Lecture 13

Generative Models: Introduction

Speaker: Kaiming He

Overview: Lectures on Generative Models

- Today: **Introduction**

- 10/23: "**Classical" methods**
  - Variational Autoencoder (VAE)
  - Generative Adversarial Network (GAN)

- 10/30: **"Modern" methods**
  - Diffusion Models
  - Flow Matching

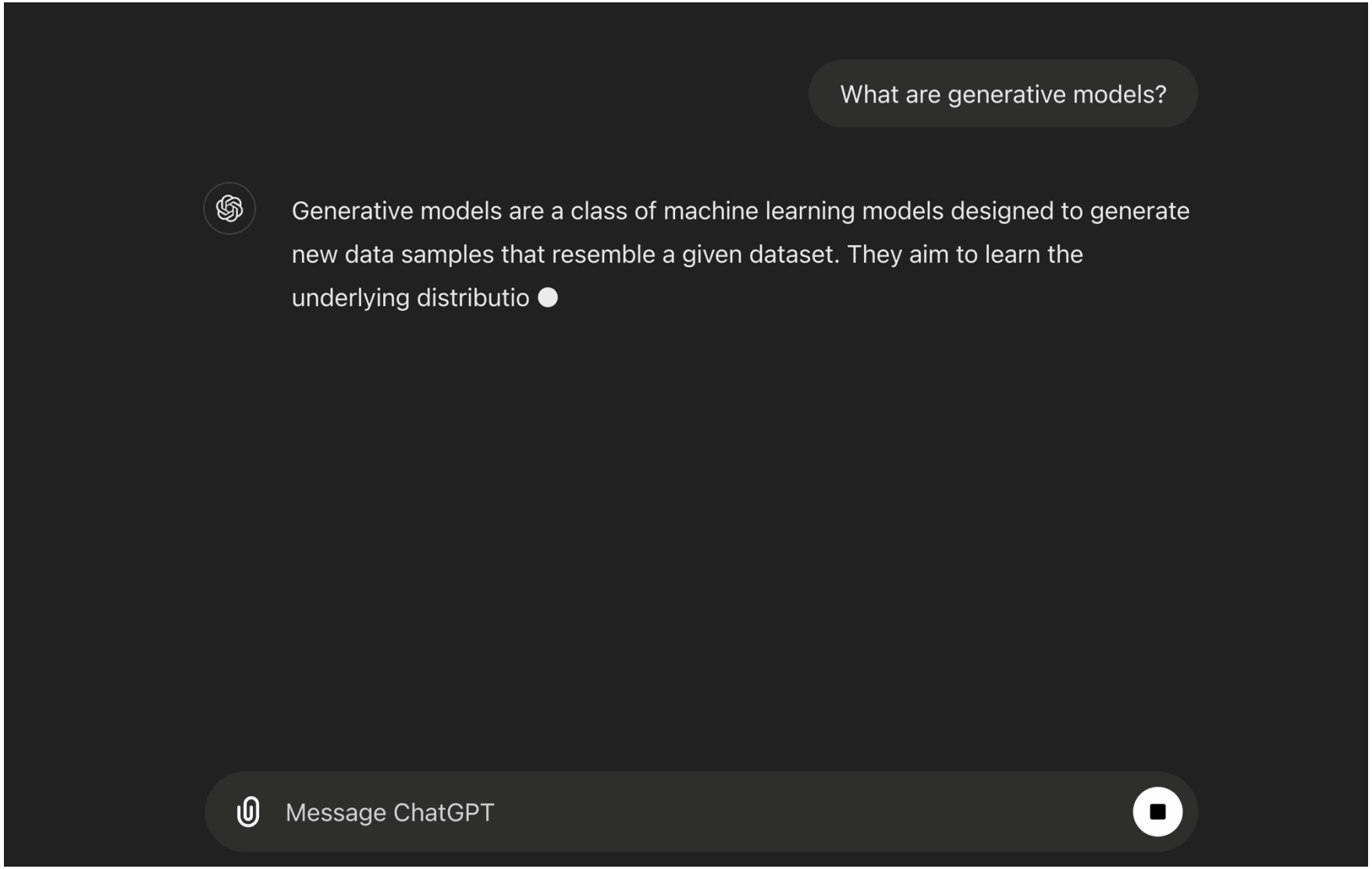
- Previous lectures: Autoregressive (AR) models - RNN, Transformer, LLM, ...

Today: Introduction to Generative Models

- What are Generative Models?
- Generative Modeling of Real-world Problems
- Families of Generative Models

The “GenAI” Era

Chatbot and natural language conversation



The “GenAI” Era

Text-to-image generation



Generated by Stable Diffusion 3 Medium.
Prompt: teddy bear teaching a course, with "generative models" written on blackboard

The “GenAI” Era

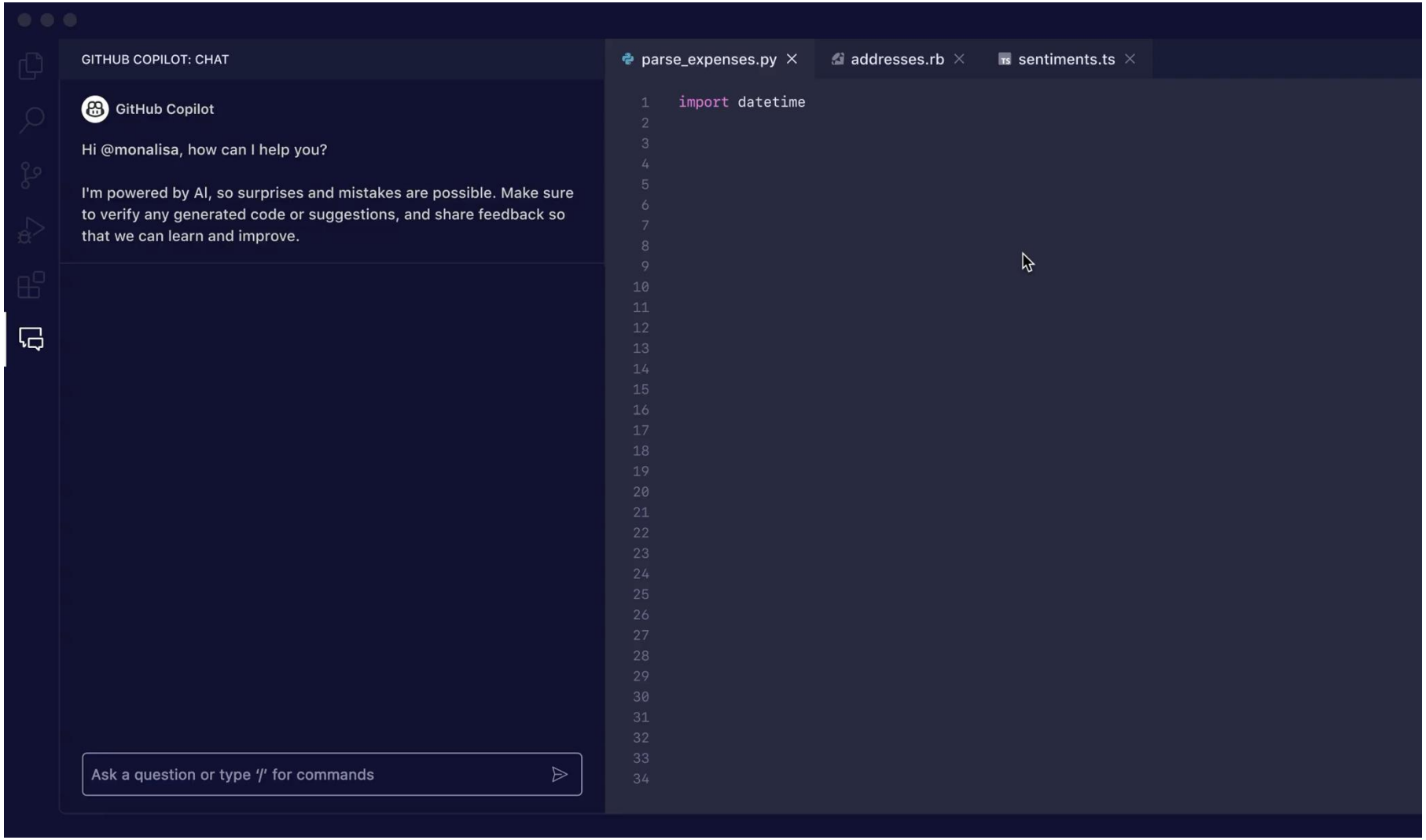
Text-to-video generation



Generated by Sora

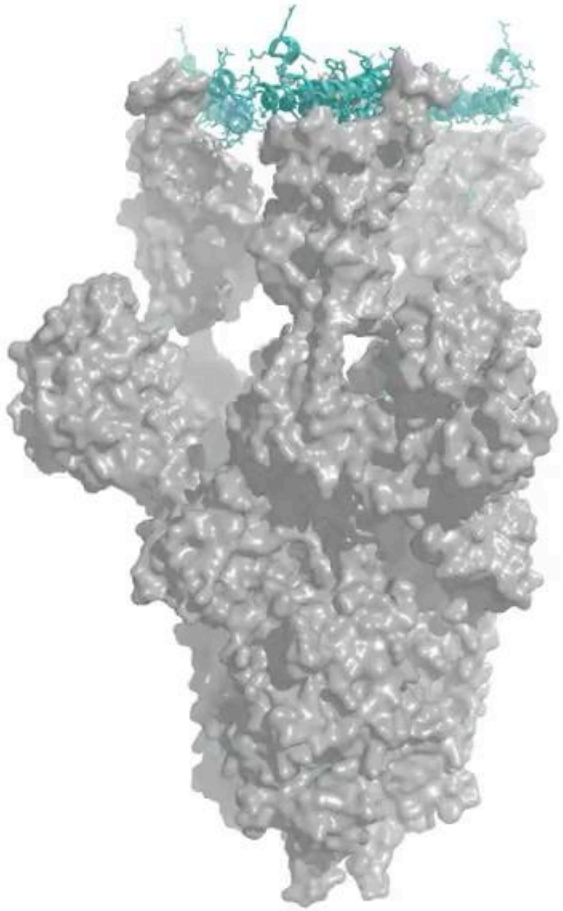
The “GenAI” Era

AI assistant for code generation



The “GenAI” Era

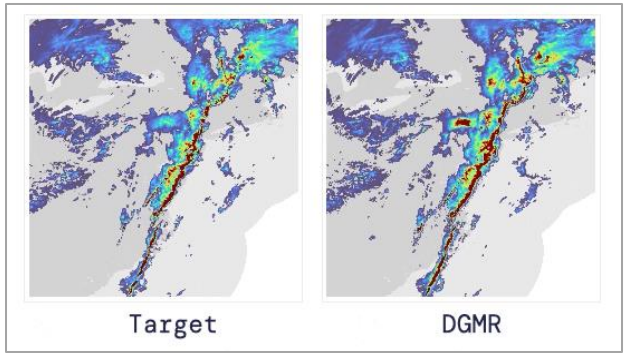
Protein design and generation



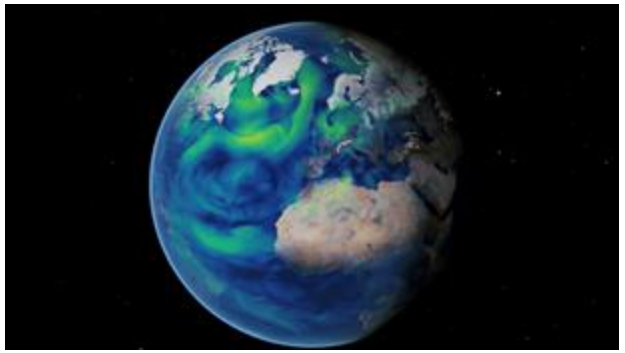
Watson, et al. De novo design of protein structure and function with RFdiffusion, Nature 2023

The “GenAI” Era

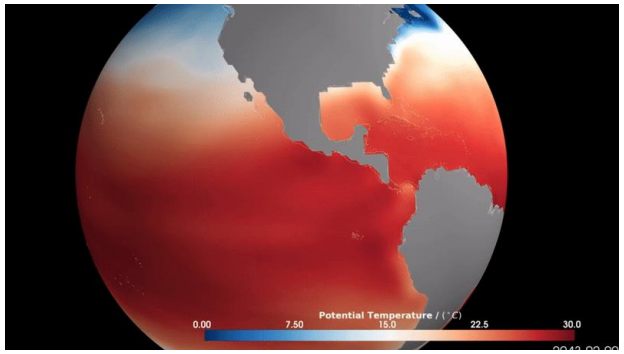
Scientific Problems



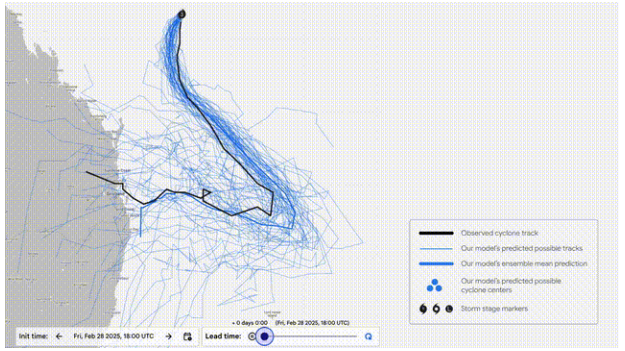
precipitation



weather



ocean

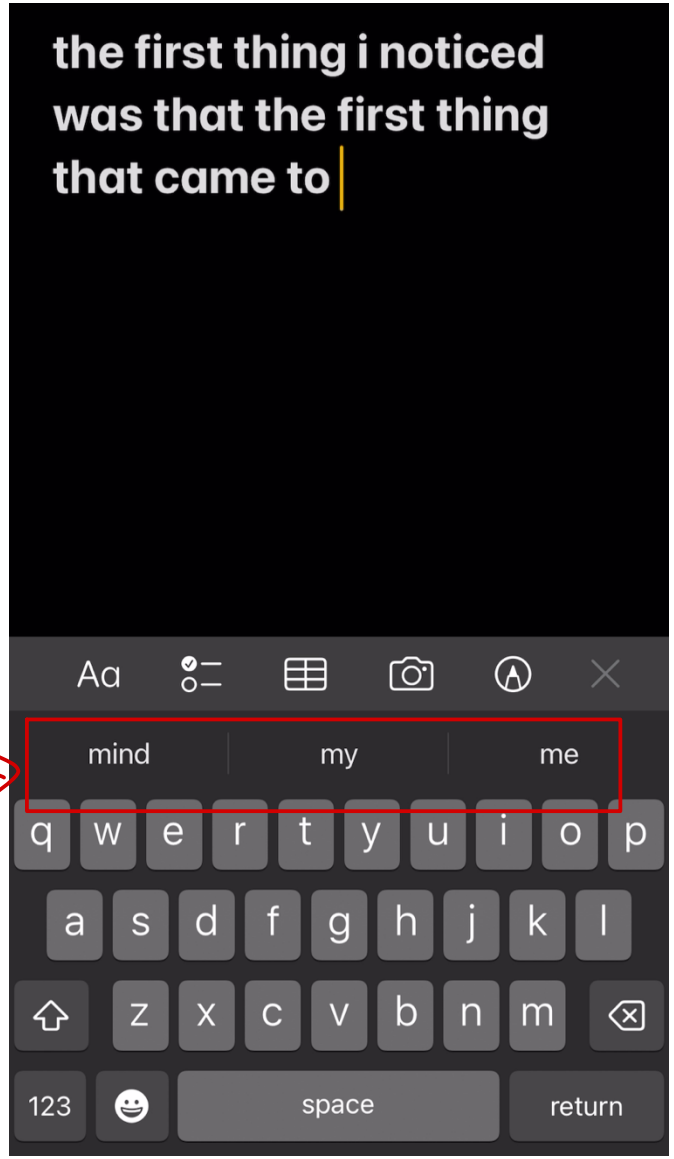


cyclone

Skilful precipitation nowcasting using deep generative models of radar. S Ravuri, et al., Nature, 2021.
Meet “Samudra”: M²LInES’ AI Global Ocean Emulator for Climate. L Zanna, 2024.
How we're supporting better tropical cyclone prediction with AI. DeepMind, 2025.
FourCastNet 3 Enables Fast and Accurate Large Ensemble Weather Forecasting with Scalable Geometric ML. B Bonev et al., 2025.

