



# I. Introduction

Young-geun Kim

Department of Statistics and Probability

STT 997 (SS 2025)

# Outline

- 1 GENERATIVE MODEL
- 2 STATISTICAL REASONING WITH GENERATIVE MODEL
- 3 CHRONICLE OF DEEP GENERATIVE MODELS: A FOCUS ON STATISTICAL DISTANCES

# Generative Model

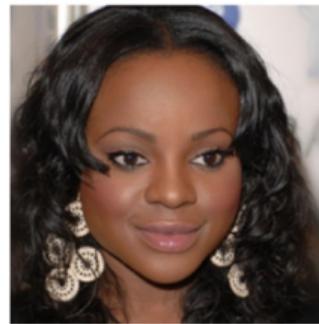
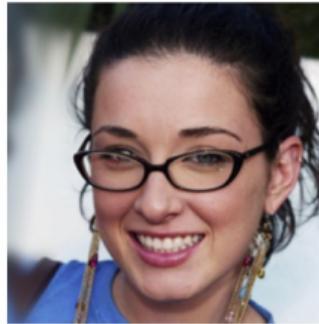


- The advent of high-dimensional and large-scale data has increased the richness of information within the data, even without human-annotated labels, emphasizing the need for advanced statistical methods.
- Recent advancements have enabled the learning and generation of complex data. These models are now commonly referred to as *Generative AI* or *Generative Models*.

A screenshot of a generated video by OpenAI Sora. More examples can be found at <https://openai.com/sora/>.

# Generative Model

1,024



1,024

- Let  $\vec{X} = (X_1, \dots, X_p)^T$  be observations following  $p(\vec{x})$ , e.g.,  
 $p = 1,024 \times 1,024 \times 3 \approx 3.15M$  for color images shown above.
- Generative models learn the joint distribution of all observed variables,  $\vec{X}$ :

$$p(\vec{x}). \quad (1)$$

With  $p(\vec{x})$ , we can augment data, investigate cluster structures, and detect anomalies.

Images are from CelebA dataset (Liu et al., 2015).

# Generative Model

- In this course, the term ‘Generative Model’ refers to statistical models that learn the distribution of observations either without human-annotated labels or with auxiliary information.<sup>1</sup>
- Deep generative models have shown remarkable performance as:
  - ① simulators by generating realistic data,
  - ② dimension reduction tools by extracting low-dimensional representations,
  - ③ inference tools by translating observations to other domains.

---

<sup>1</sup>For example, demographic information in medical data or timestamps in temporal data.

# Application: Image Generation



- After training, generative models can synthesize realistic data without prior information such as age and race in face generation.
- These models can be used to augment data, generate privacy-free samples, and enhance virtual reality experiences.

