Information Theoretic Foundations of Generative Models

Part 2: Diffusion Models and Metrics for Generative Models

Lalitha Sankar and Monica Welfert

ISIT 2025 Tutorial, Ann Arbor, MI

June 19, 2025

Supported in part by NSF grants CIF-1815361, CIF-2007688, DMS-2134256, and SCH-2205080

Overview of Part II

Maximum Likelihood Estimation

From VAEs to Diffusion Models via ELBO

Diffusion Models
- Forward and Reverse Processes

Conditional Diffusion Models

Evaluation Metrics: Motivation and Limitations

Diffusion Models: Practice

Additional Materials
- Brief History of Generative Diffusion Models

Metrics: Additional Details

Maximum Likelihood Estimation

From Log Likelihood to Evidence Lower Bound (ELBO)

- Key idea: ambient data $x \in \mathcal{X}$ can be mapped to a different (e.g., latent or noisy) data $z \in \mathcal{Z}$
- True distribution $p(x)$: unknown; so are mappings $p(x|z)$ and $q(z|x)$

Figure from Luo (2022)

From Log Likelihood to Evidence Lower Bound (ELBO)

- Key idea: ambient data $x \in \mathcal{X}$ can be mapped to a different (e.g., latent or noisy) data $z \in \mathcal{Z}$
- True distribution $p(x)$: unknown; so are mappings $p(x|z)$ and $q(z|x)$
- Mappings from X to Z and Z to X can be learned as parametrized models $p_\theta(x|z)$ and $q_\phi(z|x)$

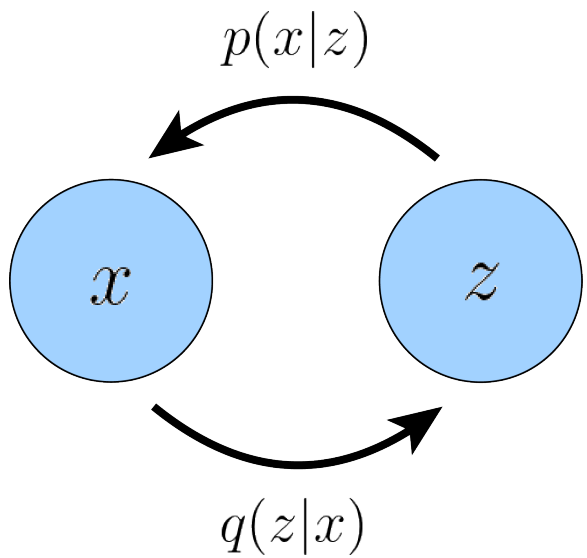


Figure from Luo (2022)

From Maximum Likelihood to Evidence Lower Bound (ELBO)

Log Likelihood aka Evidence:

$$
\begin{aligned}
\log p(x) &= \log p(x) \int q_\phi(z \mid x)\, dz \\
&= \int q_\phi(z \mid x) \big(\log p(x)\big)\, dz \\
&= \mathbb{E}_{q_\phi(z|x)}\big[\log p(x)\big] \\
&= \mathbb{E}_{q_\phi(z|x)}\Big[\log \frac{p(x,z)}{p(z \mid x)}\Big] \\
&= \mathbb{E}_{q_\phi(z|x)}\Big[\log \frac{p(x,z)\, q_\phi(z \mid x)}{p(z \mid x)\, q_\phi(z \mid x)}\Big] \\
&= \mathbb{E}_{q_\phi(z|x)}\Big[\log \frac{p(x,z)}{q_\phi(z \mid x)}\Big] \;+\; D_{\mathrm{KL}}\big(q_\phi(z \mid x) \,\|\, p(z \mid x)\big) \\
&\geq \underbrace{\mathbb{E}_{q_\phi(z|x)}\Big[\log \frac{p(x,z)}{q_\phi(z \mid x)}\Big]}_{\textbf{ELBO}} \quad \text{(KL divergence always } \geq 0\text{)}
\end{aligned}
$$

From ELBO to Variational AutoEncoders (VAEs)

- z: Captures the low-dimensional latent structure of x
- Goal: learn parameters of variational posterior $q_\phi(z|x)$ to match $q(z|x)$
  - Achieved by minimizing $D_{\mathrm{KL}}\big(q_\phi(z \mid x) \,\|\, p(z \mid x)\big)$
  - Not possible to do so directly as $p(z|x)$ is unknown
- Since evidence is independent of ϕ, maximizing ELBO minimizes $D_{\mathrm{KL}}\big(q_\phi(z \mid x) \,\|\, p(z \mid x)\big)$

Maximizing ELBO: helps learn both the latent representation and estimate the evidence!

From ELBO to Variational AutoEncoders (VAEs)

- variational: learn $q_\phi(z|x)$ over a parametrized family
- autoencoder: from ambient x to latent (bottleneck) z back to x

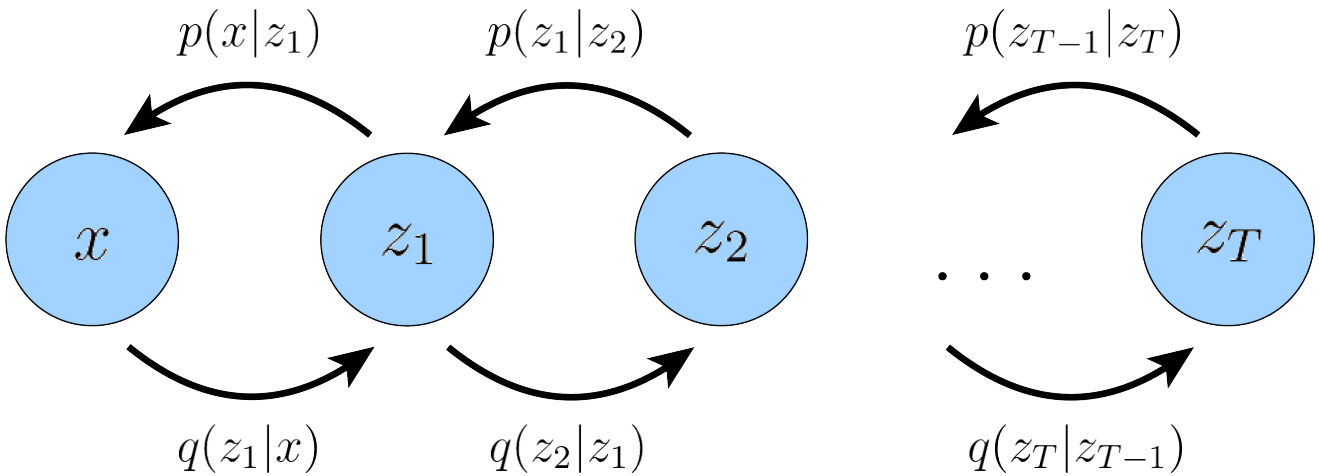
ELBO simplifies to:

$$\begin{aligned}\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right] &= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(z)}{q_\phi(z|x)}\right] \\ &= \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right]}_{\text{reconstruction term}} - \underbrace{D_{\mathrm{KL}}\big(q_\phi(z \mid x) \,\|\, p(z)\big)}_{\text{prior matching term}}\end{aligned}$$

VAEs learn two mappings: Encoder: $q_\phi(z \mid x)$ Decoder: $p_\theta(x \mid z)$