Generative Models before the “GenAI” Era

Your keyboard



자동완성

Generative Models before the “GenAI” Era

2009, PatchMatch: Photoshop’s Content-aware Fill



p(빈 픽셀 | 주변 픽셀)

→ GenAI (X)

randomized search algorithm

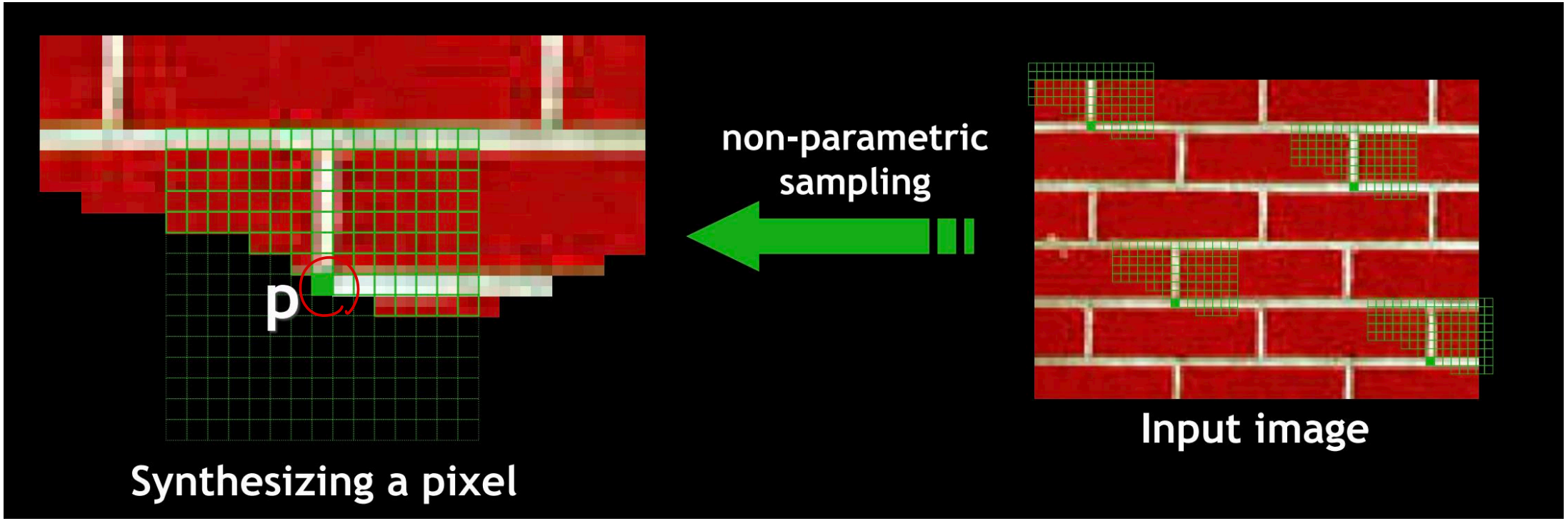
PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing, SIGGRAPH 2009

Generative Models before the “GenAI” Era

1999, the Efros-Leung algorithm for texture synthesis

In today’s word: this is an **Autoregressive** model

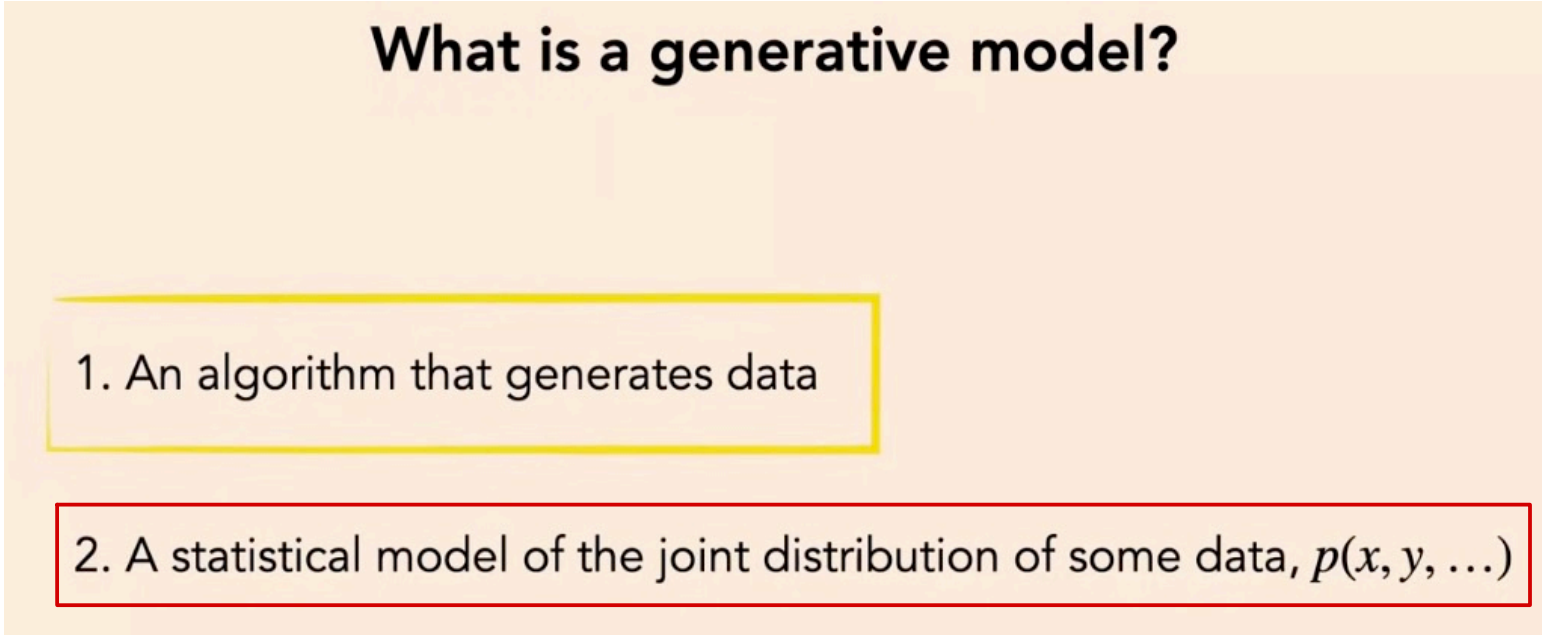
이전의 생성 결과물을 조건으로 다음 결과는 순차적으로 생성.



현재까지 생성된 이웃이 주어졌을 때 다음 픽셀의 조건부분포는 input image로부터 추정.
(여러 후보군의 것은 Sampling)

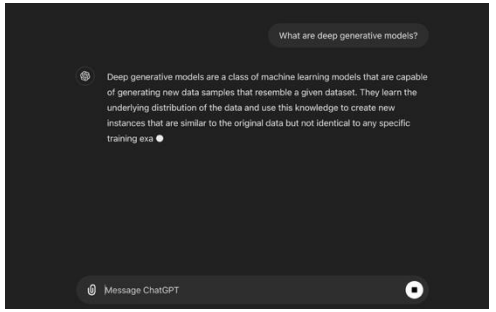
Texture Synthesis by Non-parametric Sampling, ICCV 1999

What are Generative Models?



What do these scenarios have in common?

- There are **multiple** predictions to one input.
- Some predictions are more **plausible**.
- Training data may contain **no solution**.
- Output may be **more complex**, more informative, and higher-dimensional



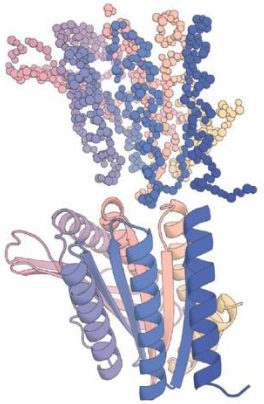
Chatbot



Image generation



Video generation



Protein generation

Discriminative Models vs. Generative Models

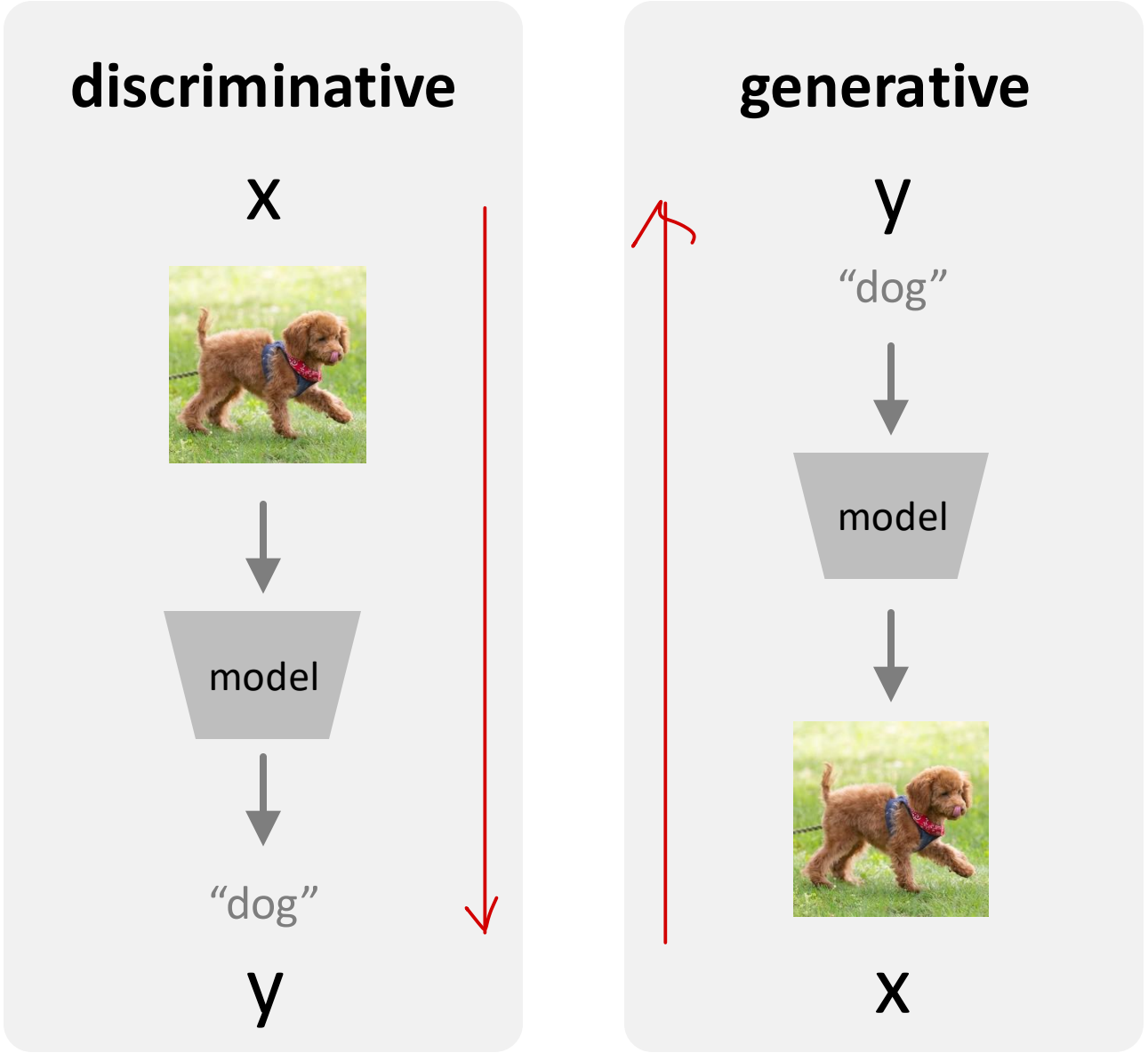
Discriminative Model

- "sample" x ⇨ "label" y
- one desired output

Image classifier: $f : \mathbb{R}^{N \cdot M \cdot C} \rightarrow \{dog, \cdots, cat\}$

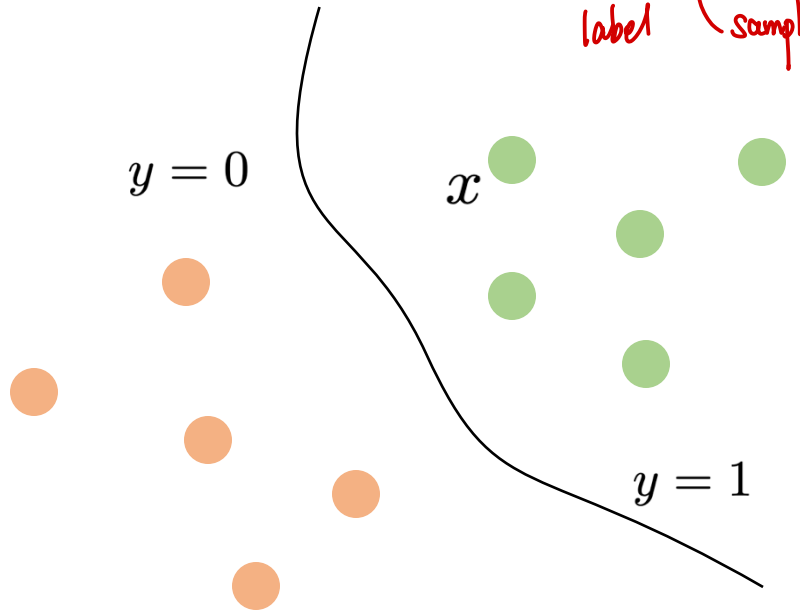
Generative Model

- "label" y ⇨ "sample" x
- many possible outputs

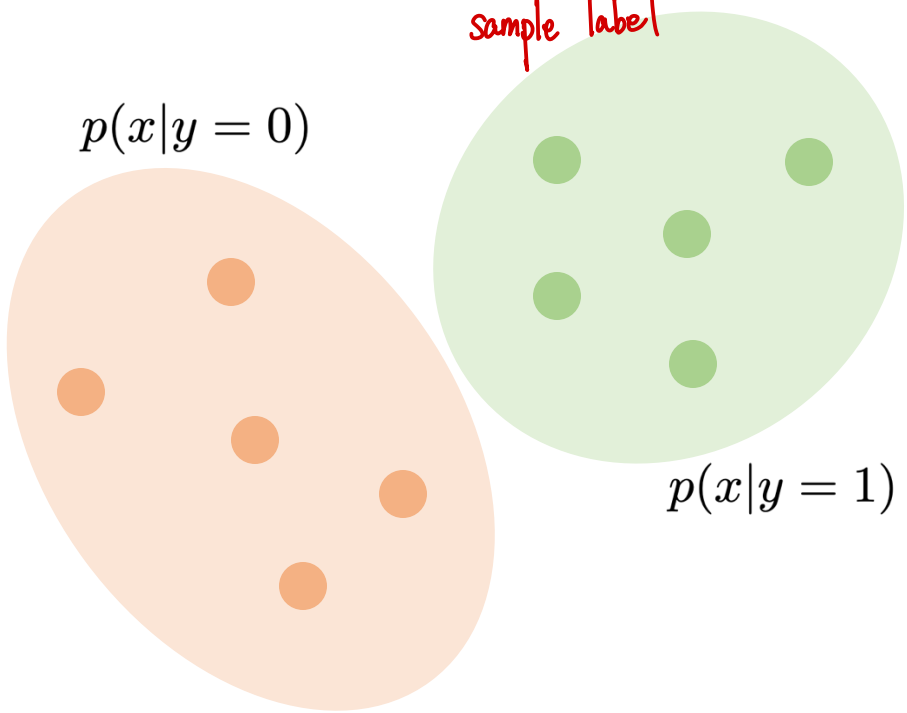


Discriminative Models vs. Generative Models

discriminative $p(y|x)$



generative $p(x|y)$



- Generative models can be discriminative: Bayes' rule
- Can discriminative models be generative?

- Generative models can be discriminative: Bayes' rule

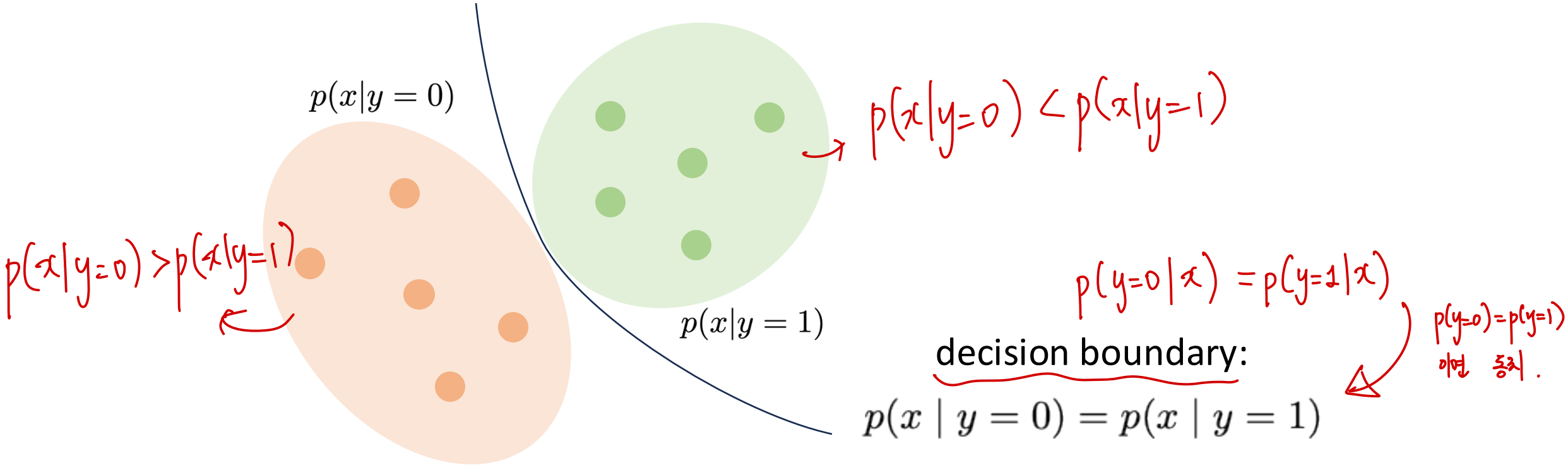
$$p(y|x) = p(x|y)\frac{p(y)}{p(x)}$$

assuming known prior (uniform categories)

constant for given x

discriminative

generative



- Generative models can be discriminative: Bayes' rule

$$p(y|x) = p(x|y)\frac{p(y)}{p(x)}$$

discriminative generative

assuming known prior (uniform categories)

constant for given x

- Can discriminative models be generative?

$$p(x|y) = p(y|x)\frac{p(x)}{p(y)}$$

generative discriminative

unknown prior distribution of x ("natural" images?)

constant for given y

discriminative : data → label : answer one way

generative : label → data : various ⇓ 한 label 당 data 분포 modeling.