---

The generated image is from <http://thispersondoesnotexist.com/>

# Application: Text-to-Image Generation

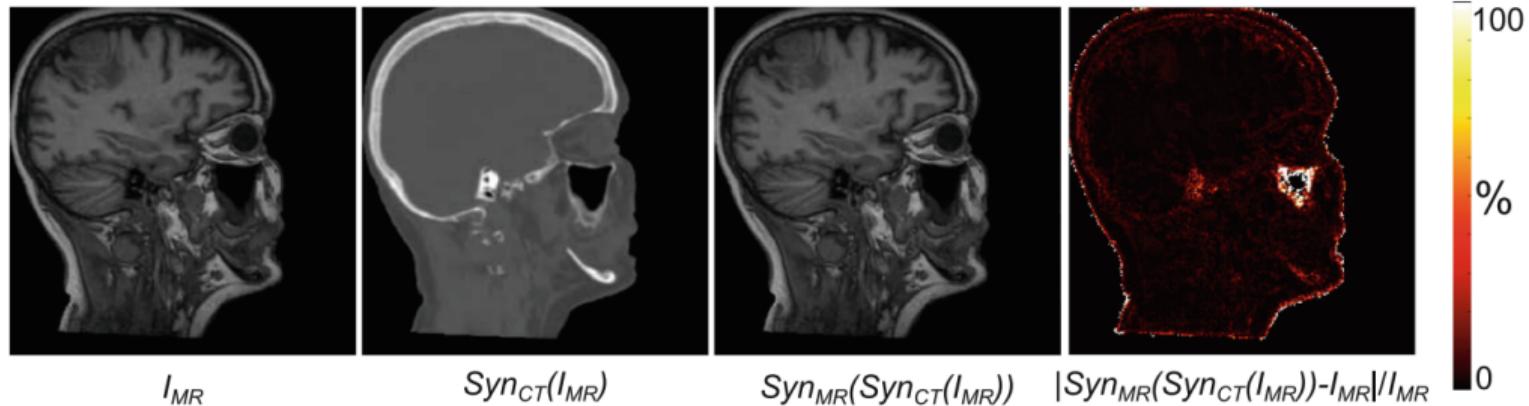


- Images are generated by DALL-E-2 using  
*"There is a clean desk in the middle. Outside the window,  
a whale shark is swimming in the dark night sky above Manhattan."*
- Generated images reflect semantic information in text descriptions.

---

Images are generated with DALL-E-2 (Ramesh et al., 2022).

# Application: Cross Modality Transfer

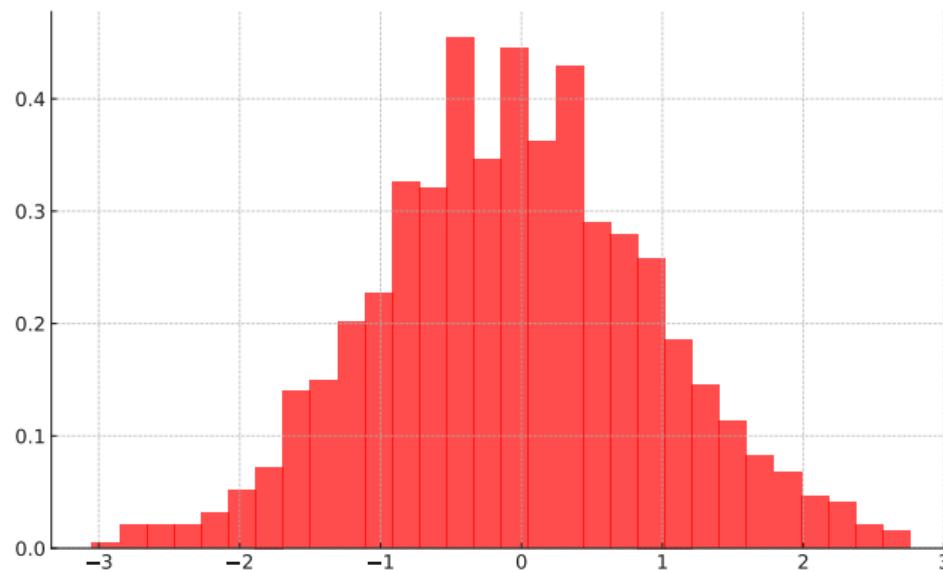


- Generative models are also useful in imputing missing modalities, e.g.,  $MR \rightarrow CT$ , or enhancing data resolution.

Images are from Wolterink et al. (2017).

# How to Learn Distributions: Toy Example

- Let's begin with a basic example. Assume we observed  $n$  univariate samples  $x_1, \dots, x_n$  having the following histogram:



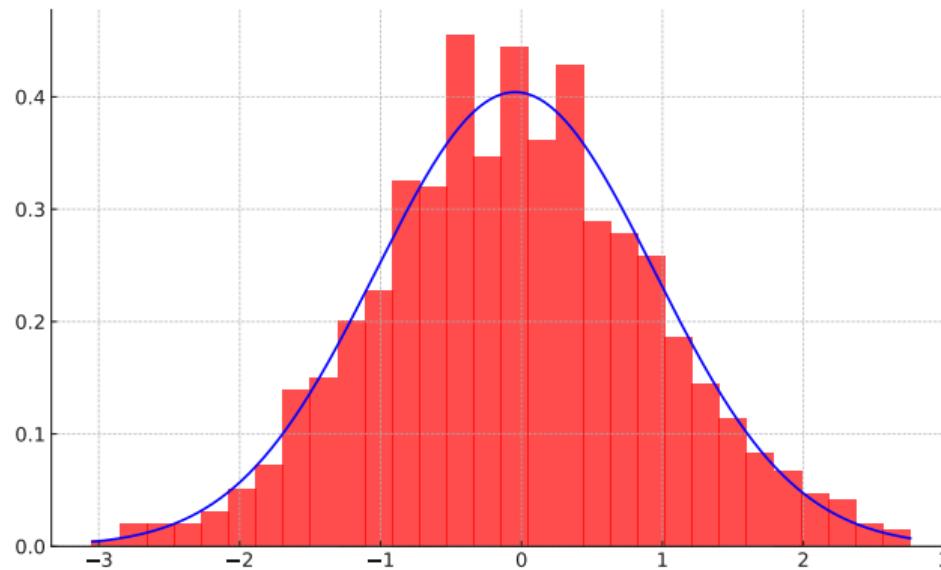
# How to Learn Distributions: Toy Example

