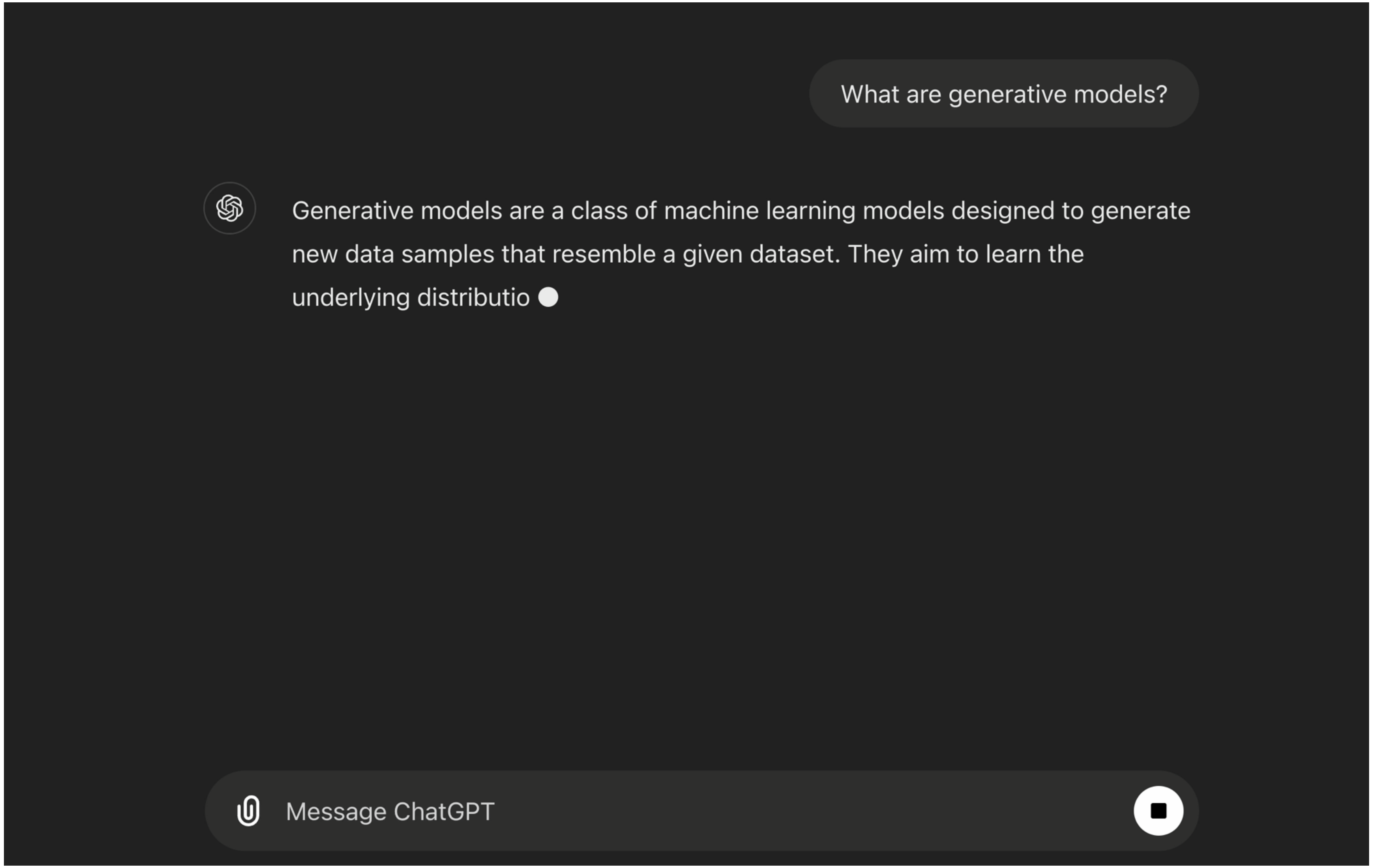
Lecture 13

Generative Models: Introduction

Speaker: Kaiming He

The “GenAI” Era

Chatbot and natural language conversation



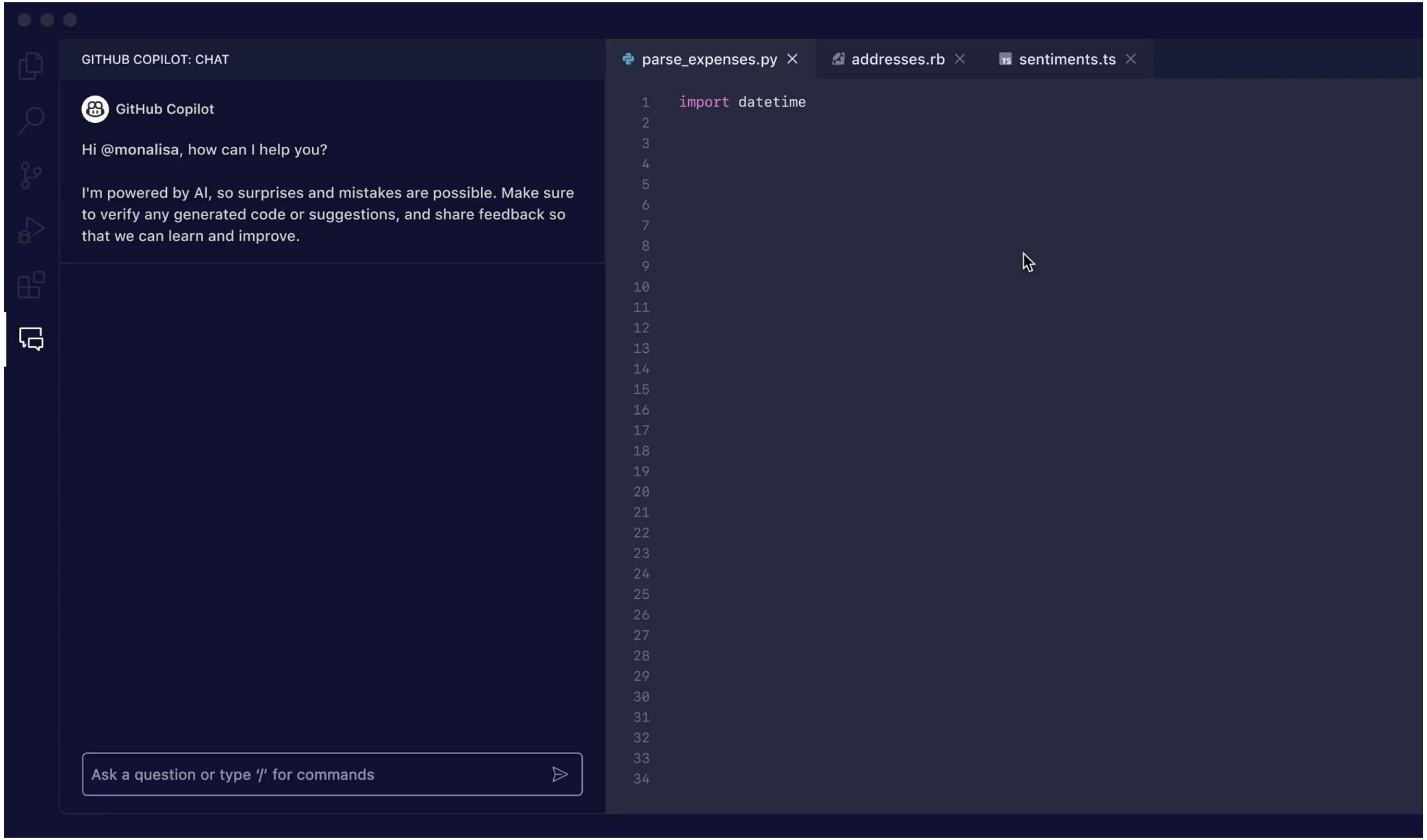
The “GenAI” Era

Text-to-image generation

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

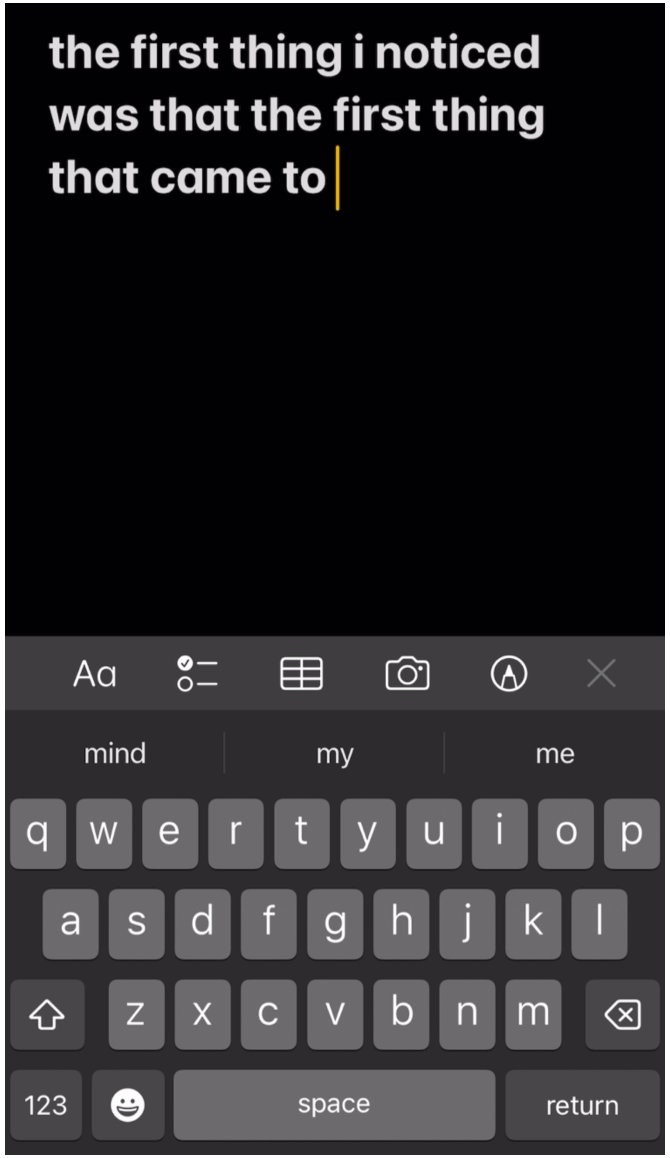
The "GenAI" Era

AI assistant for code generation



Generative Models before the “GenAI” Era

Your keyboard



Generative Models before the "GenAI" Era

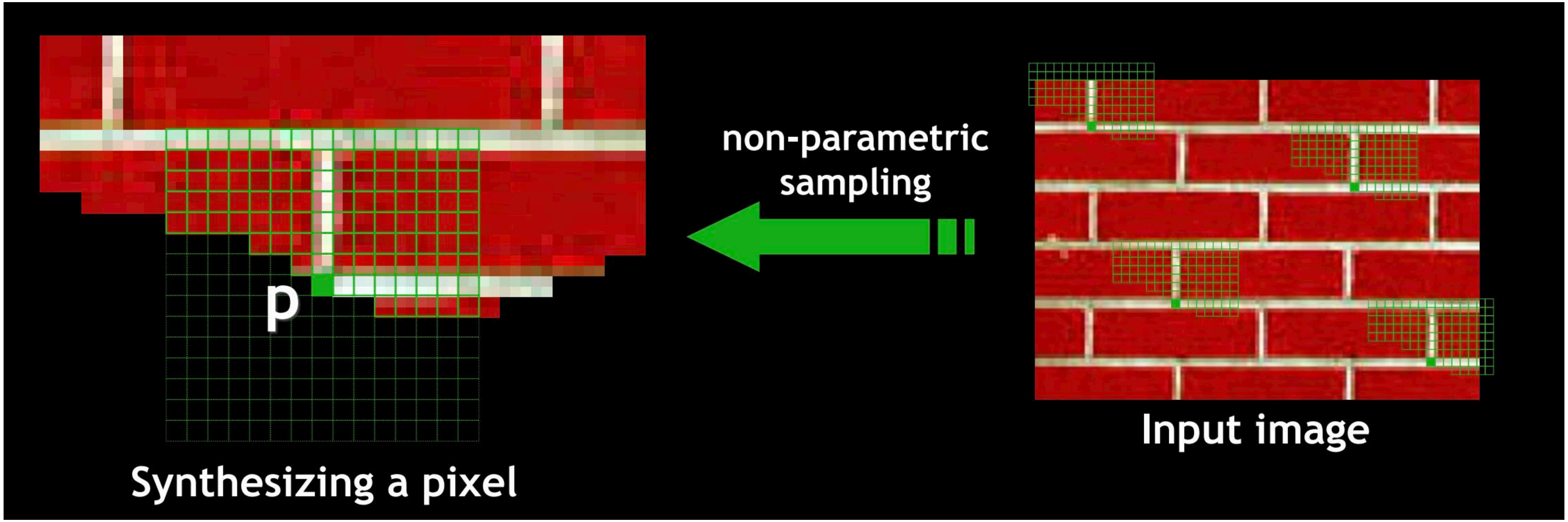
2009, PatchMatch: Photoshop's Content-aware Fill

