# Deep Generative Models

## Lecture 7

Roman Isachenko

Moscow Institute of Physics and Technology
Yandex School of Data Analysis

2025, Autumn

# Recap of Previous Lecture

## Likelihood-Free Learning

- Likelihood isn't a perfect metric for generative models.
- Likelihood may be intractable.

Imagine we have two sets of samples:

- $\{\mathbf{x}_i\}_{i=1}^{n_1} \sim \pi(\mathbf{x})$ – real samples;
- $\{\mathbf{x}_i\}_{i=1}^{n_2} \sim p(\mathbf{x}|\boldsymbol{\theta})$ – generated (fake) samples.

$p(y = 1|\mathbf{x}) = P(\mathbf{x} \sim \pi(\mathbf{x}))$; $\quad p(y = 0|\mathbf{x}) = P(\mathbf{x} \sim p(\mathbf{x}|\boldsymbol{\theta}))$

## Assumption

The generative distribution $p(\mathbf{x}|\boldsymbol{\theta})$ matches the true distribution $\pi(\mathbf{x})$ if we can't distinguish between them using a discriminative model $p(y|\mathbf{x})$.

- **Generator:** a generative model $\mathbf{x} = \mathbf{G}(\mathbf{z})$ that produces more realistic samples.
- **Discriminator:** a classifier $D(\mathbf{x}) \in [0, 1]$ distinguishing real from generated samples.

# Recap of Previous Lecture

## GAN Optimality Theorem

The minimax game

$$\min_G \max_D \Big[\underbrace{\mathbb{E}_{\pi(\mathbf{x})} \log D(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D(\mathbf{G}(\mathbf{z})))}_{V(G,D)}\Big]$$

has a global optimum at $\pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta})$, and then $D^*(\mathbf{x}) = 0.5$.

$$\min_G V(G, D^*) = \min_G \left[2\mathrm{JSD}(\pi \| p) - \log 4\right] = -\log 4, \quad \pi(\mathbf{x}) = p(\mathbf{x}|\boldsymbol{\theta}).$$

If the generator can be **any** function and the discriminator is **optimal** at each step, then the generator is **guaranteed to converge** to the data distribution.

---

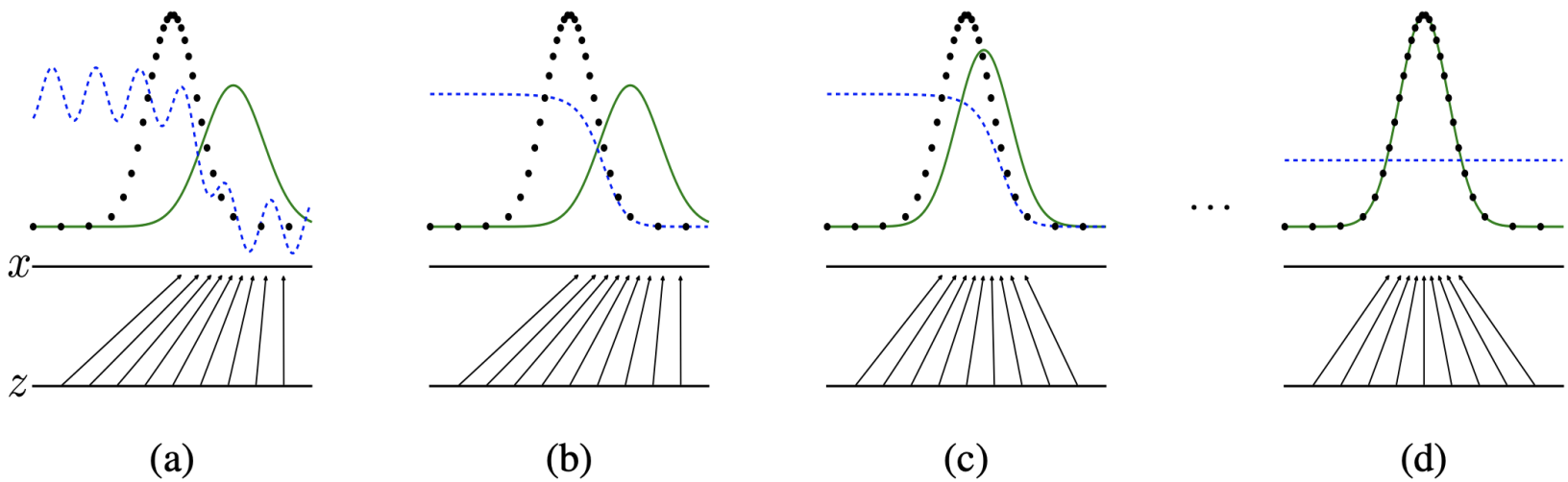
*Goodfellow I. J. et al. Generative Adversarial Networks, 2014*

# Recap of Previous Lecture

- The generator is updated in the parameter space; the discriminator isn't optimal at every iteration.
- Both generator and discriminator loss typically oscillate during GAN training.

