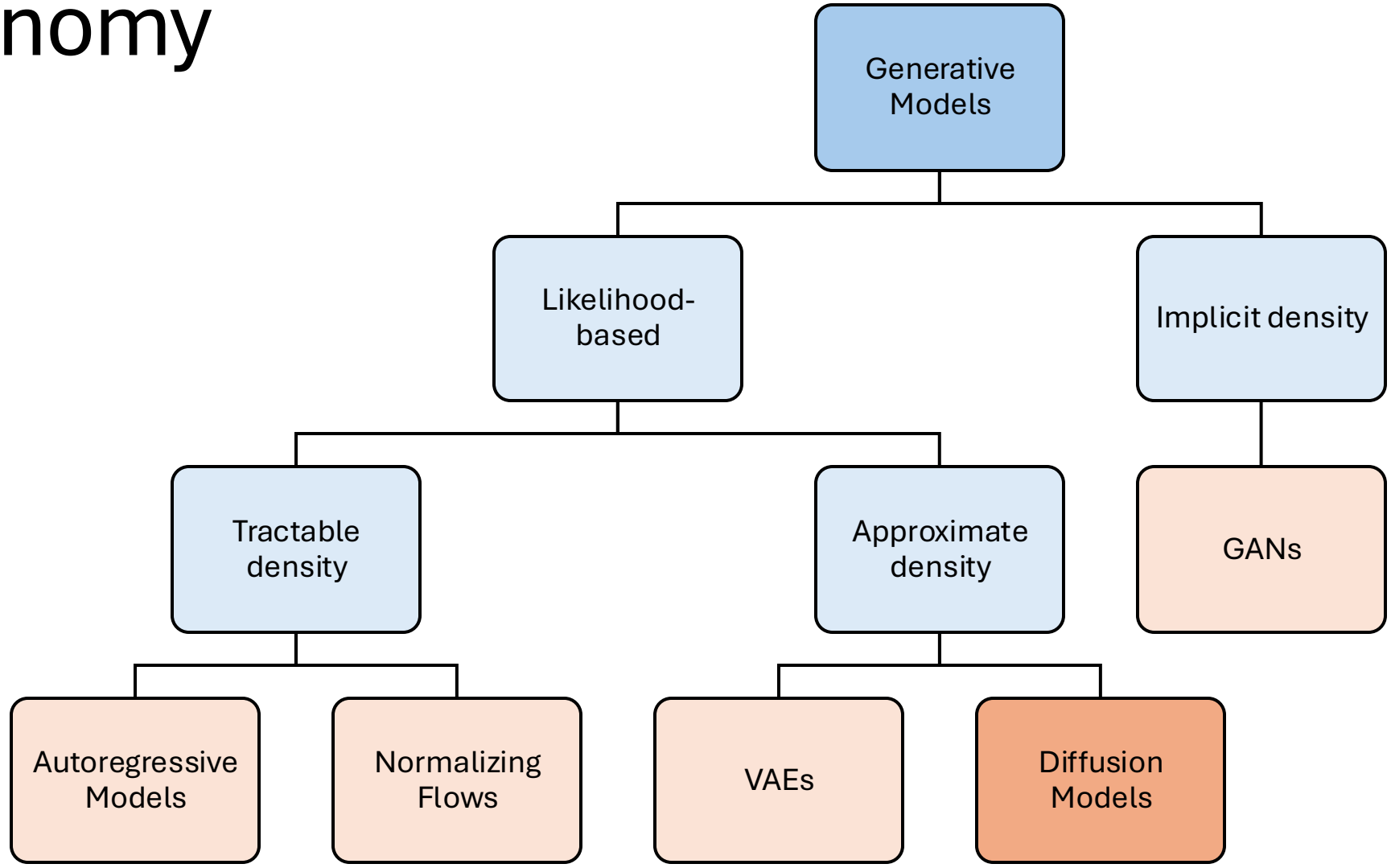
* Extra: Modern Architectures

Contents

- Diffusion Models
- Flow Matching
- Guidance
- Latent Diffusion Models

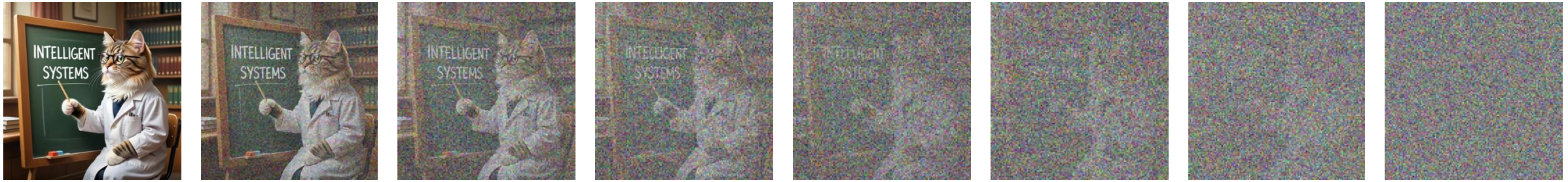
Diffusion Models

Taxonomy


Generative Models
Likelihood-based
Implicit density
Tractable density
Approximate density
GANs
Autoregressive Models
Normalizing Flows
VAEs
Diffusion Models

Diffusion Models: General Idea

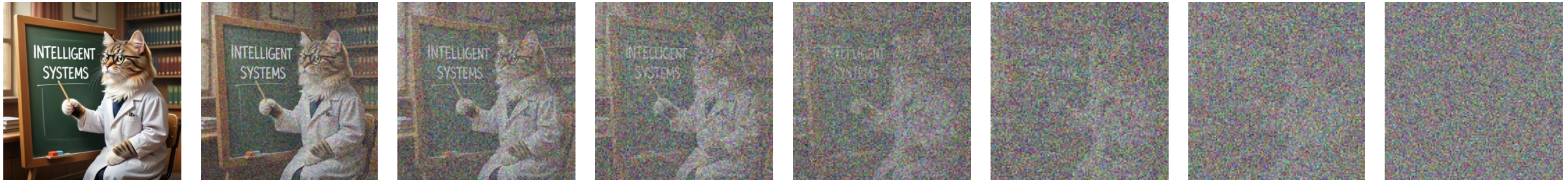
Forward process: gradually add noise to input



Reverse process: learn to generate data by denoising

Diffusion Models: General Idea

Forward process: gradually add noise to input



Reverse process: learn to generate data by denoising

Q1: How to define a forward process?
Q2: How to learn a reverse denoising process?

Latent Variable Models (VAE-like)

Monte Carlo estimation

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \int p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} = \mathbb{E}_{p(\mathbf{z})} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) \approx \frac{1}{K} \sum_{k=1}^{K} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}^{(k)}), \quad \mathbf{z}^{(k)} \sim p(\mathbf{z})$$

Prior distribution, e.g. $\mathcal{N}(\mathbf{0}, \mathbf{I})$

Can be parameterized with a NN: $f_{\boldsymbol{\theta}}(\mathbf{z}) \rightarrow \mathbf{x}$

$$\arg\max_{\boldsymbol{\theta}} \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \approx \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{x} \in \mathcal{D}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \approx \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{x} \in \mathcal{D}} \log \left[\frac{1}{K} \sum_{k=1}^{K} p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}^{(k)}) \right]$$

High variance for high-dimensional $\mathbf{z}$ → does not scale :(

Evidence Lower Bound (ELBO)

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int \frac{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) d\mathbf{z} \geq \quad \text{(Jensen's inequality)}$$

$$\geq \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} d\mathbf{z} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} = \mathrm{ELBO}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi})$$