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

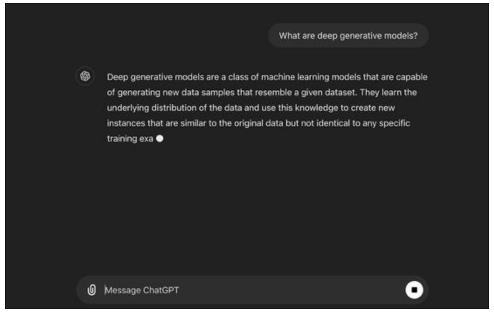


Texture Synthesis by Non-parametric Sampling, ICCV 1999

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

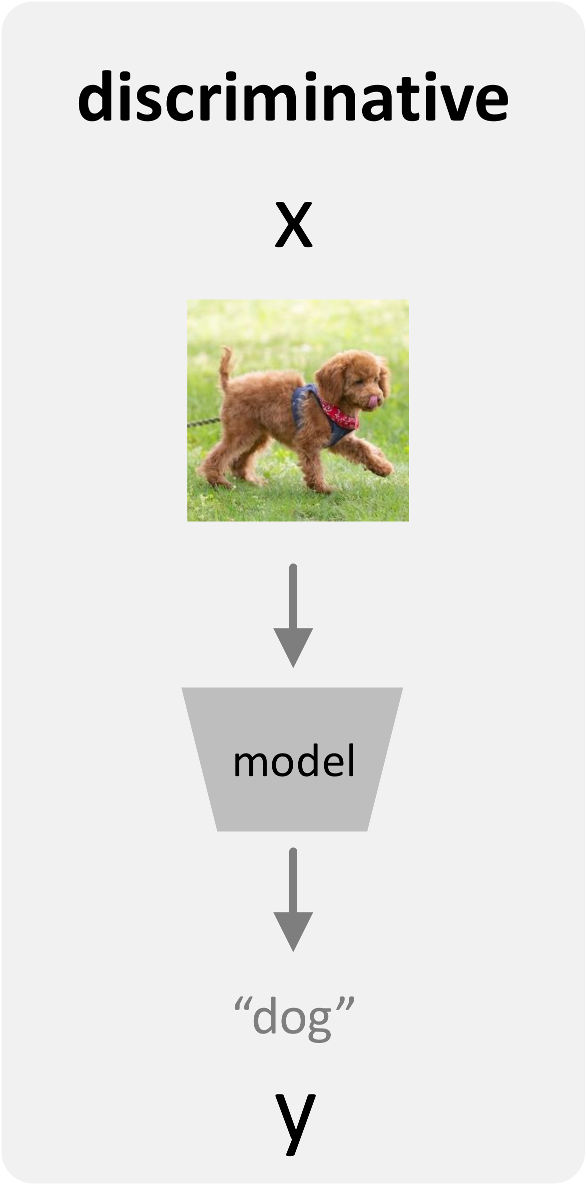
Discriminative Models vs. Generative Models

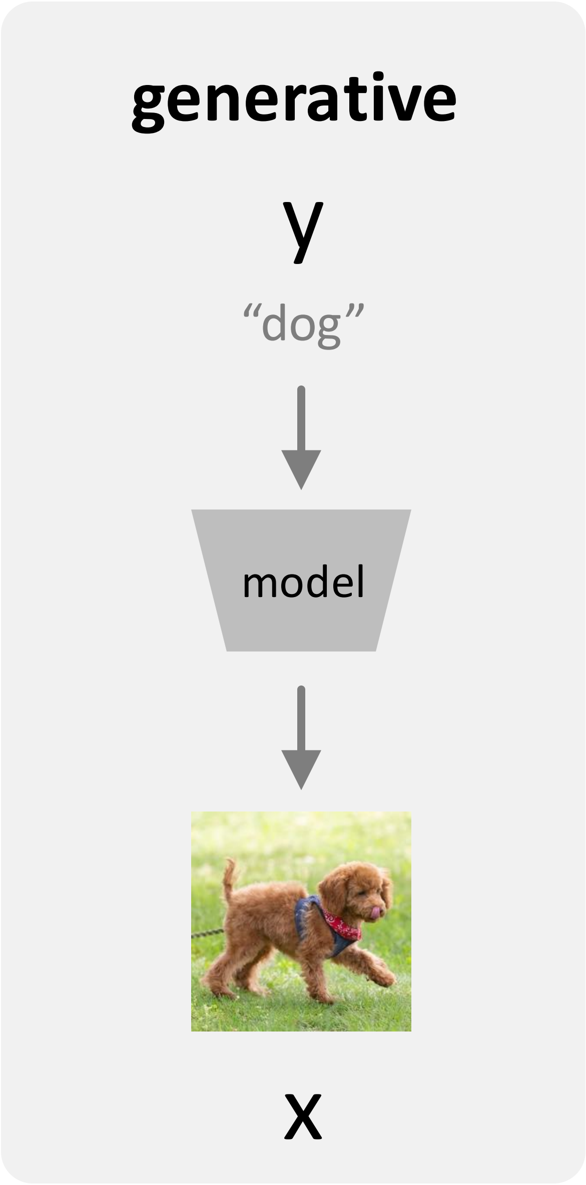
Discriminative Model

- “sample” x ⇨ “label” y
- one desired output

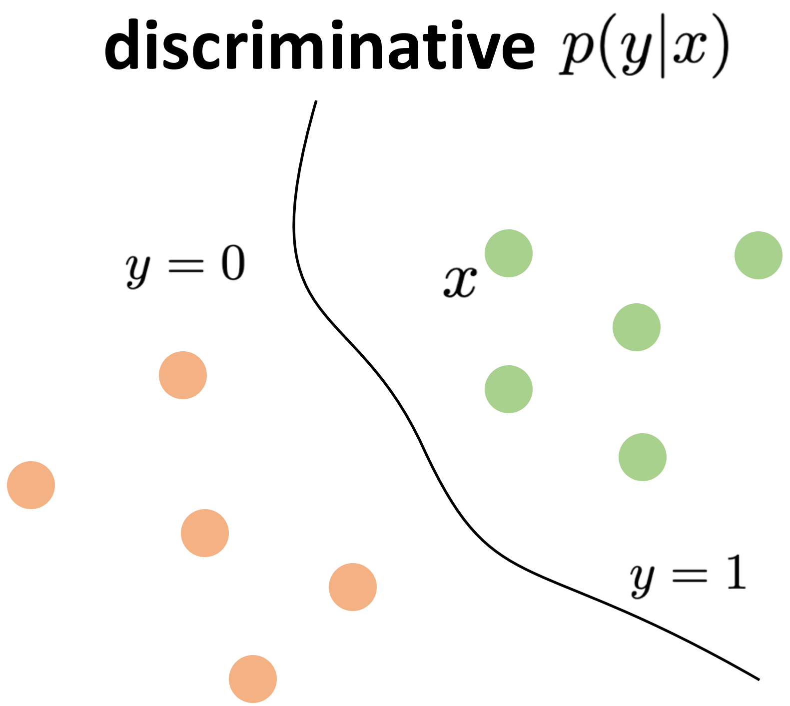
Generative Model

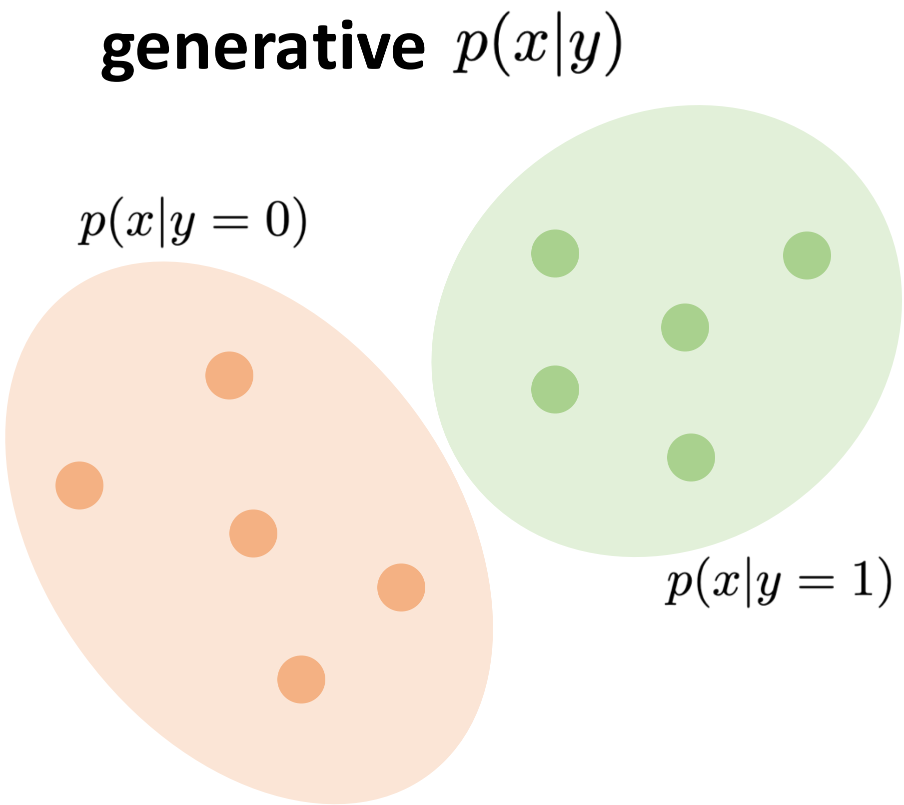
- “label” y ⇨ “sample” x
- many possible outputs





Discriminative Models vs. Generative Models





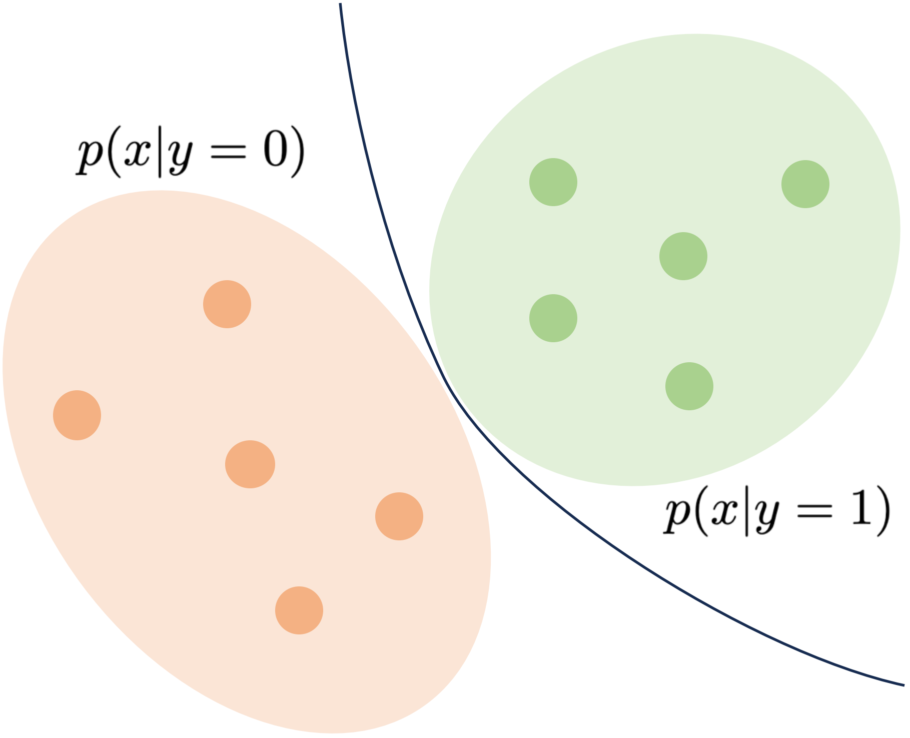
- Generative models can be discriminative: Bayes' rule
- Can discriminative models be generative?

- Generative models can be discriminative: Bayes' rule

$$\underbrace{p(y|x)}_{\text{discriminative}} = \underbrace{p(x|y)}_{\text{generative}} \frac{p(y)}{p(x)}$$

assuming known prior (uniform categories)

constant for given x



decision boundary:

$$p(x \mid y=0) = p(x \mid y=1)$$

- Generative models can be discriminative: Bayes' rule

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

assuming known prior (uniform categories)

discriminative generative

constant for given x

- Can discriminative models be generative?