## Objective

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \left[ \mathbb{E}_{\pi(\mathbf{x})} \log D_{\boldsymbol{\phi}}(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D_{\boldsymbol{\phi}}(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}))) \right]$$




*Goodfellow I. J. et al. Generative Adversarial Networks, 2014*

# Recap of Previous Lecture

## Main Issues With Standard GANs

- Vanishing gradients (solution: non-saturating GAN).
- Mode collapse (arises from Jensen-Shannon divergence).

## Standard GAN

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \left[ \mathbb{E}_{\pi(\mathbf{x})} \log D_{\boldsymbol{\phi}}(\mathbf{x}) + \mathbb{E}_{p(\mathbf{z})} \log(1 - D_{\boldsymbol{\phi}}(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}))) \right]$$

## Informal Theoretical Results

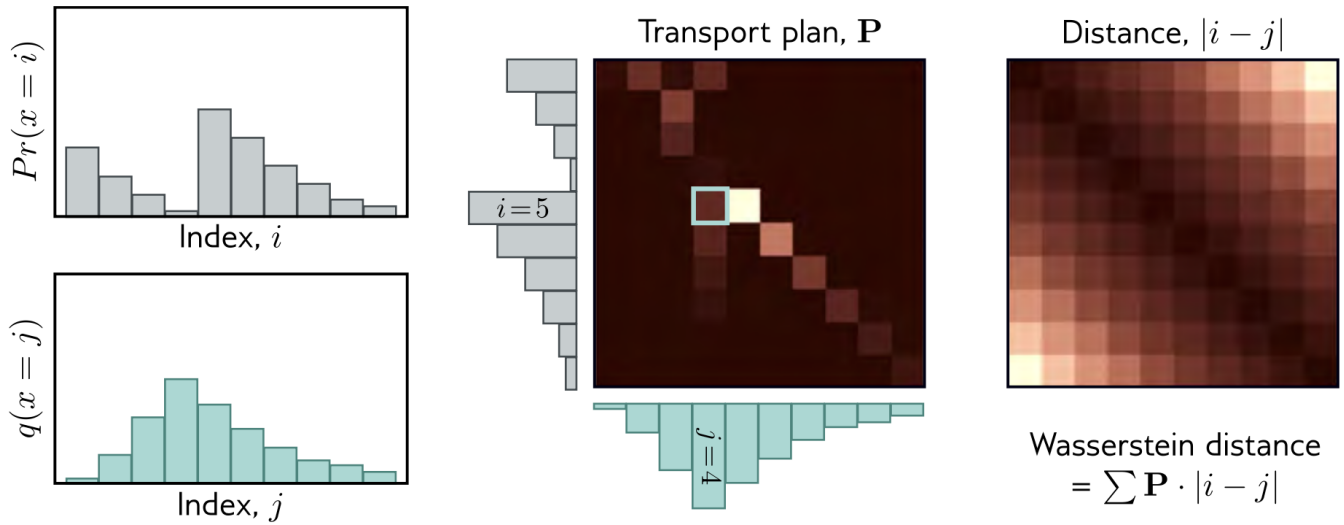
Both the data distribution $\pi(\mathbf{x})$ and the generative distribution $p(\mathbf{x}|\boldsymbol{\theta})$ are low-dimensional with disjoint supports. In such cases,

$$\mathrm{KL}(\pi \| p) = \mathrm{KL}(p \| \pi) = \infty, \quad \mathrm{JSD}(\pi \| p) = \log 2.$$

---

*Goodfellow I. J. et al. Generative Adversarial Networks, 2014*
*Arjovsky M., Bottou L. Towards Principled Methods for Training Generative Adversarial Networks, 2017*

# Recap of Previous Lecture




## Wasserstein Distance

$$W(\pi, p) = \inf_{\gamma \in \Gamma(\pi,p)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} \|\mathbf{x}-\mathbf{y}\| = \inf_{\gamma \in \Gamma(\pi,p)} \int \|\mathbf{x}-\mathbf{y}\| \gamma(\mathbf{x},\mathbf{y}) d\mathbf{x} d\mathbf{y}$$

- $\gamma(\mathbf{x}, \mathbf{y})$ – transportation plan (amount of "dirt" to transport from $\mathbf{x}$ to $\mathbf{y}$).
- $\Gamma(\pi, p)$ – set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ with marginals $\pi$ and $p$ ($\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} = p(\mathbf{y})$, $\int \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \pi(\mathbf{x})$).
- $\gamma(\mathbf{x}, \mathbf{y})$ – the amount; $\|\mathbf{x} - \mathbf{y}\|$ – the distance.

---

*Simon J.D. Prince. Understanding Deep Learning, 2023*

# Recap of Previous Lecture

## Theorem (Kantorovich-Rubinstein Duality)

$$W(\pi\|p) = \frac{1}{K} \max_{\|f\|_L \le K} \left[\mathbb{E}_{\pi(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} f(\mathbf{x})\right],$$

where $\|f\|_L \le K$ denotes $K$-Lipschitz continuous functions.