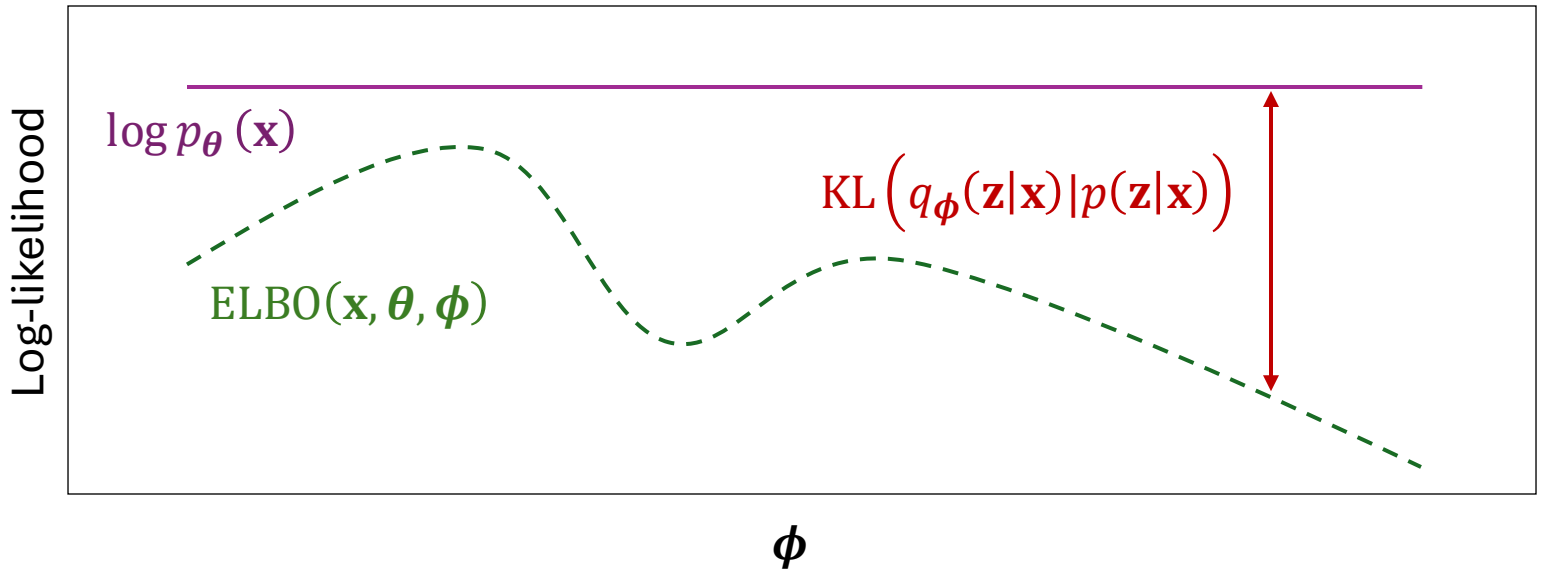
$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ – posterior distribution, parameterized with a neural network $g_{\boldsymbol{\phi}}(\mathbf{x}) \rightarrow \mathbf{z}$

How close is $\mathrm{ELBO}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\phi})$ to $\log p_{\boldsymbol{\theta}}(\mathbf{x})$?

Evidence Lower Bound (ELBO)

Unknown true posterior

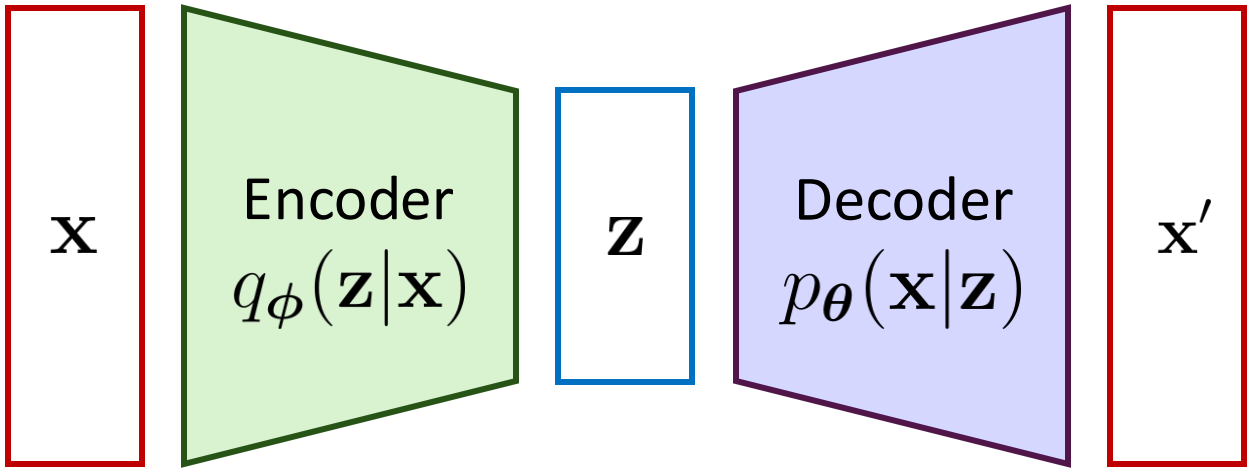
$$\log p_{\boldsymbol{\theta}}(\mathbf{x}) = \underbrace{\int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x},\mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})} d\mathbf{z}}_{\text{ELBO}(\mathbf{x},\boldsymbol{\theta},\boldsymbol{\phi})} + \underbrace{\text{KL}\left(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})\right)}_{\text{Gap between } \log p_{\boldsymbol{\theta}}(\mathbf{x}) \text{ and ELBO}(\mathbf{x},\boldsymbol{\theta},\boldsymbol{\phi})}$$



Variational Autoencoder (VAE)

Step 1: Update $\boldsymbol{\phi}$ to approximate $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ better

Step 2: Update $\boldsymbol{\theta}$ to maximize $\log p_{\boldsymbol{\theta}}(\mathbf{x})$



Denoising Diffusion Probabilistic Models

$$p_{\boldsymbol{\theta}}(\mathbf{x}_0) = \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}, \quad p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

Markov chain

Denoising Diffusion Probabilistic Models

$$p_{\boldsymbol{\theta}}(\mathbf{x}_0) = \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}, \quad p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$$ Markov chain

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}_0) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} = \log \int \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \geq$$ Jensen's inequality

not parameterized

Denoising Diffusion Probabilistic Models

$$p_{\boldsymbol{\theta}}(\mathbf{x}_0) = \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}, \quad p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$$ Markov chain

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}_0) = \log \int p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} = \log \int \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T} \geq$$ Jensen's inequality

not parameterized

$$\geq \int q(\mathbf{x}_{1:T}|\mathbf{x}_0) \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} d\mathbf{x}_{1:T} = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} = \mathrm{ELBO}(\mathbf{x}_0, \boldsymbol{\theta})$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

DDPM vs. VAE

DDPM

$$\mathrm{ELBO}_{\mathrm{DDPM}} = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}$$

$T = 1$

$$\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_0, \mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)}$$

$T \gg 1;\ q(\mathbf{x_1}|\mathbf{x_0});\ \dim(\mathbf{x_1}) = \dim(\mathbf{x_0})$

VAE

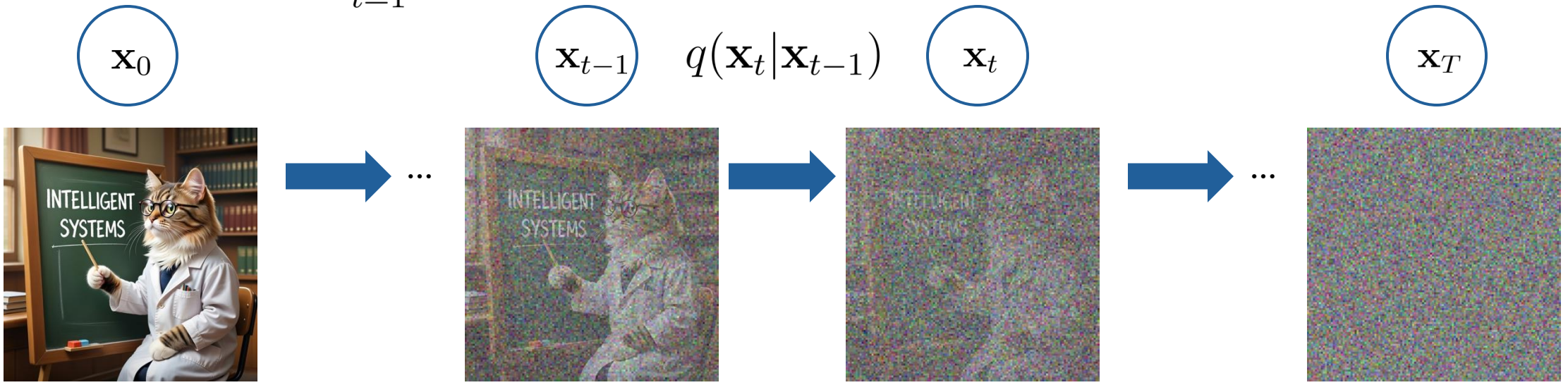
$$\mathrm{ELBO}_{\mathrm{VAE}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_0)} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_0, \mathbf{z})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_0)}$$

$T = 1;\ q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x_0});\ \dim(\mathbf{z}) < \dim(\mathbf{x_0})$

DDPM Overview

Forward process

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$



DDPM Overview

Revert it and approximate

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t) \approx p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