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

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

generative discriminative

constant for given y

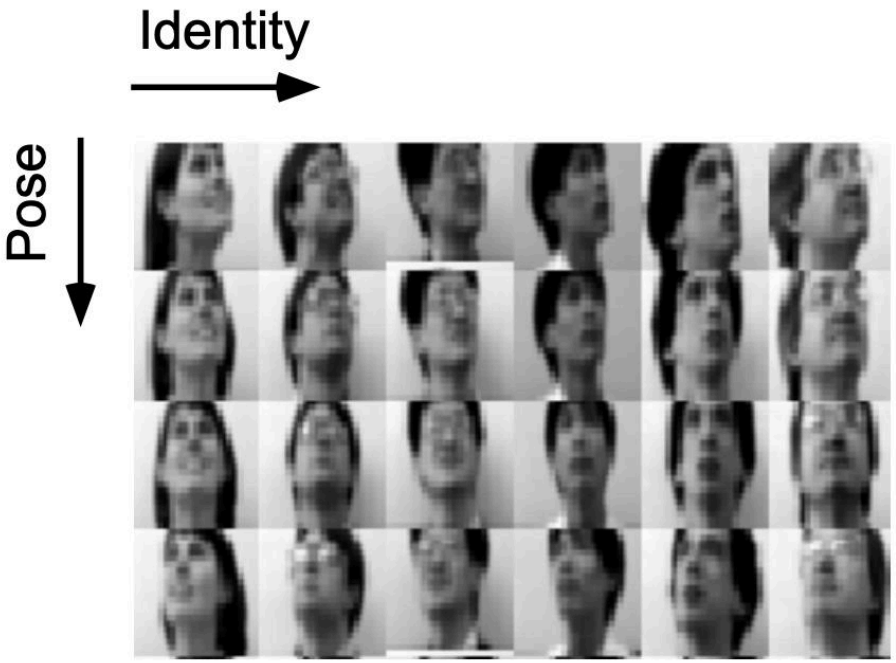
- The challenge is on **probabilistic modeling**

Probabilistic Modeling

- 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

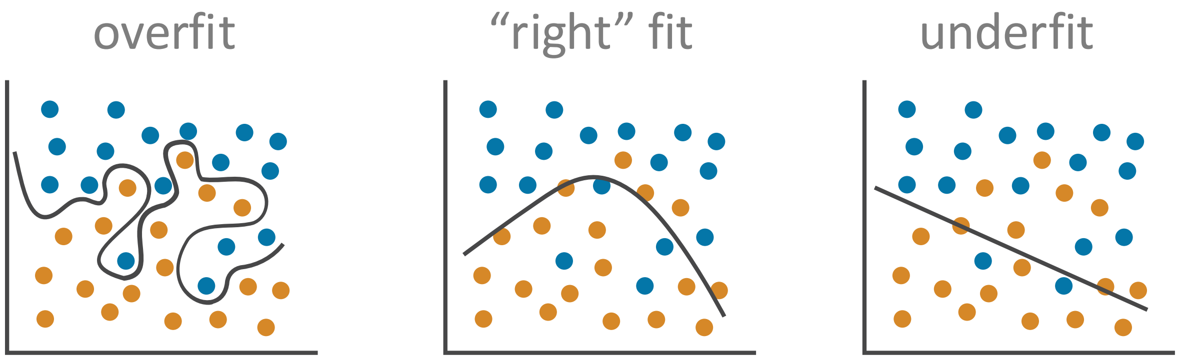


- Probability itself is a modeling assumption

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)

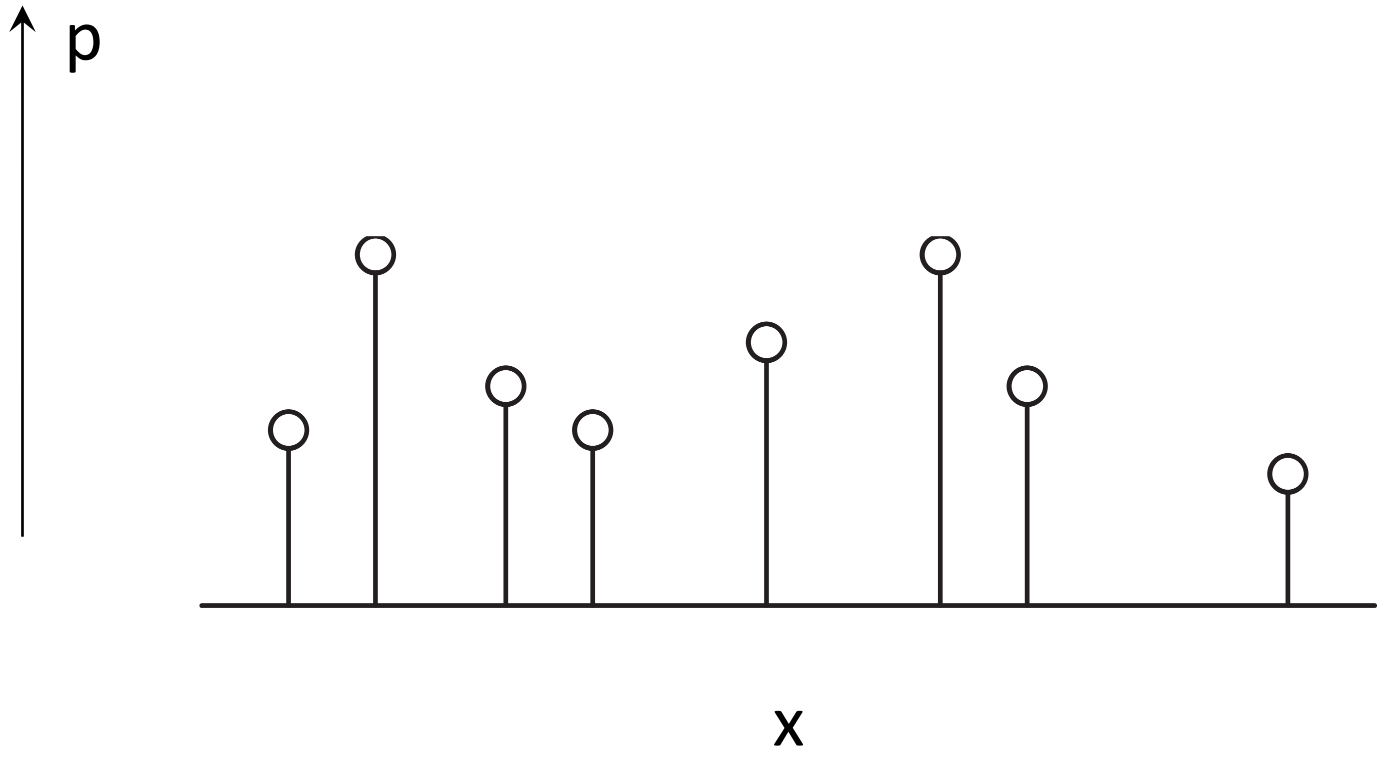


discriminative models

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

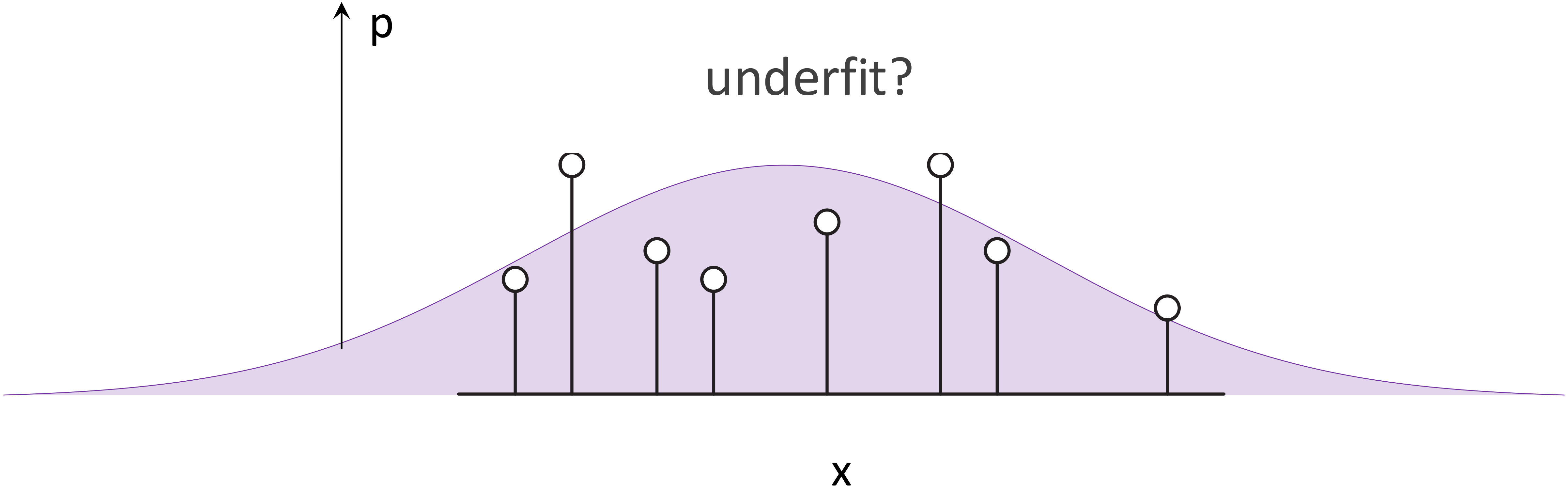
Probability itself is a modeling assumption

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



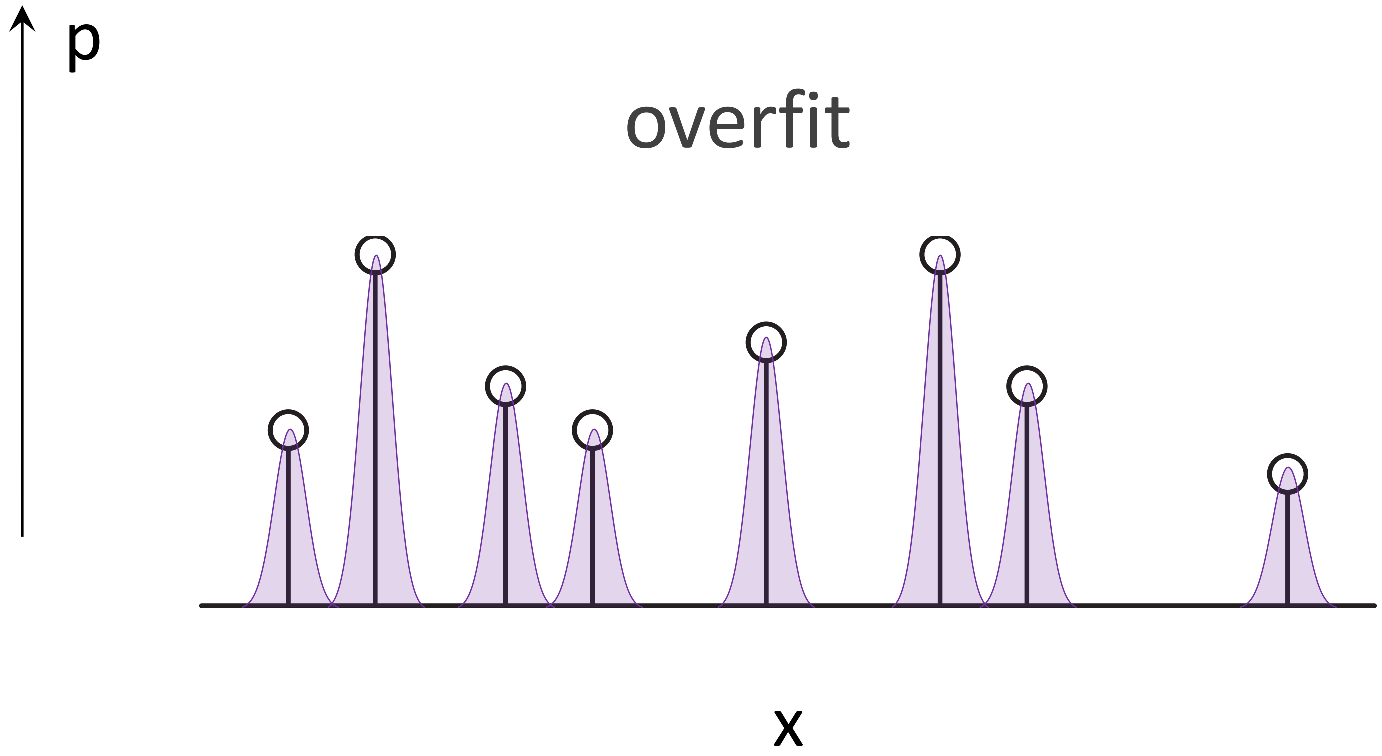
Probability itself is a modeling assumption