## WGAN Objective

$$\min_{\boldsymbol{\theta}} W(\pi\|p) = \min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}\in\boldsymbol{\Phi}} \left[\mathbb{E}_{\pi(\mathbf{x})} f_{\boldsymbol{\phi}}(\mathbf{x}) - \mathbb{E}_{p(\mathbf{z})} f_{\boldsymbol{\phi}}(\mathbf{G}_{\boldsymbol{\theta}}(\mathbf{z}))\right].$$

- The function $f$ in WGAN is called the *critic*.
- If parameters $\boldsymbol{\phi}$ lie in a compact set $\boldsymbol{\Phi} \in [-c, c]^d$, then $f(\mathbf{x}, \boldsymbol{\phi})$ is $K$-Lipschitz continuous.

$$K \cdot W(\pi\|p) = \max_{\|f\|_L \le K} \left[\mathbb{E}_{\pi(\mathbf{x})} f(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} f(\mathbf{x})\right] \ge$$

$$\ge \max_{\boldsymbol{\phi}\in\boldsymbol{\Phi}} \left[\mathbb{E}_{\pi(\mathbf{x})} f_{\boldsymbol{\phi}}(\mathbf{x}) - \mathbb{E}_{p(\mathbf{x})} f_{\boldsymbol{\phi}}(\mathbf{x})\right]$$

---

*Arjovsky M., Chintala S., Bottou L. Wasserstein GAN, 2017*

# Outline

# Outline

# Evaluation of Likelihood-Free Models

## Likelihood-Based Models

- **Train:** fit the model.
- **Validation:** tune hyperparameters.
- **Test:** assess generalization by reporting likelihood.

Not all models have tractable likelihoods
(VAE: compare ELBO values; GAN: **???**).

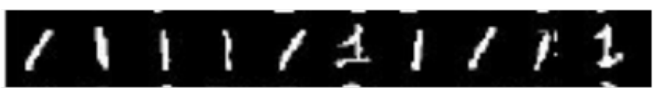
## Desirable Properties for Samples

- Sharpness

Low sharpness

High sharpness

- Diversity



Low diversity

High diversity

---

*image credit: https://deepgenerativemodels.github.io*

# Outline

# Wasserstein Metric

$$W_s(\pi, p) = \inf_{\gamma \in \Gamma(\pi, p)} \left( \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} \|\mathbf{x} - \mathbf{y}\|^s \right)^{1/s}$$

## Wasserstein GAN (Optimal Transport)

$$W(\pi, p) = \inf_{\gamma \in \Gamma(\pi, p)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} \|\mathbf{x}-\mathbf{y}\| = \inf_{\gamma \in \Gamma(\pi, p)} \int \|\mathbf{x} - \mathbf{y}\| \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}$$

## Theorem

If $\pi(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_\pi, \boldsymbol{\Sigma}_\pi)$, $p(\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$, then

$$W_2^2(\pi, p) = \|\boldsymbol{\mu}_\pi - \boldsymbol{\mu}_p\|^2 + \text{tr} \left[ \boldsymbol{\Sigma}_\pi + \boldsymbol{\Sigma}_p - 2 \left( \boldsymbol{\Sigma}_\pi^{1/2} \boldsymbol{\Sigma}_p \boldsymbol{\Sigma}_\pi^{1/2} \right)^{1/2} \right]$$

## Frechet Inception Distance

$$\text{FID}(\pi, p) = W_2^2(\pi, p)$$

---

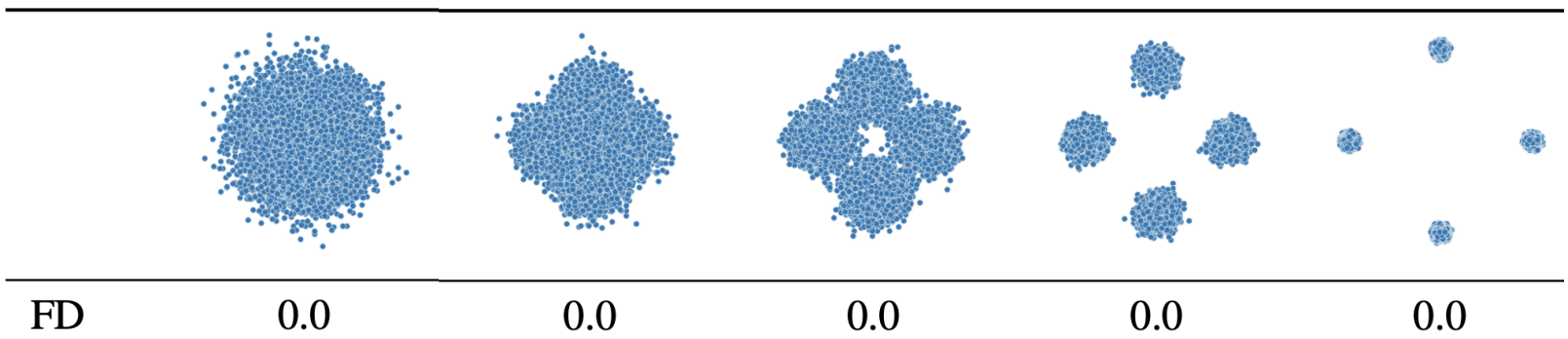
*Heusel M. et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, 2017*

