

Deep Generative Models

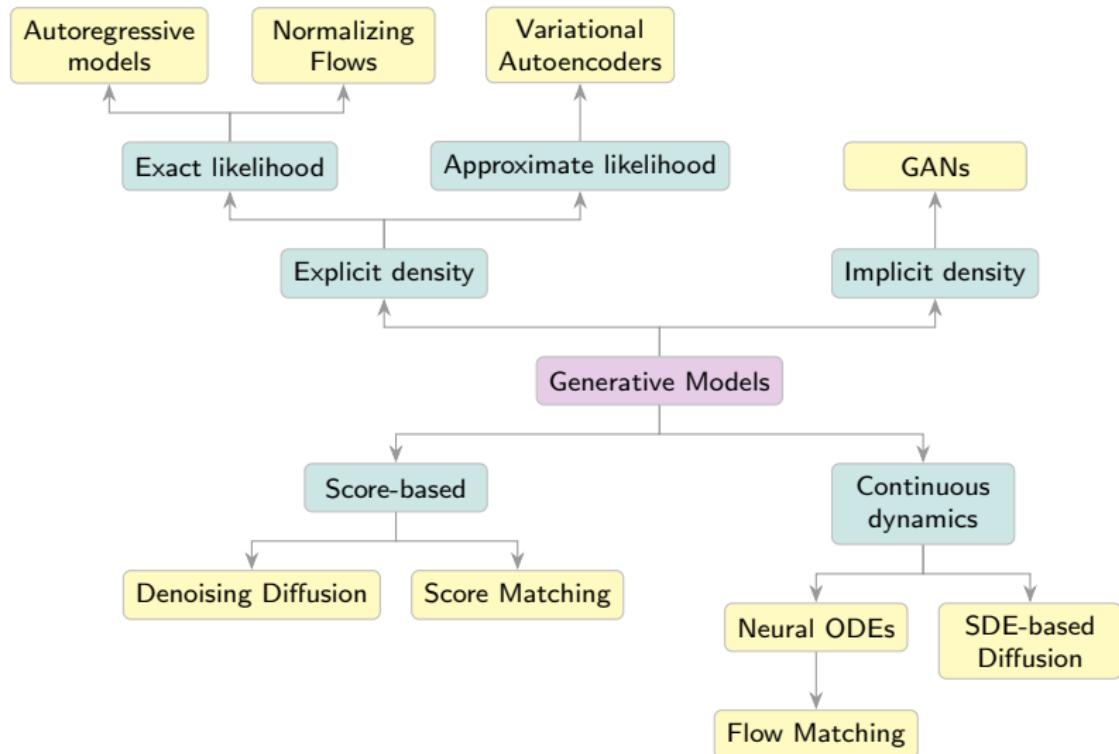
Lecture 1

Roman Isachenko



2026, Spring

Generative Models Taxonomy



Outline

1. Generative Models Overview
2. Course Tricks
3. Problem Statement
4. Divergence Minimization Framework
5. Autoregressive Modeling

Outline

1. Generative Models Overview
2. Course Tricks
3. Problem Statement
4. Divergence Minimization Framework
5. Autoregressive Modeling

VAE – The First Scalable Approach for Image Generation



DCGAN – The First Convolutional GAN for Image Generation



StyleGAN – High-Quality Face Generation



Karras T., Laine S., Aila T. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

Language Modeling at Scale

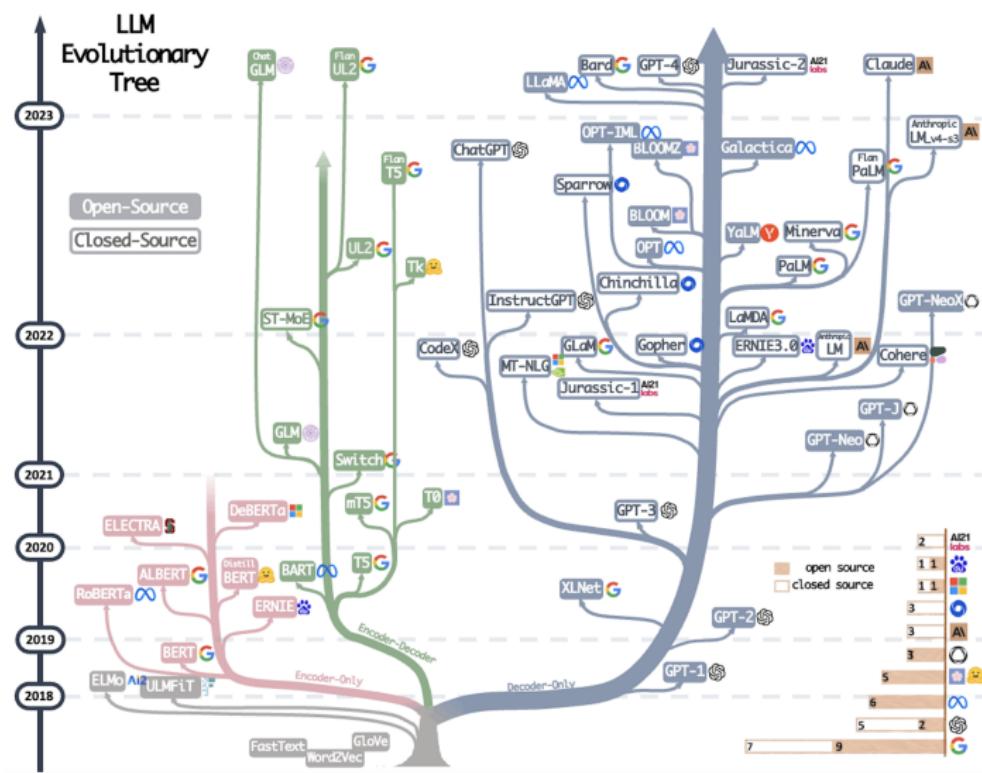
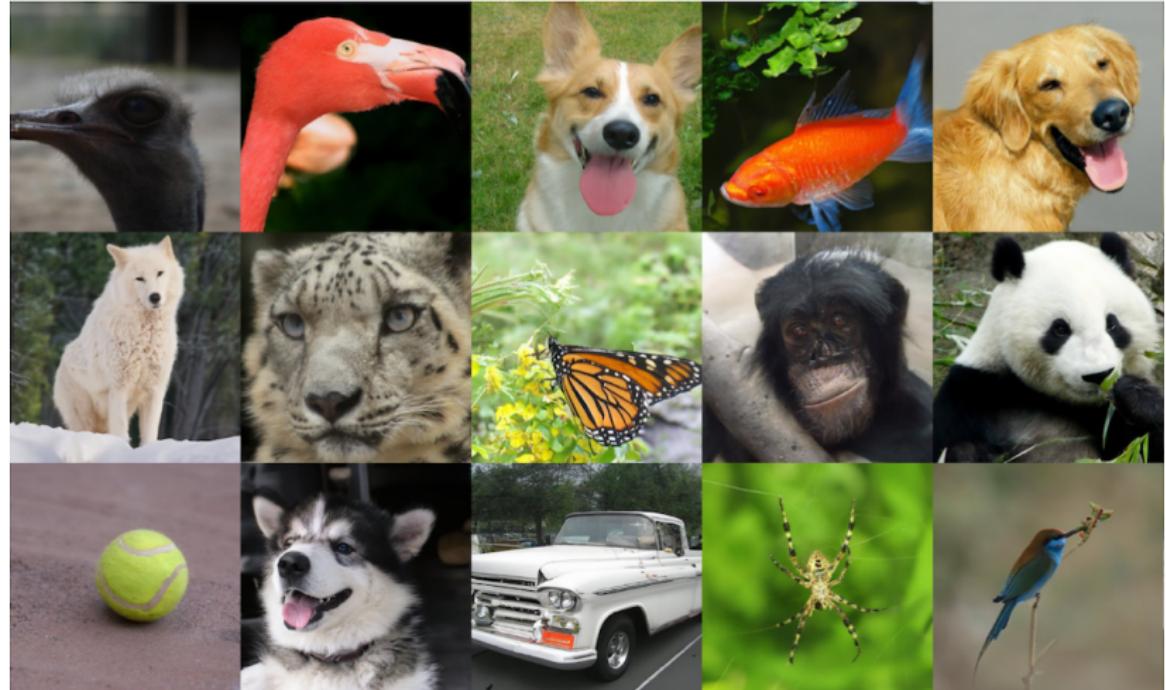