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



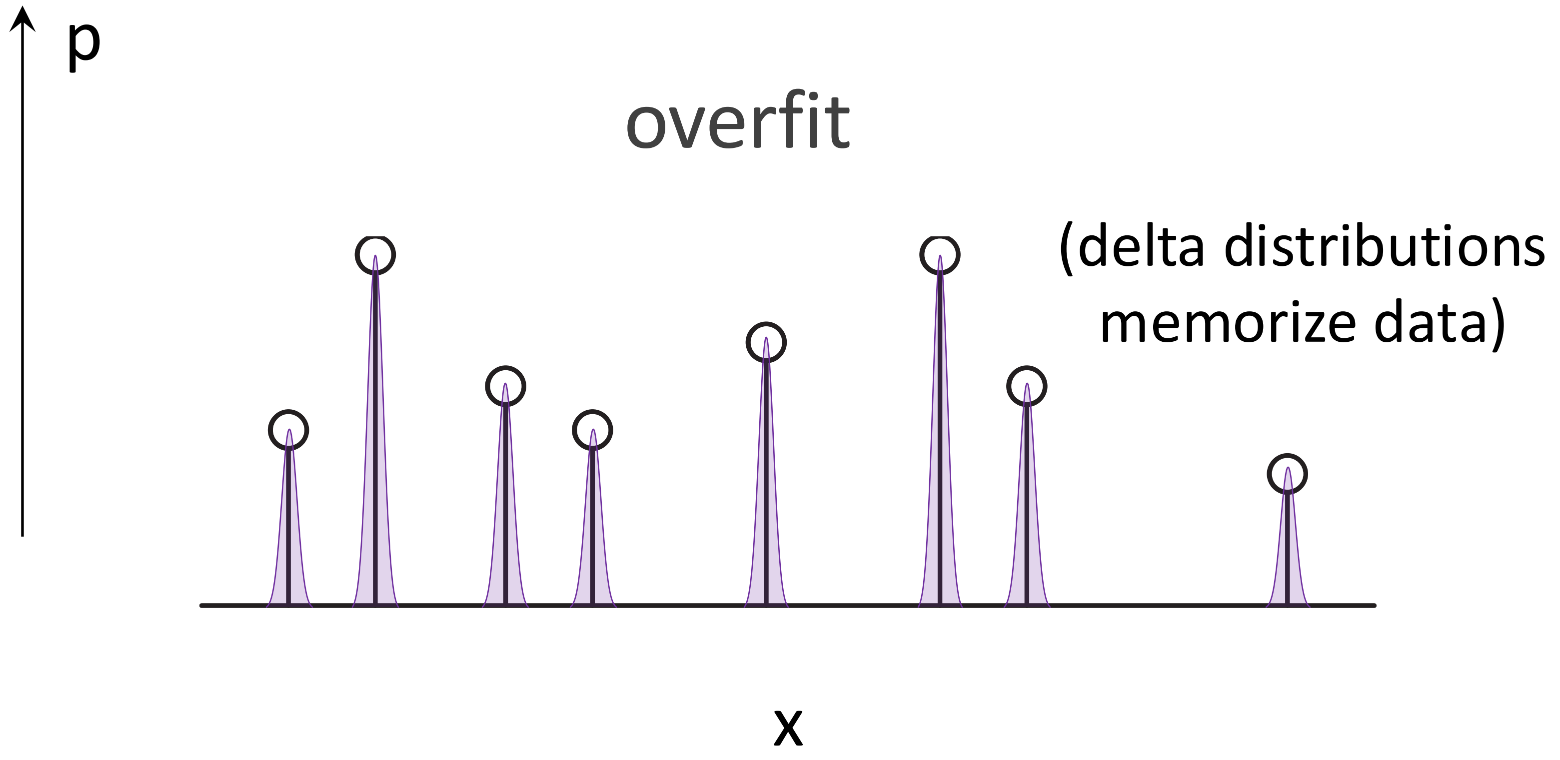
Probability itself is a modeling assumption

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



Probability itself is a modeling assumption

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



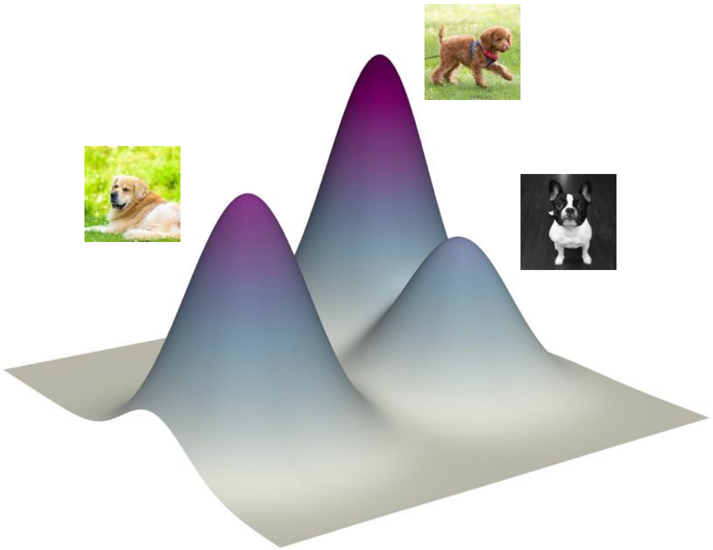
Generative Modeling is Probabilistic Modeling

data

Generative Modeling is Probabilistic Modeling

data

data distribution



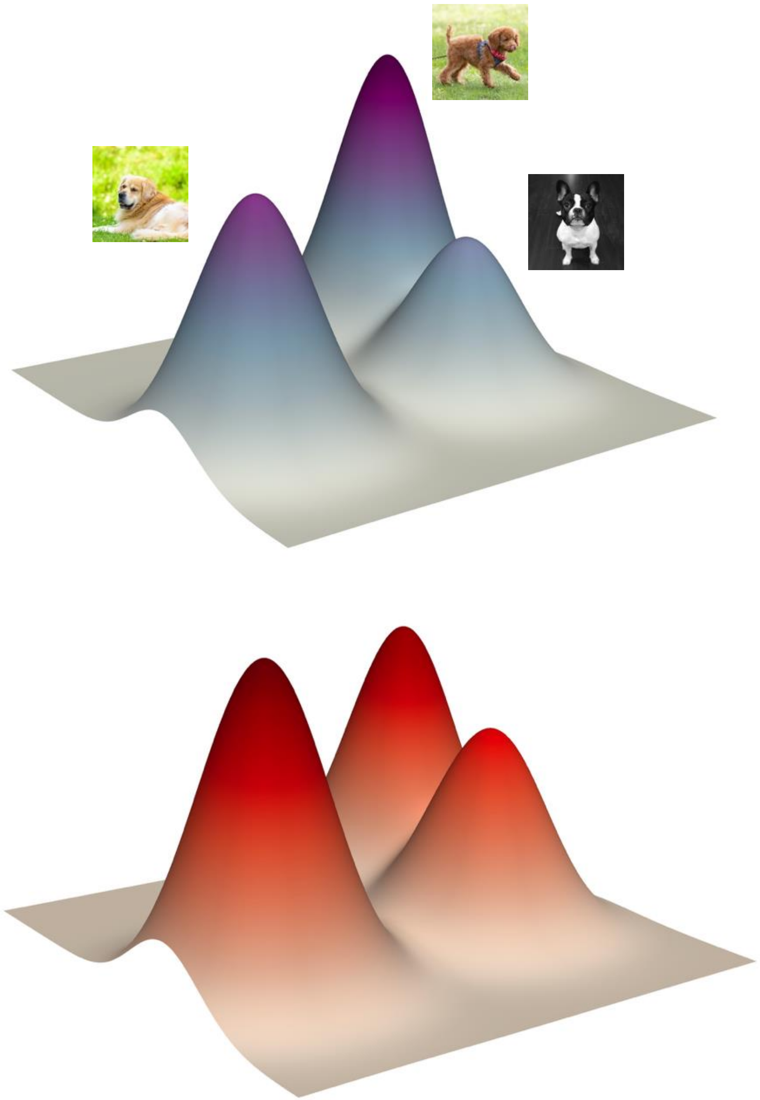
This in itself is part of modeling

Generative Modeling is Probabilistic Modeling

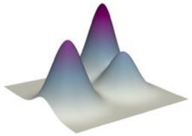
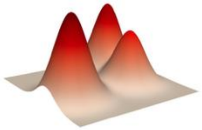
data