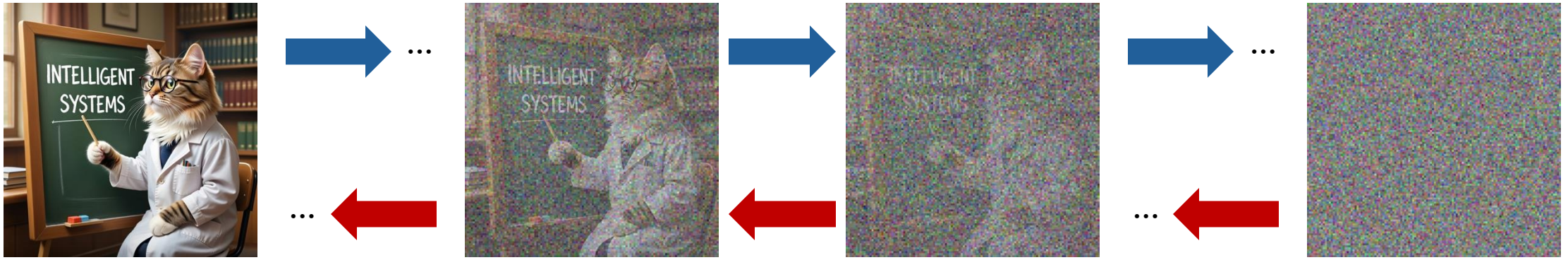
Forward process

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

Reverse process

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$$

$\mathbf{x}_0$ $\mathbf{x}_{t-1}$ $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ $\mathbf{x}_t$ $\mathbf{x}_T$



$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$

Forward Process

$$\mathbf{x}_0 = \mathbf{x} \sim p_{\text{data}}(\mathbf{x})$$

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

$$\mathbf{x}_t = \sqrt{1-\beta_t} \cdot \mathbf{x}_{t-1} + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

$$\{\beta_1, \ldots, \beta_T\} \text{ — variance schedule}, \beta_t \in (0, 1)$$

Useful property
Derive $\mathbf{x_t}$ directly from $\mathbf{x_0}$

$$\text{Let } \alpha_t = 1 - \beta_t \text{ and } \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

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

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \cdot \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \cdot \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Reverse Process

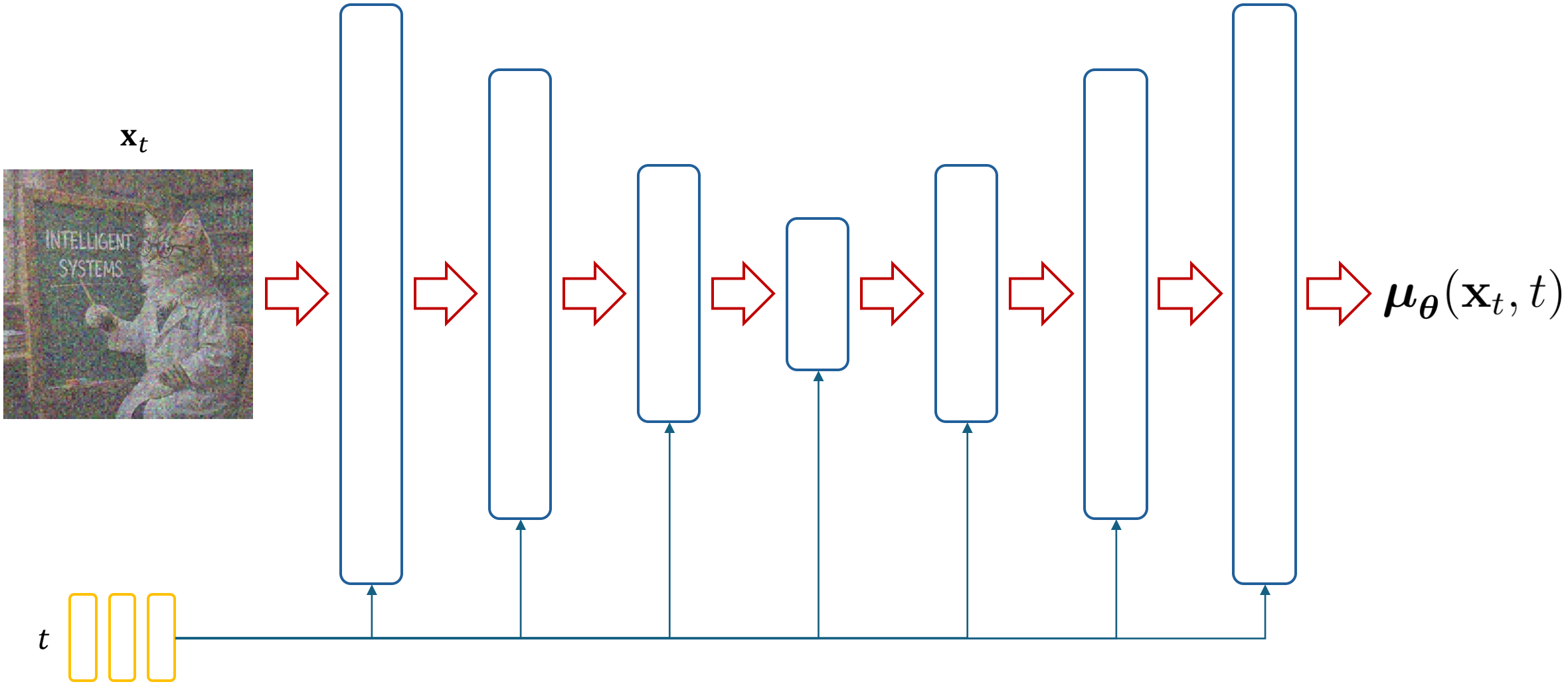
$$\mathbf{x}_T \sim p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Usually non-learnable, e.g., $\beta_t \mathbf{I}$

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right)$$

$$\mathbf{x}_{t-1} = \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) + \sqrt{\beta_t} \cdot \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Model parameterization: U-Net



DDPM training objective

$$\mathrm{ELBO}(\mathbf{x}_0, \boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log p(\mathbf{x}_T) + \sum_{t=1}^{T} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] =$$

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

Hocus-pocus: $q(\mathbf{x}_t|\mathbf{x}_{t-1}) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0) = \dfrac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}$

$q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is Markovian

Bayes' theorem

DDPM training objective

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

DDPM training objective

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

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

DDPM training objective

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

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

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

$$+ \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{x}_1) = \ldots$$

DDPM training objective

$$\ldots = \underbrace{\mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)} \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)}}_{-\mathrm{KL}(q(\mathbf{x_T}|\mathbf{x_0})|p(\mathbf{x_T}))} + \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\mathbf{x}_t,\mathbf{x}_{t-1}|\mathbf{x}_0)}}_{E_{q(\mathbf{x_t}|\mathbf{x_0})}E_{q(\mathbf{x_{t-1}}|\mathbf{x_t},\mathbf{x_0})}} \log \frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} +$$

$$+ \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{x}_1) = \ldots =$$

$$= \underbrace{-\mathrm{KL}\left(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T)\right)}_{-\mathcal{L}_T} + \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \left[-\mathrm{KL}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\|p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\right)\right]}_{-\mathcal{L}_t} +$$

$$+ \underbrace{\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)} \log p_{\boldsymbol{\theta}}(\mathbf{x}_0|\mathbf{x}_1)}_{-\mathcal{L}_0} = -\mathcal{L}$$

DDPM training objective

$$\mathcal{L}_{\text{DDPM}} = \mathcal{L}_T + \sum_{t=1}^{T-1} \mathcal{L}_t + \mathcal{L}_0$$

- $\mathcal{L}_T$ is constant
- $\mathcal{L}_0$ can be omitted for simplicity
- Let's focus on $\mathcal{L}_t$

$$\mathcal{L}_t = \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \text{KL}\left(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) || p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)\right)$$

DDPM training objective

We skip some steps for simplicity to derive the following:

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

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

Reparameterized training objective

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

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

Final training objective

Simplify

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

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

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0\sim p_{\text{data}}(\mathbf{x})}\mathbb{E}_{t\sim\mathcal{U}(1,T)}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}\left(\sqrt{\bar{\alpha}_t}\cdot\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\cdot\boldsymbol{\epsilon},t\right)-\boldsymbol{\epsilon}\right\|_2^2$$

DDPM

Training

1. Get the sample $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x})$
2. Sample timestep $t \sim \mathcal{U}(1, T)$ and the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
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 $\mathcal{L}_{\text{simple}} = \left\| \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \boldsymbol{\epsilon} \right\|_2^2$