Image credit:

<https://blog.biocomm.ai/2023/05/14/open-source-proliferation-llm-evolutionary-tree/>

Denoising Diffusion Probabilistic Model



Midjourney – Impressive Text-to-Image Results



Image credit: <https://www.midjourney.com/explore>

Sora – Video Generation



Image credit: <https://openai.com/index/sora>

GPT4o Image Editing

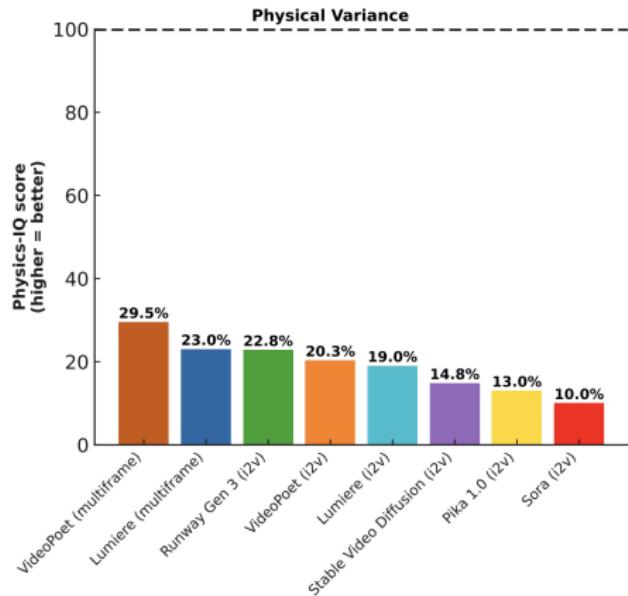
Prompt: Give this cat a detective hat and a monocle



Image credit: <https://openai.com/index/introducing-4o-image-generation/>

Open Problems in Generative Models

- ▶ Video generation
- ▶ 3D scene generation
- ▶ Understanding of physical processes
- ▶ Multimodal end-to-end models



Outline

1. Generative Models Overview
2. Course Tricks
3. Problem Statement
4. Divergence Minimization Framework
5. Autoregressive Modeling

Course Tricks I

Log-Derivative Trick

Given a differentiable function $p : \mathbb{R}^m \rightarrow \mathbb{R}$ (usually density function),

$$\nabla \log p(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \cdot \nabla p(\mathbf{x}).$$

Course Tricks I

Log-Derivative Trick

Given a differentiable function $p : \mathbb{R}^m \rightarrow \mathbb{R}$ (usually density function),

$$\nabla \log p(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \cdot \nabla p(\mathbf{x}).$$

Jensen's Inequality

If $\mathbf{x} \in \mathbb{R}^m$ is a continuous random variable with density $p(\mathbf{x})$ and $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is convex, then

$$\mathbb{E}[f(\mathbf{x})] \geq f(\mathbb{E}[\mathbf{x}]).$$

Course Tricks I

Log-Derivative Trick

Given a differentiable function $p : \mathbb{R}^m \rightarrow \mathbb{R}$ (usually density function),

$$\nabla \log p(\mathbf{x}) = \frac{1}{p(\mathbf{x})} \cdot \nabla p(\mathbf{x}).$$

Jensen's Inequality

If $\mathbf{x} \in \mathbb{R}^m$ is a continuous random variable with density $p(\mathbf{x})$ and $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is convex, then

$$\mathbb{E}[f(\mathbf{x})] \geq f(\mathbb{E}[\mathbf{x}]).$$

Monte Carlo Estimation

Let $\mathbf{x} \in \mathbb{R}^m$ be a continuous random variable with density $p(\mathbf{x})$, and $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^d$ be any vector-valued function. Then,

$$\mathbb{E}_{p(\mathbf{x})}\mathbf{f}(\mathbf{x}) = \int p(\mathbf{x})\mathbf{f}(\mathbf{x})d\mathbf{x} \approx \frac{1}{n} \sum_{i=1}^n \mathbf{f}(\mathbf{x}_i), \quad \text{where } \mathbf{x}_i \sim p(\mathbf{x}).$$

Course Tricks II

Change of Variables Theorem (CoV)

Let $\mathbf{x} \in \mathbb{R}^m$ be a random vector with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be a C^1 -diffeomorphism (\mathbf{f} and \mathbf{f}^{-1} are continuously differentiable mappings). If $\mathbf{z} = \mathbf{f}(\mathbf{x})$, then

$$p(\mathbf{x}) = p(\mathbf{z}) |\det(\mathbf{J}_\mathbf{f})| = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right|$$

Course Tricks II

Change of Variables Theorem (CoV)

Let $\mathbf{x} \in \mathbb{R}^m$ be a random vector with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be a C^1 -diffeomorphism (\mathbf{f} and \mathbf{f}^{-1} are continuously differentiable mappings). If $\mathbf{z} = \mathbf{f}(\mathbf{x})$, then

$$p(\mathbf{x}) = p(\mathbf{z}) |\det(\mathbf{J}_\mathbf{f})| = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(\mathbf{f}(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$

Course Tricks II

Change of Variables Theorem (CoV)

Let $\mathbf{x} \in \mathbb{R}^m$ be a random vector with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be a C^1 -diffeomorphism (\mathbf{f} and \mathbf{f}^{-1} are continuously differentiable mappings). If $\mathbf{z} = \mathbf{f}(\mathbf{x})$, then

$$p(\mathbf{x}) = p(\mathbf{z}) |\det(\mathbf{J}_\mathbf{f})| = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(\mathbf{f}(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$

$$p(\mathbf{z}) = p(\mathbf{x}) |\det(\mathbf{J}_{\mathbf{f}^{-1}})| = p(\mathbf{x}) \left| \det \left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}} \right) \right| = p(\mathbf{f}^{-1}(\mathbf{z})) \left| \det \left(\frac{\partial \mathbf{f}^{-1}(\mathbf{z})}{\partial \mathbf{z}} \right) \right|$$