data distribution

estimated distribution



Optimize a loss function

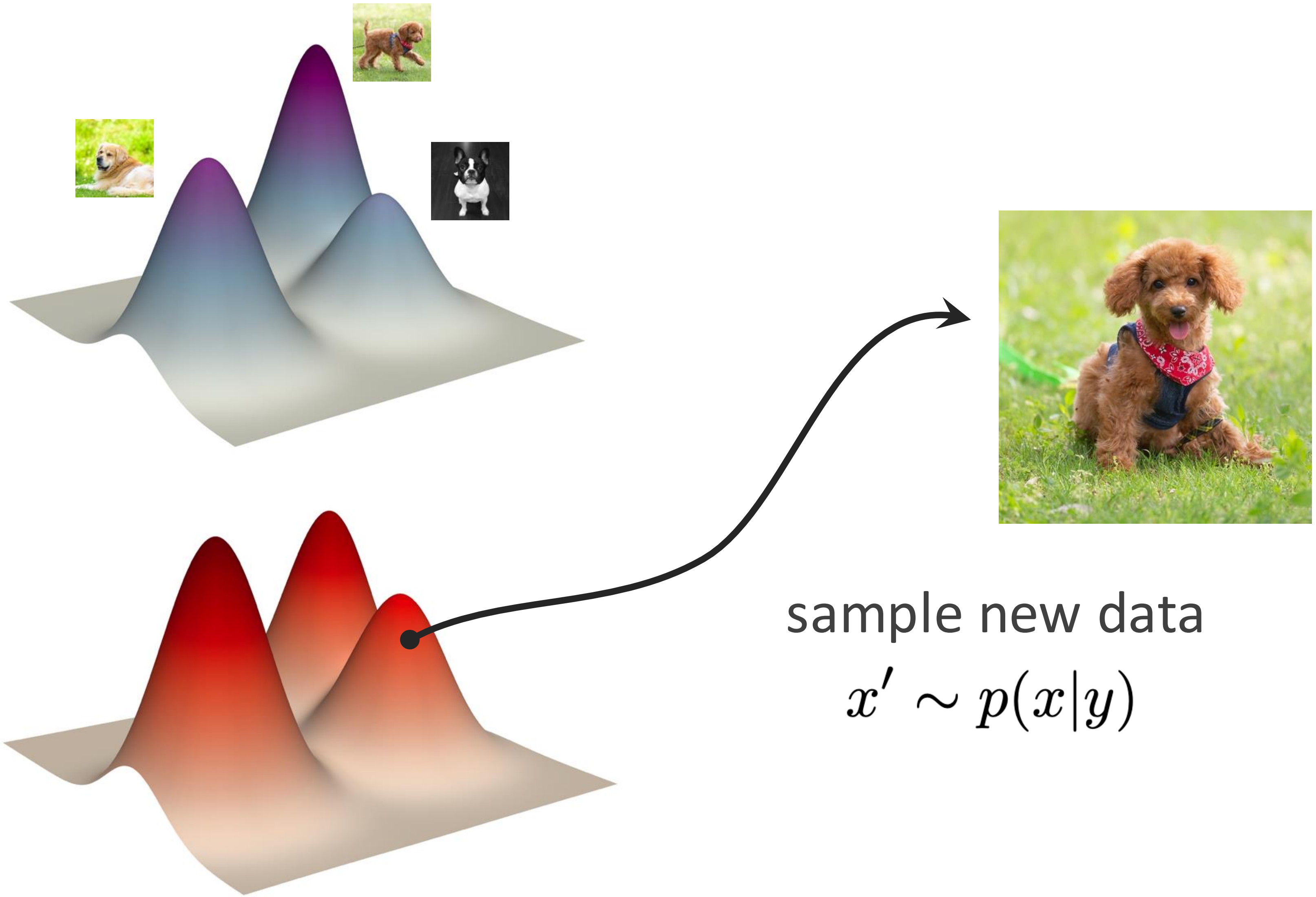
$\mathcal{L}($  ,  $)$

Generative Modeling is Probabilistic Modeling

data

data distribution

estimated distribution



sample new data

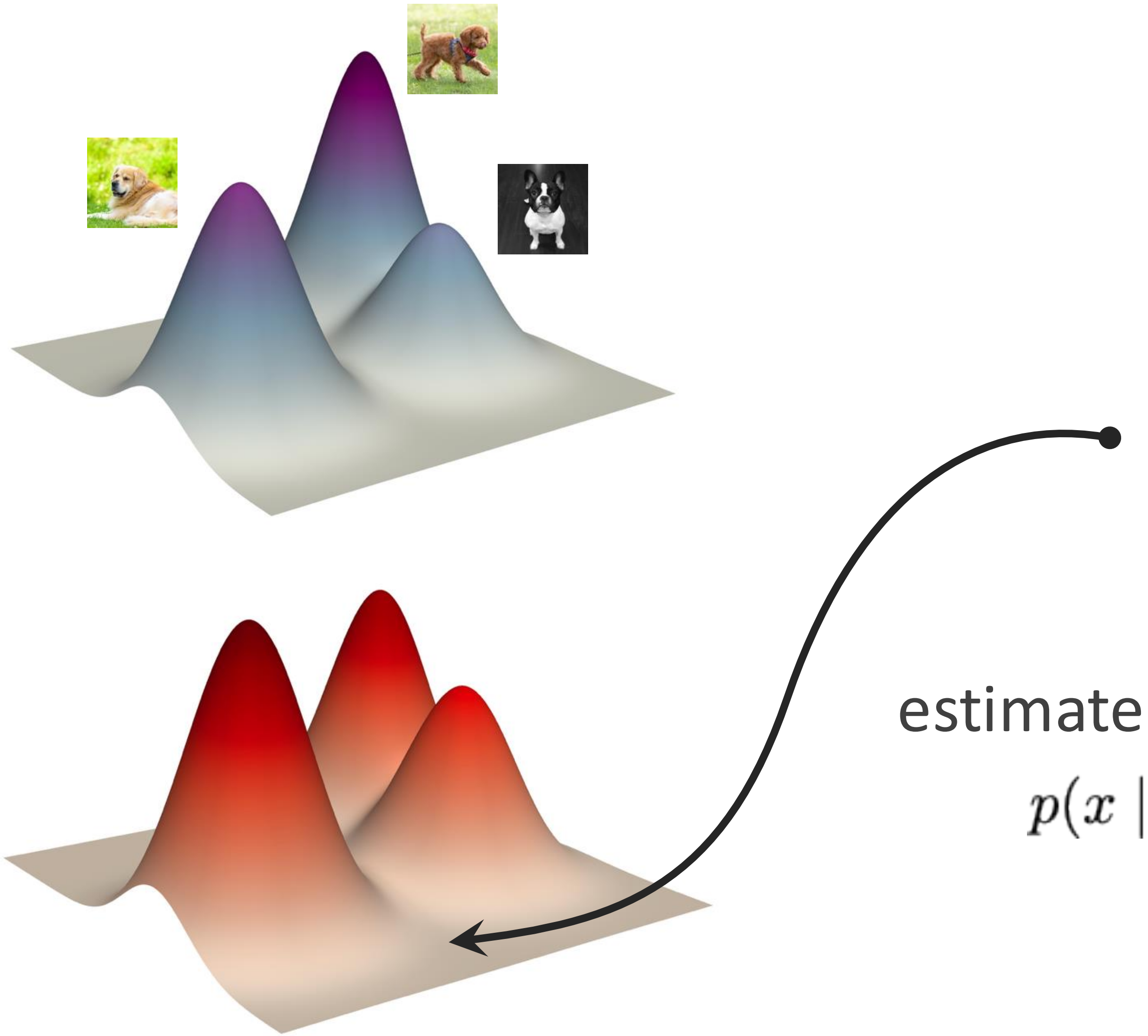
$$x' \sim p(x|y)$$

Generative Modeling is Probabilistic Modeling

data

data
distribution

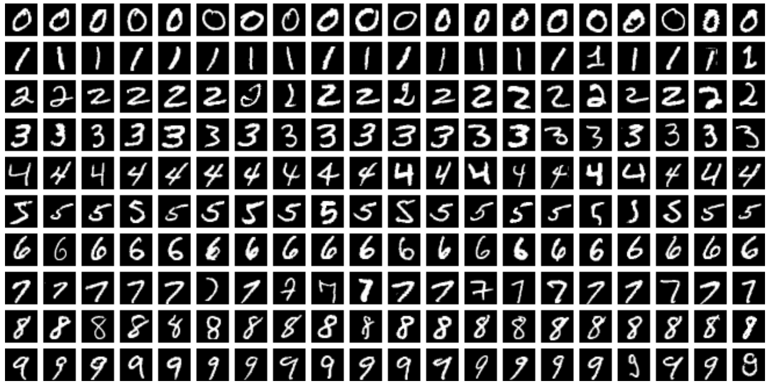
estimated
distribution



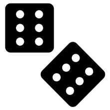
estimate prob density

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

Case study: a minimalist generative model?

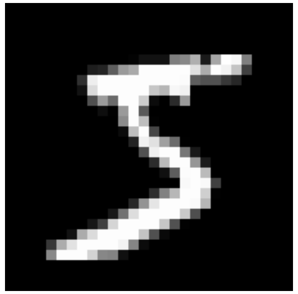


data set



random index

retrieve



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

Generative Modeling is Probabilistic Modeling

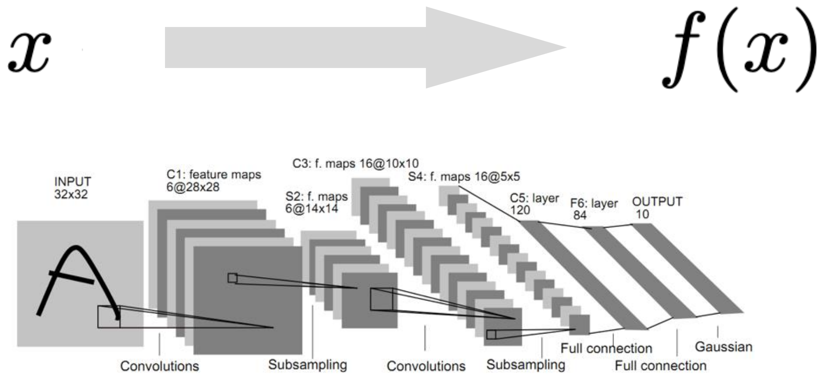
Notes:

- 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

<u>Deep</u> Generative Models

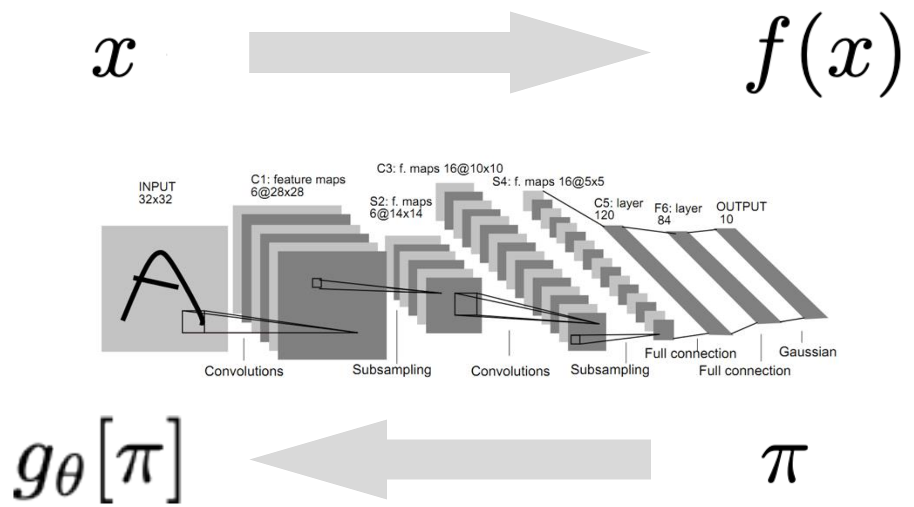
- Deep learning is **representation learning**
- **Discriminative**: represent data **instances**
  - one data point: $x \rightarrow f_\theta(x)$
  - loss w.r.t. one label: $\mathcal{L}(y, f_\theta(x))$



Deep Generative Models

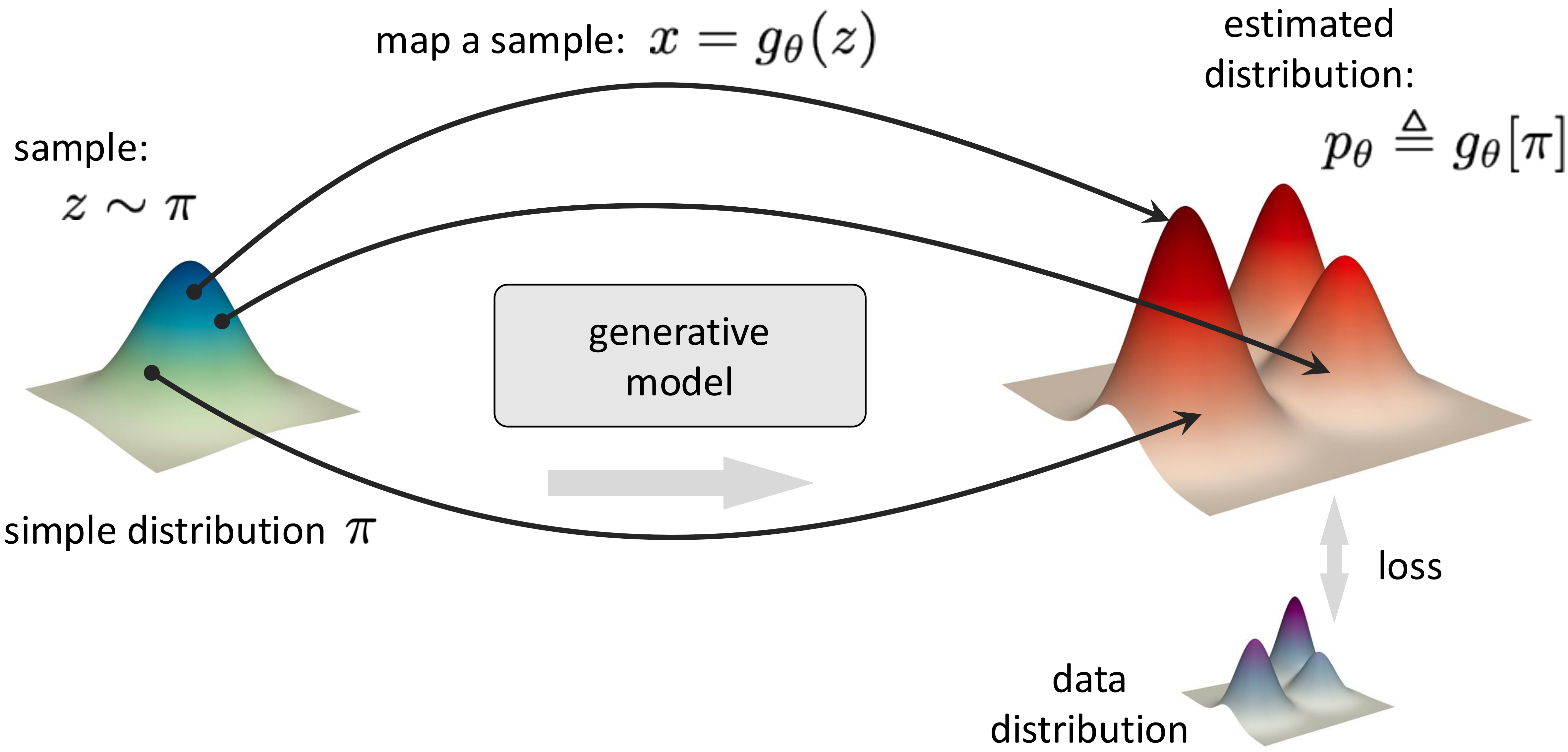
- Deep learning is **representation learning**

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



- **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: $\mathcal{L}(p_{\text{data}}, g_\theta[\pi])$

Learning to Represent Distributions

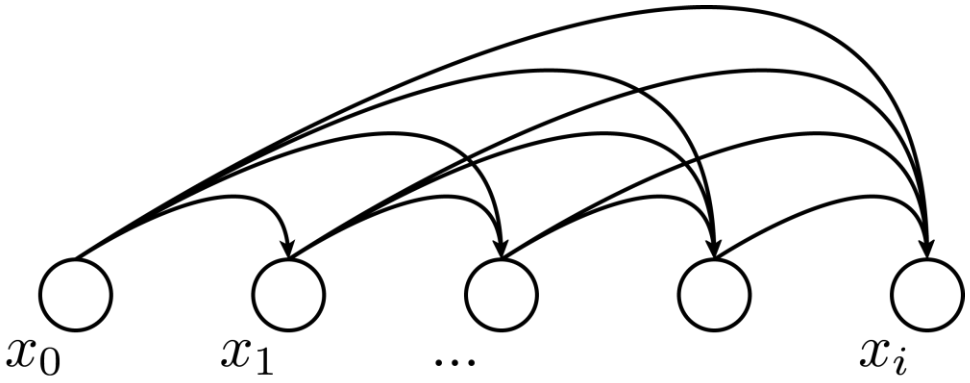


Learning to Represent Distributions

- Not all parts of a generative model are learned.