General Hierarchical VAEs

Generalize VAEs to hierarchical models with T latents $z_{1:T}$



Markov Hierarchical VAE

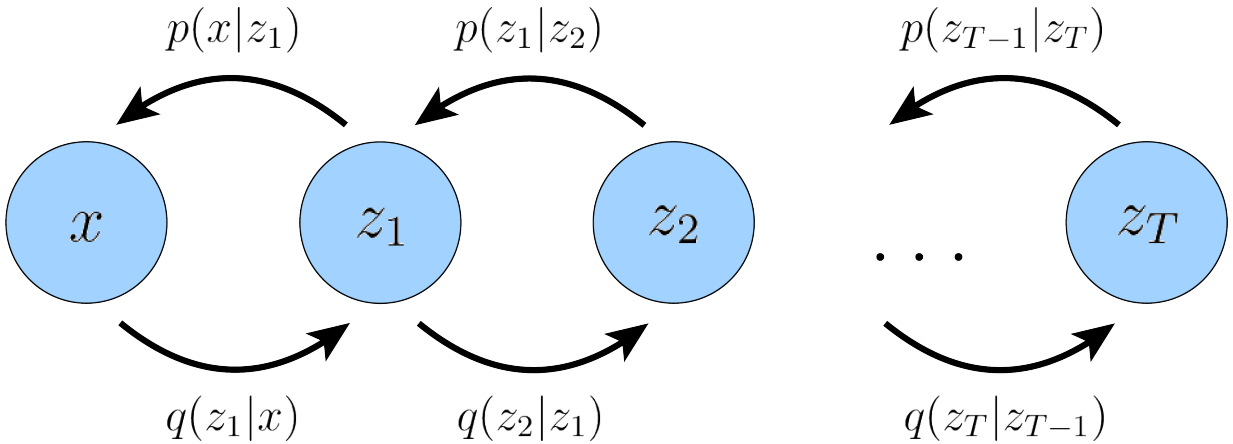
Challenge: Learning T backward $p_\theta(z_{t-1}|z_t)$ and forward $q_\phi(z_t|z_{t-1})$ models!

Figure from Luo (2022)

From VAEs to Diffusion Models via ELBO

VAEs to Diffusion Models

Markov Hierarchical VAEs



- What if the z_i variables are in the same ambient space as x?
  - Now, $x_t := z_t, t \in [1:T]$ are the intermediate variables
- Can we design the forward process to model *diffusion* from x to x_T?
  - **Problem then simplifies to only learning the reverse mappings** $q_\phi(z_{t-1}|z_t)$

Figure from Luo (2022)

Diffusion Models

Diffusion

Figure adapted from thoughtco.com

Diffusion

Diffusion described by Stochastic Differential Equations (SDEs):

$$dx_t = \underbrace{f(x_t, t)}_{\text{drift}} dt + \underbrace{g(t)}_{\text{diffusion}} dw_t, \quad dw_t \sim \mathcal{N}(0, \mathbf{I})$$

- In nature, diffusion is directional
  - Computing: we can design diffusion to increase entropy
  - But can we reverse it?

Can Diffusion be Reversed?

- Diffusion can be approximately reversed if we know the probability distribution of all possible states
- Learning $\nabla_x \log p(x, t)$ is sufficient to guide the time-reversed diffusion process
- Generative diffusion modeled using Ornstein-Uhlenbeck process (mean-reverting) for $t \in [0, T]$

Can Diffusion be Reversed?

- Diffusion can be approximately reversed if we know the probability distribution of all possible states
- Learning $\nabla_x \log p(x,t)$ is sufficient to guide the time-reversed diffusion process
- Generative diffusion modeled using Ornstein-Uhlenbeck process (mean-reverting) for $t \in [0,T]$

Ornstein-Uhlenback (Variance-Preserving) SDE (Chan, 2025):

$$\text{Forward:} \quad d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}; \qquad \frac{d\beta(t)}{dt} \geq 0$$

$$\text{Reverse:} \quad d\mathbf{x} = -\beta(t)\left[\frac{1}{2}\mathbf{x} + \underbrace{\nabla_{\mathbf{x}} \log p(\mathbf{x},t)}_{\text{score}}\right]dt + \sqrt{\beta(t)}d\tilde{\mathbf{w}}$$

Discrete-time Diffusion: Forward Process

- Data $x_0 \sim q(x_0)$ (unknown)
- $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ (additive noise modeling forward diffusion)
- $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ (variance scheduler for signal and noise)

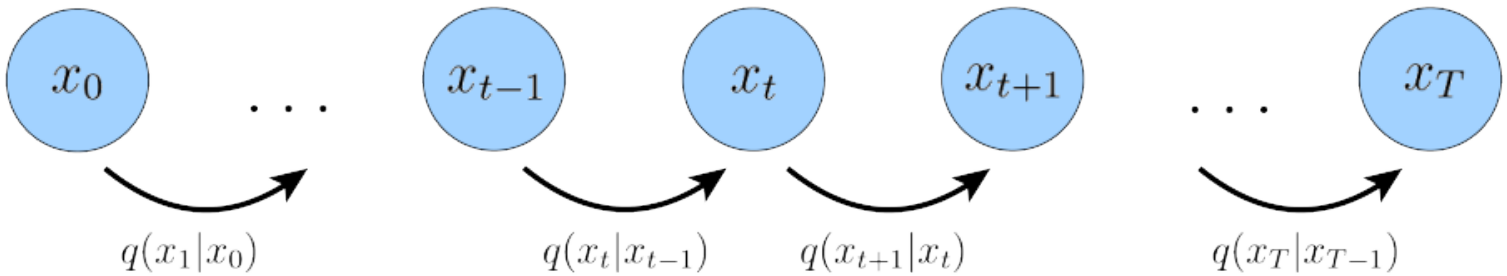
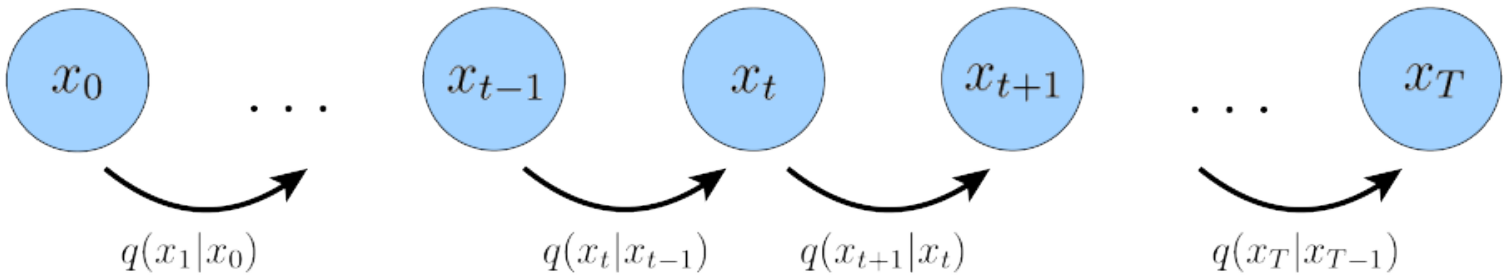


Figure from Luo (2022)

Discrete-time Diffusion: Forward Process

- Data $x_0 \sim q(x_0)$ (unknown)
- $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ (additive noise modeling forward diffusion)
- $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$ (variance scheduler for signal and noise)



Forward Processes (Variance Preserving):

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon, \qquad q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{\alpha_t} x_{t-1},\ (1-\alpha_t)\mathbf{I})$$

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \qquad q(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Figure from Luo (2022)

Revisiting Maximum Likelihood and ELBO for Diffusion

Maximum Likelihood:

$$\max_{\theta} \log p_\theta(x_0)$$

Revisiting Maximum Likelihood and ELBO for Diffusion

$$\max_\theta \log p_\theta(x_0) \geq \max_\theta \underbrace{\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right]}_{\textbf{ELBO}}$$