Course Tricks II

Change of Variables Theorem (CoV)

Let $\mathbf{x} \in \mathbb{R}^m$ be a random vector with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be a C^1 -diffeomorphism (\mathbf{f} and \mathbf{f}^{-1} are continuously differentiable mappings). If $\mathbf{z} = \mathbf{f}(\mathbf{x})$, then

$$p(\mathbf{x}) = p(\mathbf{z}) |\det(\mathbf{J}_\mathbf{f})| = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(\mathbf{f}(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$

$$p(\mathbf{z}) = p(\mathbf{x}) |\det(\mathbf{J}_{\mathbf{f}^{-1}})| = p(\mathbf{x}) \left| \det \left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}} \right) \right| = p(\mathbf{f}^{-1}(\mathbf{z})) \left| \det \left(\frac{\partial \mathbf{f}^{-1}(\mathbf{z})}{\partial \mathbf{z}} \right) \right|$$

Proof (1D)

Assume f is monotonically increasing.

$$F_Y(y) = P(Y \leq y) = P(x \leq f^{-1}(y)) = F_X(f^{-1}(y))$$

Course Tricks II

Change of Variables Theorem (CoV)

Let $\mathbf{x} \in \mathbb{R}^m$ be a random vector with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be a C^1 -diffeomorphism (\mathbf{f} and \mathbf{f}^{-1} are continuously differentiable mappings). If $\mathbf{z} = \mathbf{f}(\mathbf{x})$, then

$$p(\mathbf{x}) = p(\mathbf{z}) |\det(\mathbf{J}_\mathbf{f})| = p(\mathbf{z}) \left| \det \left(\frac{\partial \mathbf{z}}{\partial \mathbf{x}} \right) \right| = p(\mathbf{f}(\mathbf{x})) \left| \det \left(\frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}} \right) \right|$$

$$p(\mathbf{z}) = p(\mathbf{x}) |\det(\mathbf{J}_{\mathbf{f}^{-1}})| = p(\mathbf{x}) \left| \det \left(\frac{\partial \mathbf{x}}{\partial \mathbf{z}} \right) \right| = p(\mathbf{f}^{-1}(\mathbf{z})) \left| \det \left(\frac{\partial \mathbf{f}^{-1}(\mathbf{z})}{\partial \mathbf{z}} \right) \right|$$

Proof (1D)

Assume f is monotonically increasing.

$$F_Y(y) = P(Y \leq y) = P(x \leq f^{-1}(y)) = F_X(f^{-1}(y))$$

$$p(y) = \frac{dF_Y(y)}{dy} = \frac{dF_X(f^{-1}(y))}{dy} = \frac{dF_X(x)}{dx} \frac{df^{-1}(y)}{dy} = p(x) \frac{df^{-1}(y)}{dy}$$

Course Tricks III

Law of the Unconscious Statistician (LOTUS)

Let $\mathbf{x} \in \mathbb{R}^m$ be a continuous random variable with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be measurable. If $\mathbf{y} = \mathbf{f}(\mathbf{x})$, then

$$\mathbb{E}_{p(\mathbf{y})}\mathbf{g}(\mathbf{y}) = \int p(\mathbf{y})\mathbf{g}(\mathbf{y})d\mathbf{y}$$

.

Course Tricks III

Law of the Unconscious Statistician (LOTUS)

Let $\mathbf{x} \in \mathbb{R}^m$ be a continuous random variable with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be measurable. If $\mathbf{y} = \mathbf{f}(\mathbf{x})$, then

$$\mathbb{E}_{p(\mathbf{y})}\mathbf{g}(\mathbf{y}) = \int p(\mathbf{y})\mathbf{g}(\mathbf{y})d\mathbf{y} = \int p(\mathbf{x})\mathbf{g}(\mathbf{f}(\mathbf{x}))d\mathbf{x} = \mathbb{E}_{p(\mathbf{x})}\mathbf{g}(\mathbf{f}(\mathbf{x})).$$

Course Tricks III

Law of the Unconscious Statistician (LOTUS)

Let $\mathbf{x} \in \mathbb{R}^m$ be a continuous random variable with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be measurable. If $\mathbf{y} = \mathbf{f}(\mathbf{x})$, then

$$\mathbb{E}_{p(\mathbf{y})}\mathbf{g}(\mathbf{y}) = \int p(\mathbf{y})\mathbf{g}(\mathbf{y})d\mathbf{y} = \int p(\mathbf{x})\mathbf{g}(\mathbf{f}(\mathbf{x}))d\mathbf{x} = \mathbb{E}_{p(\mathbf{x})}\mathbf{g}(\mathbf{f}(\mathbf{x})).$$

Dirac Delta Function

Any deterministic variable \mathbf{x}_0 can be interpreted as a random variable with density $p(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_0)$.

$$\delta(\mathbf{x}) = \begin{cases} +\infty, & \mathbf{x} = \mathbf{x}_0 \\ 0, & \mathbf{x} \neq \mathbf{x}_0 \end{cases} \quad \int \delta(\mathbf{x})d\mathbf{x} = 1$$

Course Tricks III

Law of the Unconscious Statistician (LOTUS)

Let $\mathbf{x} \in \mathbb{R}^m$ be a continuous random variable with density $p(\mathbf{x})$, and let $\mathbf{f} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ be measurable. If $\mathbf{y} = \mathbf{f}(\mathbf{x})$, then

$$\mathbb{E}_{p(\mathbf{y})}\mathbf{g}(\mathbf{y}) = \int p(\mathbf{y})\mathbf{g}(\mathbf{y})d\mathbf{y} = \int p(\mathbf{x})\mathbf{g}(\mathbf{f}(\mathbf{x}))d\mathbf{x} = \mathbb{E}_{p(\mathbf{x})}\mathbf{g}(\mathbf{f}(\mathbf{x})).$$

Dirac Delta Function

Any deterministic variable \mathbf{x}_0 can be interpreted as a random variable with density $p(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_0)$.

$$\delta(\mathbf{x}) = \begin{cases} +\infty, & \mathbf{x} = \mathbf{x}_0 \\ 0, & \mathbf{x} \neq \mathbf{x}_0 \end{cases} \quad \int \delta(\mathbf{x})d\mathbf{x} = 1$$

$$\mathbb{E}_{p(\mathbf{x})}\mathbf{f}(\mathbf{x}) = \int \delta(\mathbf{x} - \mathbf{x}_0)\mathbf{f}(\mathbf{x})d\mathbf{x} = \mathbf{f}(\mathbf{x}_0)$$

Outline

1. Generative Models Overview
2. Course Tricks
3. Problem Statement
4. Divergence Minimization Framework
5. Autoregressive Modeling

Problem Statement

We're given **finite** number of i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ drawn from an **unknown** distribution $p_{\text{data}}(\mathbf{x})$.

Problem Statement

We're given **finite** number of i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ drawn from an **unknown** distribution $p_{\text{data}}(\mathbf{x})$.

Objective

Our aim is to learn a distribution $p_{\text{data}}(\mathbf{x})$ that allows us to:

- ▶ Generate new samples from $p_{\text{data}}(\mathbf{x})$ (sample $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$) — **generation**.

Problem Statement

We're given **finite** number of i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ drawn from an **unknown** distribution $p_{\text{data}}(\mathbf{x})$.