- The challenge is on **probabilistic modeling**

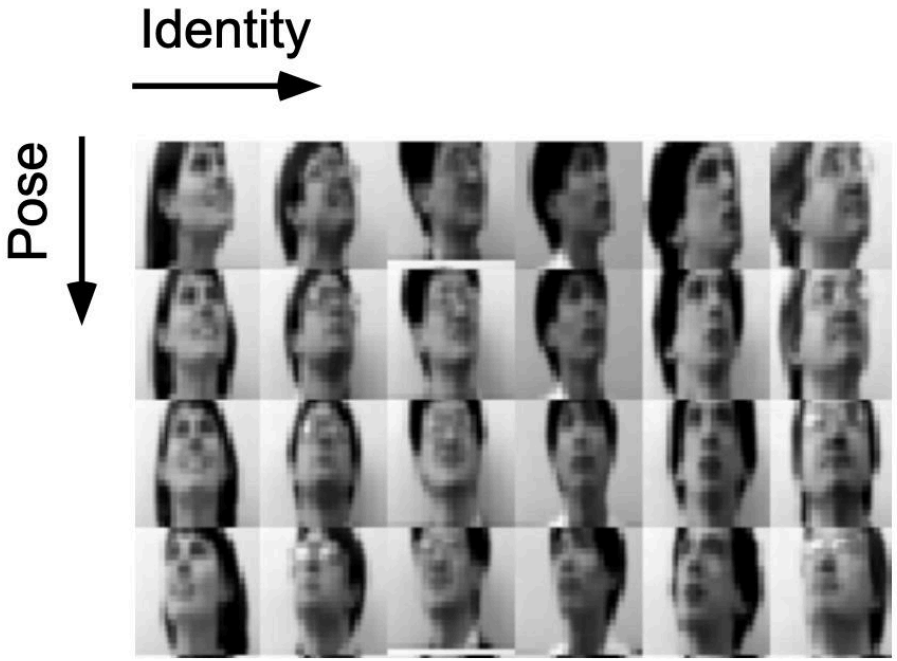
Probabilistic Modeling

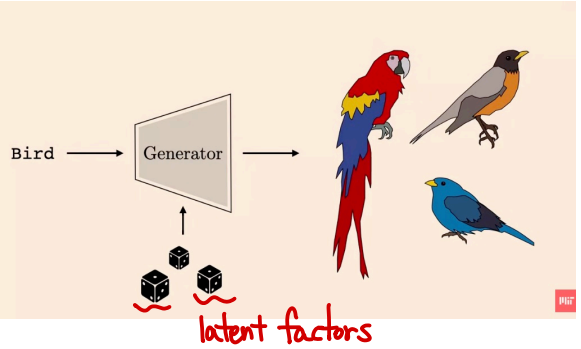
latent z $\xrightarrow{g(z)}$ x

- Where does probability come from?
- Assuming underlying **distributions of data generation process**

example:

- latent factors z (pose, lighting, scale, ...)
- z has simple distributions
- observations x are rendered by a graphics model, (a function on z)
- x has complex distributions



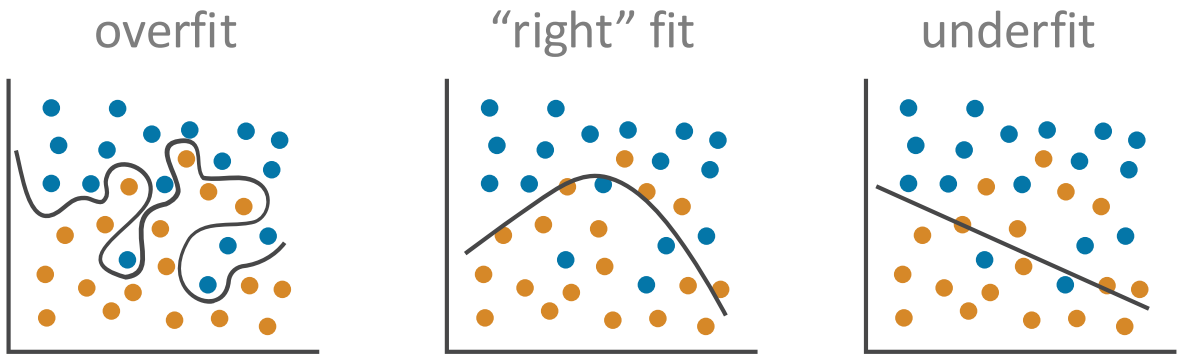


- Probability itself is a modeling assumption

Figure from: W. T. Freeman, J. B. Tenenbaum, “Learning Bilinear Models for Two-Factor Problems in Vision”, 1996

Probability itself is a modeling assumption

- There may be **no true "underlying" distribution** behind the data.
- Even if it exists, we can only observe **a finite sample set** of it.
- Models must **extrapolate** from observations to represent a distribution.
- Leads to **overfitting vs. underfitting** (just as in discriminative models)

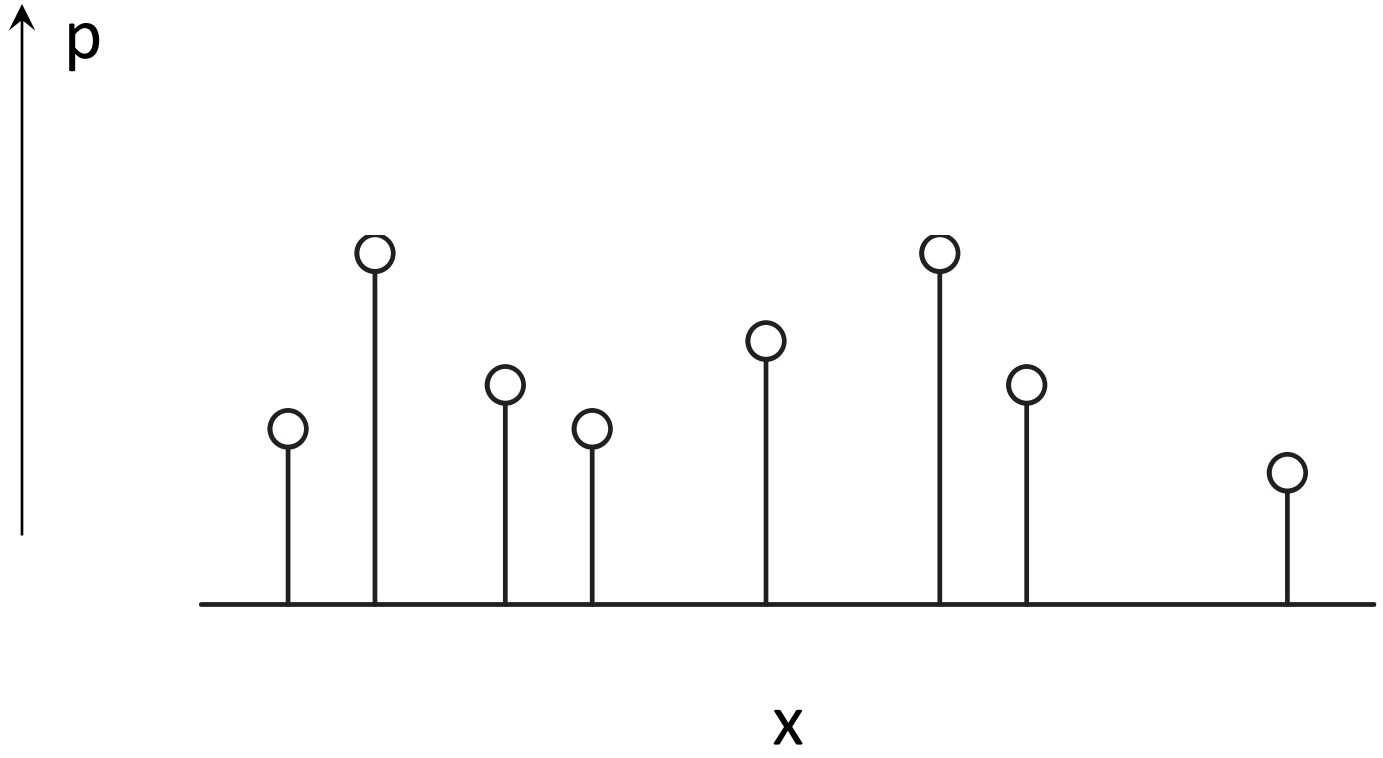


discriminative models

Figure credit: https://www.mathworks.com/discovery/overfitting.html

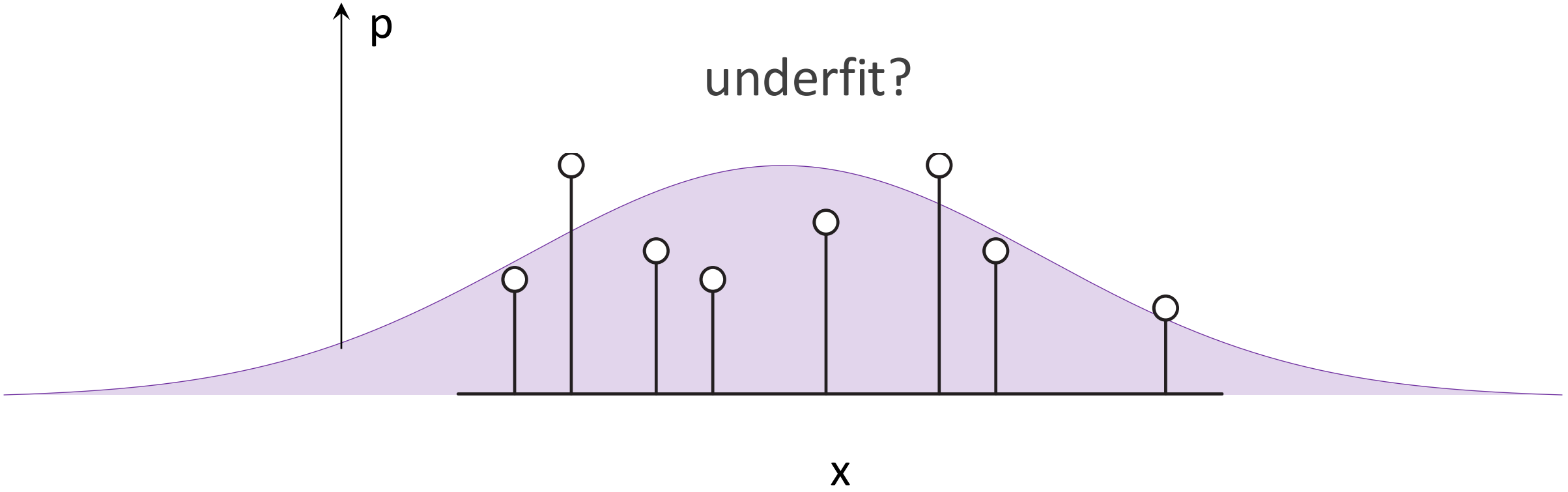
Probability itself is a modeling assumption

- We can only observe **a finite sample set** of an unknown distribution.



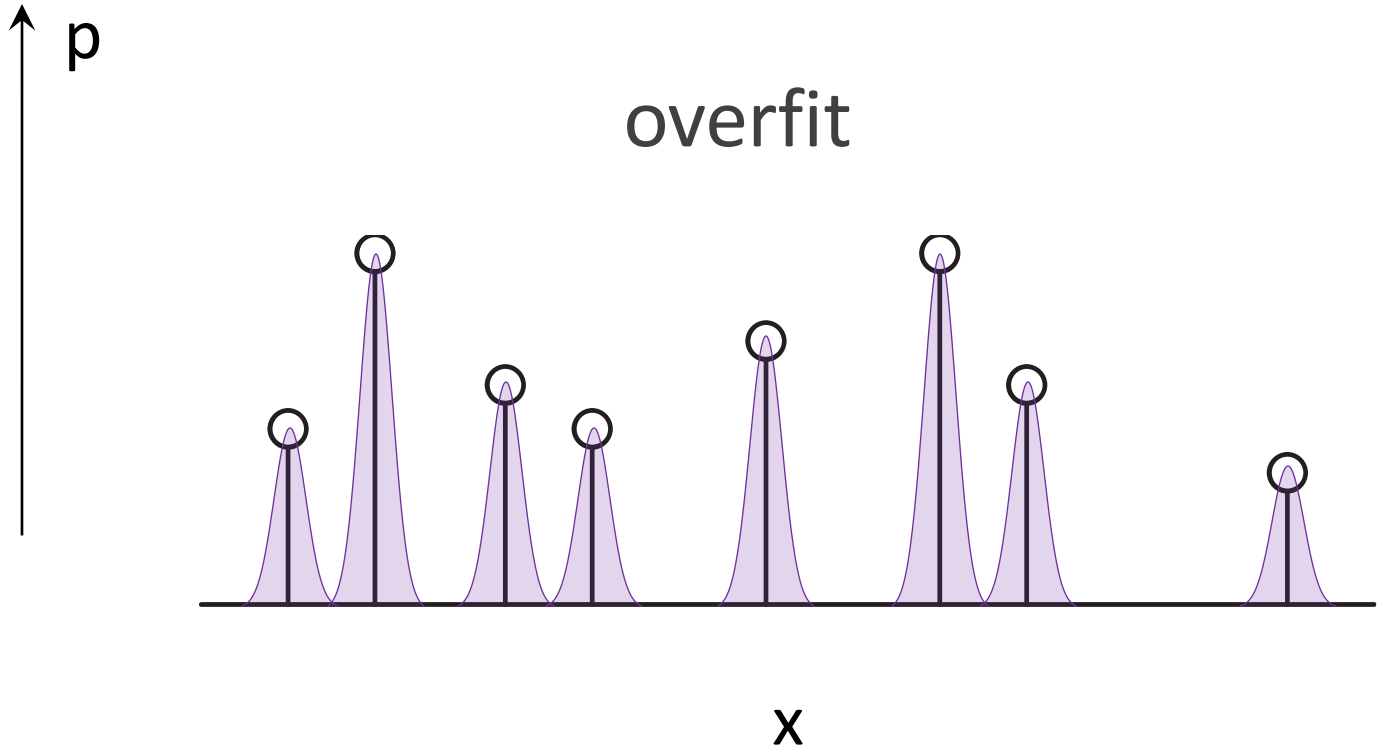
Probability itself is a modeling assumption

- **Underfitting**: the probabilistic model is too simple.



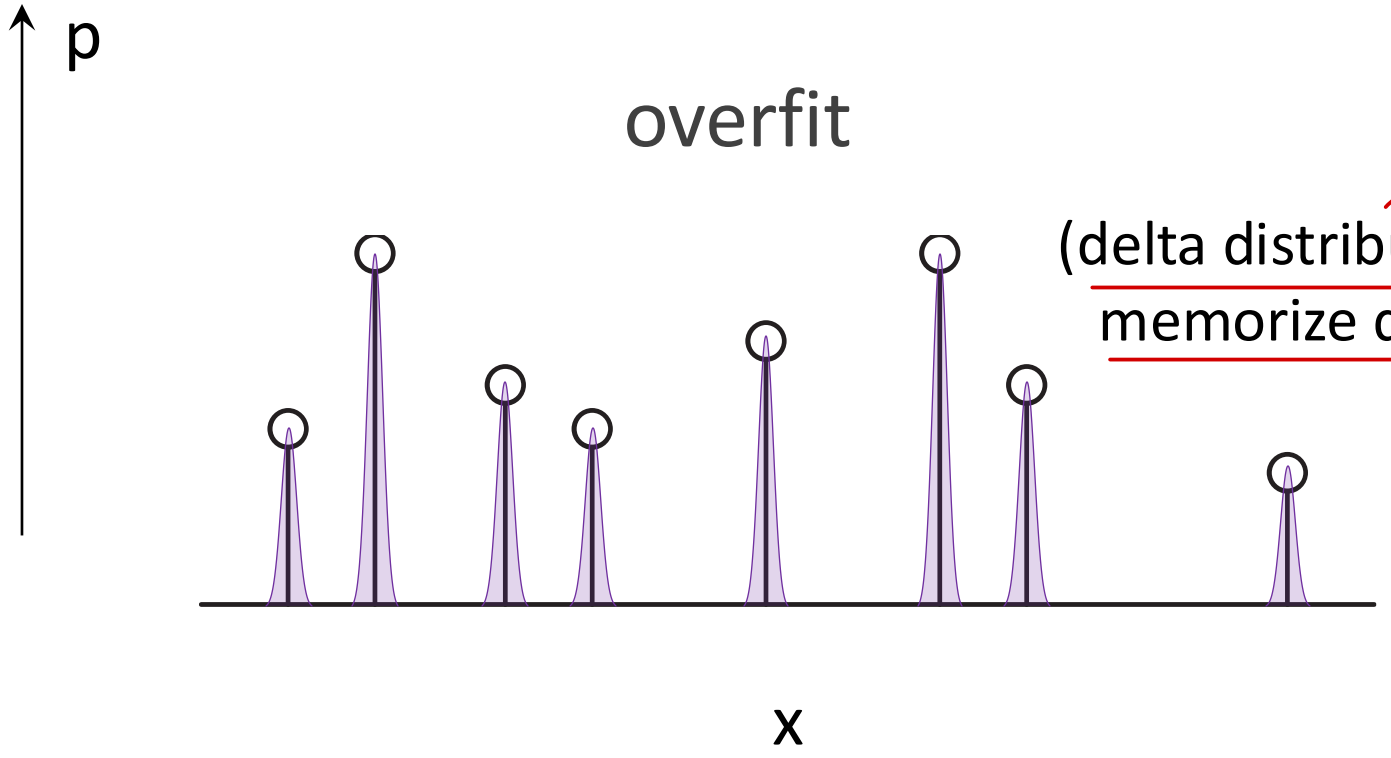
Probability itself is a modeling assumption

- **Overfitting**: the probabilistic model does not extrapolate/generalize



Probability itself is a modeling assumption

- **Overfitting**: the probabilistic model does not extrapolate/generalize



특정 값에 확률 집중이
focus.