# Frechet Inception Distance (FID)

$$\mathrm{FID}(\pi, p) = \|\boldsymbol{\mu}_\pi - \boldsymbol{\mu}_p\|^2 + \mathrm{tr}\left[\boldsymbol{\Sigma}_\pi + \boldsymbol{\Sigma}_p - 2\left(\boldsymbol{\Sigma}_\pi^{1/2}\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_\pi^{1/2}\right)^{1/2}\right]$$

- FID is computed in the latent space $\mathbf{z}$.
- We use a pretrained image embedder to get latent representations $\mathbf{z} = \mathbf{f}(\mathbf{x})$.
- $\boldsymbol{\mu}_\pi$, $\boldsymbol{\Sigma}_\pi$ and $\boldsymbol{\mu}_p$, $\boldsymbol{\Sigma}_p$ are statistics of latent representations for samples from $\pi(\mathbf{x})$ and $p(\mathbf{x}|\boldsymbol{\theta})$.

*FID*$(p(\mathbf{x}), \mathcal{N}(0, \mathbf{I}))$




*Jayasumana S. et al. Rethinking FID: Towards a Better Evaluation Metric for Image Generation, 2024*

# Frechet Inception Distance (FID)

$$\mathrm{FID}(\pi, p) = \|\boldsymbol{\mu}_\pi - \boldsymbol{\mu}_p\|^2 + \mathrm{tr}\left[\boldsymbol{\Sigma}_\pi + \boldsymbol{\Sigma}_p - 2\left(\boldsymbol{\Sigma}_\pi^{1/2}\boldsymbol{\Sigma}_p\boldsymbol{\Sigma}_\pi^{1/2}\right)^{1/2}\right]$$

## Drawbacks

- Depends on the pretrained classification network.
- Uses the normality assumption.
- May not correlate with human evaluation.

| Model | Model-A | Model-B |
|---|---|---|
| FID | 21.40 | 18.42 |
| $\mathrm{FID}_\infty$ | 20.16 | 17.19 |
| Human rater preference | 92.5% | 6.9% |

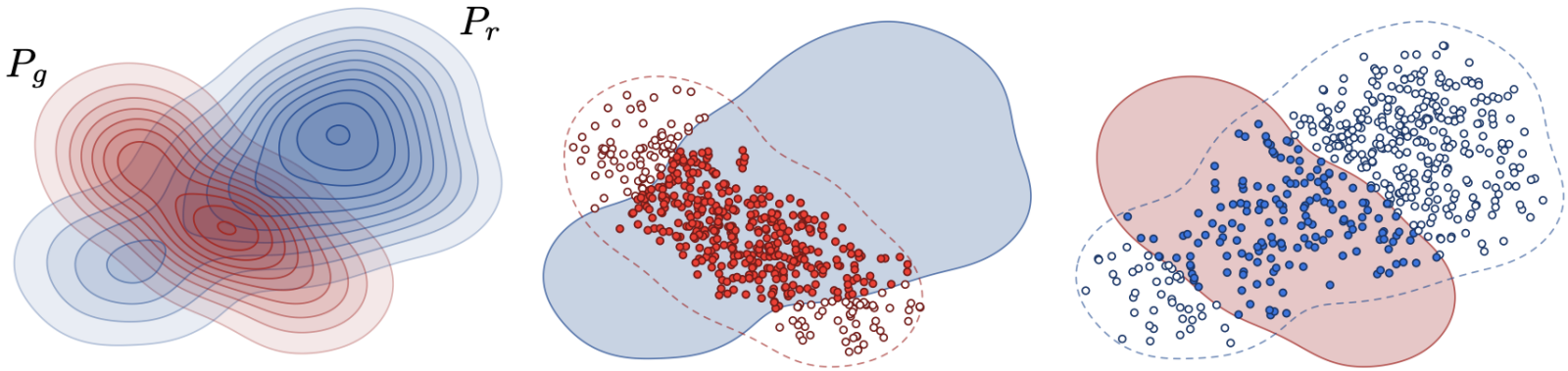
*Jayasumana S. et al. Rethinking FID: Towards a Better Evaluation Metric for Image Generation, 2024*

# Outline

# Precision-Recall

## Desirable Properties for Generated Samples

- **Sharpness**: generated samples should possess high visual quality.
- **Diversity**: their variation should match that in the training data.




(a) Example distributions (b) Precision (c) Recall

- **Precision** denotes the fraction of generated images that look realistic.
- **Recall** measures how well the generator covers the training data manifold.