We can expand ELBO by leveraging Markovity and Gaussianity of forward processes

$$q(x_t \mid x_{t-1}) = q(x_t \mid x_{t-1}, x_0) = \frac{q(x_{t-1} \mid x_t, x_0)\, q(x_t \mid x_0)}{q(x_{t-1} \mid x_0)}$$

Revisiting Maximum Likelihood and ELBO for Diffusion

$$q(x_t \mid x_{t-1}) = q(x_t \mid x_{t-1}, x_0) = \frac{q(x_{t-1} \mid x_t, x_0)\, q(x_t \mid x_0)}{q(x_{t-1} \mid x_0)}$$

- $q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}x_{t-1},\ (1-\alpha_t)\mathbf{I})$
- $q(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$

Revisiting Maximum Likelihood and ELBO for Diffusion

$$q(x_t \mid x_{t-1}) = q(x_t \mid x_{t-1}, x_0) = \frac{q(x_{t-1} \mid x_t, x_0)\, q(x_t \mid x_0)}{q(x_{t-1} \mid x_0)}$$

- $q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{\alpha_t}x_{t-1},\ (1-\alpha_t)\mathbf{I})$
- $q(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}\left(\mu_q(x_t, x_0),\ \Sigma_q(t)\right)$$

where

$$\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_t(1-\bar{\alpha}_{t-1})}\, x_t + \sqrt{\bar{\alpha}_{t-1}(1-\alpha_t)}\, x_0}{1-\bar{\alpha}_t}$$

$$\Sigma_q(t) = \underbrace{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}_{\sigma_q^2(t)} \mathbf{I}$$

Note that $\Sigma_q(t)$ only depends on the variance schedule α_t

ELBO Simplification for Discrete Diffusion Models

$$\max_\theta \log p_\theta(x_0) \geq \max_\theta \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right]$$

$$= \max_\theta \Bigg[\underbrace{\mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0|x_1)]}_{\textbf{reconstruction}}$$

$$- \underbrace{\mathbb{E}_{q(x_T|x_0)}[D_{\mathrm{KL}}(q(x_T|x_0)\|p(x_T))]}_{\text{prior matching}}$$

$$- \underbrace{\sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}[D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t))]}_{\textbf{denoising matching}} \Bigg]$$

ELBO Simplification for Discrete Diffusion Models

$$\max_\theta \log p_\theta(x_0) \geq \max_\theta \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} \right]$$

$$= \max_\theta \Bigg[\underbrace{\mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0|x_1)]}_{\textbf{reconstruction}}$$

$$- \underbrace{\mathbb{E}_{q(x_T|x_0)}[D_{\mathrm{KL}}(q(x_T|x_0)\|p(x_T))]}_{\text{prior matching}}$$

$$- \underbrace{\sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}[D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t))]}_{\textbf{denoising matching}} \Bigg]$$

We showed that $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is Gaussian!

Denoising Diffusion Probabilistic Model (DDPM)

$$\max_{\theta} \log p_\theta(x_0) \equiv \min_{\theta} \underbrace{\sum_{t=1}^{T} \mathbb{E}_{q(x_t|x_0)}[D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))]}_{\mathcal{L}_{\mathrm{DDPM}}}$$

Denoising Diffusion Probabilistic Model (DDPM)

$$\max_\theta \log p_\theta(x_0) \equiv \min_\theta \underbrace{\sum_{t=1}^{T} \mathbb{E}_{q(x_t|x_0)}[D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))]}_{\mathcal{L}_{\mathrm{DDPM}}}$$

Use $$q(x_{t-1}|x_t, x_0) = \mathcal{N}(\mu_q(x_t, x_0), \sigma_q^2(t)\mathbf{I})$$

to learn the reverse denoising distribution

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t, t), \sigma_q^2(t)\mathbf{I})$$

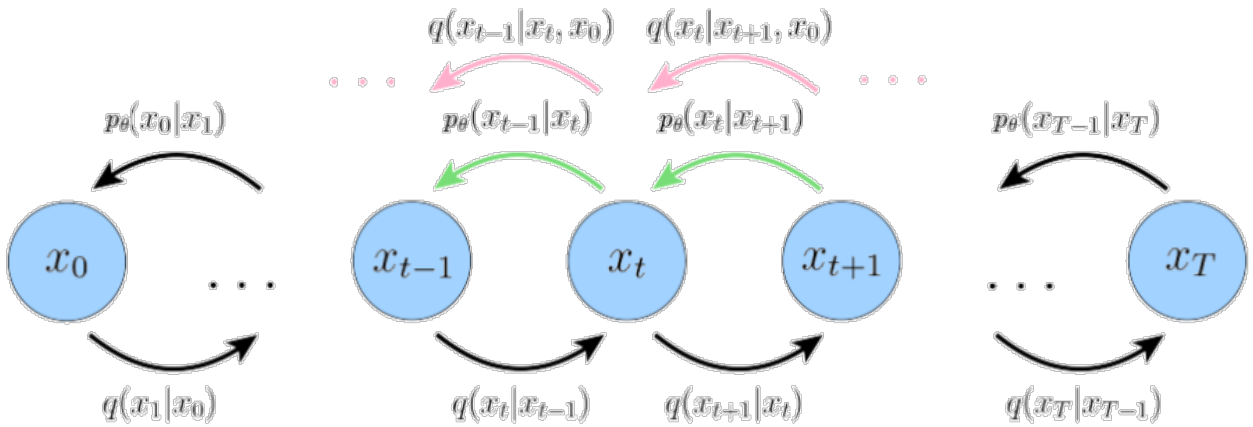


Figure from Luo (2022)

DDPM: Equivalent Objectives

- The KL divergence between two Gaussian distributions is:

$$\begin{aligned}&\mathrm{D}_{\mathrm{KL}}\left(\mathcal{N}(x;\mu_x,\Sigma_x)\,\|\,\mathcal{N}(y;\mu_y,\Sigma_y)\right)\\&=\frac{1}{2}\left(\log\frac{|\Sigma_y|}{|\Sigma_x|}-d+\mathrm{tr}(\Sigma_y^{-1}\Sigma_x)+(\mu_y-\mu_x)^\top\Sigma_y^{-1}(\mu_y-\mu_x)\right)\end{aligned}$$

- For

$$q(x_{t-1}|x_t,x_0)=\mathcal{N}(\mu_q(x_t,x_0),\sigma_q^2(t)\mathbf{I})$$

and

$$p_\theta(x_{t-1}|x_t)=\mathcal{N}(\mu_\theta(x_t,t),\sigma_q^2(t)\mathbf{I})$$

DDPM: Equivalent Objectives

- Over all observed data x_0, $\mathcal{L}_{\text{DDPM}}$ simplifies to:

$$\mathcal{L}_\mu = \mathbb{E}_{t,x_0,\epsilon}\left[\|\mu_q(x_t, x_0) - \mu_\theta(x_t, t)\|^2\right]$$

DDPM: Equivalent Objectives

- Over all observed data x_0, $\mathcal{L}_{\text{DDPM}}$ simplifies to:

$$\mathcal{L}_{\mu} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\mu_q(x_t, x_0) - \mu_\theta(x_t, t)\|^2\right]$$

- Equivalent to minimizing an alternate loss:

$$\mathcal{L}_{x_0} = \mathbb{E}_{t,x_0,\epsilon}\left[\|x_0 - x_\theta(x_t, t)\|^2\right]$$

DDPM: Equivalent Objectives

- Over all observed data x_0, $\mathcal{L}_{\text{DDPM}}$ simplifies to:

$$\mathcal{L}_\mu = \mathbb{E}_{t,x_0,\epsilon}\left[\|\mu_q(x_t, x_0) - \mu_\theta(x_t, t)\|^2\right]$$