Generative Modeling is Probabilistic Modeling

data



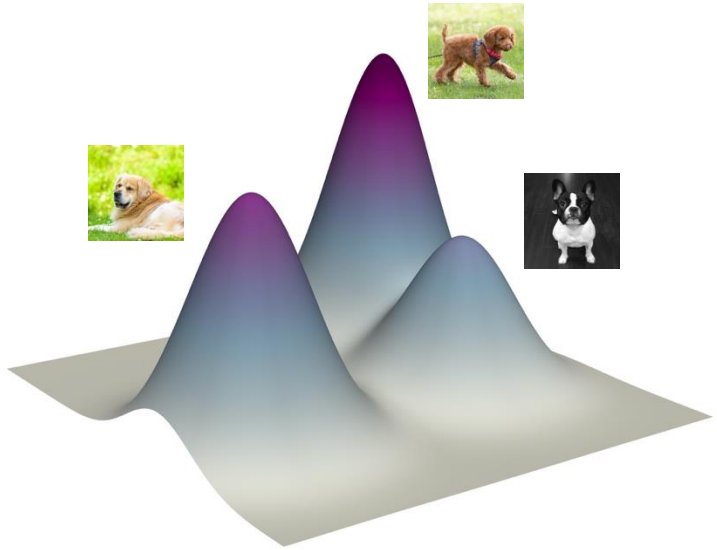




Generative Modeling is Probabilistic Modeling

data

data distribution

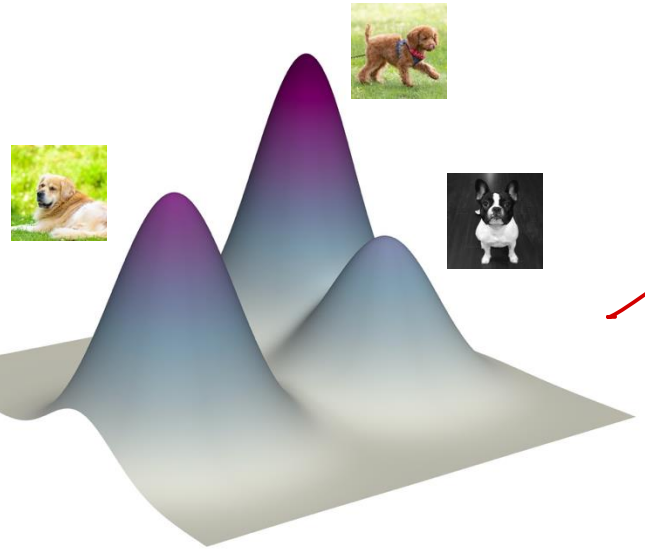


This in itself is part of modeling

data가 특정 분포를 따른다고 가정.

Generative Modeling is Probabilistic Modeling

data

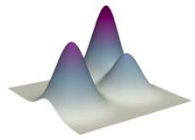
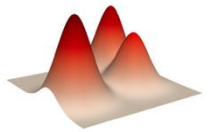


data distribution

$p_{data}(x)$

data 분포를 modeling.

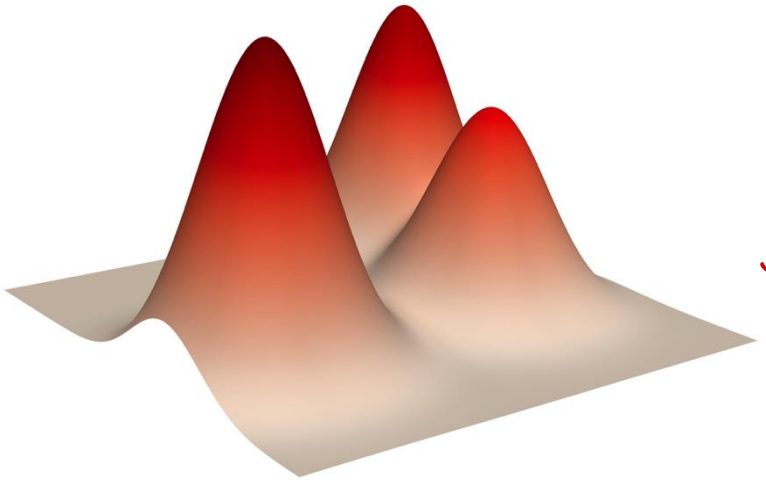
Optimize a loss function

$\mathcal{L}($  ,  $)$

Generative Model's

estimated distribution

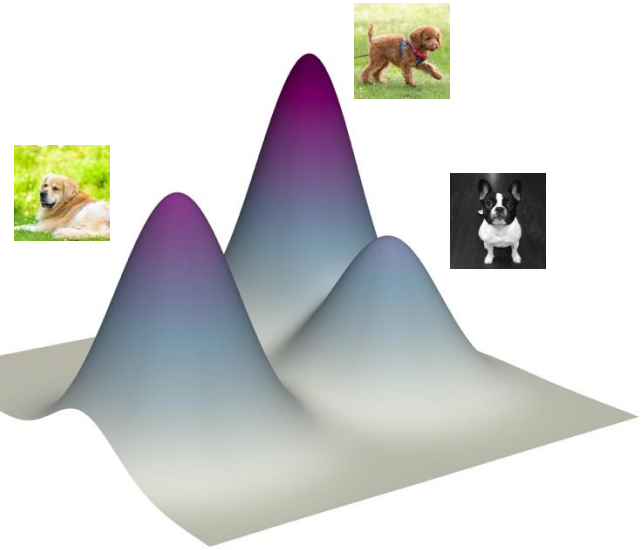
$p_\theta(x)$



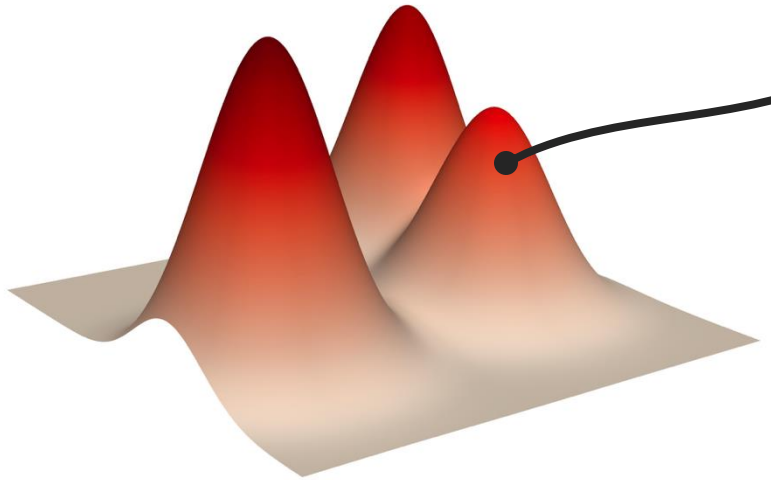
modeling한 분포를 학습.

Generative Modeling is Probabilistic Modeling

data



data distribution



estimated distribution



sample new data

$$x' \sim p(x|y)$$

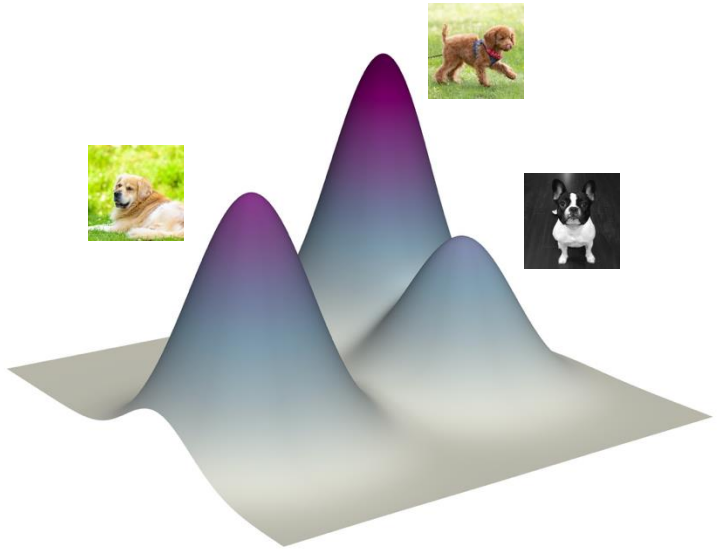
생성된 data 생성

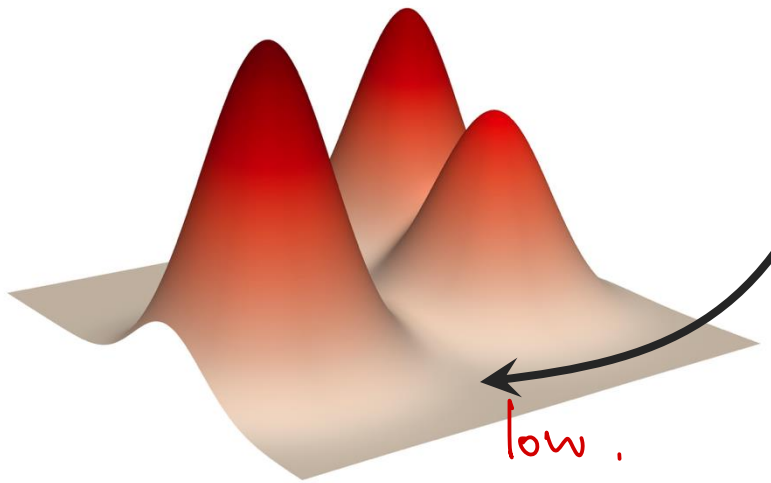
Generative Modeling is Probabilistic Modeling

data

data distribution

estimated distribution





low.



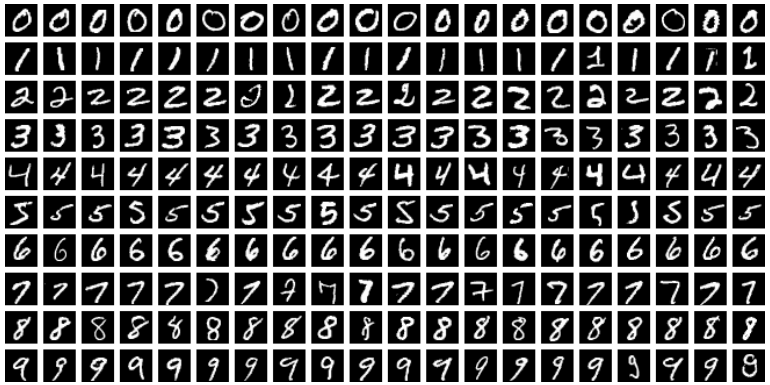
estimate prob density

$$p(x \mid y{=}\text{dog}) = ?$$

이 이미지가 dog과 얼마나 같은가

(⇒ 분류문제)

Case study: a minimalist generative model?

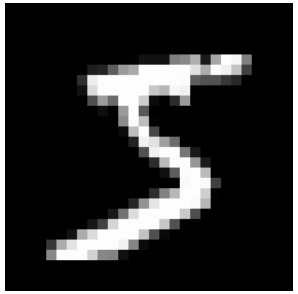


data set



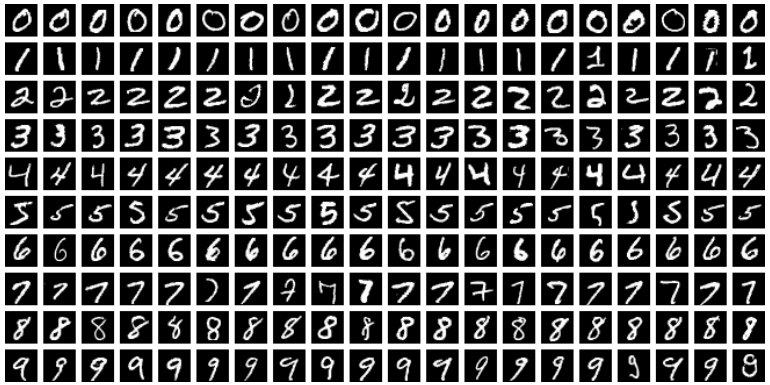
random index

retrieve



- Is this a generative model? Yes
- Is this a "good" generative model? No - not generalize?

Case study: a minimalist generative model?



data set



random index

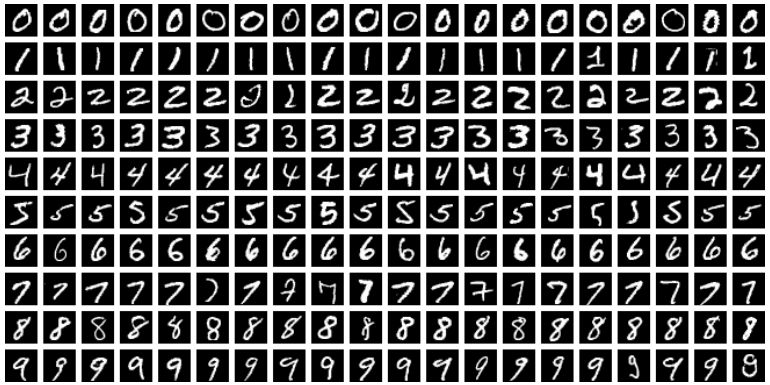
retrieve
+ random noise



- Is this a new sample? Yes
- Is this a “good” new sample? No - not “within distribution”?

분포 안에 포함이 안될수있음

Case study: a minimalist generative model?

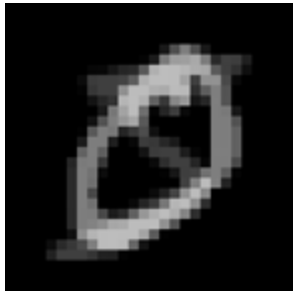


data set



2 random indexes

retrieve and merge



pixel

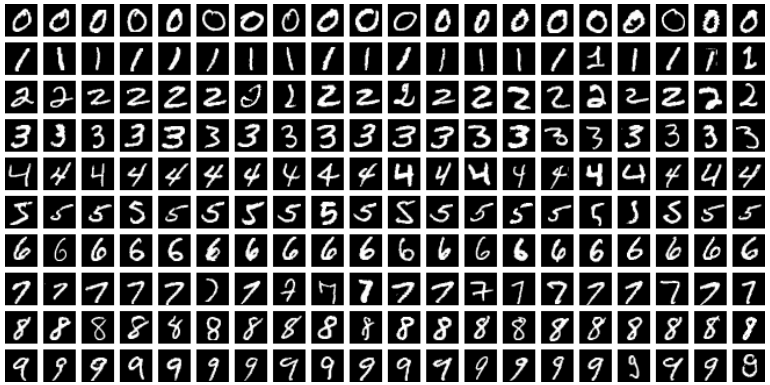
$$\left(\frac{\text{image 1} + \text{image 2}}{2}\right)$$

$$\neq$$

value (")

- Is this a new sample? Yes
- Is this a "good" new sample? No - not "within distribution"?

Case study: a minimalist generative model?



data set



random index

retrieve and **colorize**



- Is this a new sample? Yes
- Is this a "good" new sample? No? It depends.

목적에 따라 달라짐.

ex) goal = MNIST분포에서, out of distribution.

goal = colorful한 숫자 그림 ⇒ good sample.