Sampling

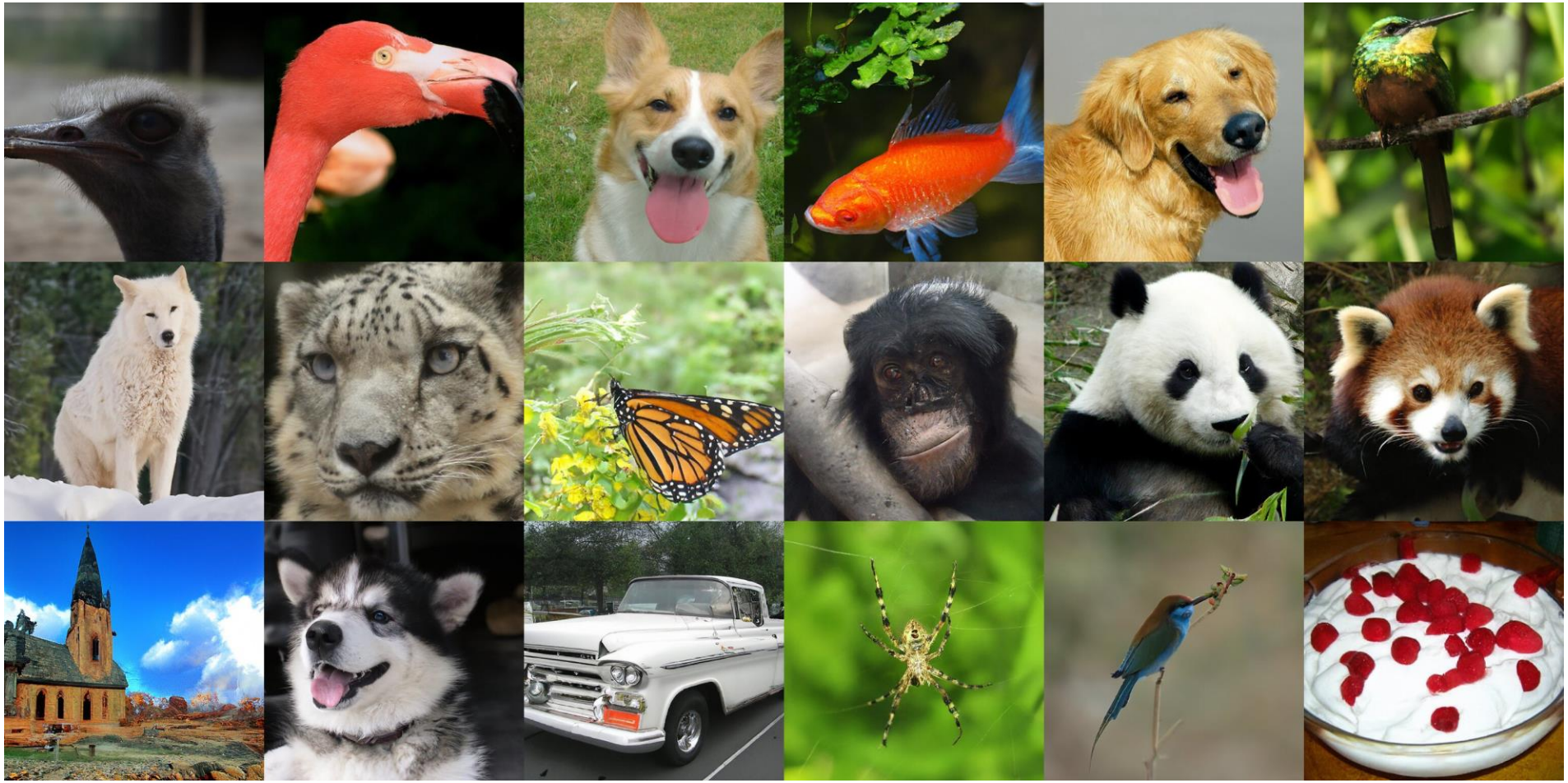
Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

For t in $T, \ldots, 1$:

1. Compute mean of $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \tilde{\beta}_t \mathbf{I}\right) \longrightarrow \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \cdot \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right)$
2. Get denoised image

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

Samples



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

DDPM summary

DDPMs are latent variable models, similar to VAE

Forward process – Markov chain $p(\mathbf{x_0}) \rightarrow p(\mathbf{x_T}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$

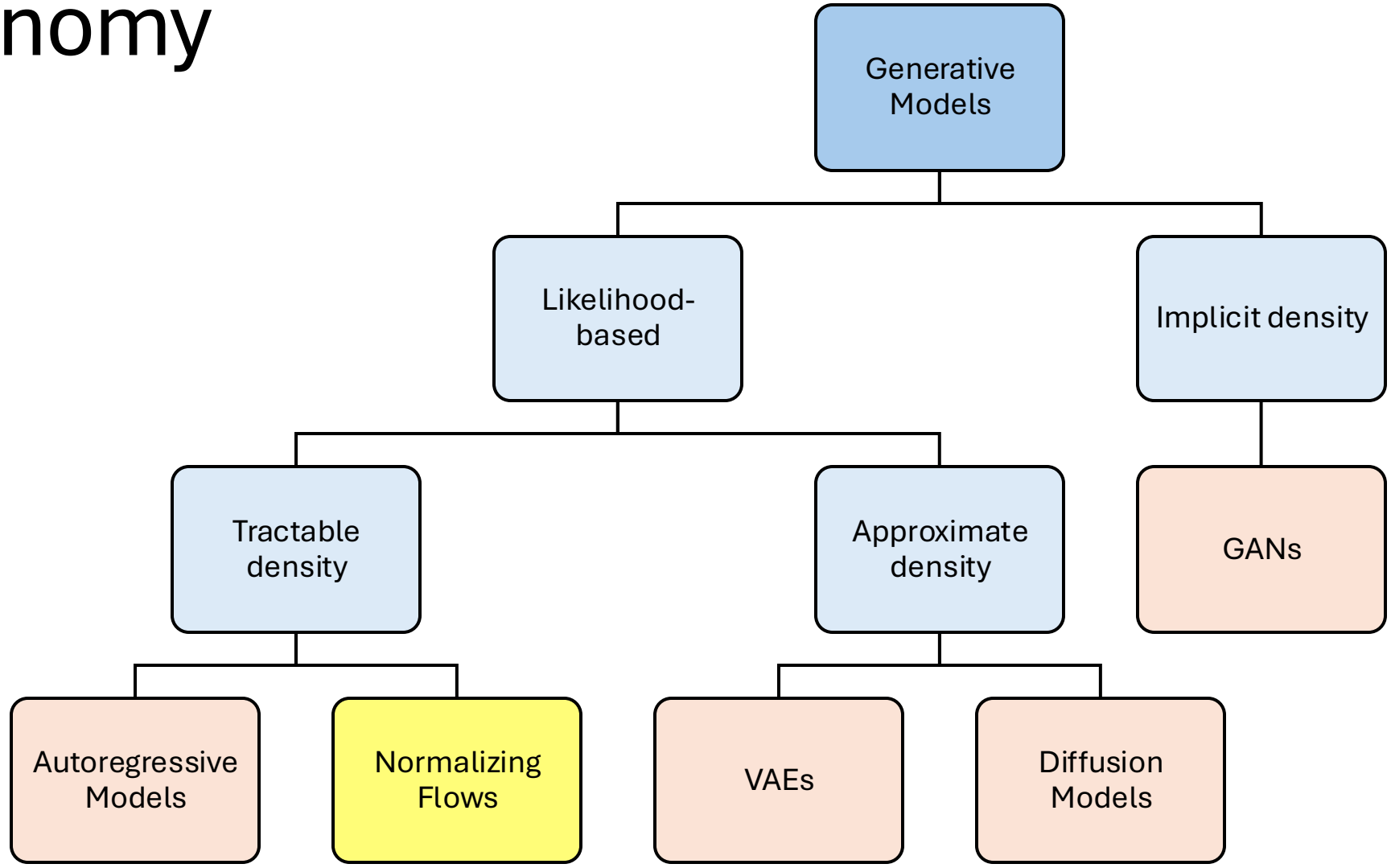
Reverse process reverts the forward process $p(\mathbf{x_T}) \rightarrow p(\mathbf{x_0})$

Costly iterative sampling with $\mathrm{T} \approx 1000$ steps

Loss – MSE between predicted and added noise at different noise levels

Flow Matching

Taxonomy


Generative Models
Likelihood-based
Implicit density
Tractable density
Approximate density
GANs
Autoregressive Models
Normalizing Flows
VAEs
Diffusion Models

Ordinary Differential Equations (ODEs)

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}(t), t), \quad \mathbf{x}(t_0) = \mathbf{x}_0$$

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

Euler Update Step (ODESolve)

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

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

ODE dynamics for random samples

Let's consider an ODE dynamics $\mathbf{x}(t)$ in the interval $t \in [0,1]$:

- $\mathbf{x}_0 \sim p_0(\mathbf{x}) = p(\mathbf{x})$, $\mathbf{x}_1 \sim p_1(\mathbf{x}) = \pi(\mathbf{x})$;
- $p(\mathbf{x})$ is a base distribution (e.g., $\mathcal{N}(\mathbf{0}, \mathbf{I})$), and $\pi(\mathbf{x})$ is the true data distribution

Fjelde, Tor, et al. An Introduction to Flow Matching, 2024 https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

Flow Matching

$$\mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{\mathbf{x}_1\sim\pi(\mathbf{x})}\mathbb{E}_{\mathbf{x}_0\sim p(\mathbf{x})}\left\|\mathbf{f}(\mathbf{x}_t,t)-\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)\right\|_2^2\rightarrow\min_{\boldsymbol{\theta}}$$

- Approximate the true vector field $\mathbf{f}(\mathbf{x},t)$ using $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x},t)$
- Use $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x},t)$ for deterministic sampling from the ODE

But…

- There are infinitely many possible $\mathbf{f}(\mathbf{x},t)$ between $\pi(\mathbf{x})$ and $p(\mathbf{x})$
- The true vector field $\mathbf{f}(\mathbf{x},t)$ is **unknown**
- We need to select the “best” $\mathbf{f}(\mathbf{x},t)$ and make the objective tractable

Here We Go… Again – Latent Variable Model

Let's introduct the latent variable $\mathbf{z}$:

$$p_t(\mathbf{x}) = \int p_t(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$

Then a **conditional vector field** will be:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, t) \Rightarrow \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{z}, t)$$

Theorem

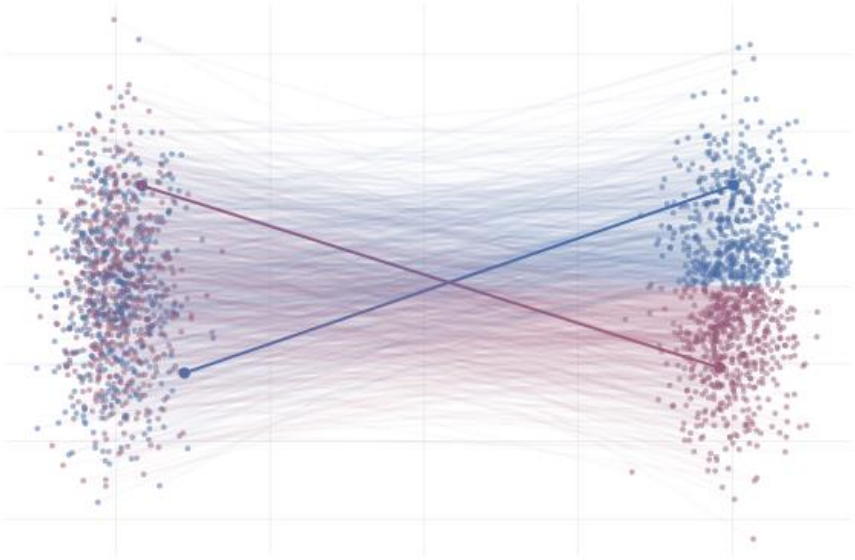
$$\arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{CFM}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\mathrm{FM}}$$

How should we choose the conditioning latent variable?

Linear Interpolation: $\mathbf{z} = (\mathbf{x}_0, \mathbf{x}_1)$

$$p_t(\mathbf{x}|\mathbf{x}_0, \mathbf{x}_1) = \mathcal{N}\left(t\mathbf{x}_1 + (1-t)\mathbf{x}_0, \sigma^2\mathbf{I}\right), \quad \mathbf{x}_t = t\mathbf{x}_1 + (1-t)\mathbf{x}_0$$

$$\frac{d\mathbf{x}_t}{dt} = \mathbf{f}(\mathbf{x}_t, \mathbf{x}_0, \mathbf{x}_1, t) = \mathbf{x}_1 - \mathbf{x}_0$$

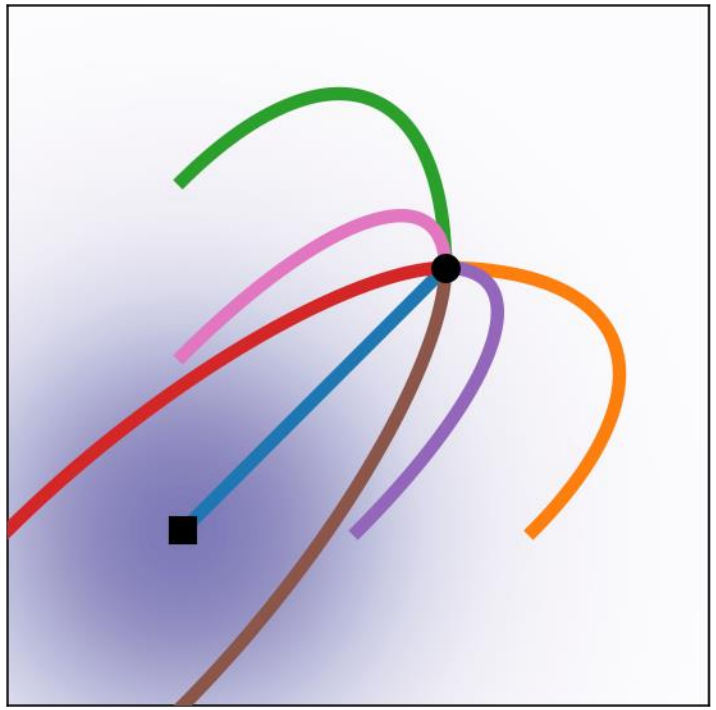


The conditional vector field $\mathbf{f}(\mathbf{x_t}, \mathbf{x_0}, \mathbf{x_1}, t)$ defines straight lines between $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\pi(\mathbf{x})$

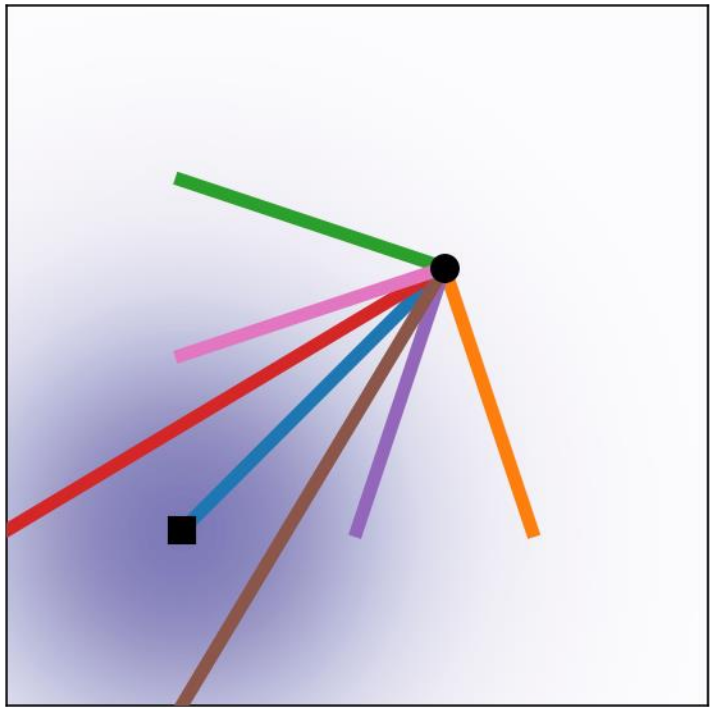
Fjelde, Tor, et al. An Introduction to Flow Matching, 2024

https://mlg.eng.cam.ac.uk/blog/2024/01/20/flow-matching.html

DDPM vs. Flow Matching



Diffusion



Flow Matching

Lipman Y. et al. Flow matching for generative modeling, 2022

Rectified Flow (Linear Interpolation)

$$\mathbb{E}_{t\sim\mathcal{U}[0,1]}\mathbb{E}_{\mathbf{x}_1\sim\pi(\mathbf{x})}\mathbb{E}_{\mathbf{x}_0\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left\|(\mathbf{x}_1-\mathbf{x}_0)-\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)\right\|_2^2\to\min_{\boldsymbol{\theta}}$$

Training

1. Sample $\mathbf{x}_1\sim\pi(\mathbf{x})$
2. Sample time $t\sim\mathcal{U}[0,1]$ and $\mathbf{x}_0\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
3. Obtain the noisy image $\mathbf{x}_t=t\mathbf{x}_1+(1-t)\mathbf{x}_0$
4. Compute the loss $\mathcal{L}=\left\|(\mathbf{x}_1-\mathbf{x}_0)-\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_t,t)\right\|_2^2$