- Equivalent to minimizing an alternate loss:

$$\mathcal{L}_{x_0} = \mathbb{E}_{t,x_0,\epsilon}\left[\|x_0 - x_\theta(x_t, t)\|^2\right]$$

- Finally, from $x_t = \sqrt{\bar{\alpha}}x_0 + \sqrt{1-\bar{\alpha}}\epsilon$, we get:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

From DDPM to Score-based Diffusion Models

- Most commonly used loss for DDPM is:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

- Score function related to the noise prediction $\epsilon_\theta(\cdot)$ as:

$$\boxed{\nabla_{x_t} \log p_t(x_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)}$$

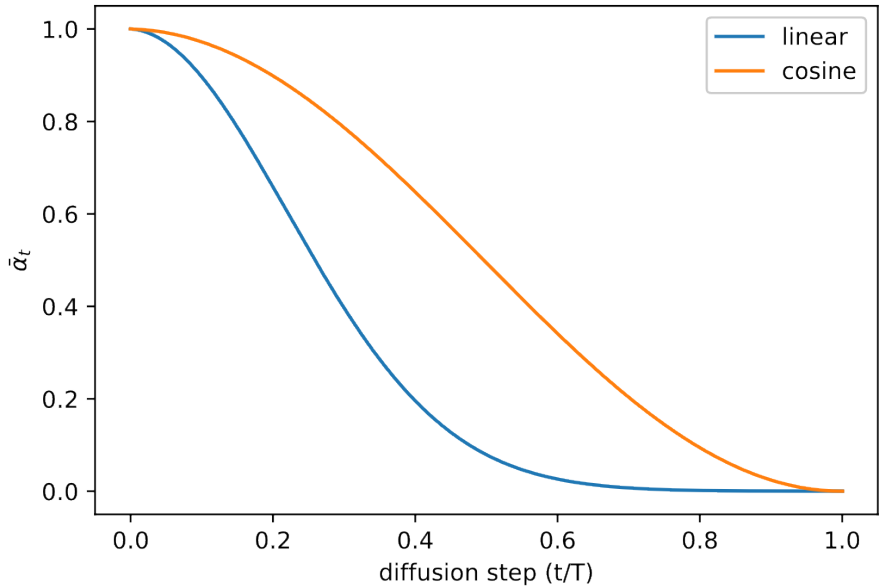
- We then obtain the general score-based diffusion model loss as:

$$\mathcal{L}_{\text{score}} = \frac{1}{2\sigma_q^2(t)} \cdot \frac{(1-\alpha_t)^2}{\alpha_t} \|s_\theta(x_t, t) - \nabla \log p(x_t)\|_2^2$$

- In practice, score-based objective does not use true score[1]

[1] Hyvärinen (2005)

Noise Schedulers in Diffusion Models



- $\bar{\alpha}_1 > \bar{\alpha}_2 > \ldots > \bar{\alpha}_T$
- $\bar{\alpha}_1 \approx 1$ and $\bar{\alpha}_T \approx 0$
- $\bar{\alpha}_T$ sufficiently small to ensure $x_T \approx \mathcal{N}(0, \mathbf{I})$

Images from Dhariwal and Nichol (2021)

Diffusion Model Overview

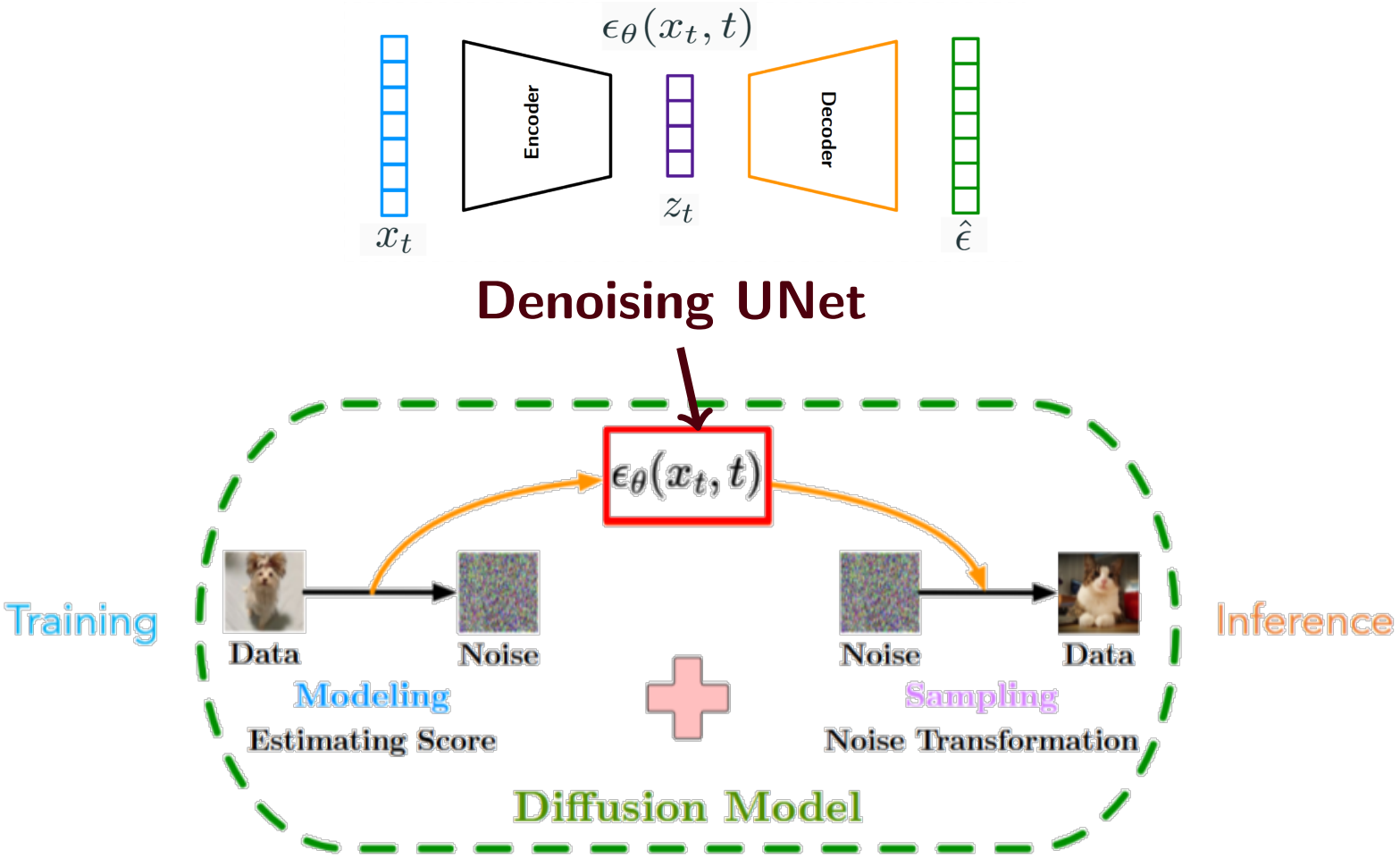


Figure from Wang and Chen

Training and Sampling Algorithms

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** $\mathbf{x}_0$

Algorithms from Jonathan Ho et al. "Denoising Diffusion Porbabilistic Models"

Conditional Diffusion Models

Conditional Diffusion Models

Conditional Generation: $p_\theta(x_0|y)$ where y is the conditioning information

Conditional Diffusion Models

Conditional Generation: $p_\theta(x_0|y)$ where y is the conditioning information

Class labels

"An astronaut riding a horse, by Hiroshige"