Generative Modeling is Probabilistic Modeling

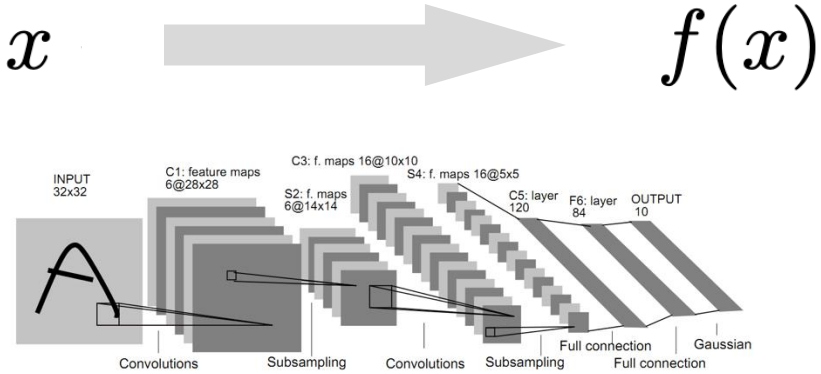
Notes:

- Generative models involve probabilistic models hypothesized by humans.
- Probabilistic modeling is not just the work of neural nets (even in the context of deep learning).
- Probabilistic modeling is a popular way, but not the only way.
- "*All models are wrong, but some are useful.*"*

* Quote George Box

Deep Generative Models

- Deep learning is **representation learning**
- **Discriminative**: represent data **instances**
  - one data point: $x \rightarrow f_\theta(x)$
  - loss w.r.t. one label: $\mathcal{L}(y, f_\theta(x))$

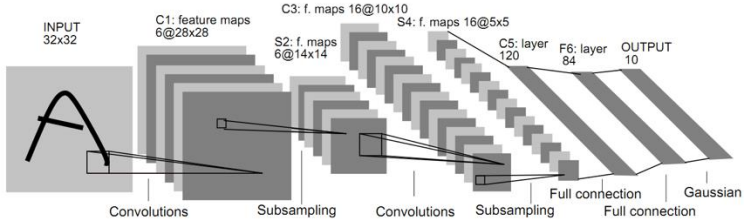


Deep Generative Models

- Deep learning is **representation learning**
- **Discriminative**: represent data **instances**
  - one data point: $x \rightarrow f_\theta(x)$
  - loss w.r.t. one label: $\mathcal{L}(y, f_\theta(x))$

discriminative

x → $f(x)$



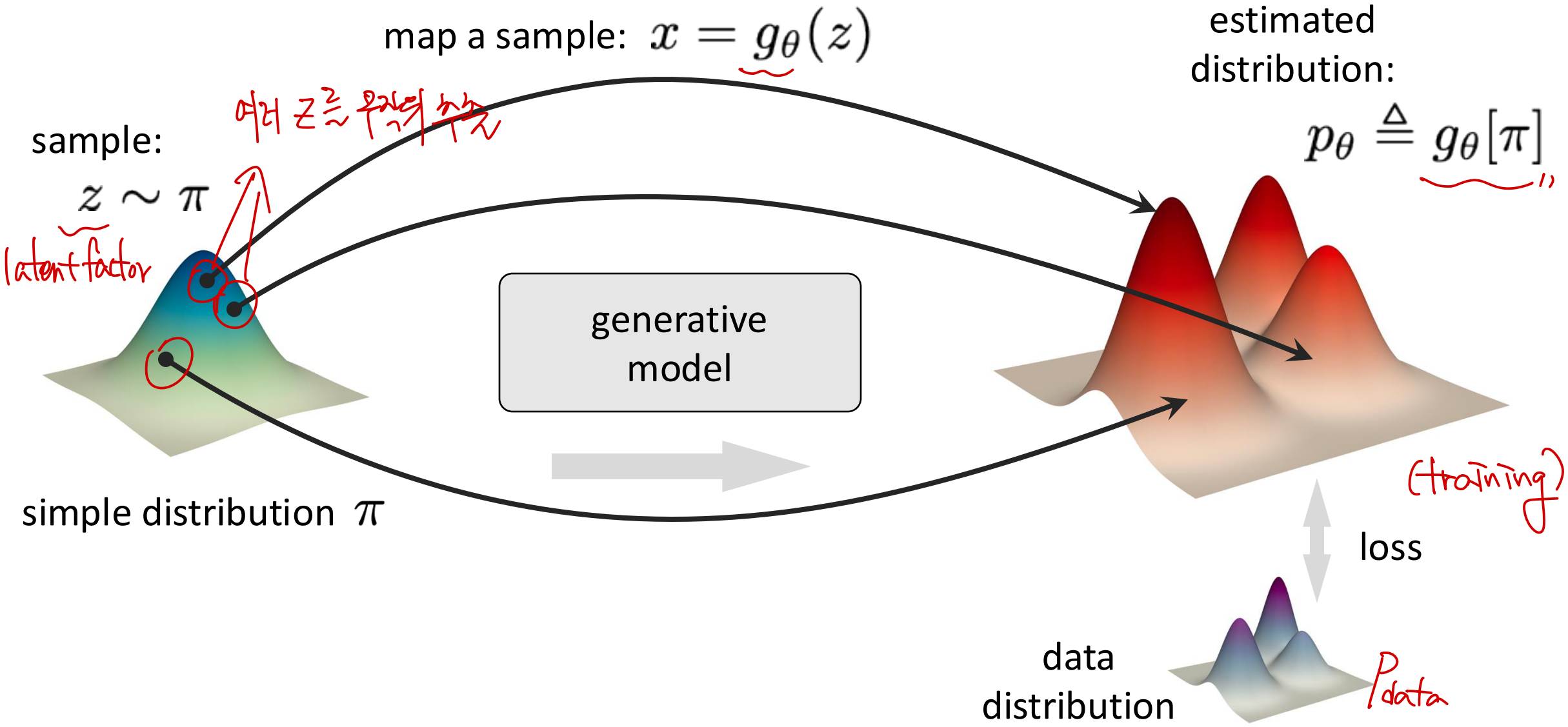
$g_\theta[\pi]$ ← π

Generative

- **Generative**: represent data **distribution**
  - distribution-to-distribution mapping: $\pi \rightarrow g_\theta[\pi]$ (example: π is Gaussian)
  - loss w.r.t. true (but known) distribution: $\mathcal{L}(p_{\text{data}}, g_\theta[\pi])$

sample data distribution

Learning to Represent Distributions



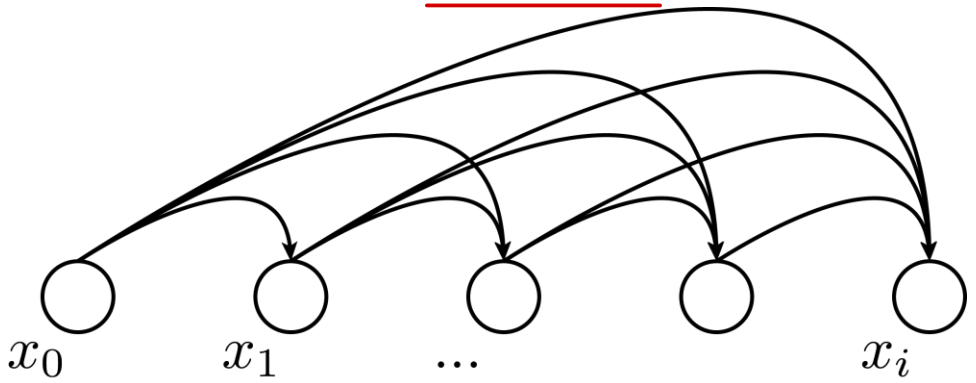
Learning to Represent Distributions

- Not all parts of a generative model are learned.

밀도함수 자체는 design
→ 확률은 학습

Case study:
Autoregressive model

This dependency structure is designed (not learned).

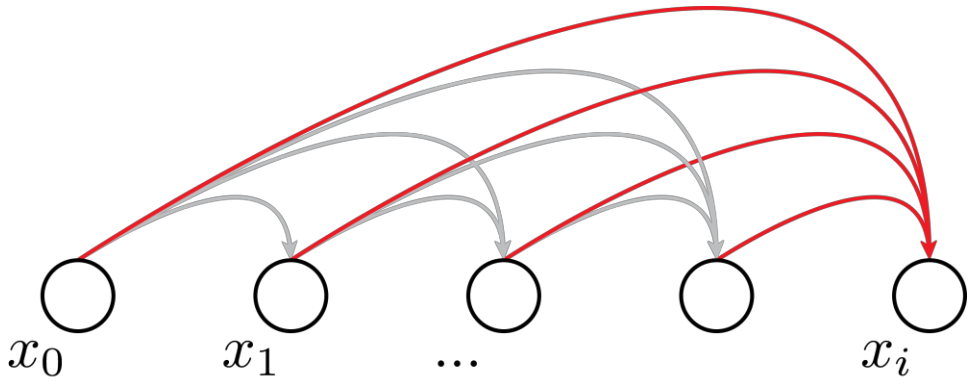


Learning to Represent Distributions

- Not all parts of a generative model are learned.

Case study:
Autoregressive model

This probability is learned



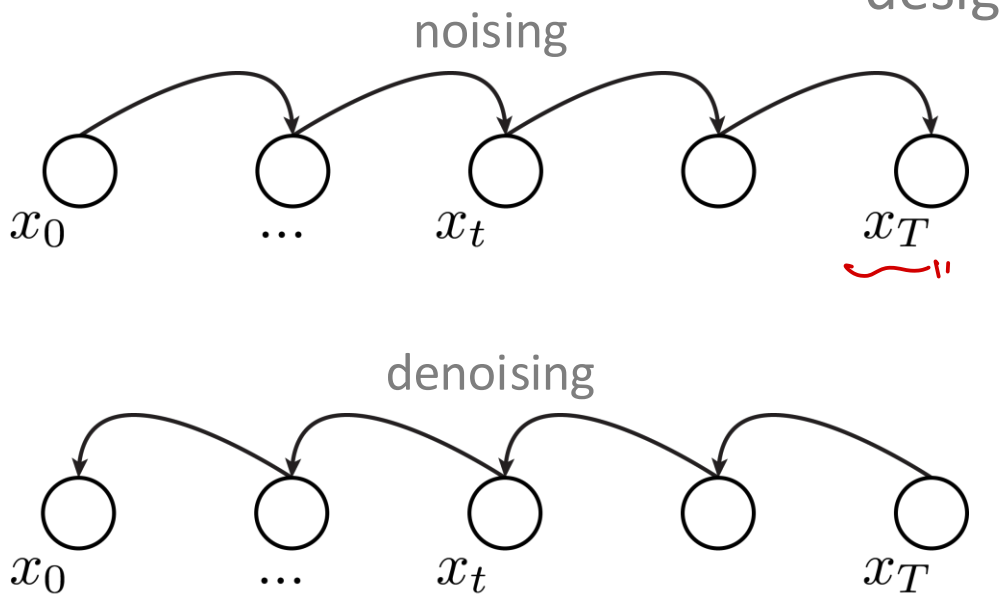
$$p(x_i \mid x_0, x_1, \ldots)$$

Learning to Represent Distributions

- Not all parts of a generative model are learned.

Case study:
Diffusion model

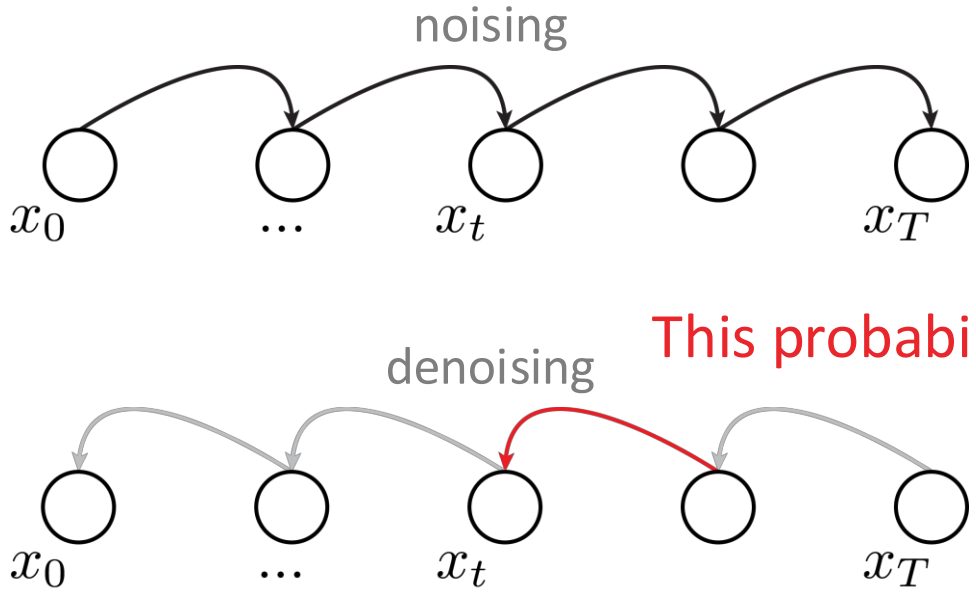
This noising process is designed (not learned).



Learning to Represent Distributions

- Not all parts of a generative model are learned.

Case study:
Diffusion model



This probability is learned

$p_\theta(x_{t-1} \mid x_t)$

노이즈로부터 이전 단계의 샘플을
(노이즈가 제거되는) 얻을 확률

Objective Functions in Generative Models

- The objective is a *functional*: it operates on functions



a functional

a function

$p_{\text{data}}(x)$

input data distribution

another function

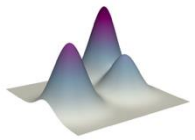
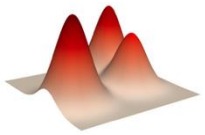
$p_\theta(x)$

estimated

→ 분포는 비교할 loss function 필요.

Objective Functions in Generative Models

- Case study: Kullback–Leibler (KL) Divergence

$\mathcal{L}($  ,  $)$

x is a dummy variable

$$\mathcal{D}_{\mathrm{KL}}(p_{\mathrm{data}} \,||\, p_\theta) \triangleq \mathbb{E}_{x\sim p_{\mathrm{data}}}\left[\log \frac{p_{\mathrm{data}}(x)}{p_\theta(x)}\right]$$

① p_{data} 로부터 x 샘플

② 값 비교.

$\left(p_{data}(x) = p_\theta(x),\ \log\frac{p_{data}(x)}{p_\theta(x)} = 0\right)$

③ 모든 x에 대한 $\log\frac{p_{data}}{p_\theta}$ 평균.

$\mathcal{D}$'s inputs: $p_\theta, p_{\mathrm{data}}$

Objective Functions in Generative Models

- Likelihood: $-\mathbb{E}_{p_{\text{data}}}[\log p_\theta(x)]$ — MLE, VAE, Normalizing Flows

- Divergence: $\mathcal{D}_{\text{KL}}(p_{\text{data}} \,||\, p_\theta)$

- Adversarial: $\mathcal{D}_f(p_{\text{data}},\ p_\theta)$ — GAN

- Score-based: $\mathbb{E}_{p_{\text{data}}} \|\nabla_x \log p_{\text{data}} - \nabla_x \log p_\theta\|^2$ — Diffusion, Energy-based, ...

same

KL divergence와 likelihood의 관계

식을 분리하면 다음과 같습니다.

$$D_{\text{KL}}(p_{\text{data}} \,\|\, p_\theta) = \mathbb{E}_{p_{\text{data}}}[\log p_{\text{data}}(x)] - \mathbb{E}_{p_{\text{data}}}[\log p_\theta(x)]$$

첫 번째 항은 θ와 관계없는 상수입니다. 따라서 KL divergence를 최소화하는 것은 다음 negative log-likelihood를 최소화하는 것과 같은 최적해를 가집니다.

$$-\mathbb{E}_{x\sim p_{\text{data}}}[\log p_\theta(x)]$$

실제로는 p_{data}의 수식을 모르므로 학습 데이터로 이를 근사합니다.

$$-\frac{1}{N}\sum_{i=1}^{N} \log p_\theta(x_i)$$

discriminator

gradient의 차이 (density가 변화하는 방향의 차이)



What makes the objectives difficult to design?