Sampling

1. Sample $\mathbf{x}_0\sim\mathcal{N}(\mathbf{0},\mathbf{I})$
2. Solve the ODE to obtain $\mathbf{x}_1=\mathrm{ODESolve}_{\mathbf{f}}(\mathbf{x}_0,\boldsymbol{\theta},t_0=0,t_1=1)$

FLUX – cutting-edge flow matching model

https://flux1ai.com/dev

Guidance

Guidance

- Up to now, we have focused on **unconditional** generative models
- In practice, most generative models are **conditional** (in diffusion era it is call guided): $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})$
- Here, $\mathbf{y}$ might denote a class label or **text** (as in text-to-image task)

Prompt: Gorgeous fat fluffy cat in a knitted sweater with text “ИГРИСТОЕ”

Prompt: A stylish old man with long gray hair and a beard sitting at his laptop in the office wearing a T-shirt that says “Джун”

Prompt: A white Porsche 911 is standing in the forest in the fog. On his license plate there is the inscription “Андрюха”

https://huggingface.co/kandinskylab/Kandinsky-5.0-T2I-Lite

Label Guidance

VQ-VAE (Proposed) BigGAN deep

Razavi A., Van den Oord A., Vinyals O. Generating diverse high-fidelity images with VQ-VAE-2, 2019

Text Guidance

(c) GLIDE (CLIP guidance, scale 2.0)

(d) GLIDE (Classifier-free guidance, scale 3.0)

Nichol A. et al. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models, 2021

Guidance in Generative Models

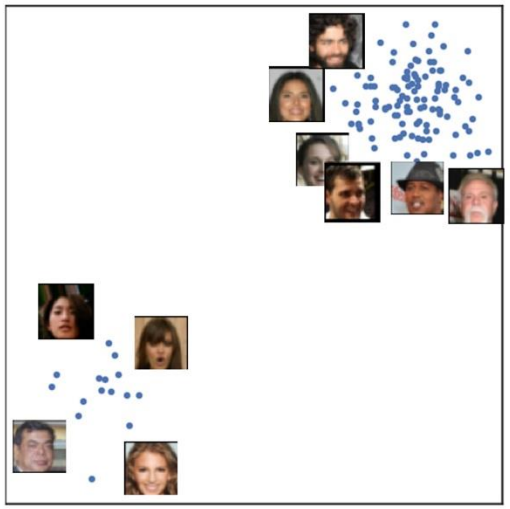
How to make guided model?

Instead of sampling from $p_{\boldsymbol{\theta}}(\mathbf{x})$, we sample from $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{y})$.

Given supervised data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, we can treat $\mathbf{y}$ as an additional model input:

- $p(x_i|\mathbf{x}_{<i}, \mathbf{y})$ for Autoregressive models
- Encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}, \mathbf{y})$ and Decoder $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}, \mathbf{y})$ for VAEs
- $G_{\boldsymbol{\theta}}(\mathbf{z}, \mathbf{y})$ for GANs
- $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{y})$ for DDPMs
- $\mathbf{f}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{y})$ for Flow Matching

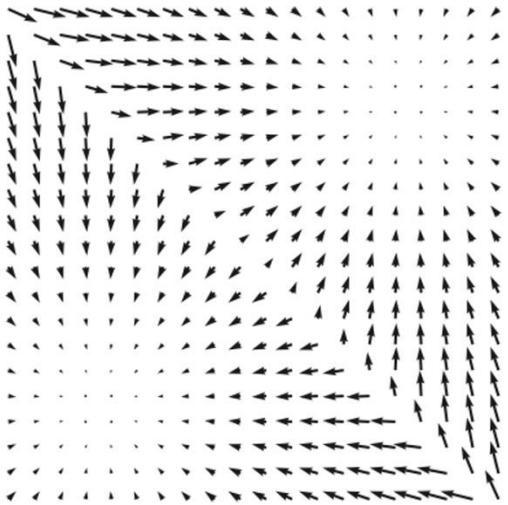
Score Matching



Data samples

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

score matching



Scores

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

Langevin dynamics

New samples

Diffusion Models are hidden Score Matching Models…

$$\nabla_{\mathbf{x}} \log \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \sigma^2\mathbf{I}) = -\frac{\mathbf{x}-\boldsymbol{\mu}}{\sigma^2} = -\frac{\boldsymbol{\epsilon}}{\sigma}$$

$$\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t|\mathbf{x}_0) = -\frac{\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)}{\sqrt{1-\bar{\alpha}_t}}$$

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

Classifier Guidance

Guided Generation

$$\begin{aligned}\nabla_{\mathbf{x}_t} \log p_{\boldsymbol{\theta}}(\mathbf{x}_t|\mathbf{y}) &= \nabla_{\mathbf{x}_t} \log \left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}_t) p(\mathbf{y}|\mathbf{x}_t)}{p(\mathbf{y})} \right) \\ &= \nabla_{\mathbf{x}_t} \log p_{\boldsymbol{\theta}}(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) \\ &= \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)\end{aligned}$$

Guidance Scale – scale up the guidance contribution

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

Solution: Train an additional classifier $p(\mathbf{y}|\mathbf{x_t})$ on noisy data

Classifier-Free Guidance (CFG)

Classifier Guidance: $\mathbf{s}_{\boldsymbol{\theta}}^{\gamma}(\mathbf{x}_t, t, \mathbf{y}) = \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) + \gamma \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$

Bayes Theorem

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

Classifier-Free Guidance (without explicit classifier)

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

Classifier-Free Guidance (CFG)

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

- Introduce “the absence of conditioning” label $\mathbf{y} = \emptyset$
- Use it to get unguided score function $\mathbf{s}_{\boldsymbol{\theta},t}(\mathbf{x}_t) = \mathbf{s}_{\boldsymbol{\theta},t}(\mathbf{x}_t, \emptyset)$
- Train a single model $\mathbf{s}_{\boldsymbol{\theta},t}(\mathbf{x}_t, \mathbf{y})$ using **supervised** data, but artificially drop the labels $\mathbf{y}$ with some fixed probability (simulating the case of $\mathbf{y} = \emptyset$)
- Apply the **model twice** during inference to get $\mathbf{s}_{\boldsymbol{\theta},t}(\mathbf{x}_t, \mathbf{y})$ and $\mathbf{s}_{\boldsymbol{\theta},t}(\mathbf{x}_t, \emptyset)$

Latent Diffusion Models

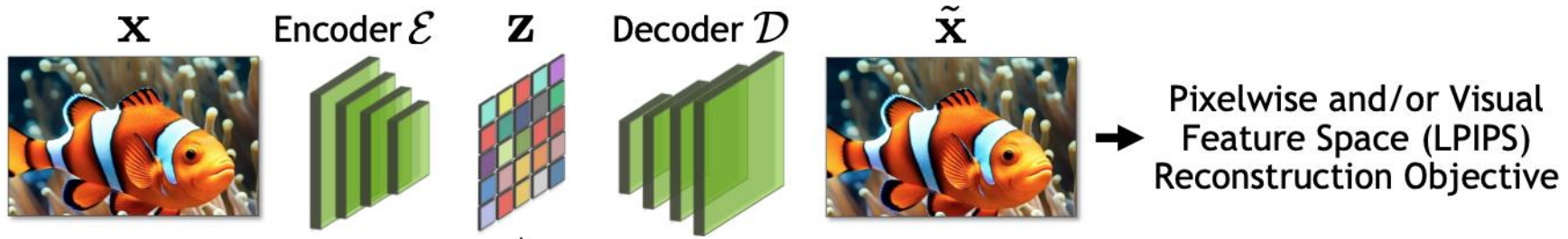
Latent Diffusion Models

Map Data into Compressed Latent Space. Train Diffusion Model efficiently in Latent Psace.

