

Lecture 1

# Introduction

6.S978 Deep Generative Models

Kaiming He  
EECS, MIT



# The “GenAI” Era

## Chatbot and natural language conversation

What are deep generative models?



Deep generative models are a class of machine learning models that are capable of generating new data samples that resemble a given dataset. They learn the underlying distribution of the data and use this knowledge to create new instances that are similar to the original data but not identical to any specific training example.



Message ChatGPT



# 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

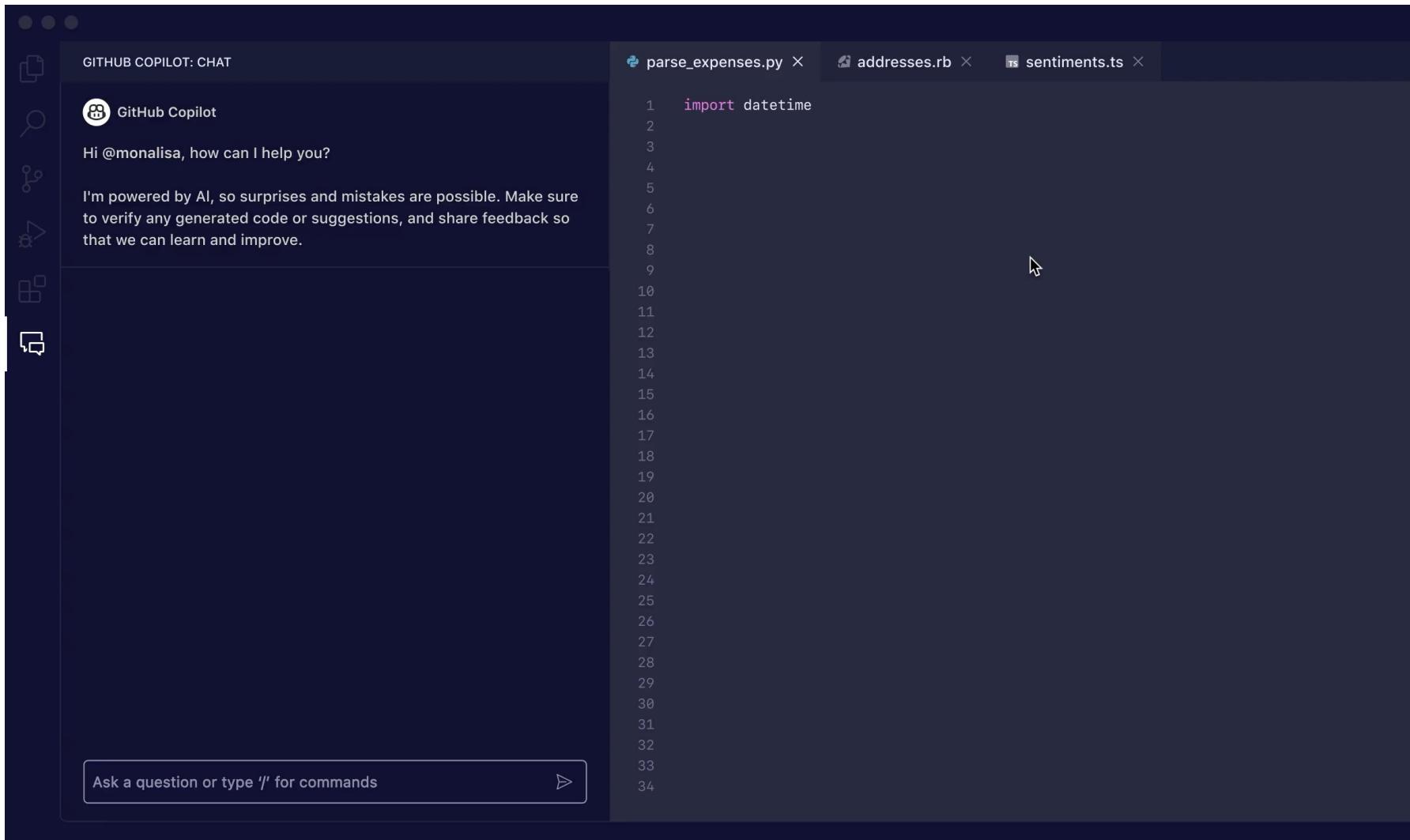
Text-to-video generation



Generated by Sora

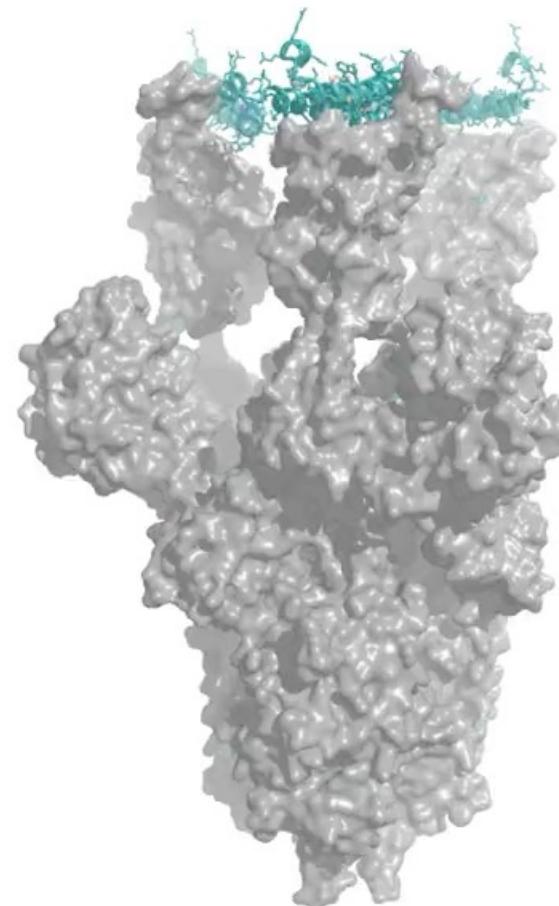
# The “GenAI” Era

## AI assistant for code generation



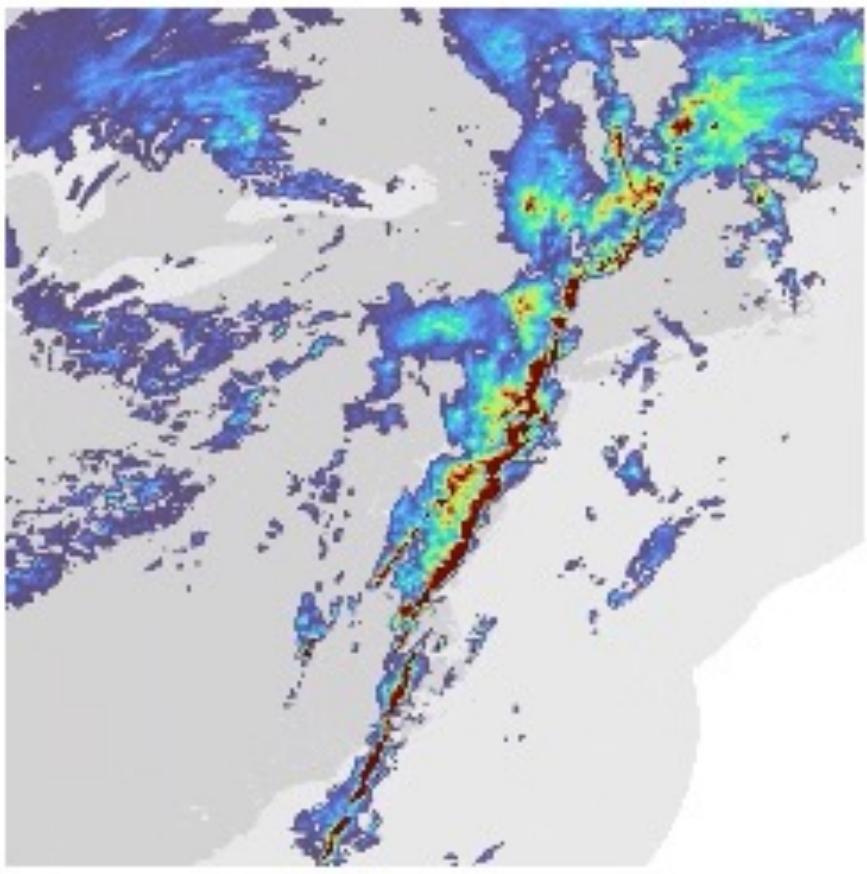
# The “GenAI” Era

Protein design and generation

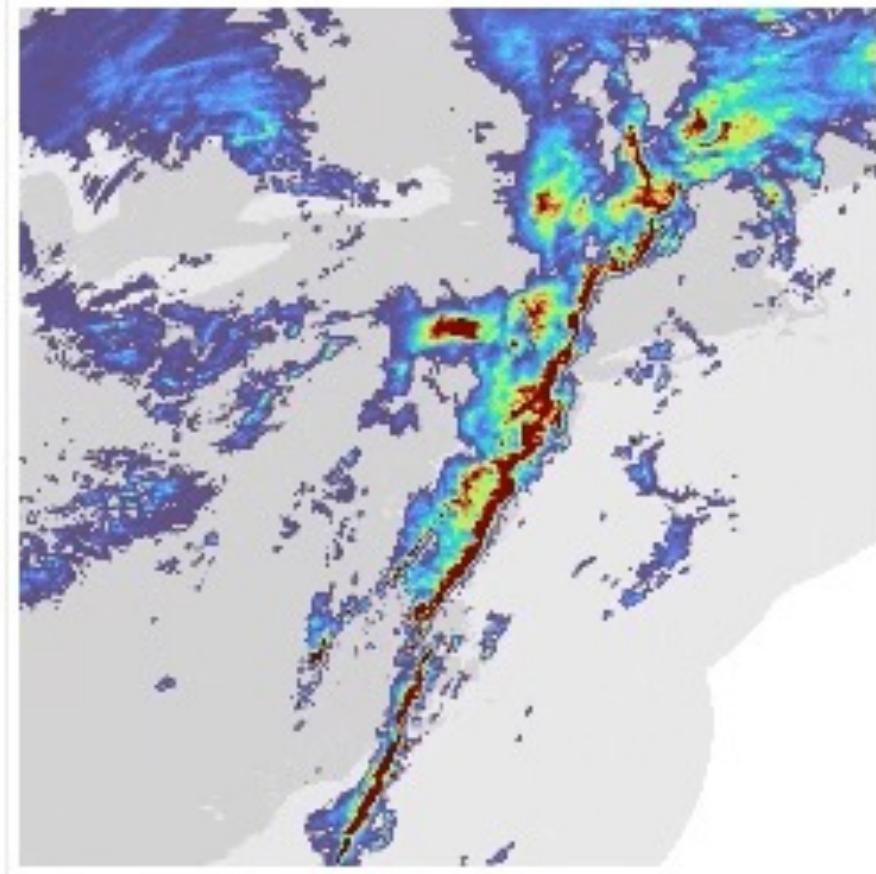