- Analytical formulations
  - Only for limited families (Gaussian, uniform, categorical, ...)
  - Low-dimensional
  - Example: Gaussian => L2 loss

simple distribution

(고차원에서 복잡성↑, 계산 X)

↳ $\Sigma(x-\hat{x})^2$

- Monte Carlo estimation
  - Distributions unknown in analytical form
  - We can only sample from them → 표본이 유한함. (추정값에 오차와 변동이 있음)
  - Example: $x, y \sim p_{\text{data}}(x, y)$

- Divergence measures
  - often not directly computable (a surrogate is used)
  - Example: GAN

대안

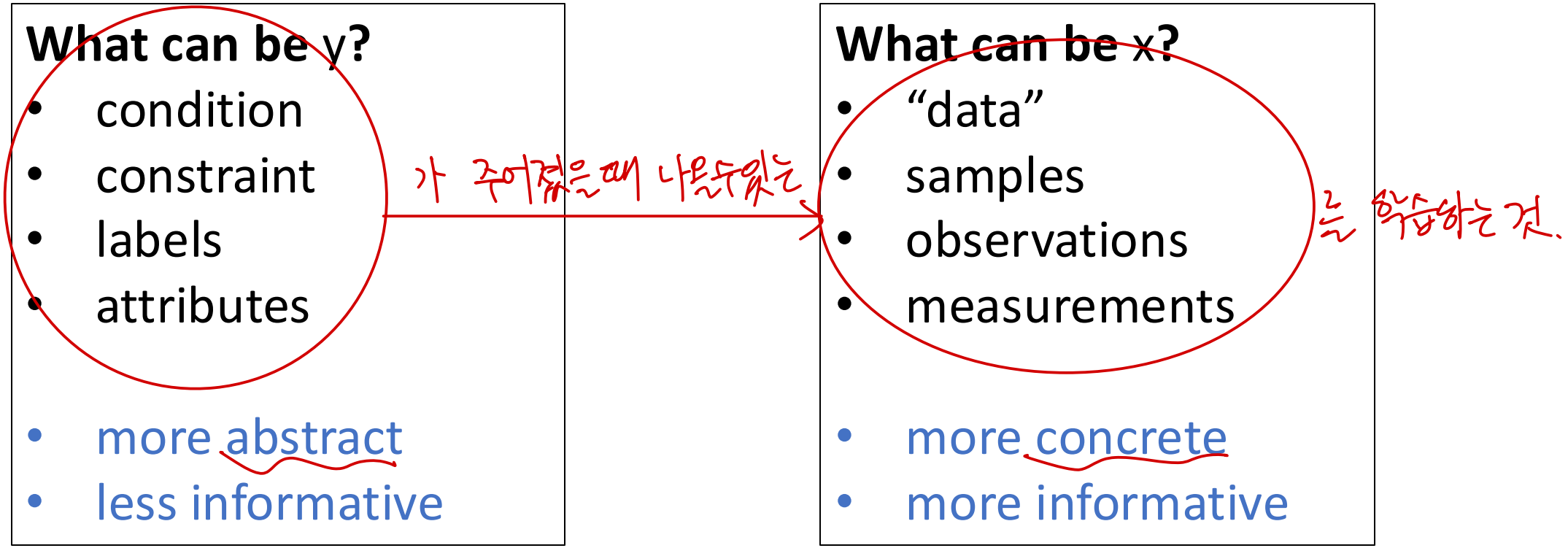
Elements of Deep Generative Models

- **Formulation:** a real-world problem => probabilistic modeling
  - designed structures
  - learnable components
- **Representation**: use deep neural nets to represent distributions
- **Objectives**: discrepancy between distributions
- **Optimization**: differentiable, computable, and tractable
- **Inference**:
  - sampler: able to produce new samples
  - density estimator: (optional) able to evaluate $p(x)$

Generative Modeling of Real-world Problems

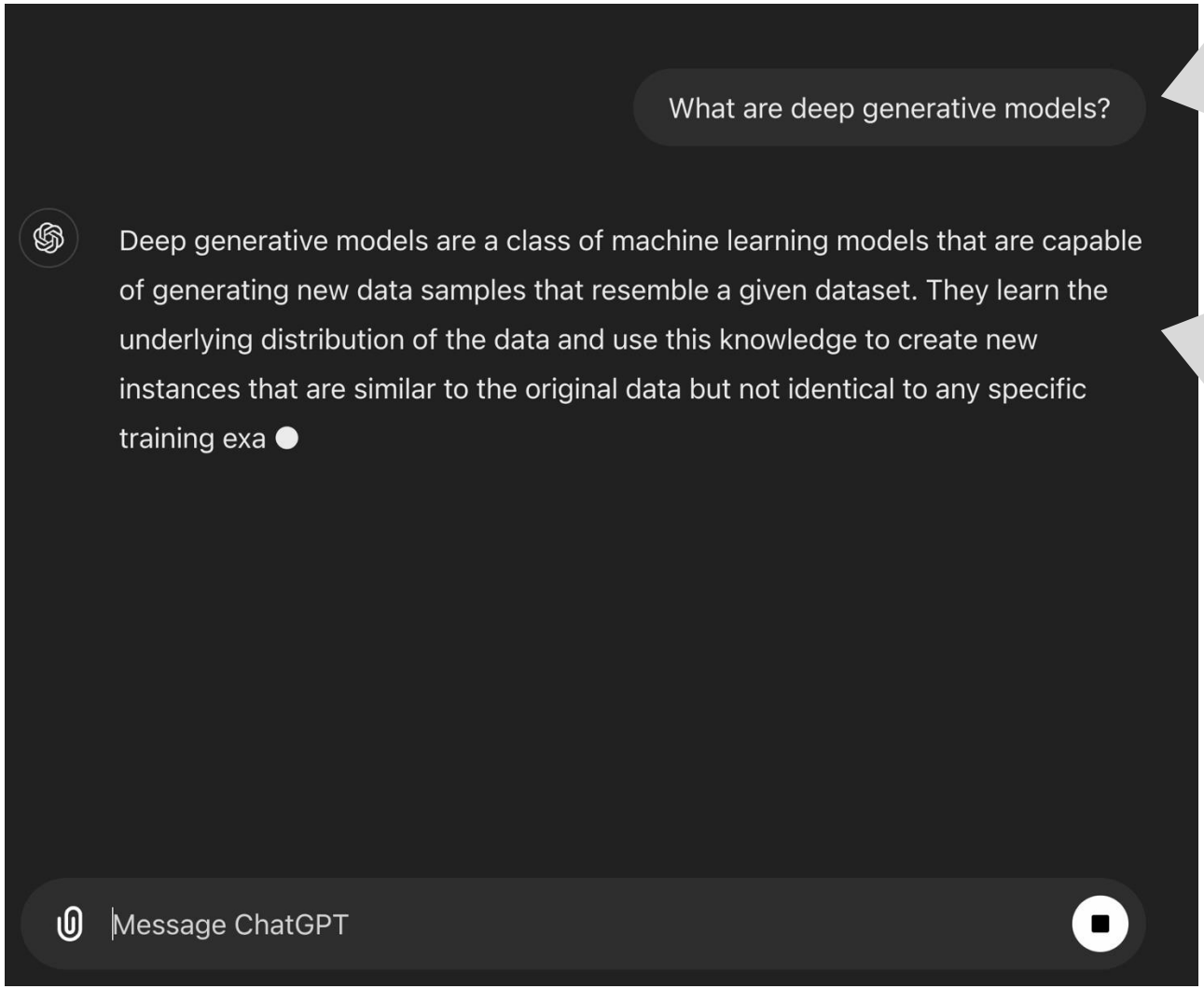
Generative Modeling of Real-world Problems

- Generative models are about $p(x|y)$



Generative Modeling as p(x|y): Case Study

- **Natural language conversation**



y: prompt

x: response of the chatbot

Generative Modeling as p(x|y): Case Study

- **Text-to-image/video generation**

Prompt: teddy bear teaching a course, with "generative models" written on blackboard

y: text prompt



x: generated visual content

Image generated by Stable Diffusion 3 Medium

Generative Modeling as p(x|y): Case Study

- **Text-to-3D structure generation**



"motorcycle"



"mech suit"



"ghost lantern"



"furry fox head"

x: generated 3D structures

y: text prompt



"dresser"



"swivel chair"



"astronaut"



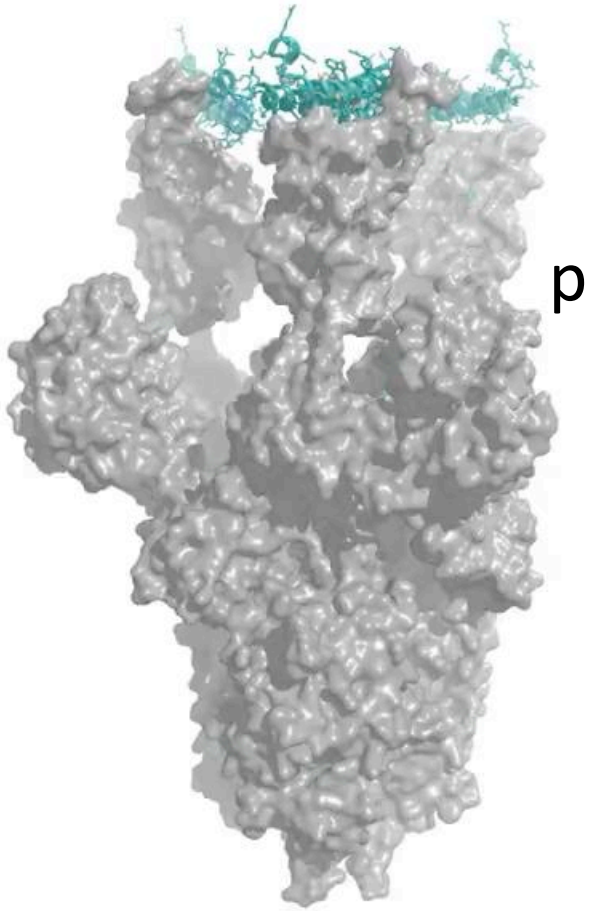
"mushroom house"

Figure credit: Tang, et al. LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation. ECCV 2024

Generative Modeling as p(x|y): Case Study

- **Protein structure generation**

y: condition/constraint
(e.g., symmetry)



x: generated
protein structures

Watson, et al. De novo design of protein structure and function with RFdiffusion, Nature 2023

Generative Modeling as p(x|y): Case Study

- **Class-conditional image generation**

“red fox”



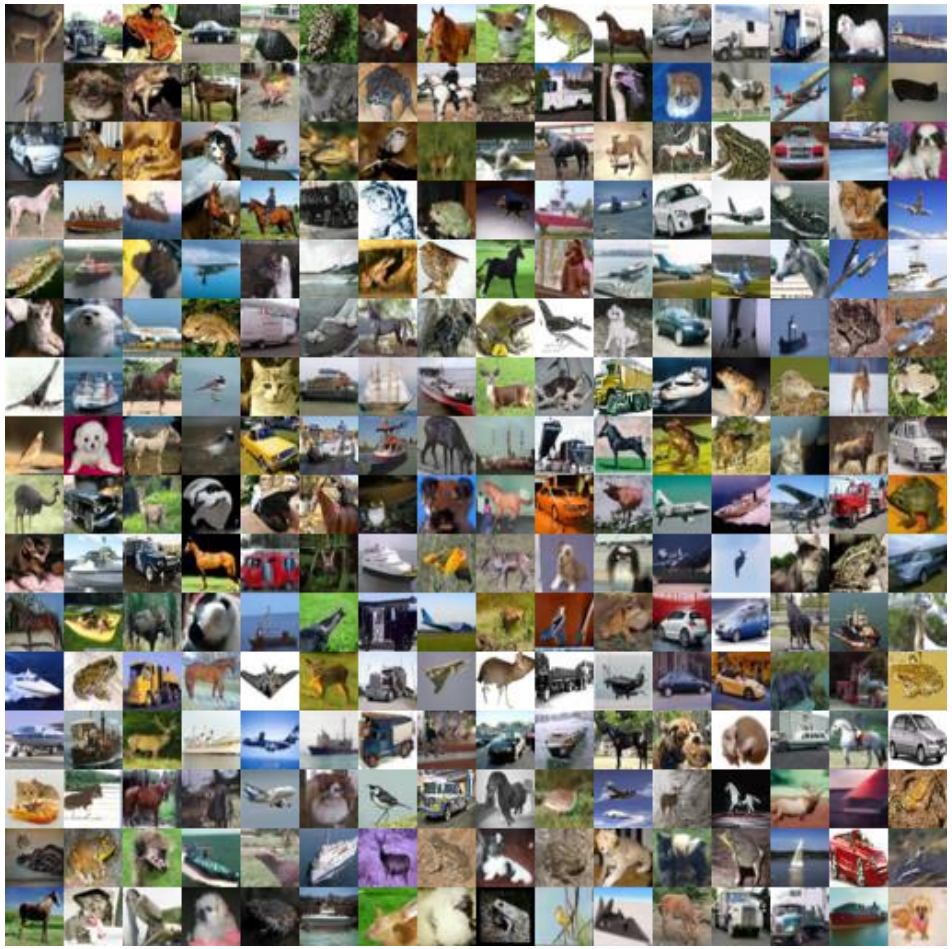
y: class label

x: generated image

Image generated by: Li, et al. Autoregressive Image Generation without Vector Quantization, 2024

Generative Modeling as p(x|y): Case Study

- **"Unconditional" image generation**



y: an implicit condition

"images following CIFAR10 distribution"

x: generated CIFAR10-like images

- p(x|y): images ~ CIFAR10
- p(x): all images

Images generated by: Karras, et al. Elucidating the Design Space of Diffusion-Based Generative Models, NeurIPS 2022

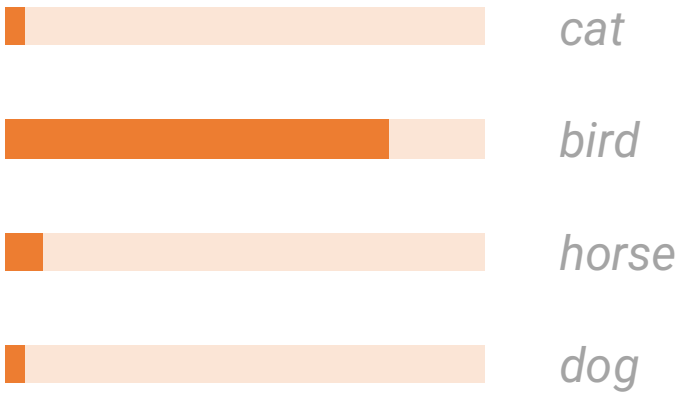
Generative Modeling as p(x|y): Case Study

- **Classification** (often not viewed as generative)

y: an image as the "condition"



x: probability of classes conditioned on the image



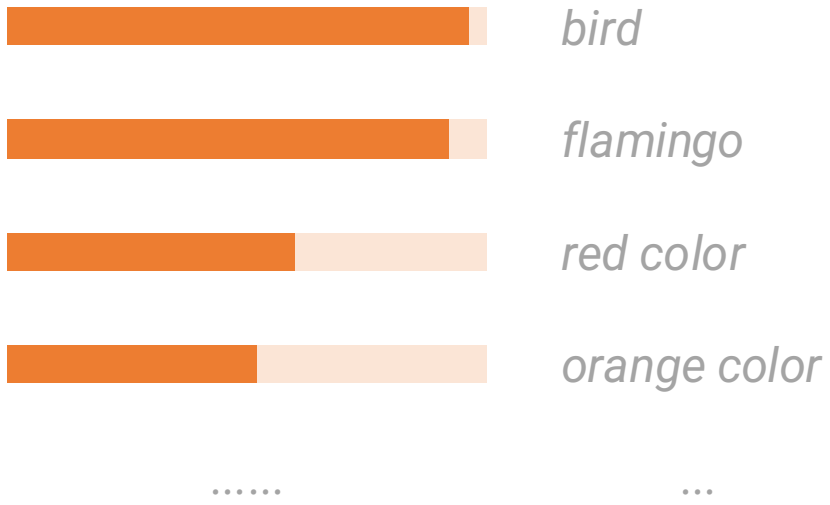
Generative Modeling as p(x|y): Case Study

- **Open-vocabulary recognition**

y: an image as the “condition”



x: plausible descriptions conditioned on the image



Generative Modeling as p(x|y): Case Study

- **Image captioning**

y: an image as the “condition”



x: plausible descriptions conditioned on the image

```
a baseball player with a catcher and umpire on top of a baseball field.
a baseball player is sliding into a base.
a baseball player swings at a pitch with the pitcher and umpire behind him.
baseball player with bat in the baseball game.
a batter in the process on the bat in a baseball game.
```

figure credit: https://github.com/GoogleCloudPlatform/asl-ml-immersion/blob/master/notebooks/multi_modal/solutions/image_captioning.ipynb

Generative Modeling as p(x|y): Case Study

- **Visual dialogue**

User What is unusual about this image?



Source: https://www.barnorama.com/wp-content/uploads/2016/12/03-Confusing-Pictures.jpg

GPT-4 The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

y: image and text prompt

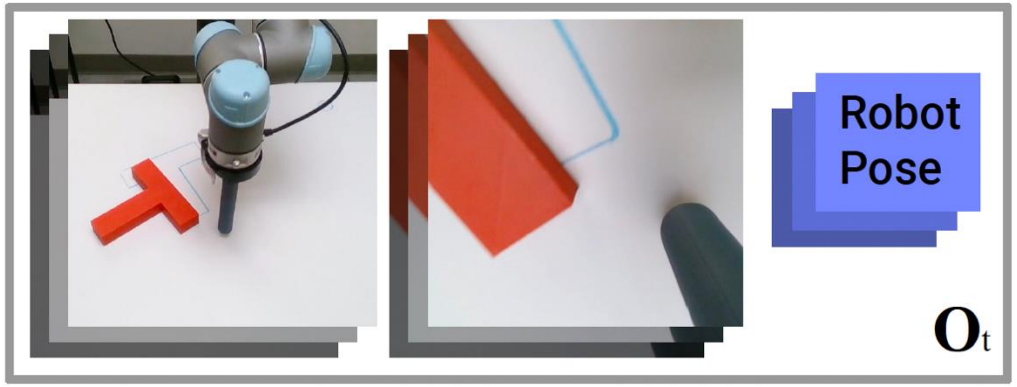
x: response of the chatbot

Figure from: GPT-4 Technical Report, 2023

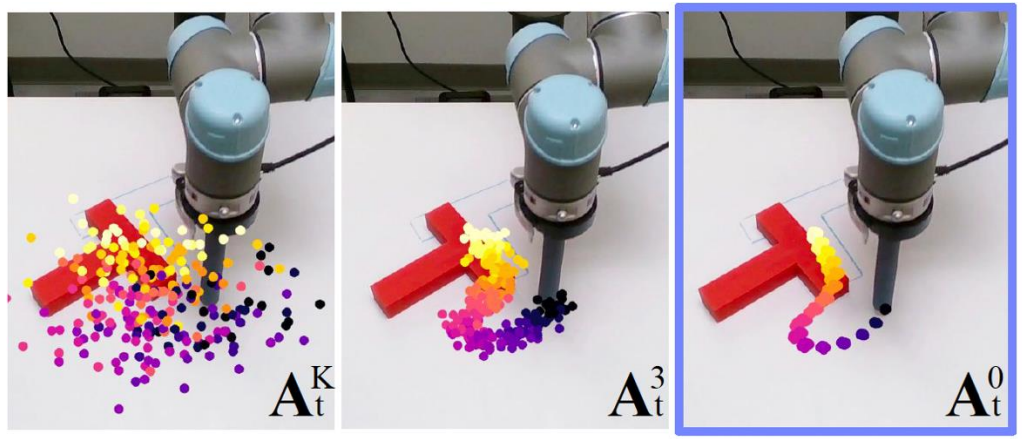
Generative Modeling as p(x|y): Case Study

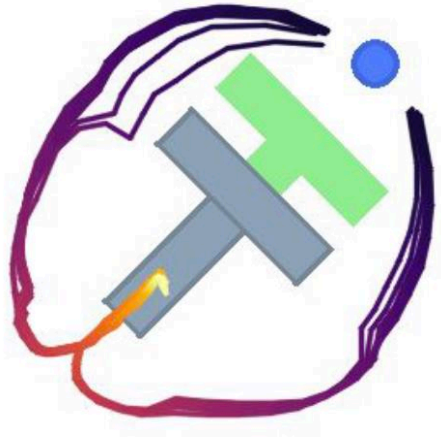
- **Policy Learning in Robotics**

y: visual and other sensory observations



x: policies (probability of actions)





Chi, et al. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion, RSS 2023

Generative Modeling of Real-world Problems

- Generative models are about $p(x|y)$

What can be y?