- How to learn the distribution where  $x_i$  comes from?
- One way is to model it as Gaussian and find the optimal mean and std parameters.
- $p_\theta(x) := (2\pi\sigma^2)^{-1/2} \exp\left((x - \mu)^2/(2\sigma^2)\right)$  where  $\theta = (\mu, \sigma^2)^T \in \mathbb{R} \times \mathbb{R}^+$ .

**Q:** What are good evaluation criteria for optimality? How do we determine which parameter values are superior?

# How to Learn Distributions: Toy Example

- $I_n(\theta) := \sum_{i=1}^n \log p_\theta(x_i) = -(n/2) \log \sigma^2 - \sum_{i=1}^n (x_i - \mu)^2 / (2\sigma^2) - (n/2) \log 2\pi$
- $\arg \max_{\theta} I_n(\theta) = (\hat{\mu}_n, \hat{\sigma}_n^2)^T = \left( \bar{x}, n^{-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^T$



# How to Learn Distributions: Toy Example

- There is no closed-form expression for the MLEs when addressing complex data and models.
- We usually run iterative algorithms to approximate optimal parameters.

# Challenges in Generative Models



- **High-dimensional Data:** For data like 4K-resolution color images, the dimensionality is  $3,840 \times 2,160 \times 3 (\approx 24M)$ .
- **Complex Structure:** Image, video, audio, and language data exhibit complex structures.

# Challenges in Generative Models

Red Channel



Green Channel

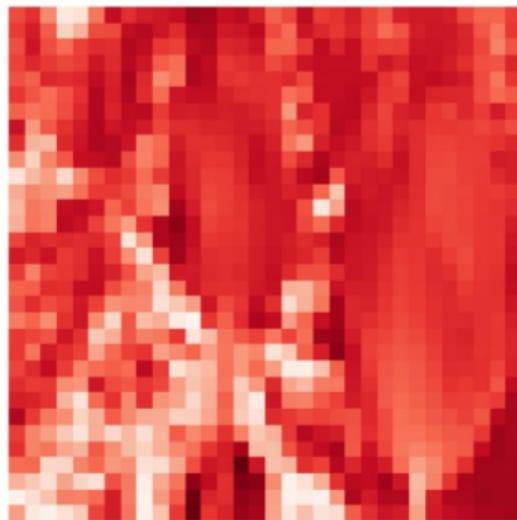


Blue Channel

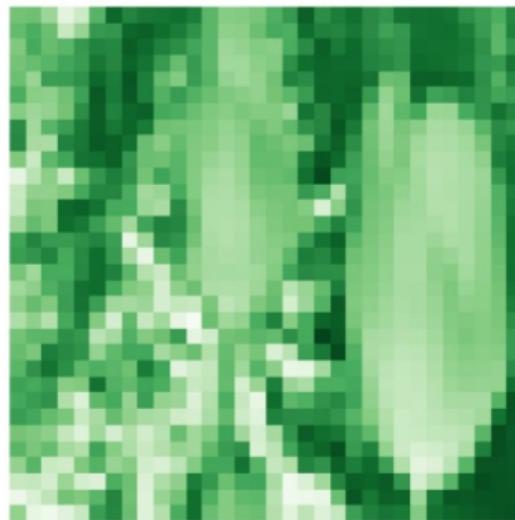


# Challenges in Generative Models

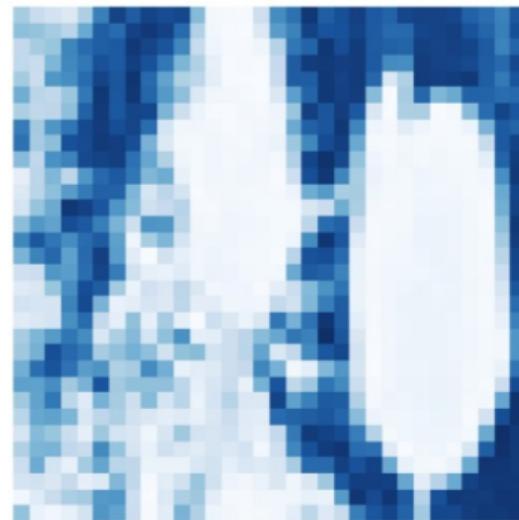
Red Channel



Green Channel



Blue Channel



# Challenges in Generative Models

## Red Channel

## Green Channel

Blue Channel

# Challenges in Generative Models



- How to model the distribution of high-dimensional data efficiently?
- How to evaluate the generated data and train the model distribution?