Case study:
Autoregressive model

This dependency structure is designed (not learned).

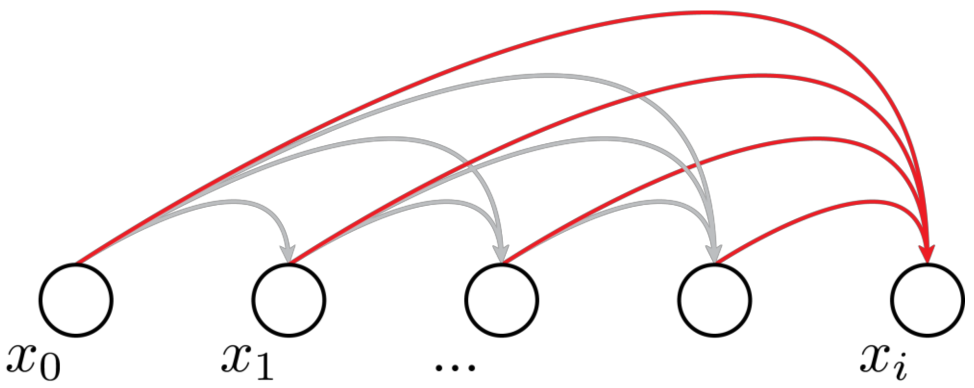


Learning to Represent Distributions

- Not all parts of a generative model are learned.

Case study:
Autoregressive model

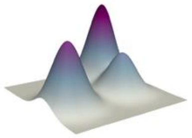
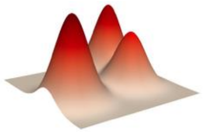
This probability is learned



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

Objective Functions in Generative Models

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

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

a functional

a function

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

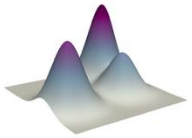
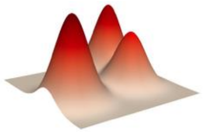
another function

$p_\theta(x)$

Objective Functions in Generative Models

- Case study: Kullback–Leibler (KL) Divergence

x is a dummy variable

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

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

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

What makes the objectives difficult to design?

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

- Monte Carlo estimation
  - Distributions unknown in analytical form
  - We can only sample from them
  - Example: $x, y \sim p_{\mathrm{data}}(x, y)$

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

See also "What Makes Modeling Information Difficult?" in Lecture 10

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 $p(x|y)$

What can be y?

- condition
- constraint
- labels
- attributes

- more abstract
- less informative

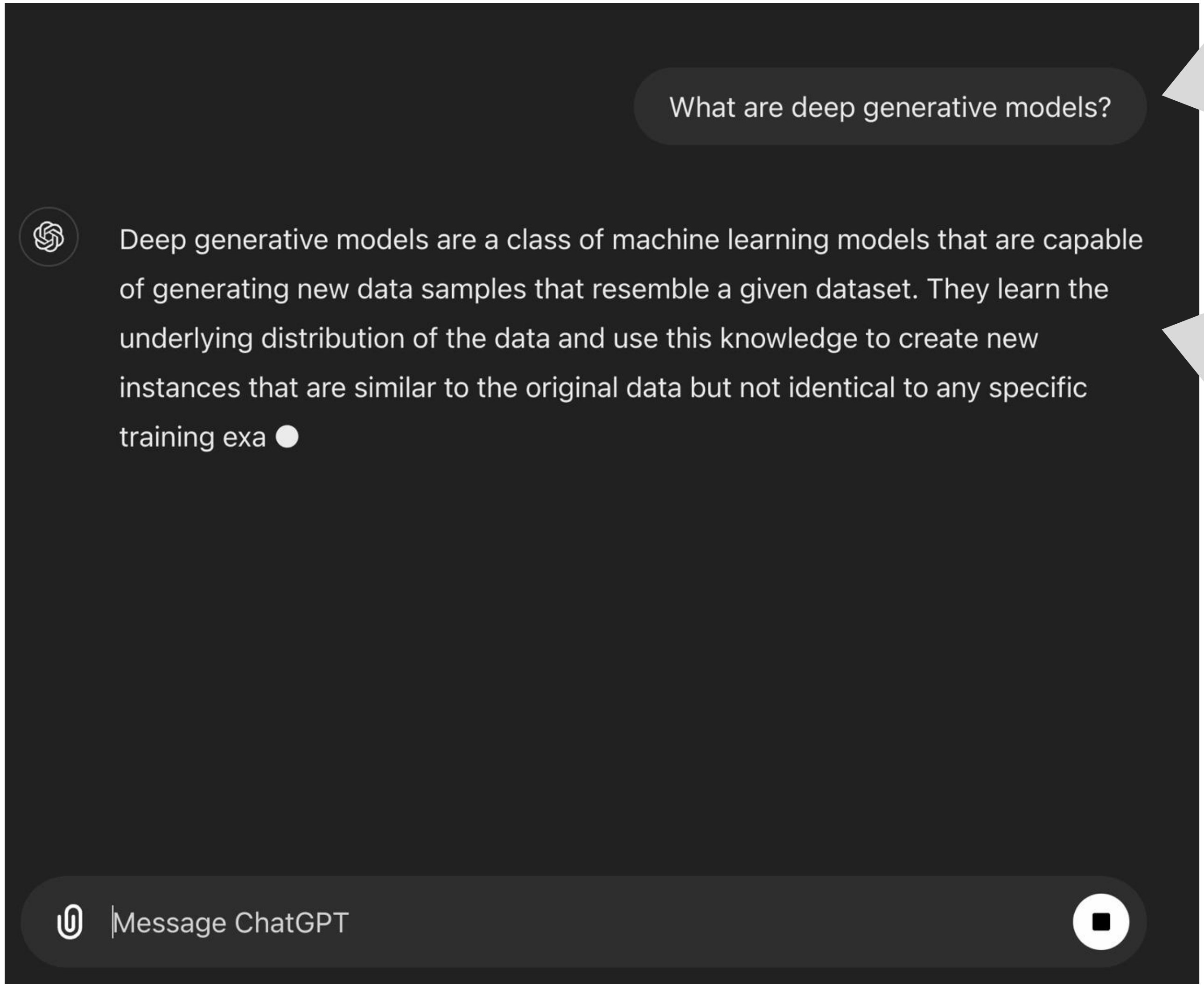
What can be x?

- “data”
- samples
- observations
- measurements

- more concrete
- more informative

Generative Modeling 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

- **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

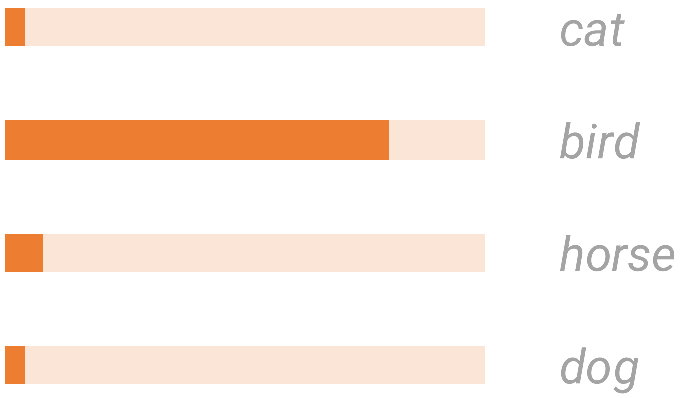
Images generated by: Karras, et al. Elucidating the Design Space of Diffusion-Based Generative Models, NeurIPS 2022

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

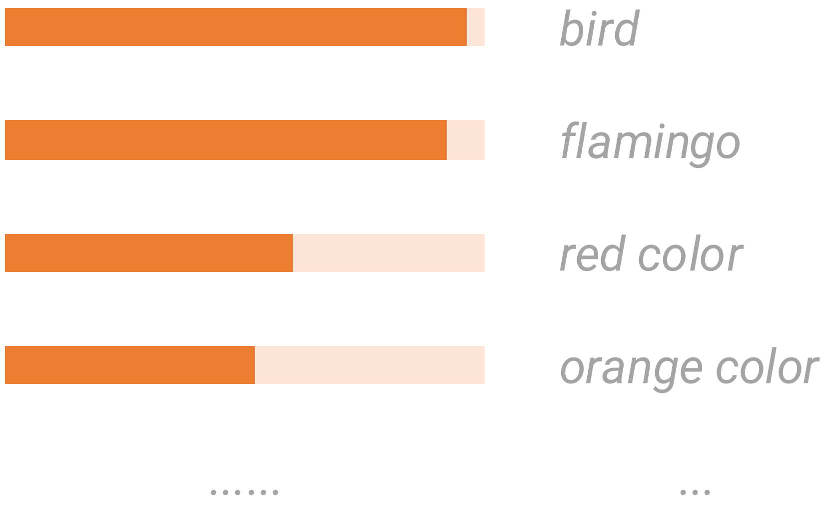


Generative Modeling as p(x|y): Case Study

- **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

y: image and text prompt

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.

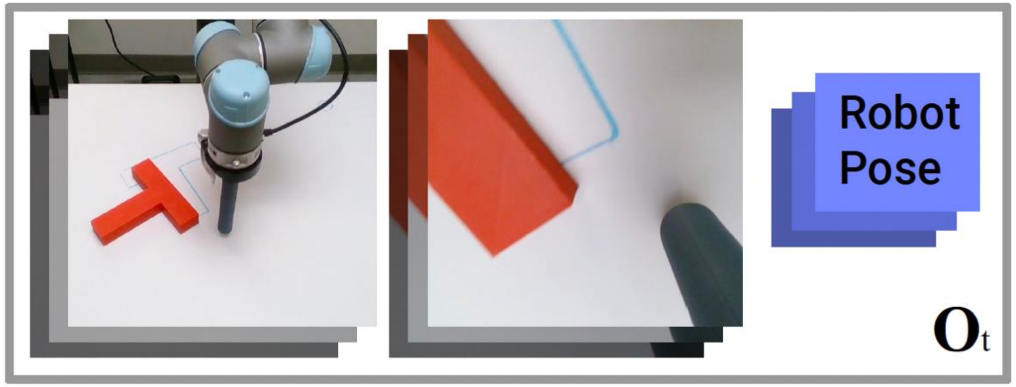
x: response of the chatbot

Figure from: GPT-4 Technical Report, 2023

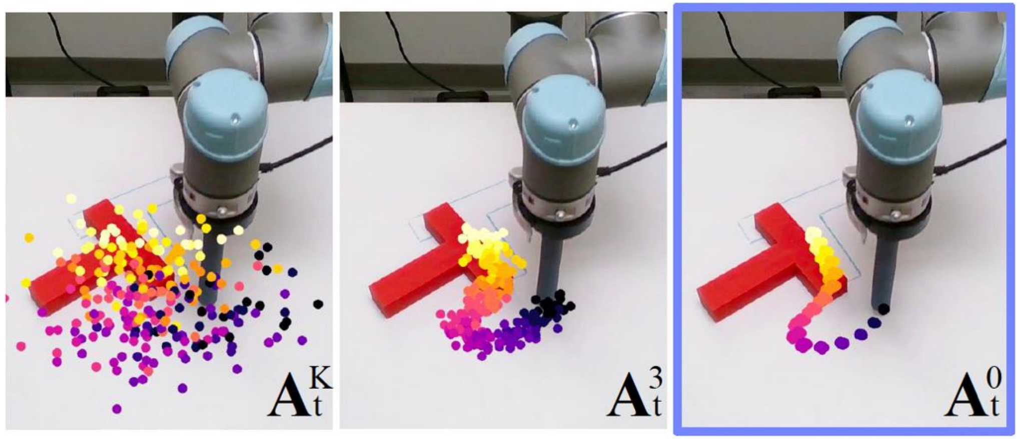
Generative Modeling as p(x|y): Case Study