Objective

Our aim is to learn a distribution $p_{\text{data}}(\mathbf{x})$ that allows us to:

- ▶ Generate new samples from $p_{\text{data}}(\mathbf{x})$ (sample $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$) — **generation**.
- ▶ Evaluate $p_{\text{data}}(\mathbf{x})$ on novel data (answering “How likely is an object \mathbf{x} ?”) — **density estimation**;

Problem Statement

We're given **finite** number of i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ drawn from an **unknown** distribution $p_{\text{data}}(\mathbf{x})$.

Objective

Our aim is to learn a distribution $p_{\text{data}}(\mathbf{x})$ that allows us to:

- ▶ Generate new samples from $p_{\text{data}}(\mathbf{x})$ (sample $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$) — **generation**.
- ▶ Evaluate $p_{\text{data}}(\mathbf{x})$ on novel data (answering “How likely is an object \mathbf{x} ?”) — **density estimation**;

Challenge

The data is high-dimensional and complex. For example, image datasets live in $\mathbb{R}^{\text{width} \times \text{height} \times \text{channels}}$. The curse of dimensionality makes accurately estimating $p_{\text{data}}(\mathbf{x})$ infeasible.

Problem Statement

We're given **finite** number of i.i.d. samples $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ drawn from an **unknown** distribution $p_{\text{data}}(\mathbf{x})$.

Objective

Our aim is to learn a distribution $p_{\text{data}}(\mathbf{x})$ that allows us to:

- ▶ Generate new samples from $p_{\text{data}}(\mathbf{x})$ (sample $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$) — **generation**.
- ▶ Evaluate $p_{\text{data}}(\mathbf{x})$ on novel data (answering “How likely is an object \mathbf{x} ?”) — **density estimation**;

Challenge

The data is high-dimensional and complex. For example, image datasets live in $\mathbb{R}^{\text{width} \times \text{height} \times \text{channels}}$. The curse of dimensionality makes accurately estimating $p_{\text{data}}(\mathbf{x})$ infeasible.

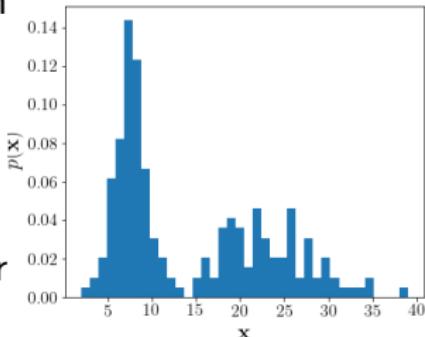
Note: here we use a strong assumption that our data is continuous (thus avoiding the domain of texts).

Histogram as a Generative Model

Assume $x \sim \text{Cat}(\pi)$. The histogram model is fully characterized by

$$\hat{\pi}_k = \hat{\pi}(x = k) = \frac{\sum_{i=1}^n [x_i = k]}{n}.$$

Curse of dimensionality: The number of bins rises exponentially.

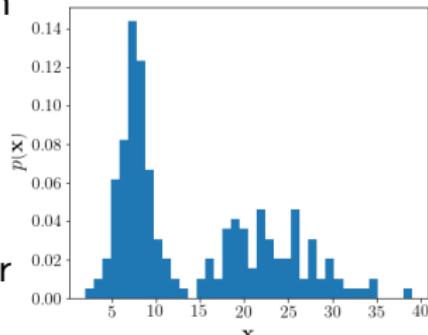


Histogram as a Generative Model

Assume $x \sim \text{Cat}(\pi)$. The histogram model is fully characterized by

$$\hat{\pi}_k = \hat{\pi}(x = k) = \frac{\sum_{i=1}^n [x_i = k]}{n}.$$

Curse of dimensionality: The number of bins rises exponentially.



MNIST example: 28×28 grayscale images, with each image $\mathbf{x} = (x_1, \dots, x_{784})$, $x_i \in \{0, 1\}$:

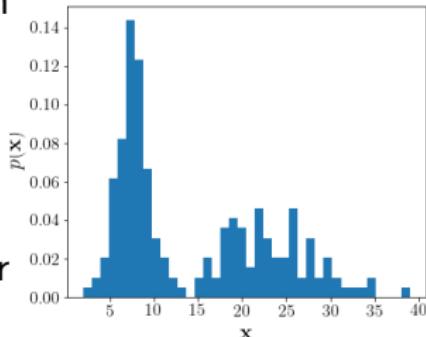
$$p_{\text{data}}(\mathbf{x}) = p(x_1) \cdot p(x_2|x_1) \cdot \dots \cdot p(x_m|x_{m-1}, \dots, x_1).$$

Histogram as a Generative Model

Assume $x \sim \text{Cat}(\pi)$. The histogram model is fully characterized by

$$\hat{\pi}_k = \hat{\pi}(x = k) = \frac{\sum_{i=1}^n [x_i = k]}{n}.$$

Curse of dimensionality: The number of bins rises exponentially.



MNIST example: 28×28 grayscale images, with each image $\mathbf{x} = (x_1, \dots, x_{784})$, $x_i \in \{0, 1\}$:

$$p_{\text{data}}(\mathbf{x}) = p(x_1) \cdot p(x_2|x_1) \cdot \dots \cdot p(x_m|x_{m-1}, \dots, x_1).$$

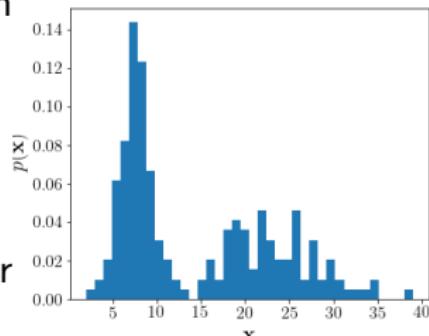
A complete histogram would require $2^{28 \times 28} - 1$ parameters for $p_{\text{data}}(\mathbf{x})$.

Histogram as a Generative Model

Assume $x \sim \text{Cat}(\pi)$. The histogram model is fully characterized by

$$\hat{\pi}_k = \hat{\pi}(x = k) = \frac{\sum_{i=1}^n [x_i = k]}{n}.$$

Curse of dimensionality: The number of bins rises exponentially.



MNIST example: 28×28 grayscale images, with each image $\mathbf{x} = (x_1, \dots, x_{784})$, $x_i \in \{0, 1\}$:

$$p_{\text{data}}(\mathbf{x}) = p(x_1) \cdot p(x_2|x_1) \cdot \dots \cdot p(x_m|x_{m-1}, \dots, x_1).$$

A complete histogram would require $2^{28 \times 28} - 1$ parameters for $p_{\text{data}}(\mathbf{x})$.

Question: How many parameters are required in these cases?

$$p_{\text{data}}(\mathbf{x}) = p(x_1) \cdot p(x_2) \cdot \dots \cdot p(x_m);$$

$$p_{\text{data}}(\mathbf{x}) = p(x_1) \cdot p(x_2|x_1) \cdot \dots \cdot p(x_m|x_{m-1}).$$

Conditional Models

In practice, we're typically interested in learning conditional models (sampling from conditional distribution $p_{\text{data}}(\mathbf{x}|\mathbf{y})$).