*Kynkäänniemi T. et al. Improved precision and recall metric for assessing generative models, 2019*

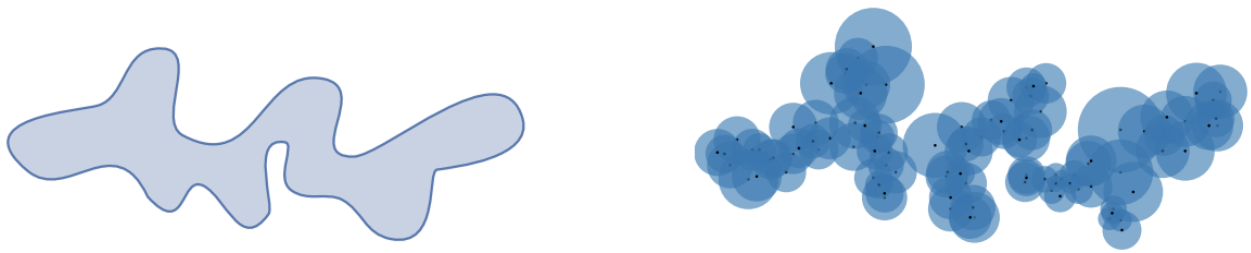
# Precision-Recall

- $\mathcal{S}_\pi = \{\mathbf{x}_i\}_{i=1}^n \sim \pi(\mathbf{x})$ – real samples;
- $\mathcal{S}_p = \{\mathbf{x}_i\}_{i=1}^n \sim p(\mathbf{x}|\theta)$ – generated samples.

Define a binary function:

$$\mathbb{I}(\mathbf{x}, \mathcal{S}) = \begin{cases} 1, & \text{if } \exists\ \mathbf{x}' \in \mathcal{S} : \|\mathbf{x} - \mathbf{x}'\|_2 \leq \|\mathbf{x}' - \mathrm{NN}_k(\mathbf{x}', \mathcal{S})\|_2; \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathrm{Precision}(\mathcal{S}_\pi, \mathcal{S}_p) = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{S}_p} \mathbb{I}(\mathbf{x}, \mathcal{S}_\pi); \quad \mathrm{Recall}(\mathcal{S}_\pi, \mathcal{S}_p) = \frac{1}{n} \sum_{\mathbf{x} \in \mathcal{S}_\pi} \mathbb{I}(\mathbf{x}, \mathcal{S}_p).$$

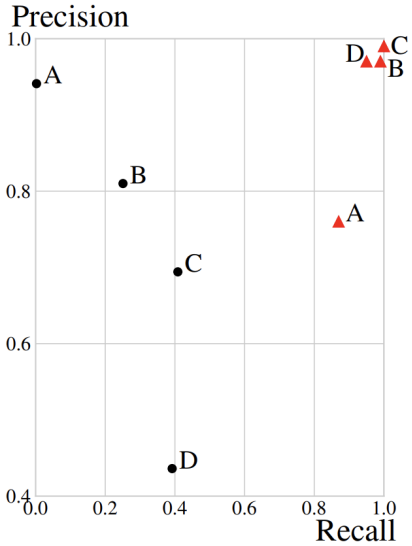


(a) True manifold (b) Approx. manifold

Embed the samples using a pretrained network (as in FID).

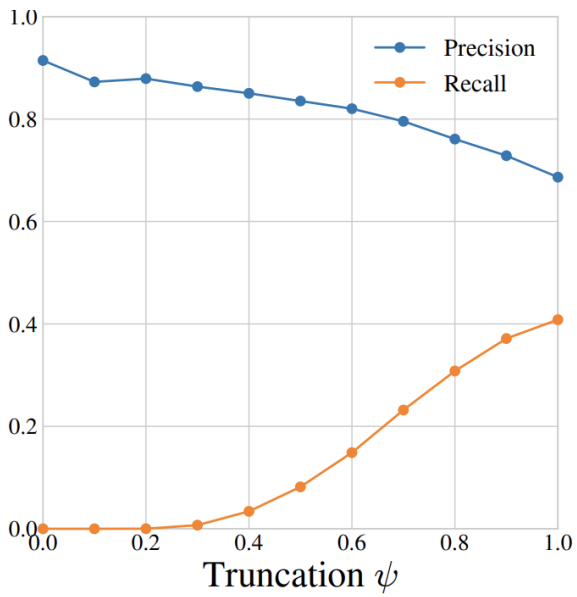
*Kynkäänniemi T. et al. Improved precision and recall metric for assessing generative models, 2019*

# Precision-Recall




**A** (FID = 91.7) **B** (FID = 16.9) **C** (FID = 4.5) **D** (FID = 16.7)

$\psi = 0.0$ $\psi = 0.3$ $\psi = 0.7$ $\psi = 1.0$



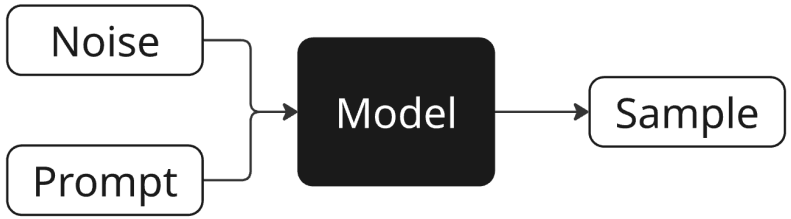

*Kynkäänniemi T. et al. Improved precision and recall metric for assessing generative models, 2019*