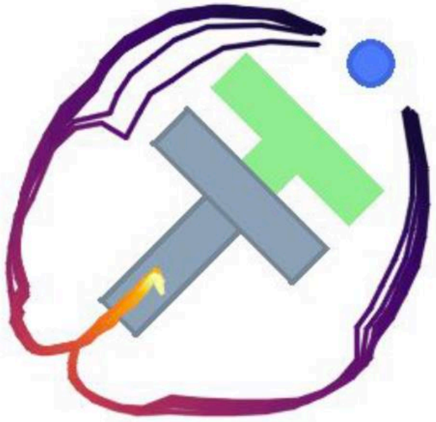
- **Policy Learning in Robotics**

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

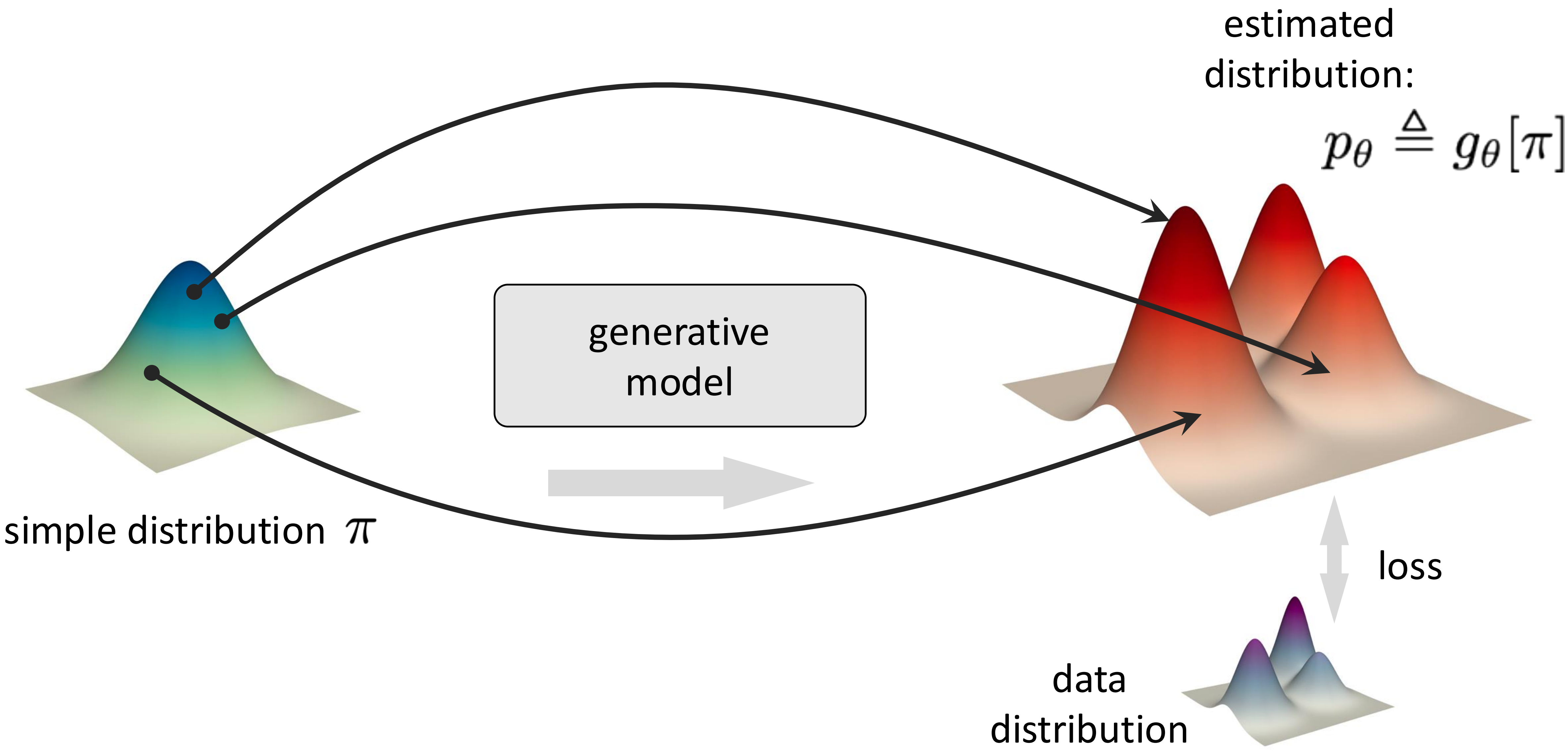
- Generative models are about $p(x|y)$

- There is no conceptual obstacle to formulating any real-world problem from a generative perspective.

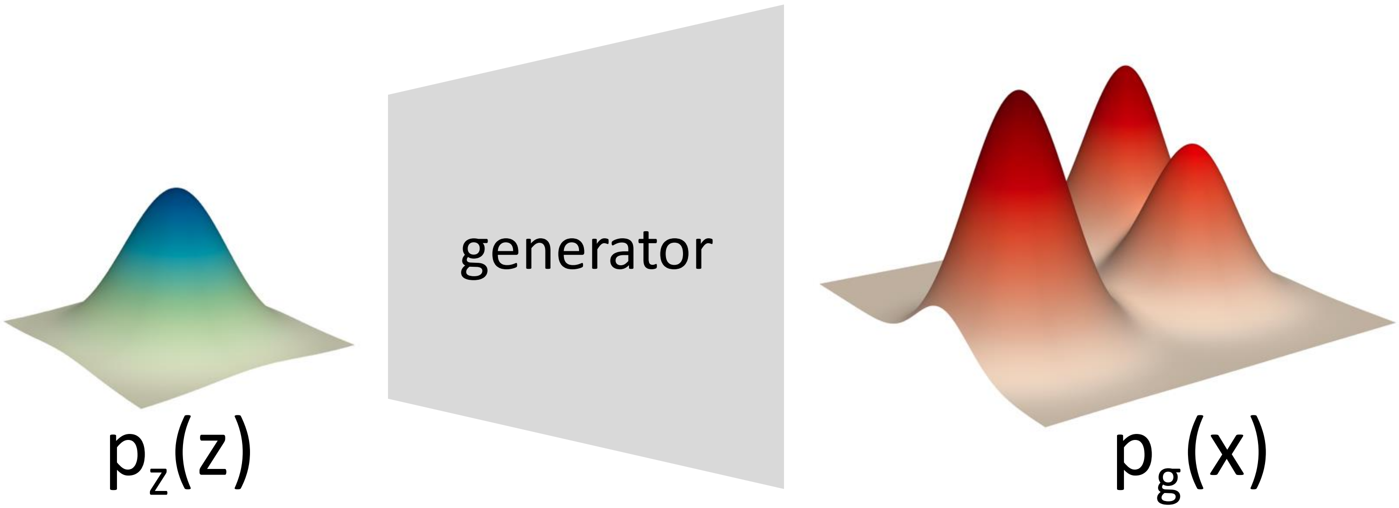
- Generative modeling is a way of problem-solving.

Families of Generative Models

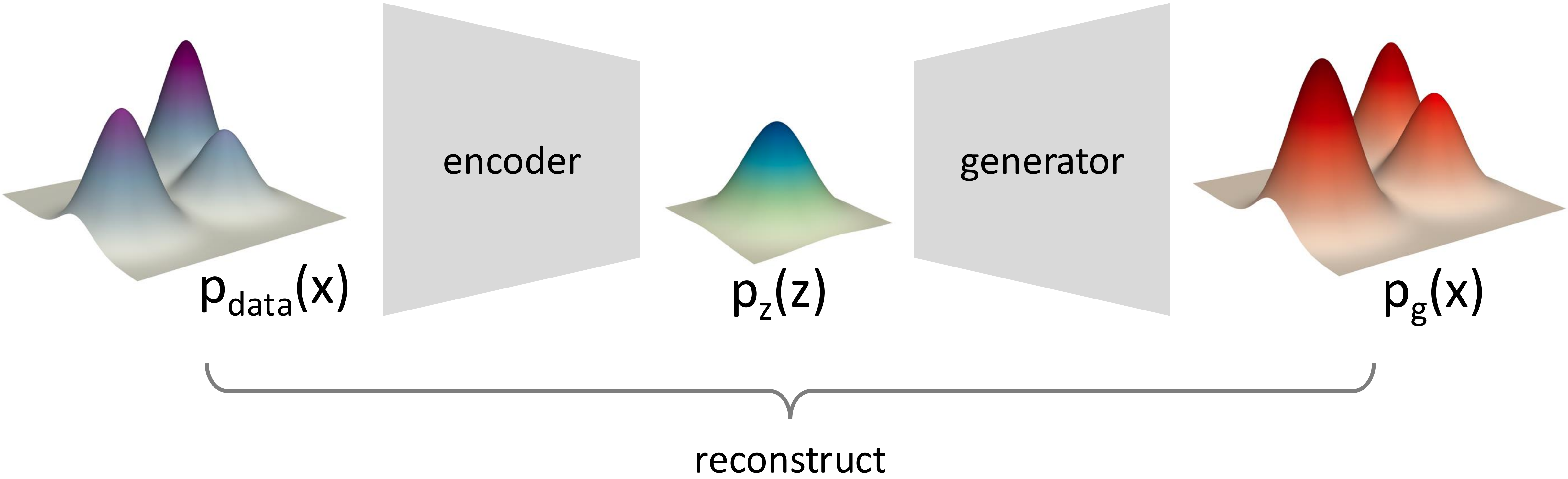
[Recap] Learning to Represent Distributions



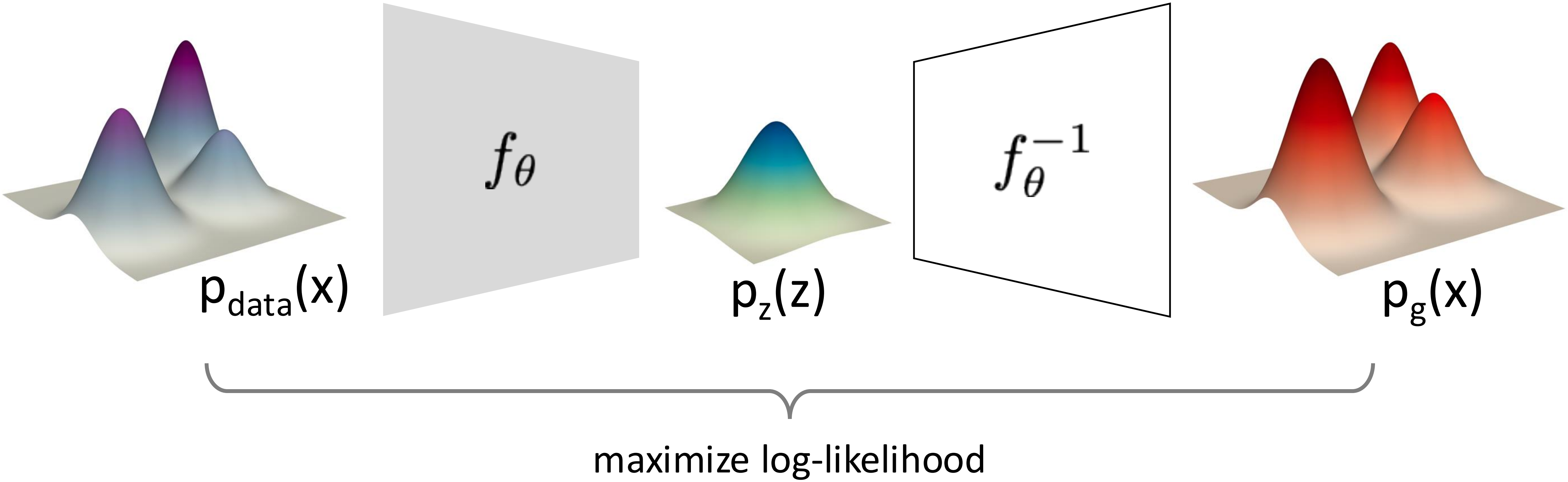
[Recap] Learning to Represent Distributions



Variational Autoencoder (VAE)