Conditional Models

In practice, we're typically interested in learning conditional models (sampling from conditional distribution $p_{\text{data}}(\mathbf{x}|\mathbf{y})$).

- ▶ $\mathbf{y} = \emptyset, \mathbf{x} = \text{image} \Rightarrow \text{unconditional image model}$
- ▶ $\mathbf{y} = \text{class label}, \mathbf{x} = \text{image} \Rightarrow \text{class-conditional image model}$
- ▶ $\mathbf{y} = \text{text prompt}, \mathbf{x} = \text{image} \Rightarrow \text{text-to-image model}$
- ▶ $\mathbf{y} = \text{image}, \mathbf{x} = \text{image} \Rightarrow \text{image-to-image model}$
- ▶ $\mathbf{y} = \text{image}, \mathbf{x} = \text{text} \Rightarrow \text{image-to-text (image captioning) model}$
- ▶ $\mathbf{y} = \text{English text}, \mathbf{x} = \text{Russian text} \Rightarrow \text{sequence-to-sequence model (machine translation) model}$
- ▶ $\mathbf{y} = \text{sound}, \mathbf{x} = \text{text} \Rightarrow \text{speech-to-text (automatic speech recognition) model}$
- ▶ $\mathbf{y} = \text{text}, \mathbf{x} = \text{sound} \Rightarrow \text{text-to-speech model}$

Outline

1. Generative Models Overview
2. Course Tricks
3. Problem Statement
4. Divergence Minimization Framework
5. Autoregressive Modeling

Divergences

- ▶ Let us fix a probabilistic model $p_{\theta}(\mathbf{x})$ from a parametric family of distributions $\{p(\mathbf{x}|\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}$.

Divergences

- ▶ Let us fix a probabilistic model $p_{\theta}(\mathbf{x})$ from a parametric family of distributions $\{p(\mathbf{x}|\theta) \mid \theta \in \Theta\}$.
- ▶ Instead of searching among all possible distributions for the true $p_{\text{data}}(\mathbf{x})$, we seek a functional approximation $p_{\theta}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$.

Divergences

- ▶ Let us fix a probabilistic model $p_{\theta}(\mathbf{x})$ from a parametric family of distributions $\{p(\mathbf{x}|\theta) \mid \theta \in \Theta\}$.
- ▶ Instead of searching among all possible distributions for the true $p_{\text{data}}(\mathbf{x})$, we seek a functional approximation $p_{\theta}(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$.

What is a Divergence?

Let \mathcal{P} be the set of all probability distributions. A mapping $D : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$ is called a **divergence** if

- ▶ $D(\pi \| p) \geq 0$ for all $\pi, p \in \mathcal{P}$
- ▶ $D(\pi \| p) = 0$ if and only if $\pi \equiv p$

Divergences

- ▶ Let us fix a probabilistic model $p_\theta(\mathbf{x})$ from a parametric family of distributions $\{p(\mathbf{x}|\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \Theta\}$.
- ▶ Instead of searching among all possible distributions for the true $p_{\text{data}}(\mathbf{x})$, we seek a functional approximation $p_\theta(\mathbf{x}) \approx p_{\text{data}}(\mathbf{x})$.

What is a Divergence?

Let \mathcal{P} be the set of all probability distributions. A mapping $D : \mathcal{P} \times \mathcal{P} \rightarrow \mathbb{R}$ is called a **divergence** if

- ▶ $D(\pi \| p) \geq 0$ for all $\pi, p \in \mathcal{P}$
- ▶ $D(\pi \| p) = 0$ if and only if $\pi \equiv p$

Divergence Minimization Problem

$$\min_{\theta} D(p_{\text{data}} \| p_{\theta})$$

where $p_{\text{data}}(\mathbf{x})$ is the true data distribution and $p_\theta(\mathbf{x})$ is the model distribution.

Forward KL vs Reverse KL (Kullback-Leibler Divergence)

Forward KL

$$\text{KL}(p_{\text{data}} \| p_{\theta}) = \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \rightarrow \min_{\theta}$$

Forward KL vs Reverse KL (Kullback-Leibler Divergence)

Forward KL

$$\text{KL}(p_{\text{data}} \| p_{\theta}) = \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \rightarrow \min_{\theta}$$

Reverse KL

$$\text{KL}(p_{\theta} \| p_{\text{data}}) = \int p_{\theta}(\mathbf{x}) \log \frac{p_{\theta}(\mathbf{x})}{p_{\text{data}}(\mathbf{x})} d\mathbf{x} \rightarrow \min_{\theta}$$

Forward KL vs Reverse KL (Kullback-Leibler Divergence)

Forward KL

$$\text{KL}(p_{\text{data}} \| p_{\theta}) = \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \rightarrow \min_{\theta}$$

Reverse KL

$$\text{KL}(p_{\theta} \| p_{\text{data}}) = \int p_{\theta}(\mathbf{x}) \log \frac{p_{\theta}(\mathbf{x})}{p_{\text{data}}(\mathbf{x})} d\mathbf{x} \rightarrow \min_{\theta}$$

What's the practical distinction between these two objectives?

Forward KL vs Reverse KL (Kullback-Leibler Divergence)

Forward KL

$$\text{KL}(p_{\text{data}} \| p_{\theta}) = \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \rightarrow \min_{\theta}$$

Reverse KL

$$\text{KL}(p_{\theta} \| p_{\text{data}}) = \int p_{\theta}(\mathbf{x}) \log \frac{p_{\theta}(\mathbf{x})}{p_{\text{data}}(\mathbf{x})} d\mathbf{x} \rightarrow \min_{\theta}$$

What's the practical distinction between these two objectives?

Maximum Likelihood Estimation (MLE)

Let $\{\mathbf{x}_i\}_{i=1}^n$ be i.i.d. observed samples.

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^n p_{\theta}(\mathbf{x}_i) = \arg \max_{\theta} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i).$$

Forward KL vs Reverse KL: MLE as Forward KL

Forward KL

$$\text{KL}(p_{\text{data}} \| p_{\theta}) = \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x}$$

Forward KL vs Reverse KL: MLE as Forward KL

Forward KL

$$\begin{aligned} \text{KL}(p_{\text{data}} \| p_{\theta}) &= \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \\ &= \int p_{\text{data}}(\mathbf{x}) \log p_{\text{data}}(\mathbf{x}) d\mathbf{x} - \int p_{\text{data}}(\mathbf{x}) \log p_{\theta}(\mathbf{x}) d\mathbf{x} \end{aligned}$$

Forward KL vs Reverse KL: MLE as Forward KL

Forward KL

$$\begin{aligned}\text{KL}(p_{\text{data}} \| p_{\theta}) &= \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \\ &= \int p_{\text{data}}(\mathbf{x}) \log p_{\text{data}}(\mathbf{x}) d\mathbf{x} - \int p_{\text{data}}(\mathbf{x}) \log p_{\theta}(\mathbf{x}) d\mathbf{x} \\ &= -\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\log p_{\theta}(\mathbf{x})] + \text{const}\end{aligned}$$