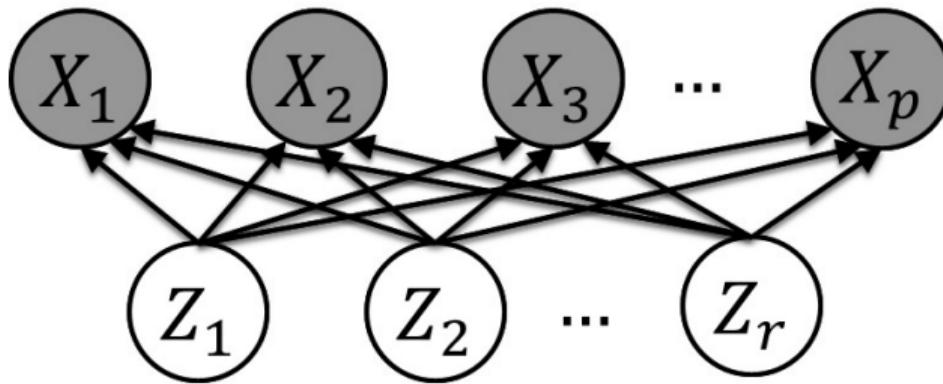
---

Example of generated images, edited from Li et al. (2024)

# Outline

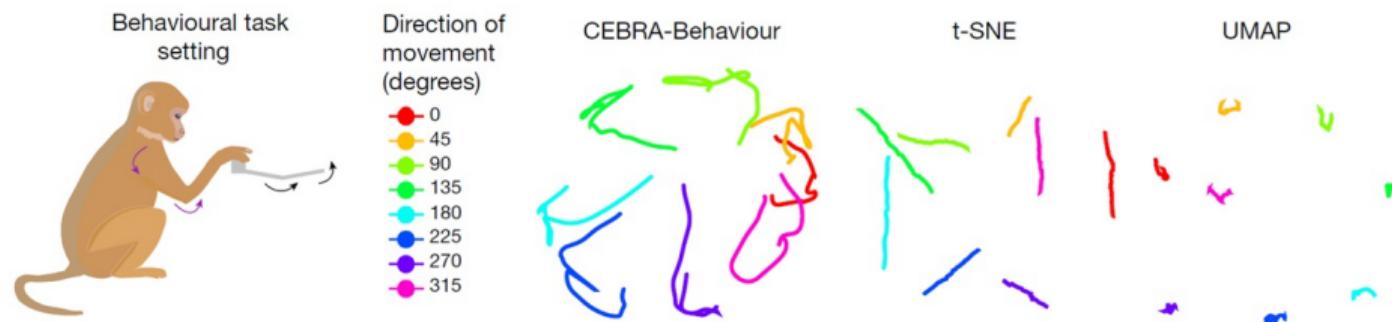
- 1 GENERATIVE MODEL
- 2 STATISTICAL REASONING WITH GENERATIVE MODEL
- 3 CHRONICLE OF DEEP GENERATIVE MODELS: A FOCUS ON STATISTICAL DISTANCES

# Latent Variable Model



- Latent variable models are popular approaches in generative models.
- For example, we can mix latent independent components to generate observations:
  - ① Latent Factors:  $\vec{Z} \sim \prod_{j=1}^r p(z_j)$
  - ② Observations:  $\vec{X} | \vec{Z} = \vec{z} \sim \prod_{j=1}^p p(x_j | \vec{Z} = \vec{z})$