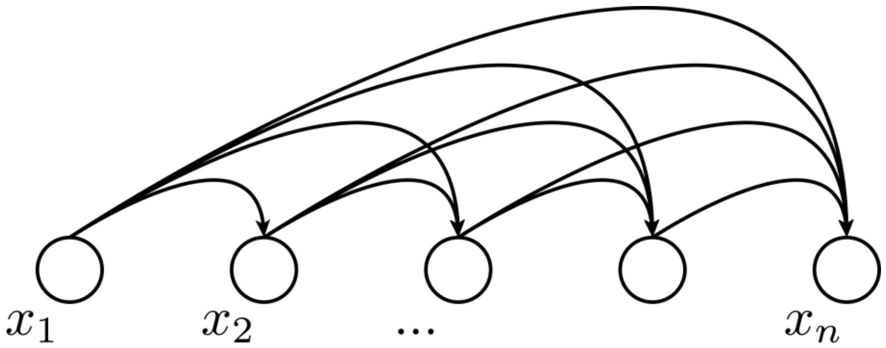


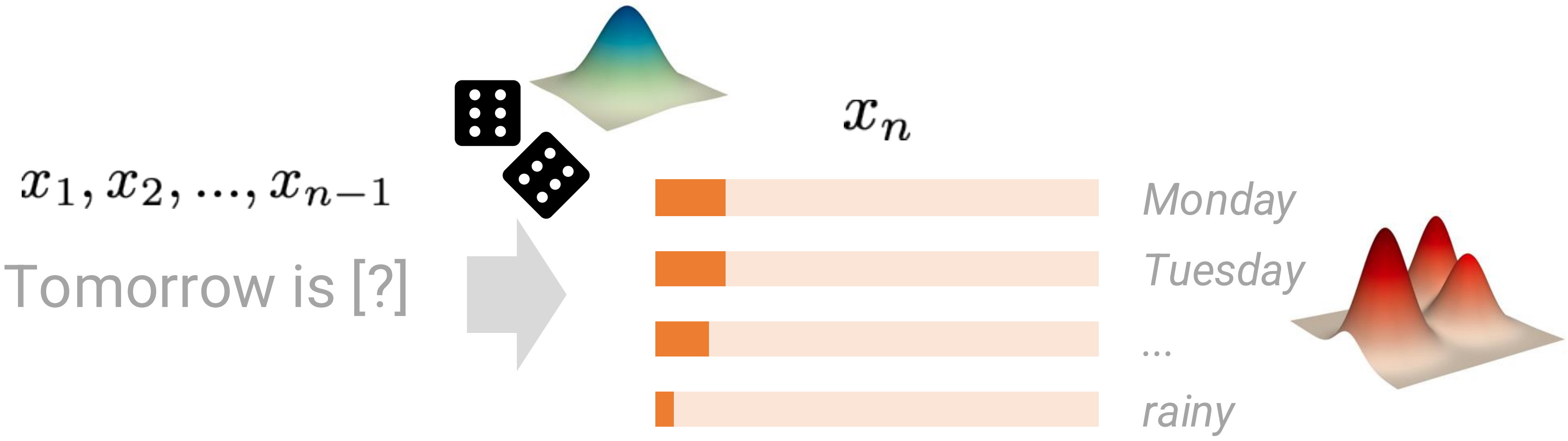
Normalizing Flows



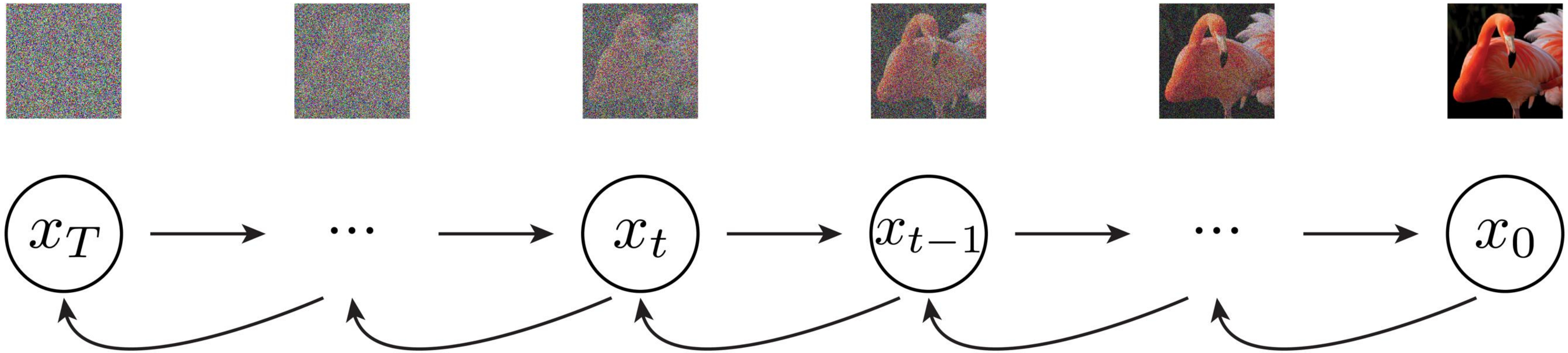
Autoregressive Models (AR)

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

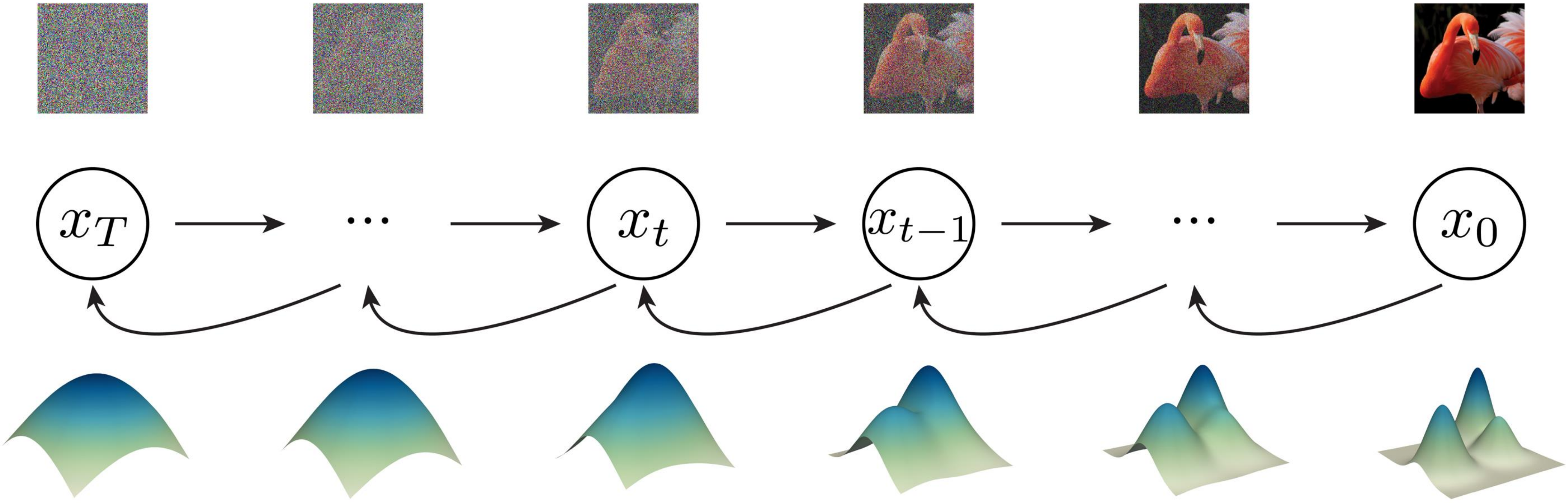




Diffusion Models



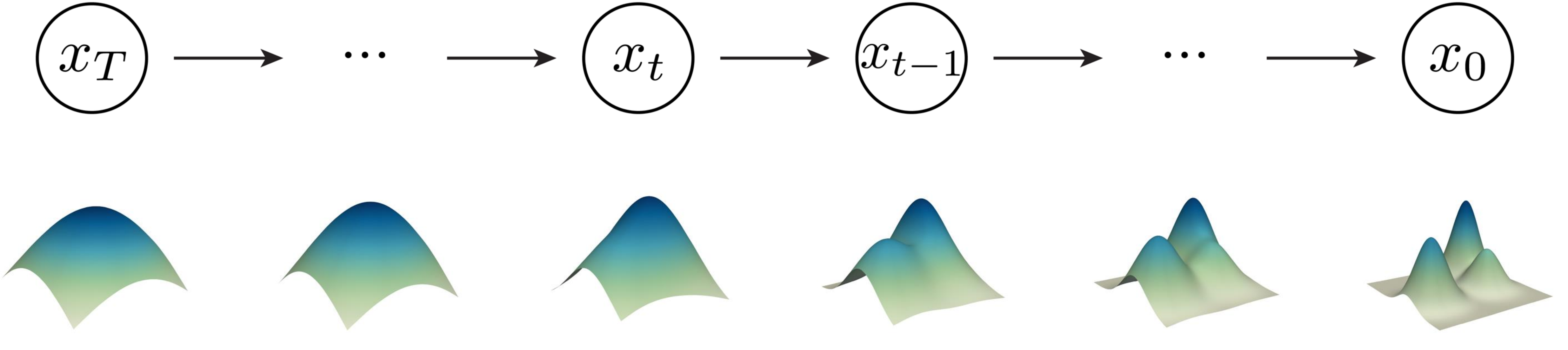
Diffusion Models



noise distribution

data distribution

Diffusion Models



Flow Matching

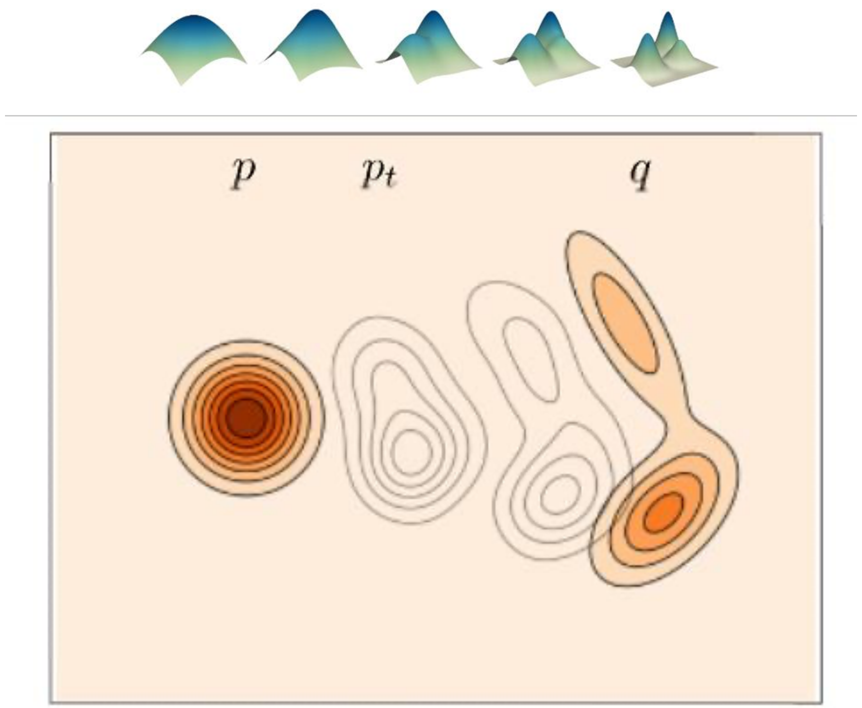
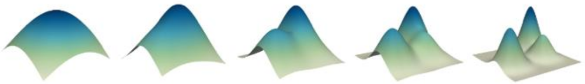


Figure adapted from: https://github.com/facebookresearch/flow_matching

Flow Matching



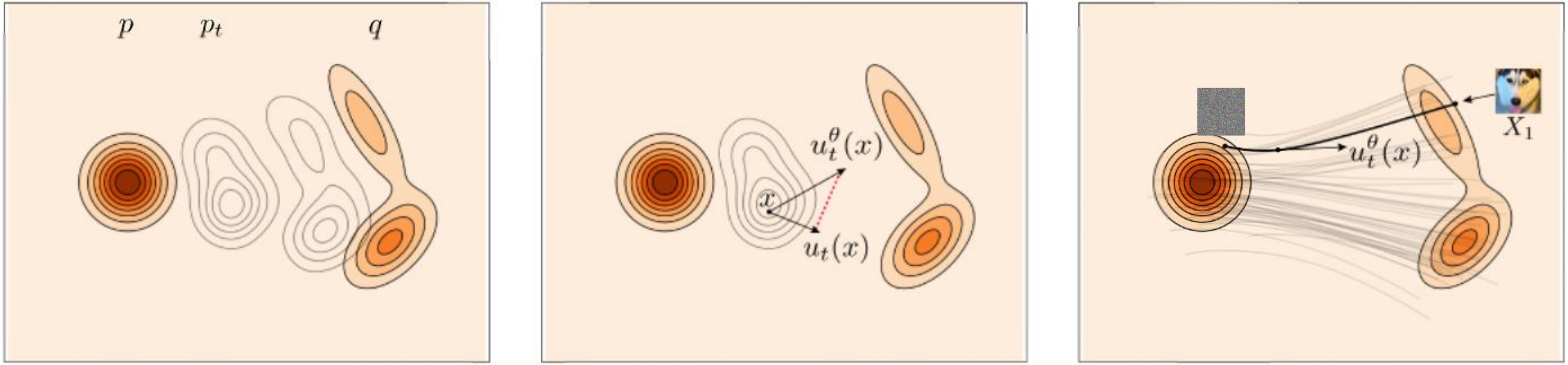


Figure adapted from: https://github.com/facebookresearch/flow_matching

Takeaways

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