Forward KL vs Reverse KL: MLE as Forward KL

Forward KL

$$\begin{aligned}\text{KL}(p_{\text{data}} \| p_{\theta}) &= \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \\ &= \int p_{\text{data}}(\mathbf{x}) \log p_{\text{data}}(\mathbf{x}) d\mathbf{x} - \int p_{\text{data}}(\mathbf{x}) \log p_{\theta}(\mathbf{x}) d\mathbf{x} \\ &= -\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\log p_{\theta}(\mathbf{x})] + \text{const} \\ &\approx -\frac{1}{n} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i) + \text{const} \rightarrow \min_{\theta}.\end{aligned}$$

Forward KL vs Reverse KL: MLE as Forward KL

Forward KL

$$\begin{aligned}\text{KL}(p_{\text{data}} \| p_{\theta}) &= \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \\ &= \int p_{\text{data}}(\mathbf{x}) \log p_{\text{data}}(\mathbf{x}) d\mathbf{x} - \int p_{\text{data}}(\mathbf{x}) \log p_{\theta}(\mathbf{x}) d\mathbf{x} \\ &= -\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\log p_{\theta}(\mathbf{x})] + \text{const} \\ &\approx -\frac{1}{n} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i) + \text{const} \rightarrow \min_{\theta}.\end{aligned}$$

Maximum likelihood estimation is thus equivalent to minimizing a Monte Carlo estimate of the forward KL divergence.

Forward KL vs Reverse KL: MLE as Forward KL

Forward KL

$$\begin{aligned}\text{KL}(p_{\text{data}} \| p_{\theta}) &= \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \\ &= \int p_{\text{data}}(\mathbf{x}) \log p_{\text{data}}(\mathbf{x}) d\mathbf{x} - \int p_{\text{data}}(\mathbf{x}) \log p_{\theta}(\mathbf{x}) d\mathbf{x} \\ &= -\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\log p_{\theta}(\mathbf{x})] + \text{const} \\ &\approx -\frac{1}{n} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i) + \text{const} \rightarrow \min_{\theta}.\end{aligned}$$

Maximum likelihood estimation is thus equivalent to minimizing a Monte Carlo estimate of the forward KL divergence.

Reverse KL

$$\text{KL}(p_{\theta} \| p_{\text{data}}) = \int p_{\theta}(\mathbf{x}) \log \frac{p_{\theta}(\mathbf{x})}{p_{\text{data}}(\mathbf{x})} d\mathbf{x}$$

Forward KL vs Reverse KL: MLE as Forward KL

Forward KL

$$\begin{aligned}\text{KL}(p_{\text{data}} \| p_{\theta}) &= \int p_{\text{data}}(\mathbf{x}) \log \frac{p_{\text{data}}(\mathbf{x})}{p_{\theta}(\mathbf{x})} d\mathbf{x} \\ &= \int p_{\text{data}}(\mathbf{x}) \log p_{\text{data}}(\mathbf{x}) d\mathbf{x} - \int p_{\text{data}}(\mathbf{x}) \log p_{\theta}(\mathbf{x}) d\mathbf{x} \\ &= -\mathbb{E}_{p_{\text{data}}(\mathbf{x})} [\log p_{\theta}(\mathbf{x})] + \text{const} \\ &\approx -\frac{1}{n} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i) + \text{const} \rightarrow \min_{\theta}.\end{aligned}$$

Maximum likelihood estimation is thus equivalent to minimizing a Monte Carlo estimate of the forward KL divergence.

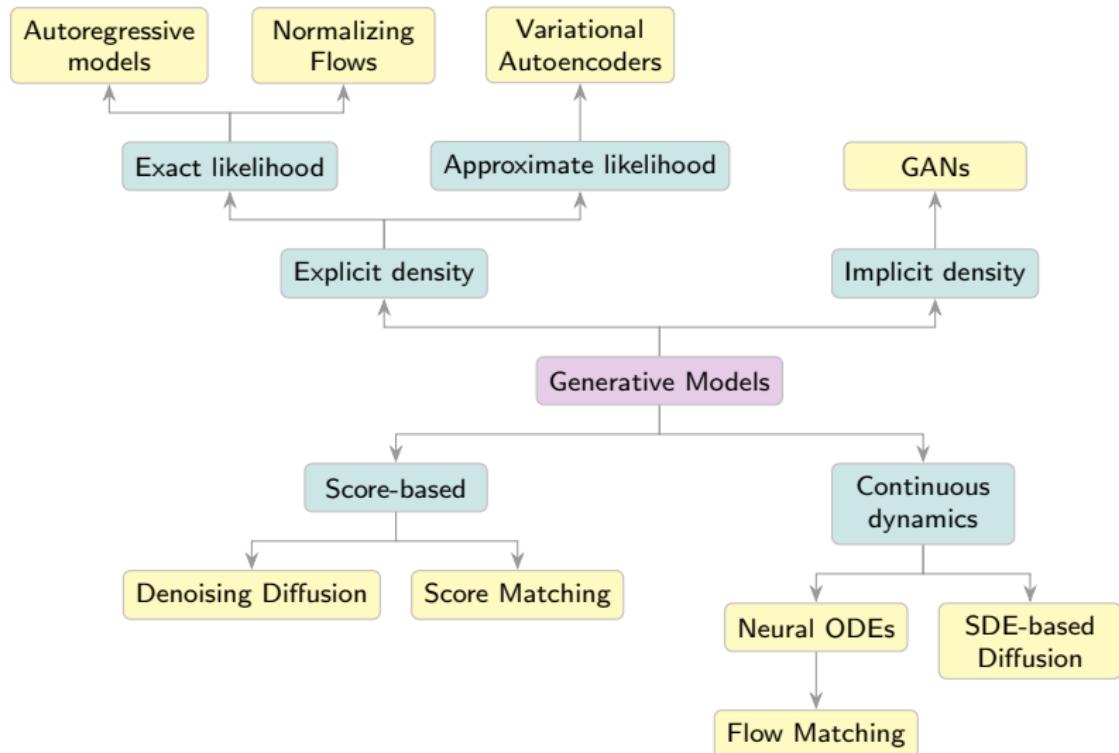
Reverse KL

$$\begin{aligned}\text{KL}(p_{\theta} \| p_{\text{data}}) &= \int p_{\theta}(\mathbf{x}) \log \frac{p_{\theta}(\mathbf{x})}{p_{\text{data}}(\mathbf{x})} d\mathbf{x} \\ &= \mathbb{E}_{p_{\theta}(\mathbf{x})} [\log p_{\theta}(\mathbf{x}) - \log p_{\text{data}}(\mathbf{x})] \rightarrow \min_{\theta}\end{aligned}$$

Outline

1. Generative Models Overview
2. Course Tricks
3. Problem Statement
4. Divergence Minimization Framework
5. Autoregressive Modeling

Generative Models Taxonomy



Autoregressive Modeling

MLE Problem

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^n p_{\theta}(\mathbf{x}_i) = \arg \max_{\theta} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i)$$

Autoregressive Modeling

MLE Problem

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^n p_{\theta}(\mathbf{x}_i) = \arg \max_{\theta} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i)$$

- ▶ This maximization is typically solved via gradient-based optimization.
- ▶ Thus, efficient computation of both $\log p_{\theta}(\mathbf{x})$ and its gradient $\frac{\partial \log p_{\theta}(\mathbf{x})}{\partial \theta}$ is crucial.