- Stage 1:

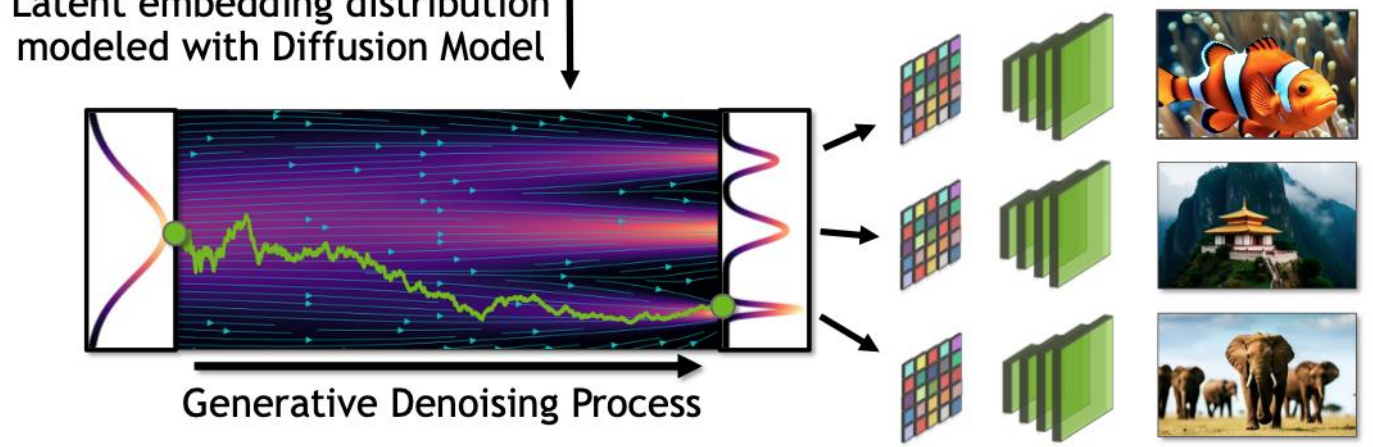
Train Autoencoder

$$\tilde{\mathbf{x}} = \mathcal{D}(\mathcal{E}(\mathbf{x}))$$



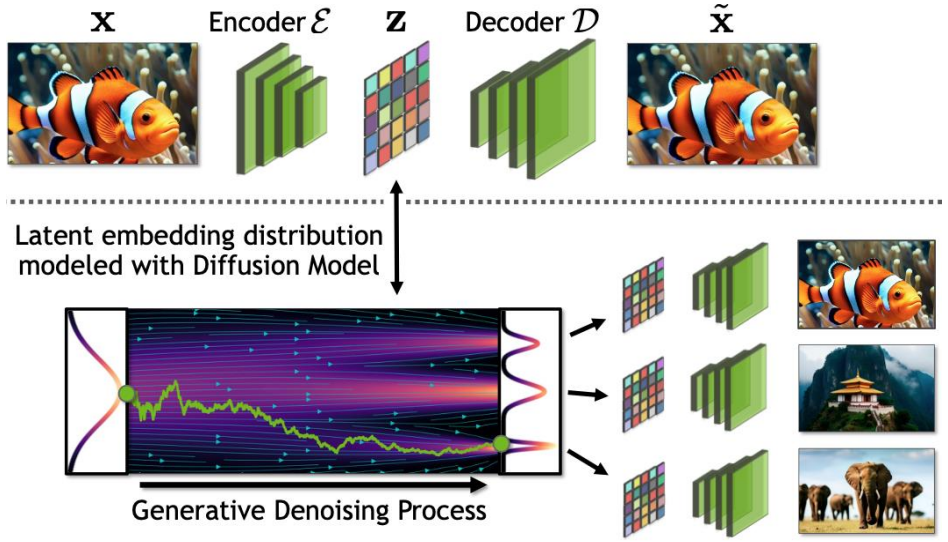
- Stage 2:

Train **Latent** Diffusion Model



Vahdat A., Kreis K., Kautz J. Score-based generative modeling in latent space, 2021
Rombach R. et al. High-resolution image synthesis with latent diffusion models, 2022

Latent Diffusion Models

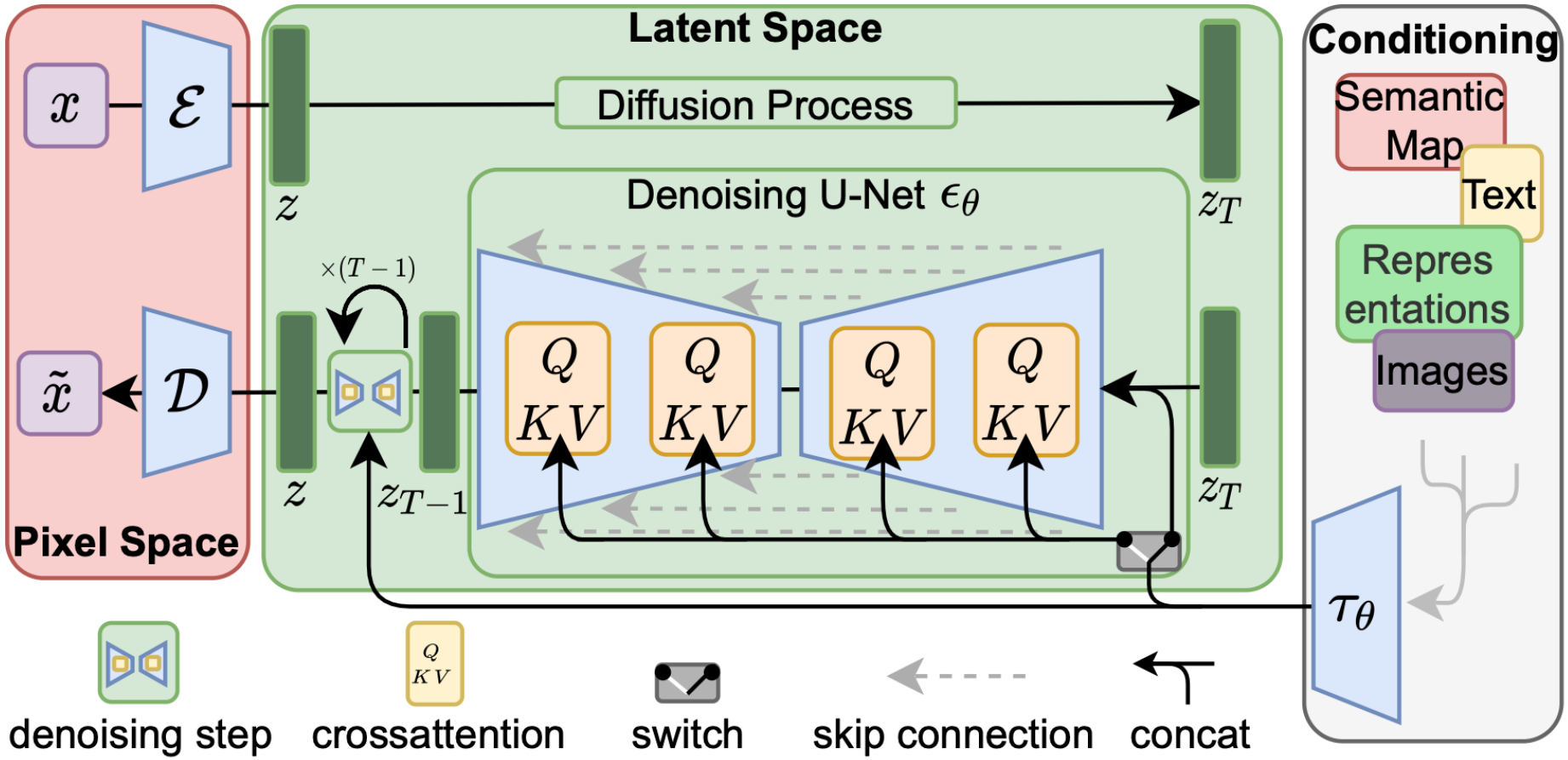


Advantages:

1. Compressed latent space: Train diffusion model in **lower resolution** latent space → **computationally more efficiently**
2. Regularized smooth/compressed latent space: **Easier task** for diffusion model and **faster sampling**
3. Flexibility: **Autoencoder can be tailored to data** (images, video, text, graphs, 3D point clous, meshes, etc.)