- condition
- constraint
- labels
- attributes

- more abstract
- less informative

What can be x?

- "data"
- samples
- observations
- measurements

- more concrete
- more informative

Generative Modeling of Real-world Problems

- Generative models are about $p(x|y)$

What can be y?

- ~~condition~~
- ~~constraint~~
- ~~labels~~
- ~~attributes~~

- ~~more abstract~~
- ~~less informative~~

What can be x?

- ~~"data"~~
- ~~samples~~
- ~~observations~~
- ~~measurements~~

- ~~more concrete~~
- ~~more informative~~

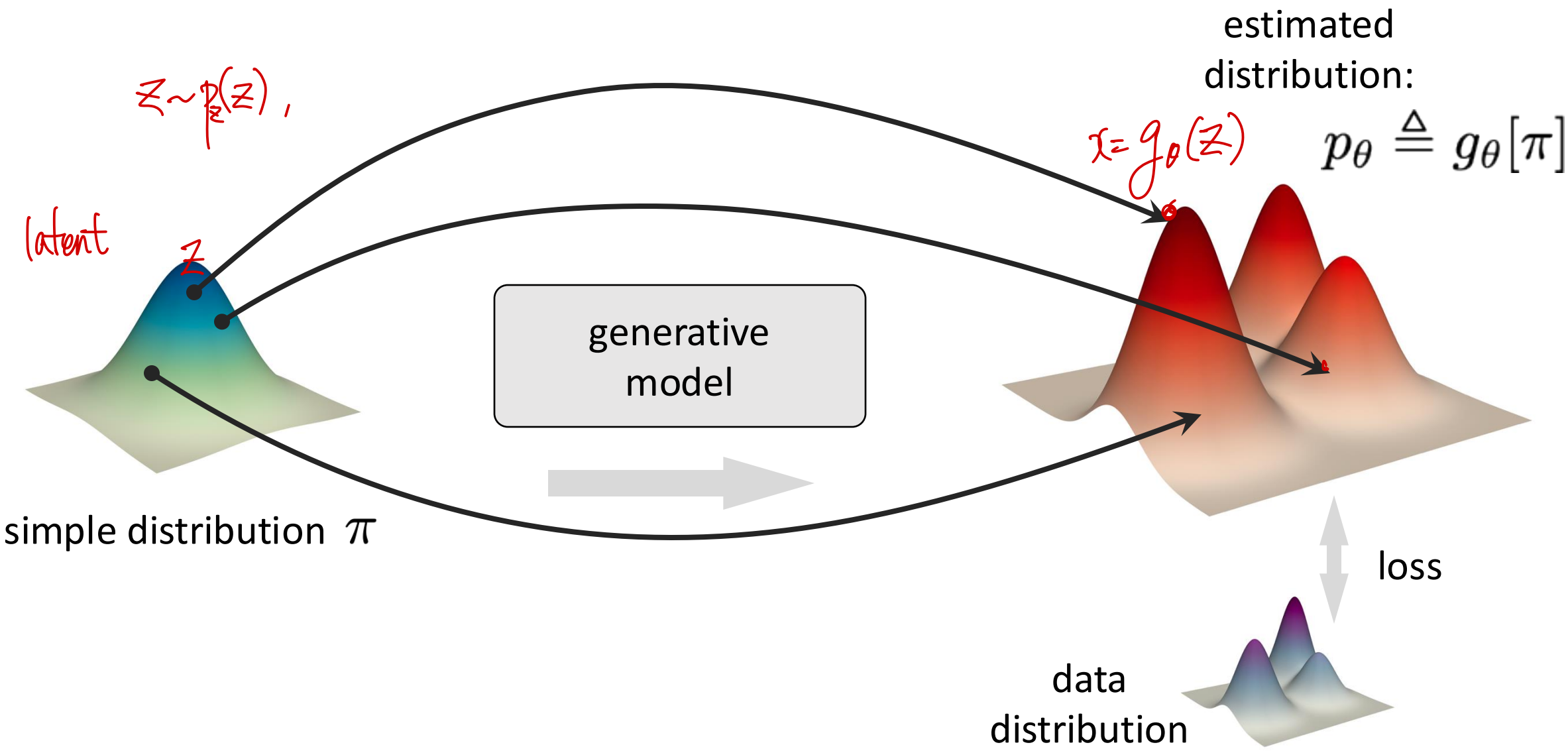
Generative Modeling of Real-world Problems

- Generative models are about $p(x|y)$ given y, model possible x

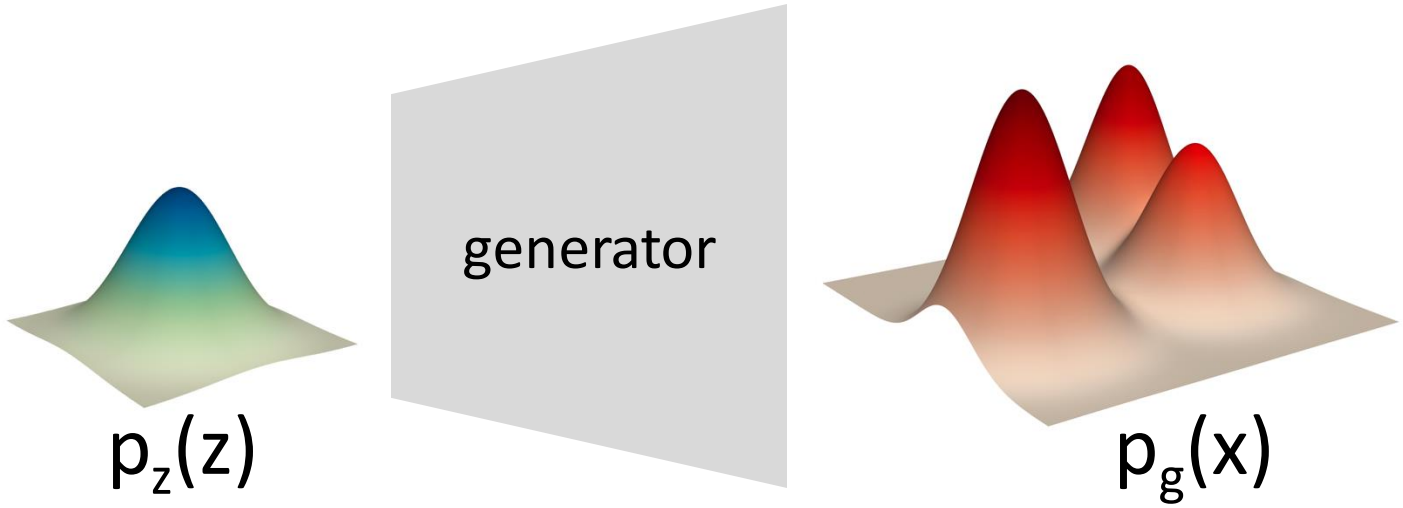
- There is no conceptual obstacle to formulating any real-world problem from a generative perspective.
- Generative modeling is a way of problem-solving.

Families of Generative Models

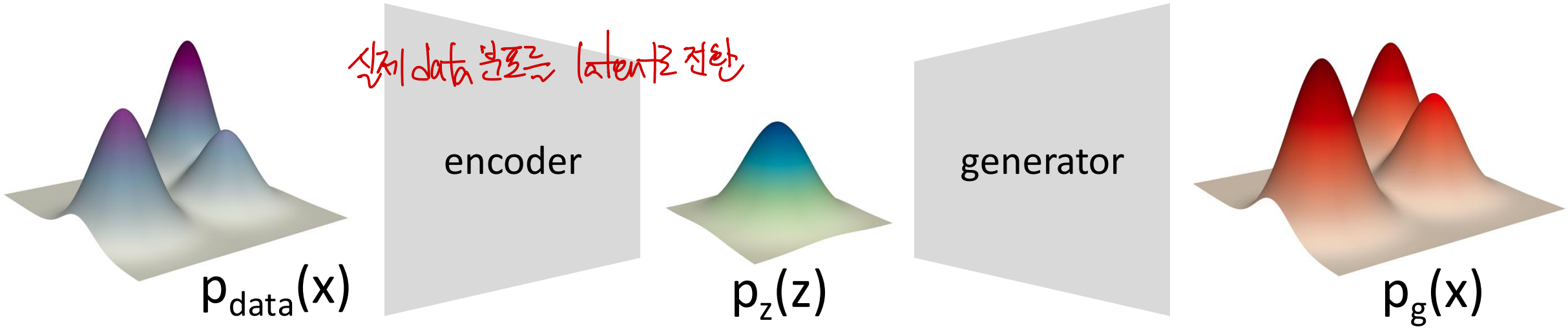
[Recap] Learning to Represent Distributions



[Recap] Learning to Represent Distributions

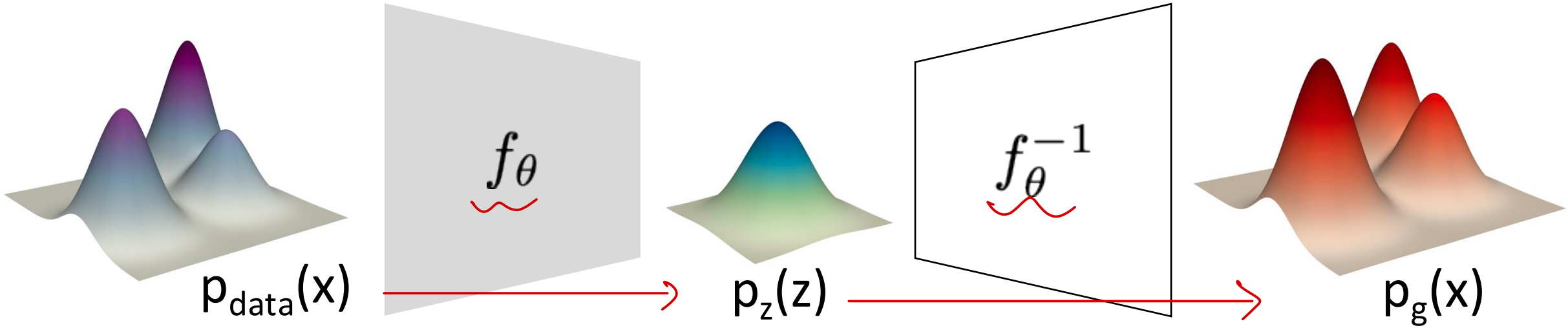


Variational Autoencoder (VAE)



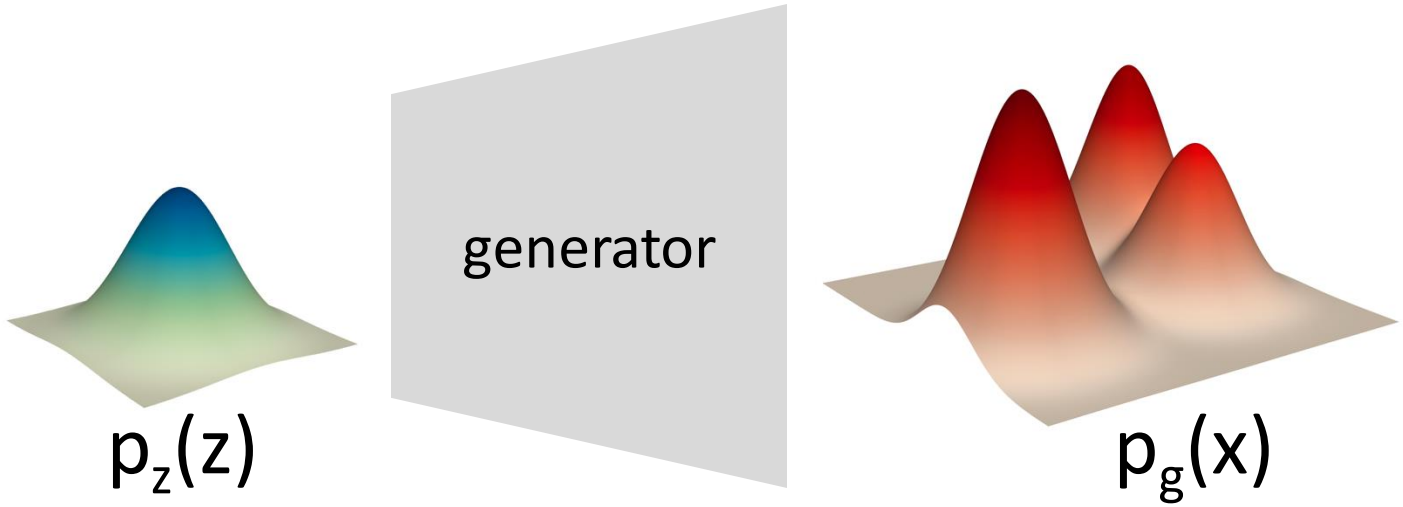
Normalizing Flows

encoder와 generator는 가역함수로 사용. → exact likelihood는 계산가능.

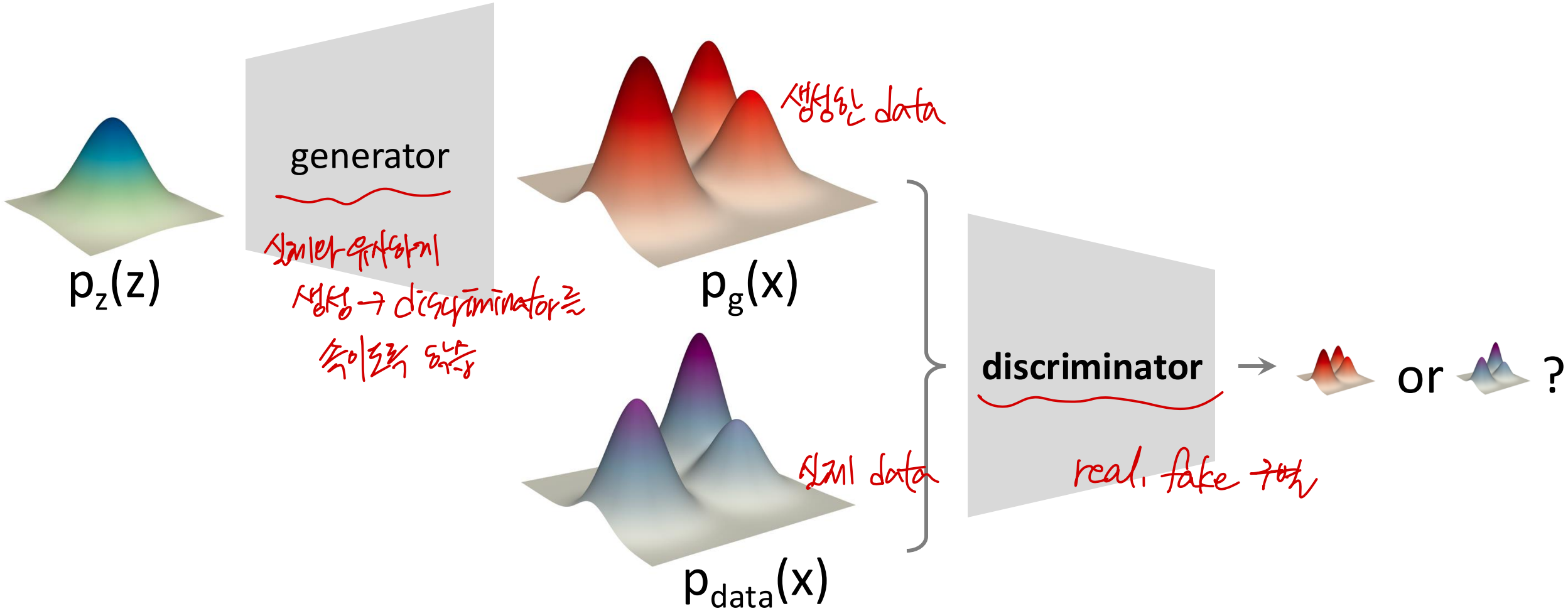


maximize log-likelihood

Generative Adversarial Net (GAN)