Autoregressive Modeling

MLE Problem

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^n p_{\theta}(\mathbf{x}_i) = \arg \max_{\theta} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i)$$

- ▶ This maximization is typically solved via gradient-based optimization.
- ▶ Thus, efficient computation of both $\log p_{\theta}(\mathbf{x})$ and its gradient $\frac{\partial \log p_{\theta}(\mathbf{x})}{\partial \theta}$ is crucial.

Likelihood as a Product of Conditionals

For $\mathbf{x} = (x_1, \dots, x_m)$, $\mathbf{x}_{1:j} = (x_1, \dots, x_j)$,

$$p_{\theta}(\mathbf{x}) = \prod_{j=1}^m p_{\theta}(x_j | \mathbf{x}_{1:j-1}); \quad \log p_{\theta}(\mathbf{x}) = \sum_{j=1}^m \log p_{\theta}(x_j | \mathbf{x}_{1:j-1})$$

Autoregressive Modeling

MLE Problem

$$\theta^* = \arg \max_{\theta} \prod_{i=1}^n p_{\theta}(\mathbf{x}_i) = \arg \max_{\theta} \sum_{i=1}^n \log p_{\theta}(\mathbf{x}_i)$$

- ▶ This maximization is typically solved via gradient-based optimization.
- ▶ Thus, efficient computation of both $\log p_{\theta}(\mathbf{x})$ and its gradient $\frac{\partial \log p_{\theta}(\mathbf{x})}{\partial \theta}$ is crucial.

Likelihood as a Product of Conditionals

For $\mathbf{x} = (x_1, \dots, x_m)$, $\mathbf{x}_{1:j} = (x_1, \dots, x_j)$,

$$p_{\theta}(\mathbf{x}) = \prod_{j=1}^m p_{\theta}(x_j | \mathbf{x}_{1:j-1}); \quad \log p_{\theta}(\mathbf{x}) = \sum_{j=1}^m \log p_{\theta}(x_j | \mathbf{x}_{1:j-1})$$

$$\theta^* = \arg \max_{\theta} \sum_{i=1}^n \left[\sum_{j=1}^m \log p_{\theta}(x_{ij} | \mathbf{x}_{i,1:j-1}) \right]$$

Autoregressive Models

$$\log p_{\theta}(\mathbf{x}) = \sum_{j=1}^m \log p_{\theta}(x_j | \mathbf{x}_{1:j-1})$$

Autoregressive Models

$$\log p_{\theta}(\mathbf{x}) = \sum_{j=1}^m \log p_{\theta}(x_j | \mathbf{x}_{1:j-1})$$

- ▶ Sampling is performed sequentially:
 - ▶ Sample $\hat{x}_1 \sim p_{\theta}(x_1)$;
 - ▶ Sample $\hat{x}_2 \sim p_{\theta}(x_2 | \hat{x}_1)$;
 - ▶ ...
 - ▶ Sample $\hat{x}_m \sim p_{\theta}(x_m | \hat{\mathbf{x}}_{1:m-1})$;
 - ▶ The generated sample is $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m)$.

Autoregressive Models

$$\log p_{\theta}(\mathbf{x}) = \sum_{j=1}^m \log p_{\theta}(x_j | \mathbf{x}_{1:j-1})$$

- ▶ Sampling is performed sequentially:
 - ▶ Sample $\hat{x}_1 \sim p_{\theta}(x_1)$;
 - ▶ Sample $\hat{x}_2 \sim p_{\theta}(x_2 | \hat{x}_1)$;
 - ▶ ...
 - ▶ Sample $\hat{x}_m \sim p_{\theta}(x_m | \hat{\mathbf{x}}_{1:m-1})$;
 - ▶ The generated sample is $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m)$.
- ▶ Each conditional $p_{\theta}(x_j | \mathbf{x}_{1:j-1})$ can be modeled using a neural network.

Autoregressive Models

$$\log p_{\theta}(\mathbf{x}) = \sum_{j=1}^m \log p_{\theta}(x_j | \mathbf{x}_{1:j-1})$$

- ▶ Sampling is performed sequentially:
 - ▶ Sample $\hat{x}_1 \sim p_{\theta}(x_1)$;
 - ▶ Sample $\hat{x}_2 \sim p_{\theta}(x_2 | \hat{x}_1)$;
 - ▶ ...
 - ▶ Sample $\hat{x}_m \sim p_{\theta}(x_m | \hat{\mathbf{x}}_{1:m-1})$;
 - ▶ The generated sample is $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m)$.
- ▶ Each conditional $p_{\theta}(x_j | \mathbf{x}_{1:j-1})$ can be modeled using a neural network.
- ▶ Modeling all conditionals separately isn't feasible. To address this, we share parameters across all conditionals.

Autoregressive Models: MLP

For large j , the conditional $p_{\theta}(x_j | \mathbf{x}_{1:j-1})$ becomes intractable as the history $\mathbf{x}_{1:j-1}$ grows variable-length.

Autoregressive Models: MLP

For large j , the conditional $p_{\theta}(x_j | \mathbf{x}_{1:j-1})$ becomes intractable as the history $\mathbf{x}_{1:j-1}$ grows variable-length.

Markov Assumption

$$p_{\theta}(x_j | \mathbf{x}_{1:j-1}) = p_{\theta}(x_j | \mathbf{x}_{j-d:j-1}), \quad d \text{ is a fixed parameter.}$$

Autoregressive Models: MLP

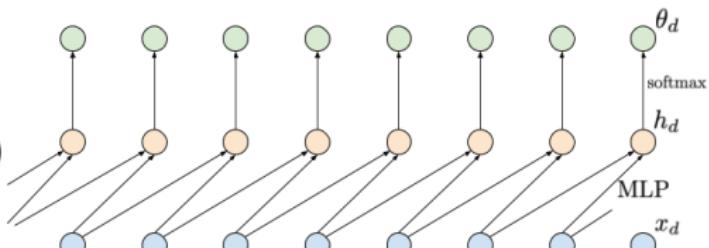
For large j , the conditional $p_{\theta}(x_j | \mathbf{x}_{1:j-1})$ becomes intractable as the history $\mathbf{x}_{1:j-1}$ grows variable-length.

Markov Assumption

$$p_{\theta}(x_j | \mathbf{x}_{1:j-1}) = p_{\theta}(x_j | \mathbf{x}_{j-d:j-1}), \quad d \text{ is a fixed parameter.}$$

Example

- ▶ $d = 2$
- ▶ $x_j \in \{0, 255\}$
- ▶ $\mathbf{h}_j = \text{MLP}_{\theta}(x_{j-1}, x_{j-2})$
- ▶ $\pi_j = \text{softmax}(\mathbf{h}_j)$
- ▶ $p_{\theta}(x_j | x_{j-1}, x_{j-2}) = \text{Cat}(\pi_j)$



Autoregressive Models: MLP

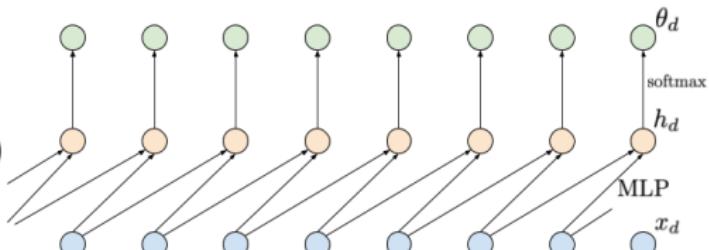
For large j , the conditional $p_{\theta}(x_j | \mathbf{x}_{1:j-1})$ becomes intractable as the history $\mathbf{x}_{1:j-1}$ grows variable-length.