Text prompt

(Left) Image from mmlab, **(Right)** Wikipedia

Conditional Diffusion: Using Classifiers to Guide

Classifier-Guided Diffusion[1]**:** Access to labeled data

$$\begin{aligned}\nabla_{x_t} \log p(x_t|y) =& \nabla_{x_t} \log \left(\frac{p(x_t)p(y|x_t)}{p(y)} \right) \\ =& \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y|x_t)\end{aligned}$$

[1] Dhariwal and Nichol (2021)

Conditional Diffusion: Using Classifiers to Guide

Classifier-Guided Diffusion:[1]

$$\begin{aligned}\nabla_{x_t} \log p(x_t|y) &= \nabla_{x_t} \log \left(\frac{p(x_t)p(y|x_t)}{p(y)} \right) \\ &= \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y|x_t)\end{aligned}$$

In practice, use a guidance strength ω:

$$\begin{aligned}\nabla_{x_t} \log p(x_t|y) &\simeq \nabla_{x_t} \log p(x_t) + \omega \nabla_{x_t} \log p(y|x_t) \\ &\simeq \underbrace{-\epsilon_\theta(x_t, t)}_{\text{unconditional score}} + \omega \cdot \underbrace{\nabla_{x_t} \log p_\phi(y|x_t)}_{\text{classifier gradient}}\end{aligned}$$

[1] Dhariwal and Nichol (2021)

Conditional Diffusion: Using Classifiers to Guide

Classifier-Guided Diffusion:[1]

$$\begin{aligned}\nabla_{x_t} \log p(x_t|y) =& \nabla_{x_t} \log \left(\frac{p(x_t)p(y|x_t)}{p(y)} \right) \\ =& \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y|x_t)\end{aligned}$$

In practice, use a guidance strength ω:

$$\begin{aligned}\nabla_{x_t} \log p(x_t|y) &\simeq \nabla_{x_t} \log p(x_t) + \omega \nabla_{x_t} \log p(y|x_t) \\ &\simeq \underbrace{-\epsilon_\theta(x_t, t)}_{\text{unconditional score}} + \omega \cdot \underbrace{\nabla_{x_t} \log p_\phi(y|x_t)}_{\text{classifier gradient}}\end{aligned}$$

Limitation of CGD: requires labels and only works for classes (can use CLIP)

[1] Dhariwal and Nichol (2021)

Conditional Diffusion: Using Classifiers to Guide

In practice:

$$\nabla_{x_t} \log p(x_t|y) \simeq \nabla_{x_t} \log p(x_t) + \omega \nabla_{x_t} \log p(y|x_t)$$

$$\simeq \underbrace{-\epsilon_\theta(x_t, t)}_{\text{unconditional score}} + \omega \cdot \underbrace{\nabla_{x_t} \log p_\phi(y|x_t)}_{\text{classifier gradient}}$$

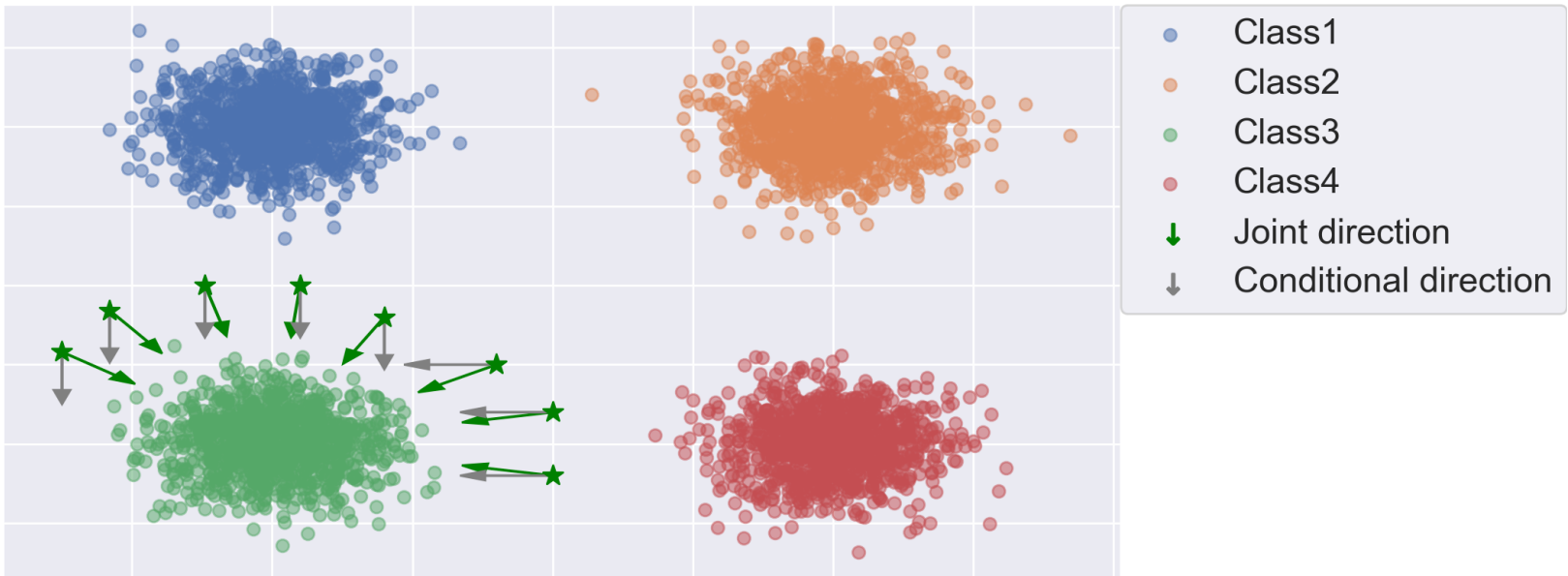


Figure from Ma et al. (2024)

Conditional Diffusion: Classifier-Free

Classifier-Free Guidance[1]:

$$\begin{aligned}
\nabla_{x_t} \log p(x_t|y) &= \nabla_{x_t} \log \left(\frac{p(x_t)p(y|x_t)}{p(y)} \right) \\
&= \nabla_{x_t} \log p(x_t) + \nabla_{x_t} \log p(y|x_t) \\
&\simeq \nabla_{x_t} \log p(x_t) + \omega \nabla_{x_t} \log p(y|x_t) \\
&= \nabla_{x_t} \log p(x_t) + \omega(\nabla_{x_t} \log p(x_t|y) - \nabla_{x_t} \log p(x_t))
\end{aligned}$$

$$\epsilon_\theta^{\text{CFG}}(x_t, t, y) = (1 - \omega) \cdot \epsilon_\theta(x_t, t, \emptyset) + \omega \cdot \epsilon_\theta(x_t, t, y)$$