# Outline

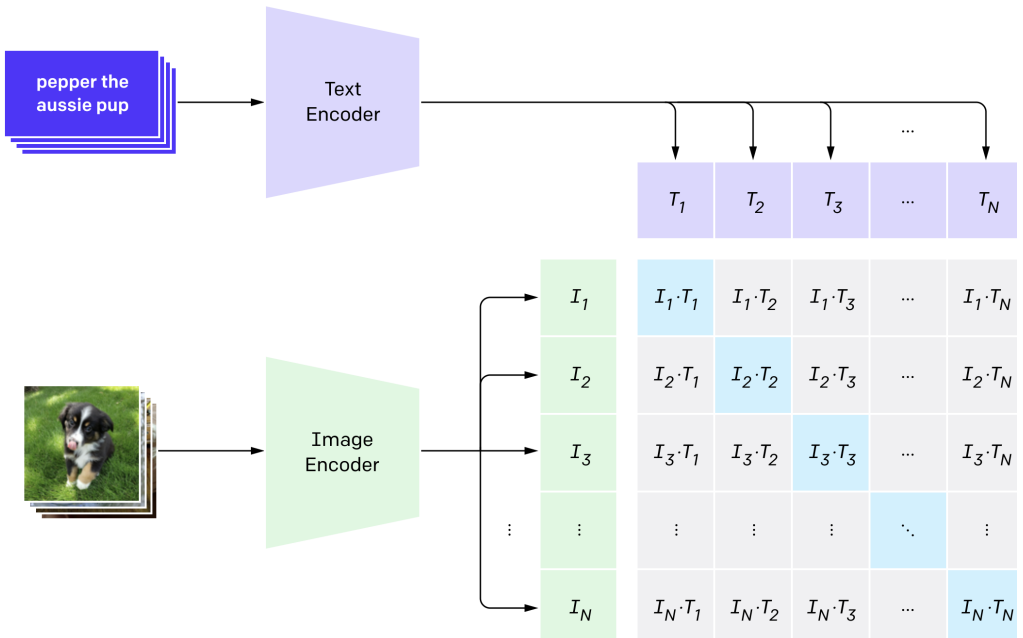
# CLIP Score

## Unconditional Model


## Conditional Model




We need a way to measure not only the quality of the generated image, but also how well it's aligned with the prompt.




---

*Radford A. et al. Learning transferable visual models from natural language supervision, 2021*

# Outline

# Human Evaluation

- No automated metric is perfect.
- The best way to evaluate generative models is by human assessment.
- It's important to assess various properties.

| Аспект | Yandex ART 2.0 | Mj 6.1 | Mj 6 | Ideogram | Recraft | Google Imagen3 | Dall-E 3 | FLUX | SBER Kandi3.1 |
|---|---|---|---|---|---|---|---|---|---|
| Релевантность | 0,59 | 0,58 | 0,63 | **0,45** | **0,51** | **0,50** | **0,50** | **0,54** | 0,75 |
| Эстетика | **0,49** | 0,55 | 0,55 | **0,51** | **0,51** | 0,61 | 0,61 | **0,54** | 0,59 |
| Комплексность | **0,44** | 0,73 | 0,70 | 0,68 | 0,76 | 0,75 | 0,75 | 0,71 | 0,74 |
| Дефектность | 0,69 | 0,57 | 0,68 | 0,55 | 0,59 | 0,63 | 0,63 | **0,50** | 0,75 |
| Предпочтение | 0,66 | 0,60 | 0,69 | **0,49** | **0,54** | 0,63 | 0,63 | **0,51** | 0,84 |

*YandexART 2.5, 2025*

# Outline

# Energy-Based Models

## Unnormalized Density

$$p(\mathbf{x}|\boldsymbol{\theta}) = \frac{\hat{p}(\mathbf{x}|\boldsymbol{\theta})}{Z_{\boldsymbol{\theta}}}, \quad \text{where } Z_{\boldsymbol{\theta}} = \int \hat{p}(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x}$$

- $\hat{p}(\mathbf{x}|\boldsymbol{\theta})$ can be any non-negative function.
- If we reparameterize as $\hat{p}(\mathbf{x}|\boldsymbol{\theta}) = \exp(-f_{\boldsymbol{\theta}}(\mathbf{x}))$, we eliminate the non-negativity constraint.

## Unnormalized Density

The gradient of the normalized log-density equals that of the unnormalized log-density:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta}) = \nabla_{\mathbf{x}} \log \hat{p}(\mathbf{x}|\boldsymbol{\theta}) - \nabla_{\mathbf{x}} \log Z_{\boldsymbol{\theta}} = \nabla_{\mathbf{x}} \log \hat{p}(\mathbf{x}|\boldsymbol{\theta})$$

- Suppose we already have this density (normalized or not) $p(\mathbf{x}|\boldsymbol{\theta})$.
- How can we sample from the model?