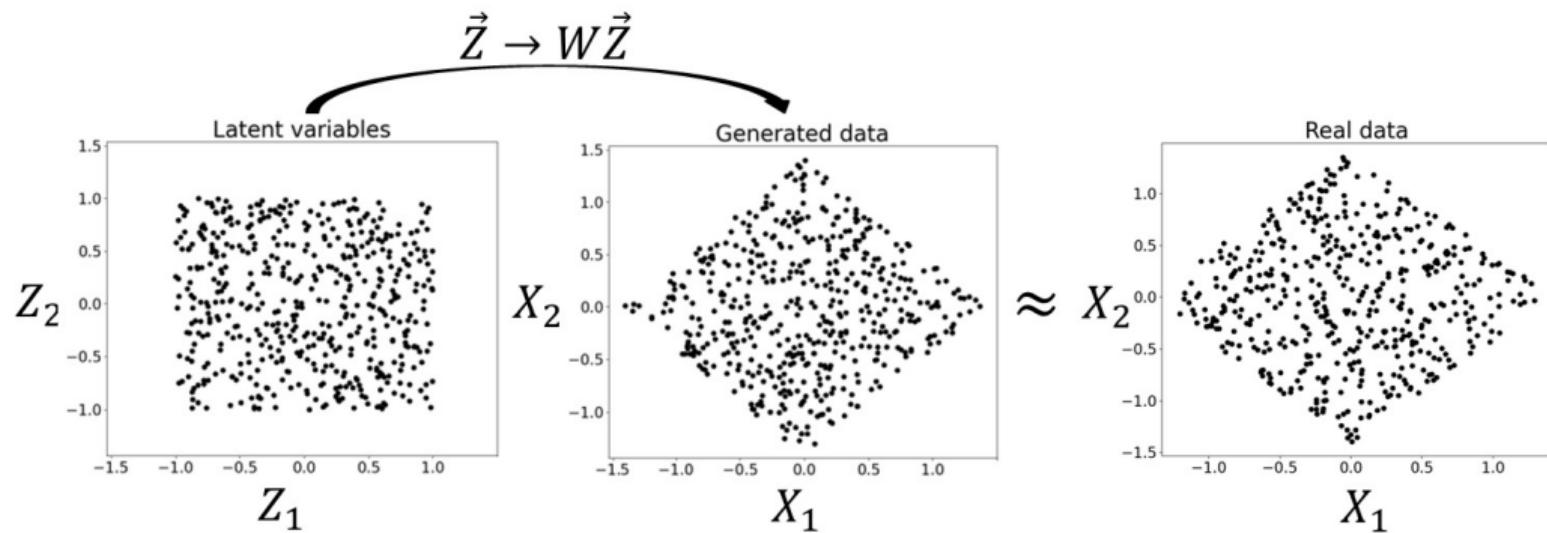
# Latent Variable Model



- Generative models can interpret the semantic meanings of factors  $\vec{Z}$  and how they affect observations  $\vec{X}$ .
- The figure shows an analysis of electrophysiology recordings from the somatosensory cortex of monkeys.
- Compared to other (nonlinear) embedding techniques, deep generative models provide more explainable results.

Images are edited from Schneider et al. (2023).

# Example: Independent Component Analysis



# Example: Independent Component Analysis

# Statistical Reasoning with Generative Model

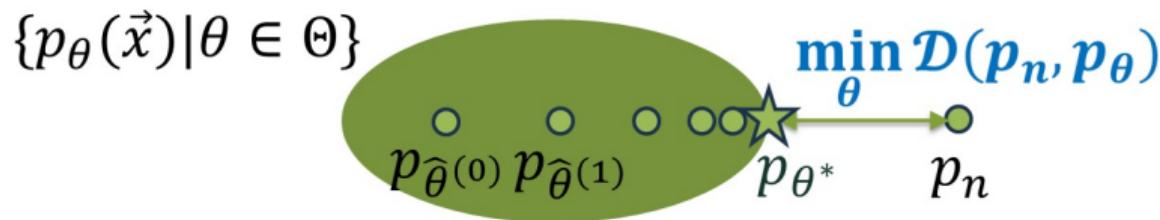
$$\{p_{\theta}(\vec{x}) | \theta \in \Theta\}$$



$p_n$

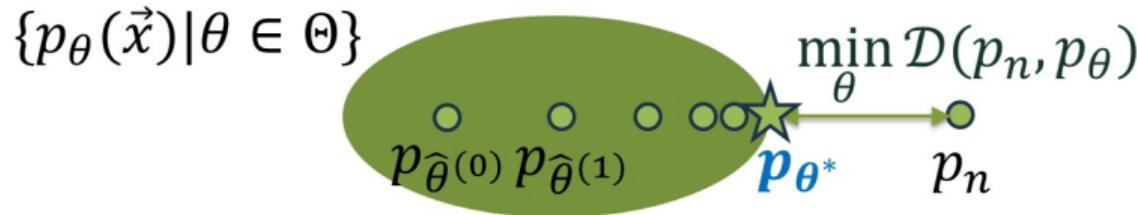
- Key elements to learn generative models include
  - ➊ **Model Class:** A flexible class of  $p_{\theta}(\vec{x})$  reflecting prior knowledge.