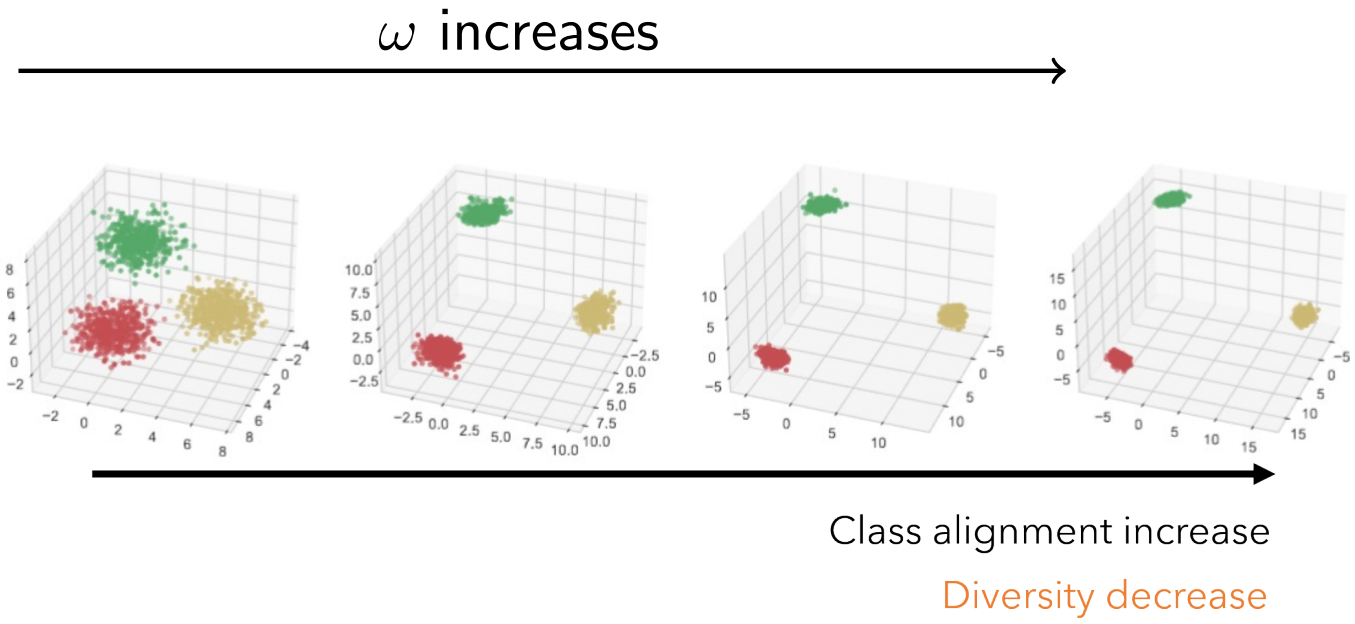
ω: classifier-free guidance parameter

[1] Ho and Salimans (2021)

Conditional Diffusion: Classifier-Free

Classifier-Free Guidance[1]:

$$\epsilon_\theta^{\mathrm{CFG}}(x_t, t, y) = (1-\omega) \cdot \epsilon_\theta(x_t, t, \emptyset) + \omega \cdot \epsilon_\theta(x_t, t, y)$$

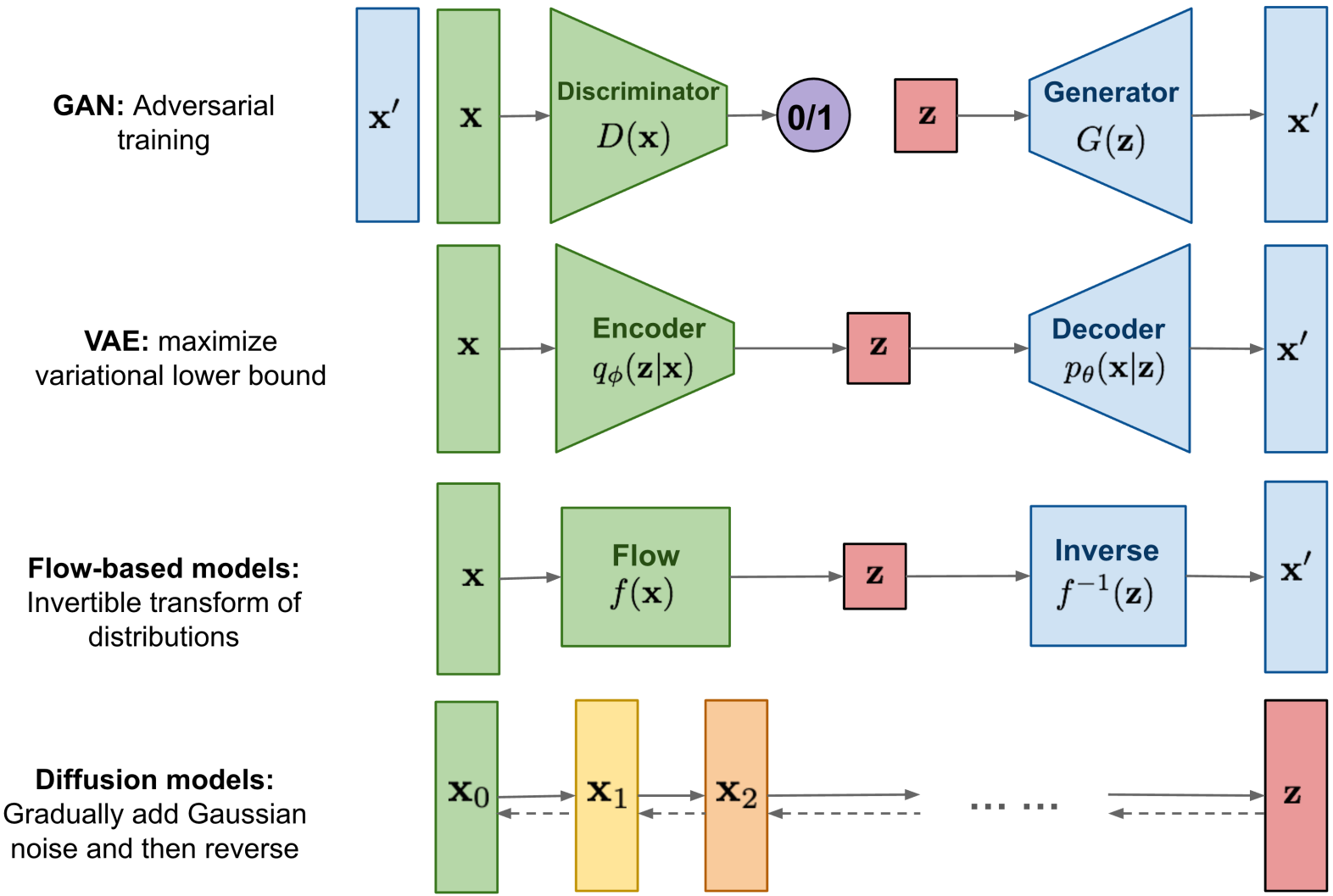


[1] Ho and Salimans (2021)

Figure from Wang and Chen

See: Brief History of Generative Diffusion Models [1]

Visual of Generative Models for Images



Evaluation Metrics: Motivation and Limitations

The Evaluation Challenge in Generative Models

How do we measure the quality of generated samples?
Human evaluation is:

- Expensive and time-consuming
- Subjective and inconsistent
- Not scalable for model development
- Difficult to reproduce across studies

Need automated metrics that correlate with human judgment while being:

- Computationally efficient
- Theoretically principled
- Robust across different domains
- Sensitive to subtle quality differences

The challenge is even greater for text-to-image models, where we must evaluate both image quality and semantic alignment with text prompts.