Markov Assumption

$$p_{\theta}(x_j | \mathbf{x}_{1:j-1}) = p_{\theta}(x_j | \mathbf{x}_{j-d:j-1}), \quad d \text{ is a fixed parameter.}$$

Example

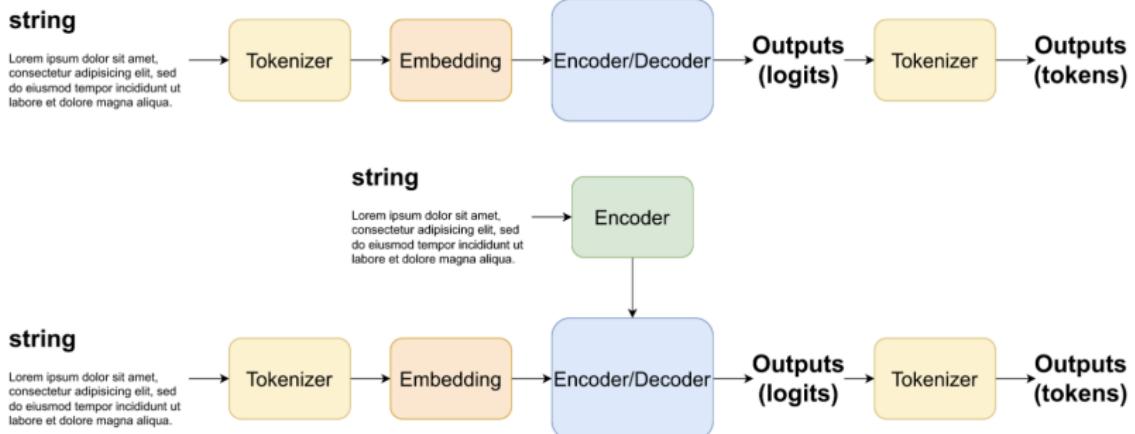
- ▶ $d = 2$
- ▶ $x_j \in \{0, 255\}$
- ▶ $\mathbf{h}_j = \text{MLP}_{\theta}(x_{j-1}, x_{j-2})$
- ▶ $\pi_j = \text{softmax}(\mathbf{h}_j)$
- ▶ $p_{\theta}(x_j | x_{j-1}, x_{j-2}) = \text{Cat}(\pi_j)$



Can we also model continuous-valued data, not just the discrete case?

Autoregressive Models: LLM

$$p_{\theta}(x_j | \mathbf{x}_{1:j-1}) = p_{\theta}(x_j | \mathbf{x}_{j-d:j-1}), \quad d \text{ is the context window.}$$



Autoregressive Models for Images

How do we model the distribution $p_{\text{data}}(\mathbf{x})$ of natural images?

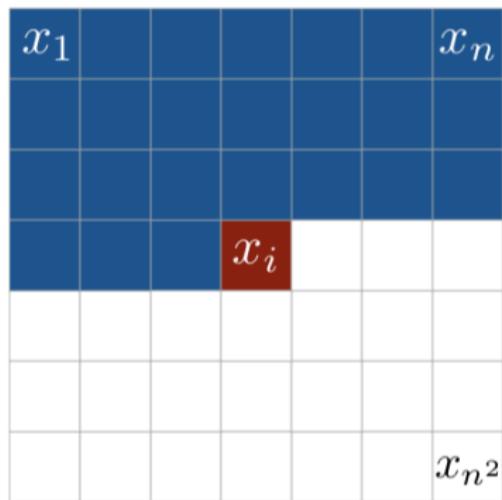
$$p_{\theta}(\mathbf{x}) = \prod_{j=1}^{\text{width} \times \text{height}} p_{\theta}(x_j | \mathbf{x}_{1:j-1})$$

Autoregressive Models for Images

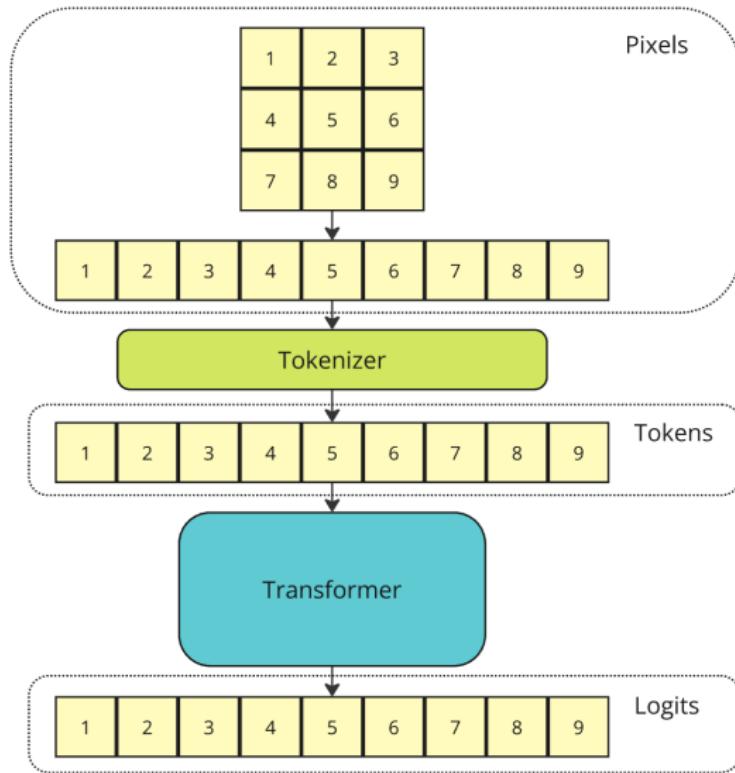
How do we model the distribution $p_{\text{data}}(\mathbf{x})$ of natural images?

$$p_{\theta}(\mathbf{x}) = \prod_{j=1}^{\text{width} \times \text{height}} p_{\theta}(x_j | \mathbf{x}_{1:j-1})$$

- ▶ A pixel ordering must be selected; the raster scan is a standard choice.
- ▶ RGB channel dependencies can be modeled explicitly as well.



Autoregressive Models: ImageGPT



Summary

- ▶ Our target is to approximate the data distribution both for density estimation and for generation.
- ▶ The divergence minimization framework offers a principled way to learn distributions that match the data.
- ▶ Minimizing the forward KL divergence is equivalent to maximum likelihood estimation.
- ▶ Autoregressive models decompose the joint distribution as a product of conditionals.
- ▶ Autoregressive sampling is simple, but inherently sequential.
- ▶ Joint density evaluation multiplies all conditional probabilities $p_{\theta}(x_j | \mathbf{x}_{1:j-1})$.
- ▶ ImageGPT applies a transformer architecture to sequences of raster-ordered image pixels.