# Statistical Reasoning with Generative Model



- Key elements to learn generative models include
  - ➊ **Model Class:** A flexible class of  $p_\theta(\vec{x})$  reflecting prior knowledge.
  - ➋ **Statistical Distance:** Measure differences between  $p_n$  and  $p_\theta$ .

# Statistical Reasoning with Generative Model



- Key elements to learn generative models include
  - ① **Model Class:** A flexible class of  $p_\theta(\vec{x})$  reflecting prior knowledge.
  - ② **Statistical Distance:** Measure differences between  $p_n$  and  $p_\theta$ .
  - ③ **Representation:** Infer knowledges with  $p_{\theta^*}(\vec{z}|\vec{x})$ .

# Outline

- 1 GENERATIVE MODEL
- 2 STATISTICAL REASONING WITH GENERATIVE MODEL
- 3 CHRONICLE OF DEEP GENERATIVE MODELS: A FOCUS ON STATISTICAL DISTANCES

# Examples of Statistical Distances

- Deep generative models learn  $p_n$  by minimizing  $\mathcal{D}(p_n, p_\theta)$ , where  $\mathcal{D}$  denotes a statistical distance.
- The effectiveness of  $\mathcal{D}$  varies depending on the type of data and the specific algorithm implementation. Each  $\mathcal{D}$  necessitates different model classes and corresponding loss functions.
- Popular choices of  $\mathcal{D}$  include:
  - ①  $f$ -divergence
  - ② Integral Probability Metric
  - ③ Wasserstein Distance<sup>2</sup>
  - ④ Fisher Divergence

<sup>2</sup>Named after “Leonid Vaserštejn”, though “Wasserstein” is more commonly used in English publications.

# Examples of Statistical Distances

## 1. *f*-divergence:

- The *f*-divergences (Rényi, 1961) are expectations of density ratios mapped by convex functions  $f : \mathbb{R}^+ \rightarrow \mathbb{R}$ , satisfying  $f(1) = 0$ . They can be expressed as:

$$\mathcal{D}_f(p||q) := \int f\left(\frac{p(\vec{x})}{q(\vec{x})}\right) q(\vec{x}) d\vec{x} \quad (2)$$

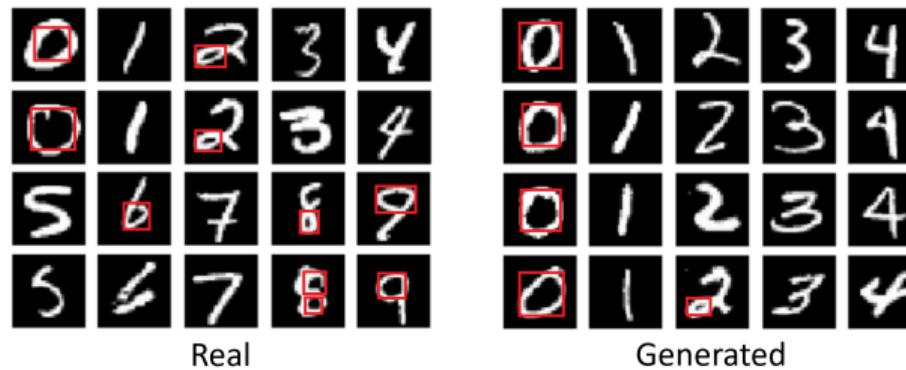
- The  $\text{KL}(p_n||p_\theta)$  equals  $\mathcal{D}_f(p_n||p_\theta)$  when  $f(u) = u \log u$ .