Evolution of Evaluation Metrics

First Generation : Limited to simple datasets and single-domain evaluation, widely adopted

- Distribution-free approach: MMD (Müller, 1997; Gretton et al., 2012)
- Inception Score (IS): Uses classifier confidence and diversity (Salimans et al., 2016)
- Fréchet Inception Distance (FID): Distribution-based comparison (Heusel et al., 2017)
- (Improved) Precision and Recall for Generative Models (Sajjadi et al., 2018; Kynkäänniemi et al., 2019)

Next Generation: Multi-modal evaluation: Quality + semantic alignment

- CLIP-based metrics: Semantic understanding for text-to-image (Hartwig et al., 2025)
- CLIP-MMD: combines empirical motivated CLIP with theoretically motivated MMD (Jayasumana et al., 2024a)

Maximum Mean Discrepancy (MMD)

MMD: Compare distributions by comparing their means in a rich feature space (Integral Probability Metric in statistics)[1]

$$\text{MMD}[\mathcal{F}, P, Q] = \sup_{f \in \mathcal{F}} \left(\mathbb{E}_{x \sim P}[f(x)] - \mathbb{E}_{y \sim Q}[f(y)] \right)$$

Properties:

- **Distribution-free:** No assumptions about the shape of P or Q
- **Characteristic:** $\text{MMD}(P, Q) = 0$ if and only if $P = Q$ (for appropriate kernels)
- **Metricity:** Satisfies triangle inequality and symmetry
- **Computational efficiency:** $O(n^2)$ vs FID's $O(d^3)$

Choice of function class $\mathcal{F}$ determines the discriminative power of metric

[1] Gretton et al. (2012); Sutherland et al. (2021)

MMD with Reproducing Kernel Hilbert Spaces

MMD has a closed form if $\mathcal{F}$ is unit ball in a RKHS with kernel k:[1]

$$\text{MMD}^2(P, Q) = \mathbb{E}_{x,x' \sim P}[k(x, x')] + \mathbb{E}_{y,y' \sim Q}[k(y, y')] - 2\mathbb{E}_{x \sim P, y \sim Q}[k(x, y)]$$

- Gaussian RBF Kernel is the most commonly used kernel for MMD:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{\sigma^2}\right)$$

- Bandwidth parameter σ controls sensitivity
  - Smaller σ focuses on local differences
  - Larger σ captures global structure

[1] A Reproducing Kernel Hilbert Space (RKHS) is a Hilbert space of functions where the evaluation of a function at a point can be represented as an inner product with a specific function, called the reproducing kernel.

MMD Empirical Estimation

Given samples $X = \{x_1, \ldots, x_m\}$ from P and $Y = \{y_1, \ldots, y_n\}$ from Q:

Unbiased U-statistic Estimator:

$$\widehat{\mathrm{MMD}}_u^2(X, Y) = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i} k(x_i, x_j) \tag{1}$$

$$+ \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} k(y_i, y_j) \tag{2}$$

$$- \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j) \tag{3}$$

Advantages:

- **Unbiased:** $\mathbb{E}[\widehat{\mathrm{MMD}}_u^2] = \mathrm{MMD}^2$ exactly
- **Consistent:** Converges to true value as sample size increases
- **Efficient:** $O(n^2)$ computation using matrix operations
- **Parallelizable:** Kernel matrix computation maps well to GPUs

MMD Statistical Properties

MMD comes with strong theoretical guarantees that FID lacks.
Finite Sample Concentration: With probability $1 - \delta$:

$$|\text{MMD}(P, Q) - \widehat{\text{MMD}}(X, Y)| \leq \epsilon_n(\delta)$$

where $\epsilon_n(\delta) = O(n^{-1/2})$.
Asymptotic Distribution: Under null hypothesis $P = Q$:

$$(m + n)\widehat{\text{MMD}}_u^2 \xrightarrow{d} \sum_{l=1}^{\infty} \lambda_l Z_l^2$$

where $Z_l \sim \mathcal{N}(0, 1)$ and λ_l are eigenvalues of the kernel operator.
Power Analysis: MMD can distinguish distributions separated by distance $\delta = O(n^{-1/2})$, matching the minimax optimal rate.
These properties enable rigorous hypothesis testing and confidence intervals.

Inception Score (IS)

Inception Score (IS) introduced to evaluate GANs in 2016[1]

IS measures how much the conditional label distribution differs from the marginal distribution

$$\text{IS}(G) = \exp\left(\mathbb{E}_{x \sim p_g}[\text{KL}(p(y|x) \| p(y))]\right)$$

- $p(y|x)$ = Inception-v3 classifier predictions for generated image x
- $p(y) = \mathbb{E}_{x \sim p_g}[p(y|x)]$ = marginal label distribution

Interpretation: Higher scores indicate both quality and diversity

[1] Heusel et al. (2017)

Advantages and Limitations of IS

Advantages

- ✓ Simple to compute and understand

Limitations

- × Only looks at generated images, not real
- × Cannot distinguish quality vs. diversity
- × Limited to ImageNet classes
- × Associates "good image" with "easy to classify"
- × Fooled by images that confuse the classifier
- × Blind to mode collapse

The Fundamental Problem: IS measures how well images fit ImageNet categories, not how realistic or high-quality they actually are

Fréchet Inception Distance

Fréchet Inception Distance (FID):

- Fit multivariate Gaussian distribution to real (p_r) and generated (p_g) images in feature space of a pretrained classifier
- μ_r, μ_g: feature means
- Σ_r, Σ_g: feature covariances

$$\mathrm{FID}(p_r, p_g) = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}),$$

Fréchet Inception Distance

Fréchet Inception Distance (FID):

- Fit multivariate Gaussian distribution to real (p_r) and generated (p_g) images in feature space of a pretrained classifier
- μ_r, μ_g: feature means
- Σ_r, Σ_g: feature covariances

$$\mathrm{FID}(p_r, p_g) = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}),$$

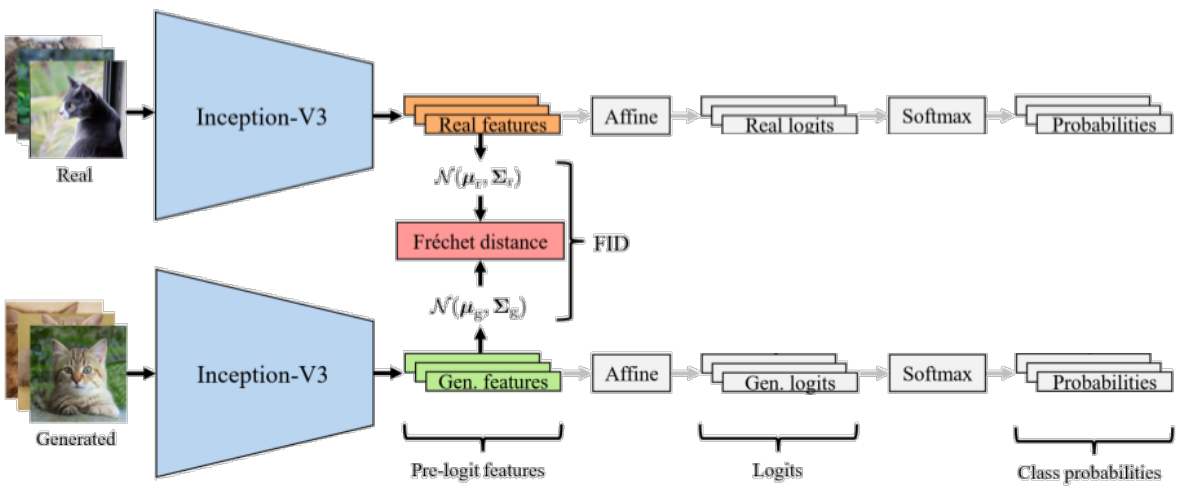


Image from Kynkäänniemi et al. (2023)

Advantages and Limitation of FID

Advantages

- ✓ Correlates with human judgment when using modern feature extractors (e.g. DINOv2)
- ✓ No labeled data required
- ✓ Computationally efficient
- ✓ Sensitive to mode collapse

Limitations

- × Affected by sample size
- × Cannot distinguish quality vs. diversity
- × Depends on feature extractor
- × Depends on image resizing implementations
- × Insensitive to noise in unimportant features

KID: Inceptionv3+MMD

Kernel Inception Distance (KID):

- FID assumes Gaussian distribution for the low-dimensional representions
- Such an assumption does not hold in practice

Kernel Inception Distance applies the MMD distance metric to the Inceptionv3 representations

- More generally, better low-dimensional representations can be obtained with autoencoders, e.g., DINOv2 (Oquab et al., 2024)

Combining meaningful encoders and distance metrics is on-going research

Why Precision and Recall for Generative Models?