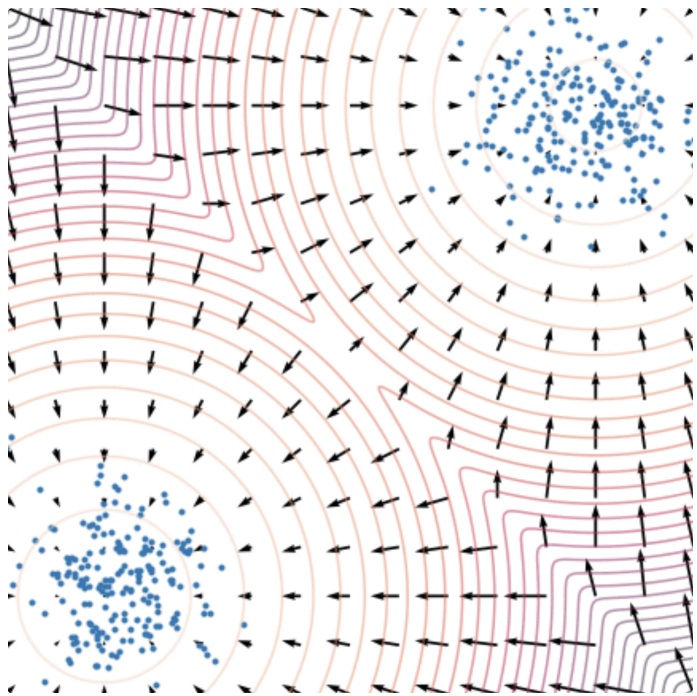
# Langevin Dynamics

## Theorem (Informal)

Let $\mathbf{x}_0$ be a random vector. Under mild regularity conditions, samples from the following dynamics will eventually follow $p(\mathbf{x}|\boldsymbol{\theta})$ (for sufficiently small $\eta$ and large $I$):

$$\mathbf{x}_{I+1} = \mathbf{x}_I + \frac{\eta}{2} \cdot \nabla_{\mathbf{x}_I} \log p(\mathbf{x}_I|\boldsymbol{\theta}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_I, \quad \boldsymbol{\epsilon}_I \sim \mathcal{N}(0, \mathbf{I}).$$

- What if $\boldsymbol{\epsilon}_I = \mathbf{0}$?
- The density $p(\mathbf{x}|\boldsymbol{\theta})$ is the **stationary** distribution of the Markov chain.
- The gradient is taken with respect to $\mathbf{x}$, not $\boldsymbol{\theta}$.
- $\nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta})$ defines a vector field.



*Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021*

# Outline

# Score Matching

## Score Function

$$\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta})$$

## Langevin Dynamics

If we know the score function $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta})$, we can generate samples from the model using Langevin dynamics:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \frac{\eta}{2} \cdot \nabla_{\mathbf{x}_l} \log p(\mathbf{x}_l|\boldsymbol{\theta}) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_l = \mathbf{x}_l + \frac{\eta}{2} \cdot \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_l) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_l.$$

## Fisher Divergence

$$D_F(\pi, p) = \frac{1}{2}\mathbb{E}_\pi \left\| \nabla_{\mathbf{x}} \log p(\mathbf{x}|\boldsymbol{\theta}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \right\|_2^2 =$$

$$= \frac{1}{2}\mathbb{E}_\pi \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