**Proof:**

$$\mathcal{D}_f(p_n||p_\theta) = \int \left(p_n(\vec{x})/p_\theta(\vec{x})\right) \log \left(p_n(\vec{x})/p_\theta(\vec{x})\right) p_\theta(\vec{x}) d\vec{x} = \int \log \left(p_n(\vec{x})/p_\theta(\vec{x})\right) p_n(\vec{x}) d\vec{x}.$$

- Thus, all likelihood maximization methods target minimizing this specific *f*-divergence.

# Examples of Statistical Distances



2. **Integral Probability Metric:** The integral probability metrics (IPMs, Müller, 1997) between distributions are the largest difference between their summary statistics.
- For example, when we use the number of circles in images as summary statistics, the difference is  $0.5$  (real) –  $0.25$  (generated) =  $0.25$ .

---

Image source: MNIST (Deng, 2012)

# Examples of Statistical Distances

## 2. Integral Probability Metric:

- We can consider many summary statistics together to precisely compare distributions.
- The IPMs can be expressed as:

$$\gamma_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) := \sup_{f \in \mathcal{F}} \left| \int f(\vec{x}) d\mathbb{P}(\vec{x}) - \int f(\vec{x}) d\mathbb{Q}(\vec{x}) \right| \quad (3)$$

where  $\mathcal{F}$  is a class of real-valued functions, and  $\mathbb{P}$  and  $\mathbb{Q}$  are probability measures.

- Examples of  $\mathcal{F}$  include the class of all 1-Lipschitz continuous functions, and functions from reproducing kernel Hilbert space (RKHS).

# Examples of Statistical Distances

	0	1	2		0	1	2	
0	1/9	1/9	1/9		0	0.10	0.30	0.40
1	1/9	1/9	1/9		1	0.40	0.05	0.25
2	1/9	1/9	1/9		2	0.25	0.30	0.10
Joint Distribution				Transportation Cost				

3. **Wasserstein Distance:** Wasserstein distance (Monge, 1781; Kantorovich, 1960)

represents the minimum expected transportation cost between two distributions.

- For example, consider the above joint distribution. The transportation cost is calculated as:

$$\begin{aligned}
 & (1/3) \times (0.10 \times (1/3) + 0.30 \times (1/3) + 0.40 \times (1/3)) \\
 & + (1/3) \times (0.40 \times (1/3) + 0.05 \times (1/3) + 0.25 \times (1/3)) \\
 & + (1/3) \times (0.25 \times (1/3) + 0.30 \times (1/3) + 0.10 \times (1/3)) = 0.24.
 \end{aligned}$$

# Examples of Statistical Distances

	0	1	2		0	1	2	
0	1/3	0	0		0.10	0.30	0.40	
1	0	1/3	0		0.40	0.05	0.25	
2	0	0	1/3		0.25	0.30	0.10	
Joint Distribution			Transportation Cost					

### 3. Wasserstein Distance

- We can consider another joint distribution. The transportation cost is

$$\begin{aligned}
 & (1/3)(0.10(1) + 0.30(0) + 0.40(0)) \\
 & + (1/3)(0.40(0) + 0.05(1) + 0.25(0)) \\
 & + (1/3)(0.25(0) + 0.30(0) + 0.10(1)) = 0.08.
 \end{aligned}$$

# Examples of Statistical Distances

## 3. Wasserstein Distance

- The  $p$ -Wasserstein distance can be expressed as

$$W_p(\mathbb{P}, \mathbb{Q}; d) := \left( \inf_{\pi \in \Pi(\mathbb{P}, \mathbb{Q})} \int d^p(\vec{x}, \vec{x}') d\pi(\vec{x}, \vec{x}') \right)^{1/p} \quad (4)$$

where  $p \in [1, \infty)$  and  $\Pi(\mathbb{P}, \mathbb{Q})$  is the set of all joint distributions whose marginals are  $\mathbb{P}$  and  $\mathbb{Q}$ .

# Examples of Statistical Distances

- 4. Fisher Divergence:** Fisher divergence (Johnson, 2004; Hyvärinen, 2005) is the expected difference between the (Stein) scores (Liu et al., 2016)<sup>3</sup> of two distributions. It can be expressed as:

$$\text{FD}(p \parallel q) = \int \|\nabla_{\vec{x}} \log p(\vec{x}) - \nabla_{\vec{x}} \log q(\vec{x})\|^2 p(\vec{x}) d\vec{x}. \quad (5)$$

- It is zero if and only if  $p = q$ .