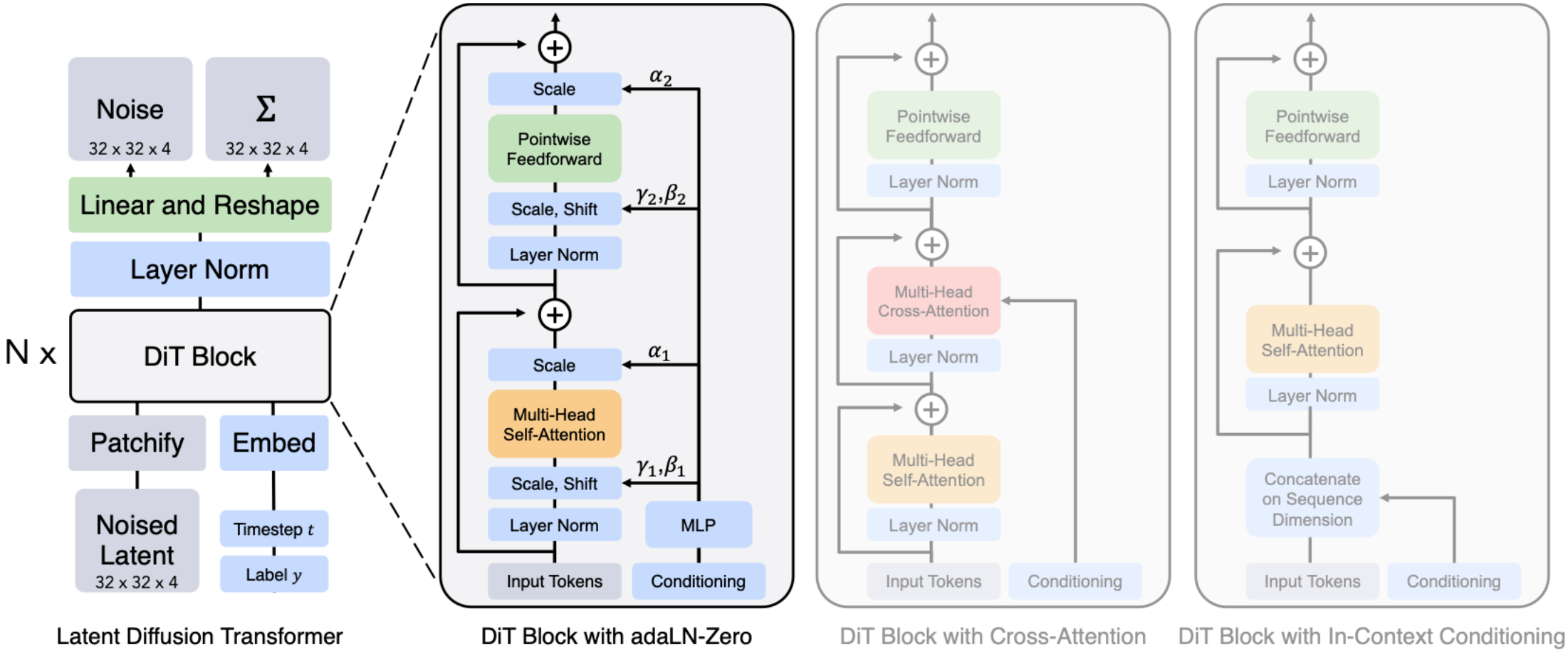
Vahdat A., Kreis K., Kautz J. Score-based generative modeling in latent space, 2021
Rombach R. et al. High-resolution image synthesis with latent diffusion models, 2022

Stable Diffusion (U-Net)



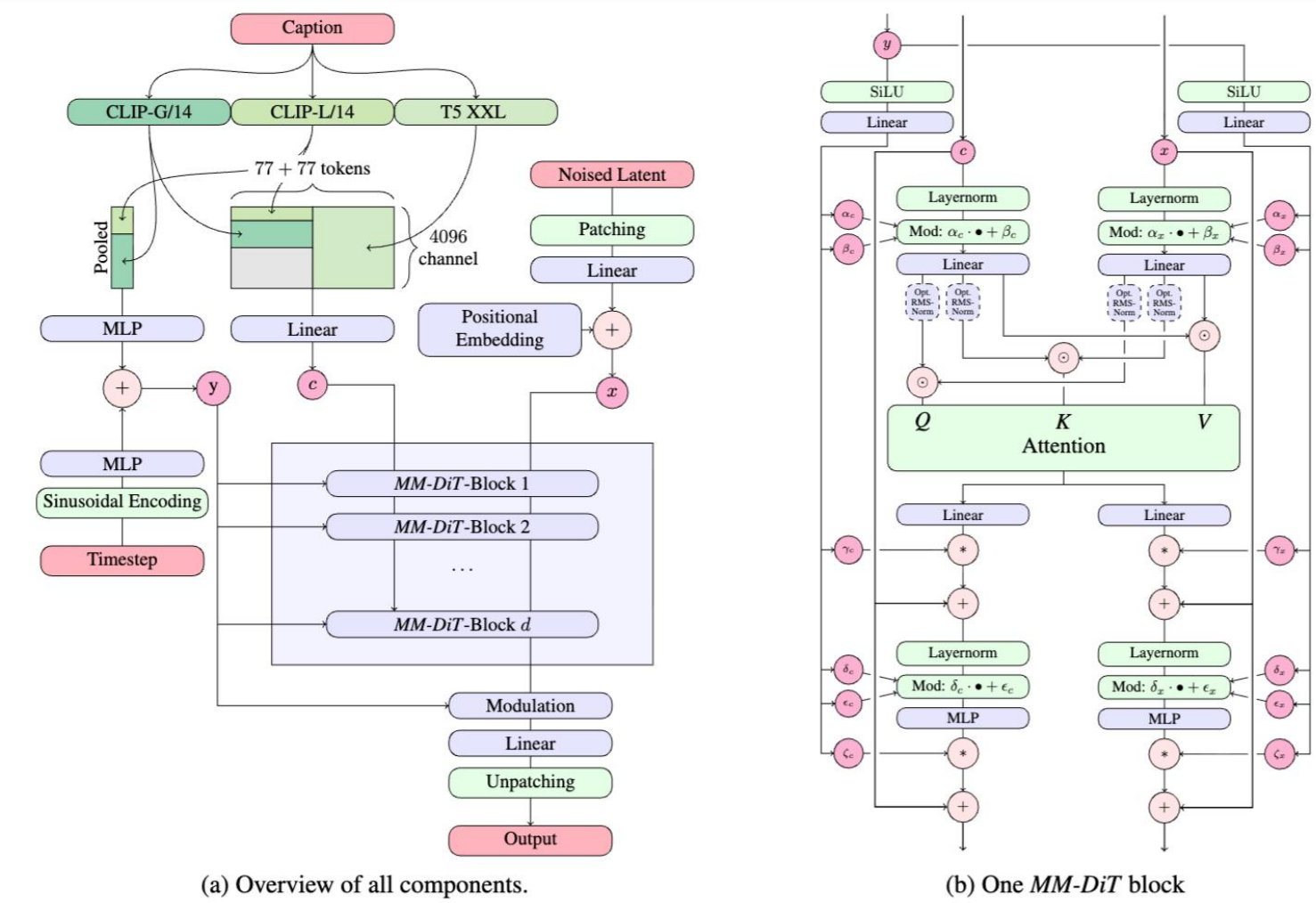
Rombach R. et al. High-resolution image synthesis with latent diffusion models, 2022

Diffusion Transformer



Peebles W., Xie S. Scalable diffusion models with transformers, 2023

Multimodal Diffusion Transformer (MM-DiT)



(a) Overview of all components.

(b) One *MM-DiT* block

Esser P. et al. Scaling rectified flow transformers for high-resolution image synthesis, 2024

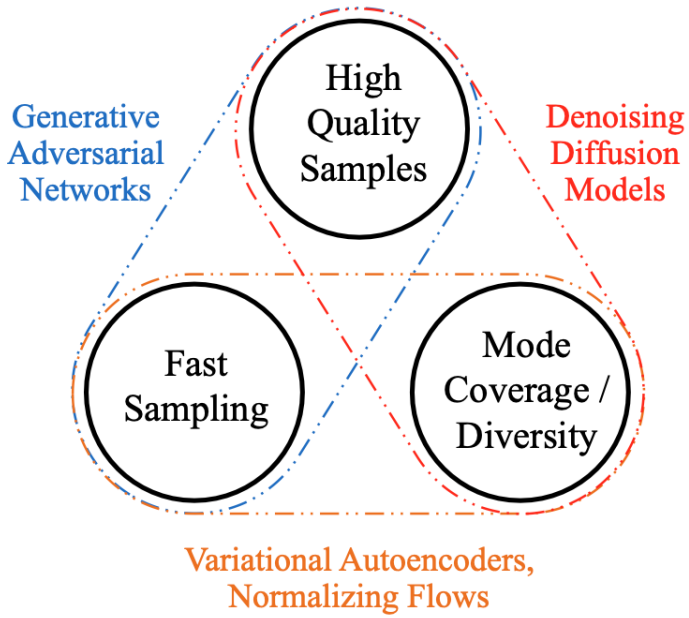
Video Diffusion and more...

Kandinsky 5.0: A Family of Foundation Models for Image and Video Generation, 2025 https://github.com/kandinskylab/kandinsky-5

Recap

- Diffusion Models
- Flow Matching
- Guidance
- Latent Diffusion Models

Overview



Model	Efficient	Sample quality	Coverage	Well-behaved latent space	Disentangled latent space	Efficient likelihood
GANs	✓	✓	✗	✓	?	n/a
VAEs	✓	✗	?	✓	?	✗
Flows	✓	✗	?	✓	?	✓
Diffusion	✗	✓	?	✗	✗	✗

Xiao Z., Kreis K., Vahdat A. Tackling the generative learning trilemma with denoising diffusion GANs, 2021
Simon J.D. Prince. Understanding Deep Learning, 2023

TIME FOR A BREAK