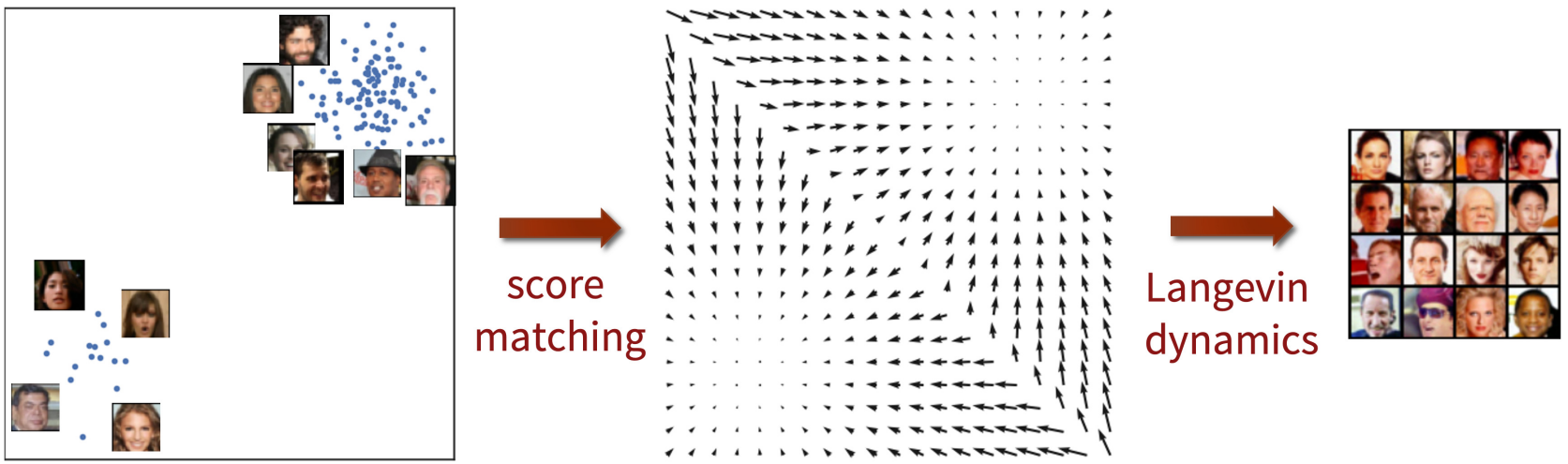
---

*Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021*

# Score Matching

## Fisher Divergence

$$D_F(\pi, p) = \frac{1}{2}\mathbb{E}_\pi \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \right\|_2^2 \to \min_{\boldsymbol{\theta}}$$




Data samples

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

Scores

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

New samples

**Problem:** We don't know $\nabla_{\mathbf{x}} \log \pi(\mathbf{x})$.

---

*Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021*

# Outline

# Denoising Score Matching

Let us perturb the original data $\mathbf{x} \sim \pi(\mathbf{x})$ with Gaussian noise:

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

$$q(\mathbf{x}_\sigma) = \int q(\mathbf{x}_\sigma|\mathbf{x})\pi(\mathbf{x})d\mathbf{x}.$$

## Assumption

The solution to

$$\frac{1}{2}\mathbb{E}_{q(\mathbf{x}_\sigma)}\left\|\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)\right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

satisfies $\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) \approx \mathbf{s}_{\boldsymbol{\theta},0}(\mathbf{x}_0) = \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})$ if $\sigma$ is sufficiently small.

- The score function of the noised data nearly matches the score function of the original data.
- The score function $\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)$ is parameterized by $\sigma$.
- **Note:** We don't know $q(\mathbf{x}_\sigma)$, just as we don't know $\pi(\mathbf{x})$.

---

*Vincent P. A Connection Between Score Matching and Denoising Autoencoders, 2010*

# Denoising Score Matching

## Theorem

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

## Gradient of the Noise Kernel

$$\mathbf{x}_\sigma = \mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}, \quad q(\mathbf{x}_\sigma|\mathbf{x}) = \mathcal{N}(\mathbf{x}, \sigma^2 \cdot \mathbf{I})$$

$$\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x}) = -\frac{\mathbf{x}_\sigma - \mathbf{x}}{\sigma^2} = -\frac{\boldsymbol{\epsilon}}{\sigma}$$

- The right-hand side doesn't require computing $\nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma)$ or even $\nabla_{\mathbf{x}_\sigma} \log \pi(\mathbf{x}_\sigma)$.
- $\mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma)$ is trained to **denoise** the noised samples $\mathbf{x}_\sigma$.

---

*Vincent P. A Connection Between Score Matching and Denoising Autoencoders, 2010*

# Denoising Score Matching

Initial objective:

$$\mathbb{E}_{\pi(\mathbf{x})} \left\| \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log \pi(\mathbf{x}) \right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

Noised objective:

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

This is equivalent to a denoising task:

$$\mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{q(\mathbf{x}_\sigma|\mathbf{x})} \left\| \mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_\sigma) - \nabla_{\mathbf{x}_\sigma} \log q(\mathbf{x}_\sigma|\mathbf{x}) \right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

$$\mathbb{E}_{\pi(\mathbf{x})} \mathbb{E}_{\mathcal{N}(0,\mathbf{I})} \left\| \mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x} + \sigma \cdot \boldsymbol{\epsilon}) + \frac{\boldsymbol{\epsilon}}{\sigma} \right\|_2^2 \to \min_{\boldsymbol{\theta}}$$

## Langevin Dynamics

$$\mathbf{x}_{l+1} = \mathbf{x}_l + \frac{\eta}{2} \cdot \mathbf{s}_{\boldsymbol{\theta},\sigma}(\mathbf{x}_l) + \sqrt{\eta} \cdot \boldsymbol{\epsilon}_l, \quad \boldsymbol{\epsilon}_l \sim \mathcal{N}(0, \mathbf{I}).$$

---

*Song Y. Generative Modeling by Estimating Gradients of the Data Distribution, blog post, 2021*

# Summary

- Frechet Inception Distance is the most popular metric for evaluating implicit generative models.
- Precision-recall allows for choosing a model that balances sample quality and diversity.
- The CLIP score is widely used to measure text-to-image alignment.
- The gold standard for evaluating generated image quality is human assessment.
- Langevin dynamics enable sampling from generative models using gradients of the log-likelihood.
- Score matching proposes minimizing Fisher divergence to estimate the score function.
- Denoising score matching optimizes Fisher divergence on noisy data, making it estimable with samples.