**Proof:**

$$\nabla_{\vec{x}} \log p(\vec{x}) = \nabla_{\vec{x}} \log q(\vec{x}) \implies p(\vec{x}) = Cq(\vec{x}) \quad (6)$$

and  $C = 1$  because  $\int p(\vec{x}) d\vec{x} = \int q(\vec{x}) d\vec{x} = 1$ .

---

<sup>3</sup>The term ‘score’ here refers to the gradient w.r.t. realizations, which differs from usual terms in parametric family distributions.

## Examples of Statistical Distances

**4. Fisher Divergence:** This is a useful measure for learning energy-based models (Teh et al., 2003), such as Boltzmann distributions:

$$p_\theta(\vec{x}) = \frac{1}{C(\theta)} \exp(-E_\theta(\vec{x})) \quad (7)$$

where  $C(\theta) := \int \exp(-E_\theta(\vec{x})) d\vec{x}$ . In this case,

$$\nabla_{\vec{x}} \log p_\theta(\vec{x}) = -\nabla_{\vec{x}} E_\theta(\vec{x}) \quad (8)$$

holds, and the normalizing constant disappears.

# Chronicle of Deep Generative Model



$f$  divergence  
-based methods

- Variational Autoencoder (VAE, Kingma and Welling, 2014)
- Generative Adversarial Network (GAN, Goodfellow et al., 2014)
- $f$ -GAN (Nowozin et al., 2016)

# Chronicle of Deep Generative Model



- Generative Moment Matching Networks (Li et al., 2015)
- Maximum Mean Discrepancy GANs (Li et al., 2017)
- Sobolev GANs (Mroueh et al., 2017)

# Chronicle of Deep Generative Model



- Wasserstein GANs (Arjovsky et al., 2017)
- Wasserstein GAN with gradient penalty (Gulrajani et al., 2017)
- Wasserstein Autoencoders (Tolstikhin et al., 2018)

# Chronicle of Deep Generative Model



- Noise Conditional Score Networks (Song and Ermon, 2019)

# Advanced Topics

- Asymptotic efficiencies according to statistical distances, e.g., minimax convergence rates of IPM-based generative models (Uppal et al., 2019).
- Methods for data domains other than images, e.g., generative pre-trained transformers (GPTs, Radford, 2018) and their variations for language data.
- Advanced graphical models reflecting domain knowledge, e.g., DALL-E (Ramesh et al., 2021) for generating images from text descriptions.

# References I

- Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.
- Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.
- Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).
- Johnson, O. (2004). *Information theory and the central limit theorem*. World Scientific.

## References II

- Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management science*, 6(4):366–422.
- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Póczos, B. (2017). Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30.
- Li, T., Tian, Y., Li, H., Deng, M., and He, K. (2024). Autoregressive image generation without vector quantization. *arXiv preprint arXiv:2406.11838*.
- Li, Y., Swersky, K., and Zemel, R. (2015). Generative moment matching networks. In *International conference on machine learning*, pages 1718–1727. PMLR.
- Liu, Q., Lee, J., and Jordan, M. (2016). A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284. PMLR.

## References III

- Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.
- Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704.
- Mroueh, Y., Li, C.-L., Sercu, T., Raj, A., and Cheng, Y. (2017). Sobolev gan. *arXiv preprint arXiv:1711.04894*.
- Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in applied probability*, 29(2):429–443.
- Nowozin, S., Cseke, B., and Tomioka, R. (2016). f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279.
- Radford, A. (2018). Improving language understanding by generative pre-training.

## References IV

- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr.
- Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics*, volume 4, pages 547–562. University of California Press.
- Schneider, S., Lee, J. H., and Mathis, M. W. (2023). Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368.
- Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

# References V

- Teh, Y. W., Welling, M., Osindero, S., and Hinton, G. E. (2003). Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4(Dec):1235–1260.
- Tolstikhin, I., Bousquet, O., Gelly, S., and Schölkopf, B. (2018). Wasserstein auto-encoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Uppal, A., Singh, S., and Póczos, B. (2019). Nonparametric density estimation & convergence rates for gans under besov ipm losses. *Advances in neural information processing systems*, 32.
- Wolterink, J. M., Dinkla, A. M., Savenije, M. H., Seevinck, P. R., van den Berg, C. A., and Išgum, I. (2017). Deep mr to ct synthesis using unpaired data. In *International workshop on simulation and synthesis in medical imaging*, pages 14–23. Springer.