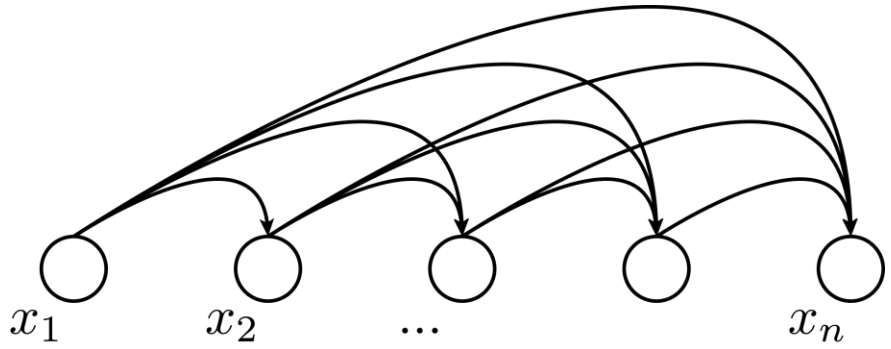


Generative Adversarial Net (GAN)

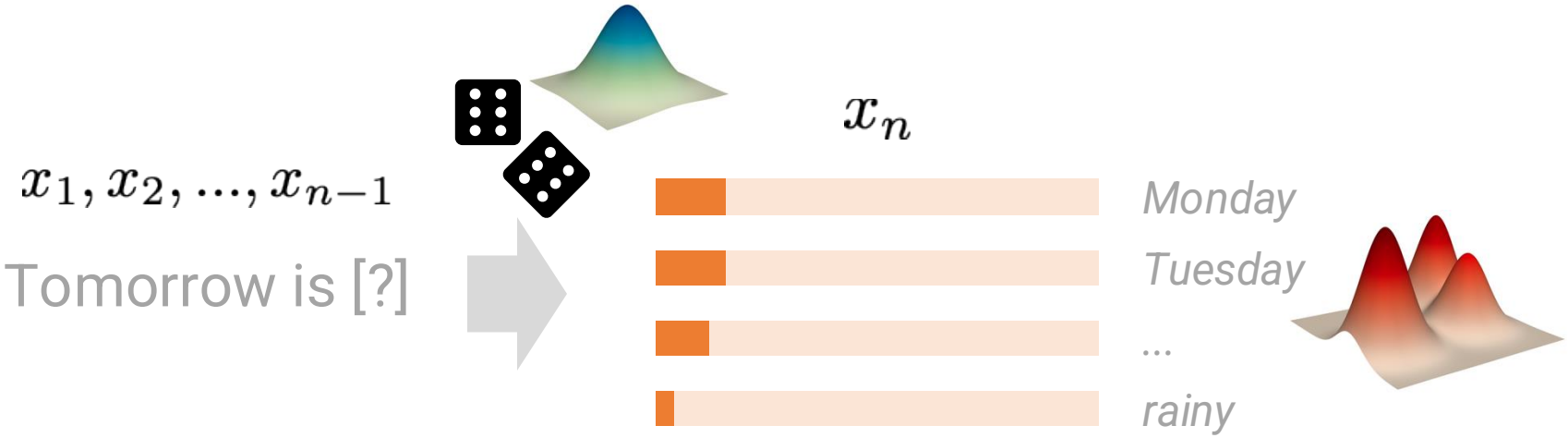


Autoregressive Models (AR)

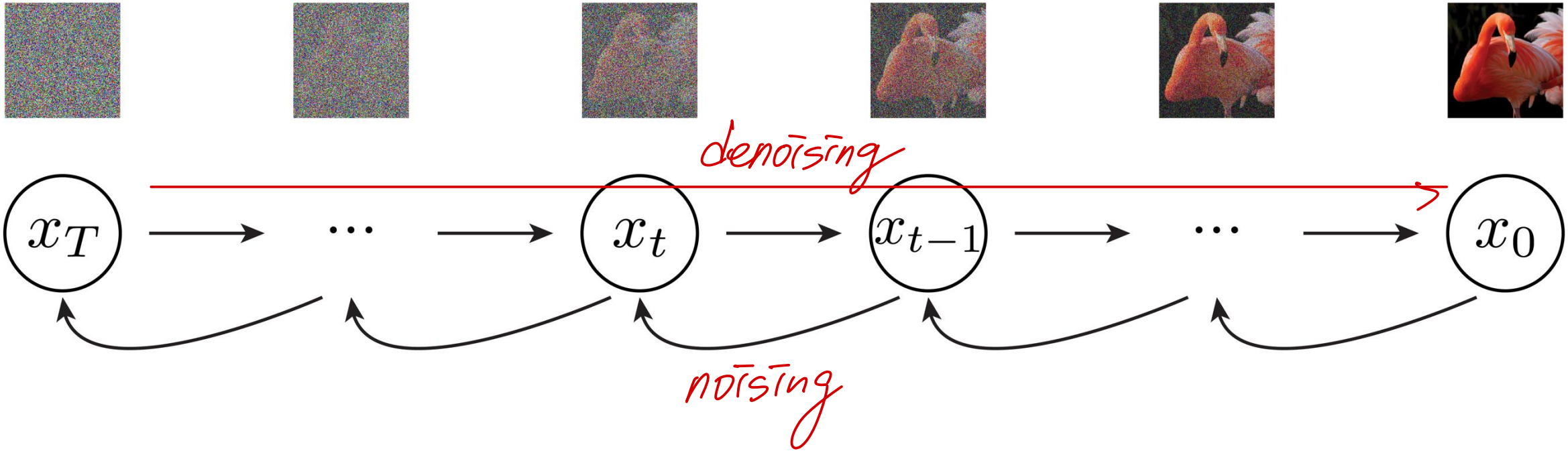
$$p(x_1, x_2, \ldots, x_n) = p(x_1)p(x_2 \mid x_1)\ldots p(x_n \mid x_1, x_2, \ldots, x_{n-1})$$



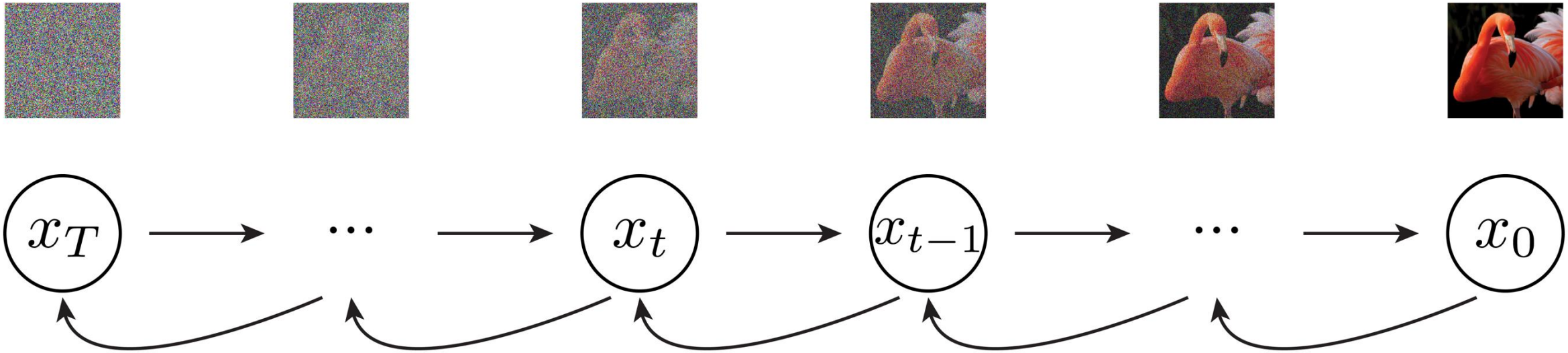
순차적 생성.

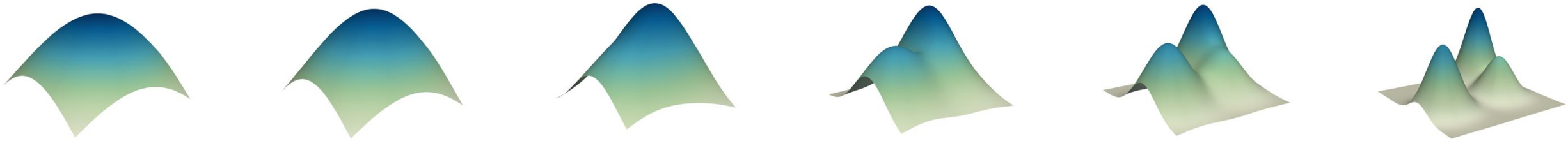


Diffusion Models



Diffusion Models



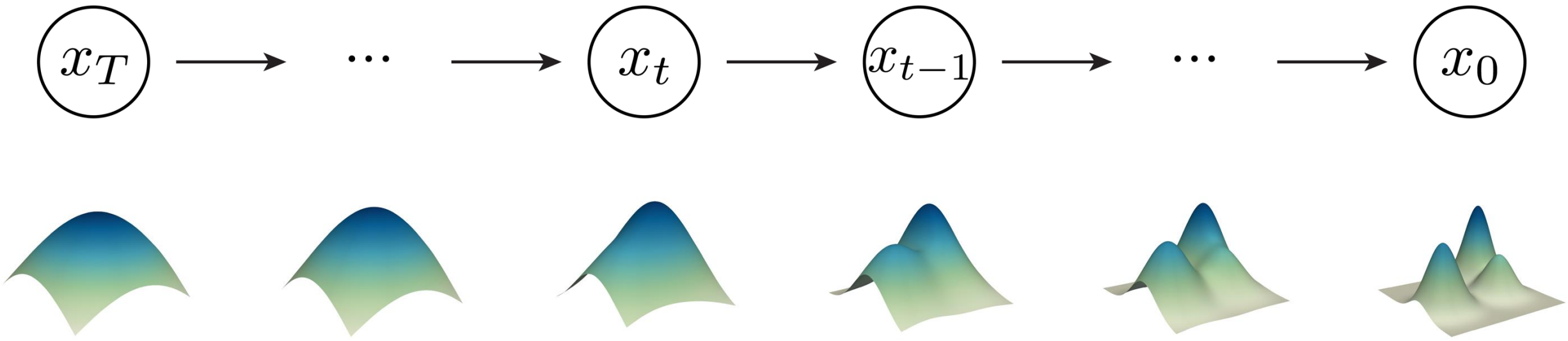


noise distribution

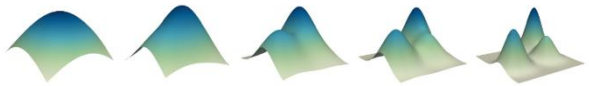
data distribution

순차적으로 변환

Diffusion Models



Flow Matching



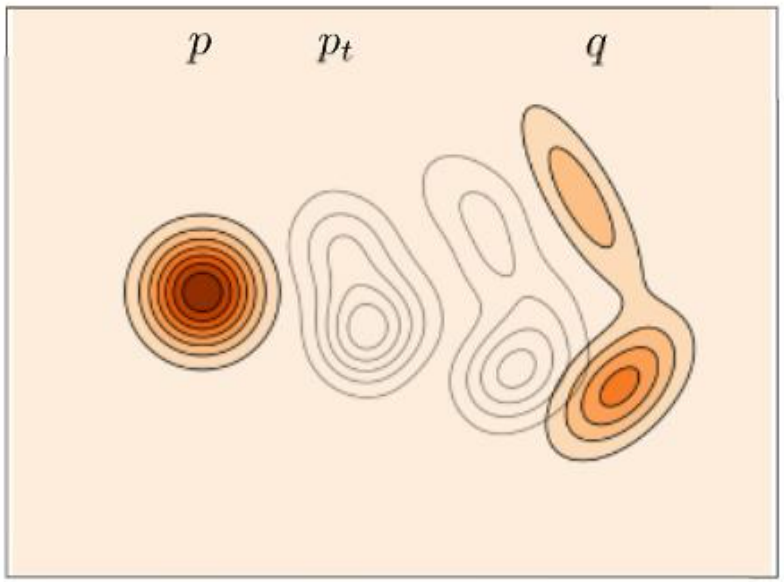
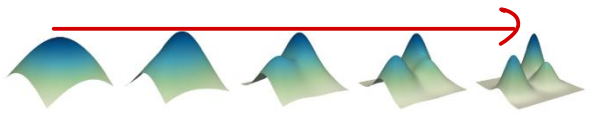


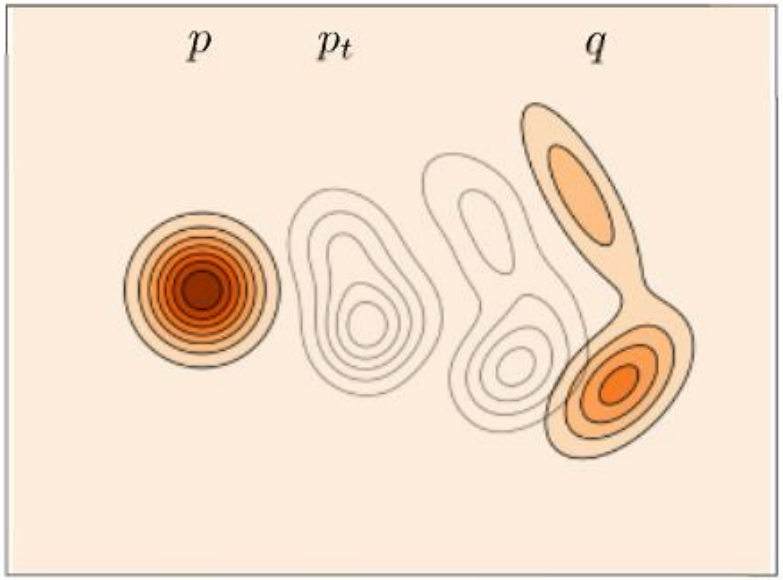
Figure adapted from: https://github.com/facebookresearch/flow_matching

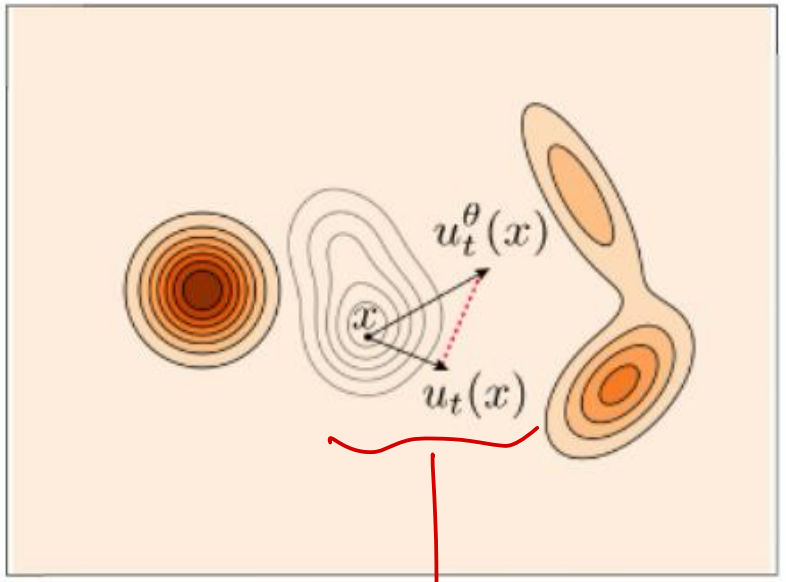
Flow Matching

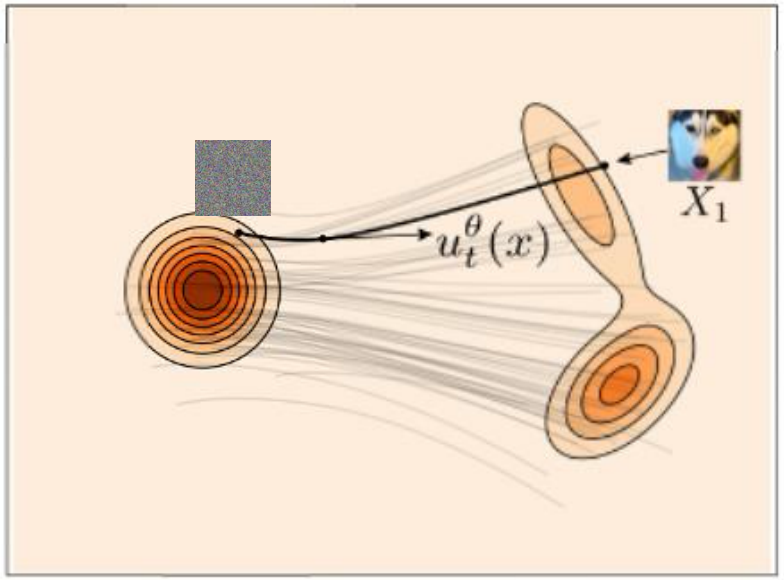


diffusion : discrete?

Flow matching: continuous?







t에 따른 vector field 학습.

(위치, 방향성)

Figure adapted from: https://github.com/facebookresearch/flow_matching

Takeaways

- Generative modeling is probabilistic modeling
- Learning to represent probabilistic distributions
- A way of problem-solving

This Lecture

- What are Generative Models?
- Generative Modeling of Real-world Problems
- Families of Generative Models