Traditional generative model evaluation often relies on single metrics like FID, which can be misleading

- A model with good FID might suffer from **mode collapse** (high quality, low diversity)
- Another model might have **poor sample fidelity** but good coverage
- Single metrics cannot distinguish between these failure modes

By separating quality from diversity, we can better understand model failures and guide improvements in generative model development

- **Precision**: Measures sample fidelity
- **Recall**: Measures sample diversity

Precision: Measuring Sample Quality

Definition: Given real $\{x_i^r\}_{i=1}^n$ and generated $\{x_j^g\}_{j=1}^m$ samples (in a representation space):

$$\text{Precision} = \frac{1}{m}\sum_{j=1}^{m} \mathbf{1}\left[x_j^g \in S(\{x_i^r\}_{i=1}^n)\right]$$

where $\mathbf{1}[\cdot]$ is the indicator function and $S(\mathcal{X})$ represents neighborhoods around samples in $\mathcal{X}$

Interpretation:

- Measures **fraction of generated samples** that fall within neighborhoods of real samples
- Answers the question: *"Are my generated samples realistic?"*
- High precision means few unrealistic or "obviously fake" samples
- Focuses on sample **fidelity** rather than diversity

Recall: Measuring Sample Diversity

Definition: Given real $\{x_i^r\}_{i=1}^n$ and generated $\{x_j^g\}_{j=1}^m$ samples (in a representation space):

$$\text{Recall} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left[x_i^r \in S(\{x_j^g\}_{j=1}^m)\right]$$

where $S(\mathcal{X})$ represents neighborhoods around samples in $\mathcal{X}$

Interpretation:

- Measures **fraction of real samples** that have nearby generated samples
- Answers the question: *"Do I cover the real distribution well?"*
- High recall means good coverage of different modes in the real data
- Focuses on sample **diversity** rather than individual quality

Defining Neighborhoods in Representation Space

The neighborhood function $S(\mathcal{X})$ is typically defined using k-nearest neighbors:

$$S(\mathcal{X}) = \bigcup_{x \in \mathcal{X}} B(x, \text{NND}_k(x))$$

where:

- $B(x, r)$ is a ball of radius r centered at x
- $\text{NND}_k(x)$ is the distance to the k-th nearest neighbor of x
- Typically $k = 3$ or $k = 5$ in practice

Defining Neighborhoods in Representation Space

For computational efficiency, this is often implemented as:

$$\text{Precision} = \frac{1}{m}\sum_{j=1}^{m} \mathbf{1}\left[\min_i d(f(x_j^g), f(x_i^r)) \leq \text{NND}_k(f(x_j^g))\right]$$

$$\text{Recall} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left[\min_j d(f(x_i^r), f(x_j^g)) \leq \text{NND}_k(f(x_i^r))\right]$$

where $f(\cdot)$ is the feature extraction function and $d(\cdot, \cdot)$ is the distance metric (usually Euclidean).

Common Failure Modes Through Precision/Recall Lens

Mode Collapse:

- **Precision**: High (generates realistic samples)
- **Recall**: Low (misses many modes of real data)
- **Problem**: Model generates high-quality but repetitive samples
- **Example**: GAN that only generates one type of digit

Mode Spreading/Blurriness:

- **Precision**: Low (generates unrealistic samples)
- **Recall**: High (covers the distribution well)
- **Problem**: Model covers all regions but with poor quality
- **Example**: VAE with overly smooth, blurry outputs

Memorization: The Need for New Metrics

The Problem: Models can achieve perfect precision and recall scores by simply memorizing training data

Precision/Recall fail to identify the following problems:

- **Memorizing a subset**: High precision, low recall (looks like mode collapse)
- **Memorizing all data**: High precision, high recall (looks like a good model!)
- Cannot distinguish memorization from learning

What We Need: Metrics that explicitly detect training data memorization, not just distributional similarity

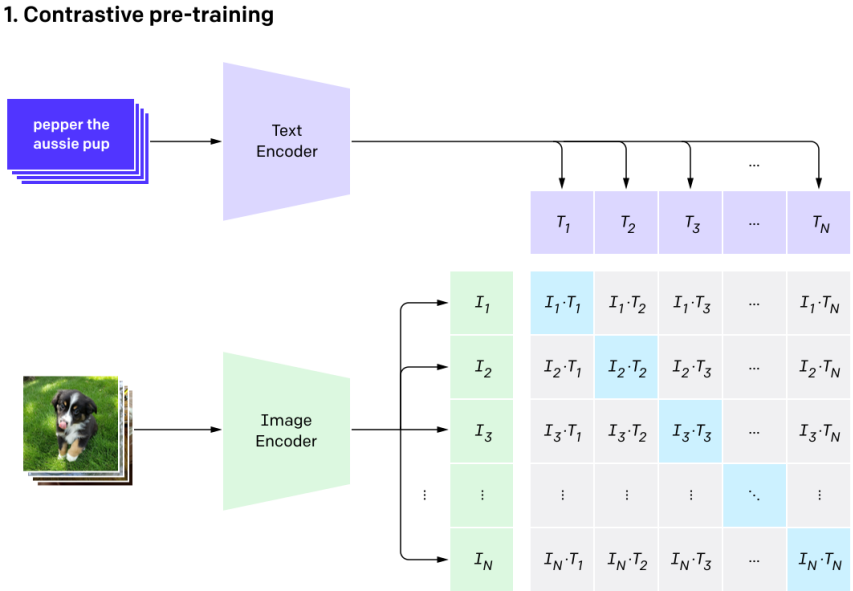
Memorization Metrics:

CLIP: Bridging Vision and Language

CLIP (Contrastive Language-Image Pre-training) revolutionized multimodal understanding by learning joint representations.

- Dataset: 400 million image-text pairs from internet
- Architecture: Separate encoders for images and text that map to the same lower dimensional space.
- Objective: Contrastive loss used to align matching pairs

Major Improvement: Instead of learning from labeled datasets with fixed categories, CLIP learns from natural language descriptions



CLIP's Advantages

Advantages

- ✓ Natural language supervision enables rich semantic understanding
- ✓ Understands complex scenes and compositions
- ✓ Joint vision-language space enables text-to-image evaluation
- ✓ CLIP features correlate better with human evaluation

CLIP in Training vs Evaluation

CLIP in Text-to-Image Training:

- CLIP text encoder conditions diffusion models via cross-attention
- Models learn to generate images aligned with CLIP's text understanding
- CLIP provides the semantic bridge between prompts and visual content

CLIP in Evaluation Metrics:

- CLIP image encoder extracts semantic embeddings from generated images
- Distribution-based metrics (CLIP-FID, CMMD) compare embedding distributions
- Alignment metrics measure text-image similarity in CLIP space

Key Distinction: CLIP text encoder guides training, while CLIP image encoder enables evaluation — related but distinct components of the same model.

CLIP-Based Metrics

Metric	Measures	Advantages	Limitations
CLIP Score Radford et al. (2021)	Text-image alignment	Simple, direct	Ignores overall quality
CLIP-FID Jayasumana et al. (2024b)	Distribution similarity	Semantic features	Gaussian assumptions
CMMD Yang et al. (2024)	Distribution similarity	Distribution free and semantic richness	Quality assumption in CLIP space
CLIP Precision/Recall Cheng et al. (2024)	Quality vs diversity	Disentangled metrics	Complex interpretation
CLIP Aesthetic Score Schuhmann et al. (2022)	Visual appeal	Human-calibrated	Subjective concept