# The “GenAI” Era

Weather forecasting



Target



DGMR

# Generative Models before the “GenAI” Era

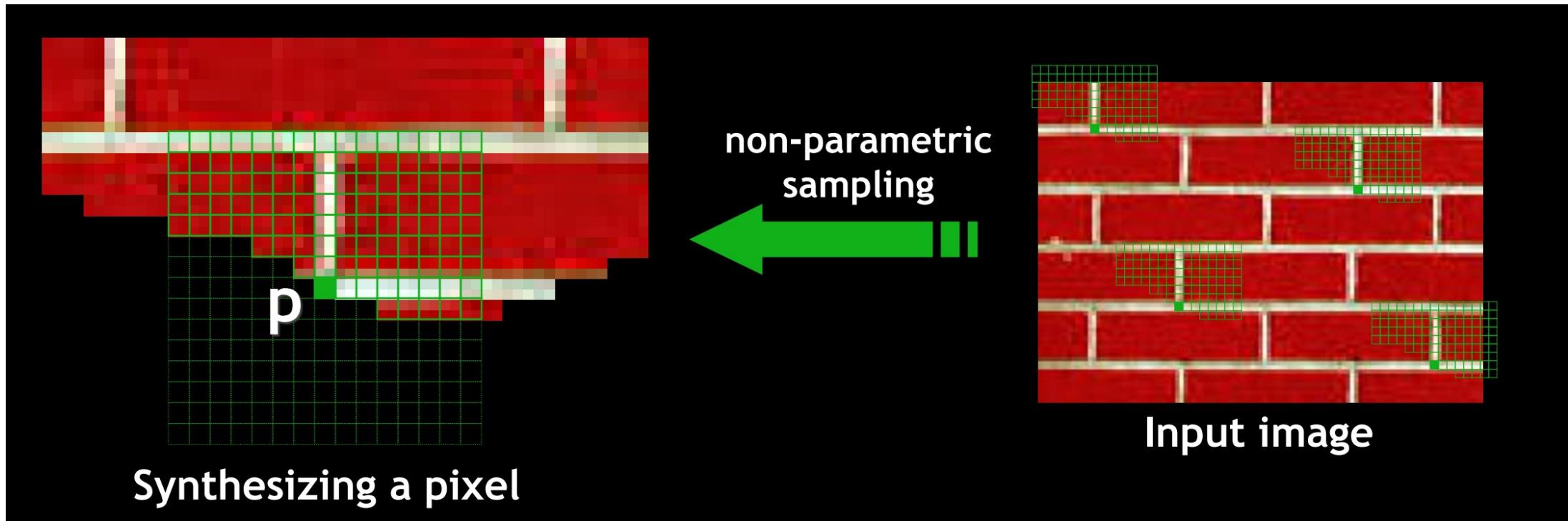
2009, PatchMatch: Photoshop’s Content-aware Fill



# Generative Models before the “GenAI” Era

1999, the Efros-Leung algorithm for texture synthesis

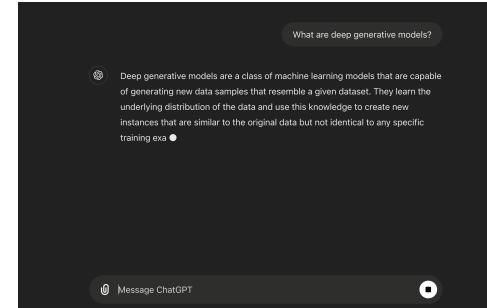
In today's word: this is an **Autoregressive** model



# **What are Generative Models?**

# What do these scenarios have in common?

- There are **multiple** or infinite predictions to one input.
- Some predictions are more “**plausible**” than some others.
- Training data may contain **no exact solution**.
- Predictions may be **more complex**, more informative, and higher-dimensional than input.



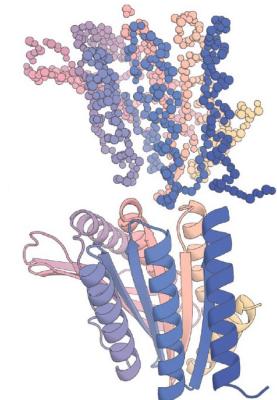
Chatbot



Image generation



Video generation



Protein generation

# Discriminative vs. Generative models

## discriminative

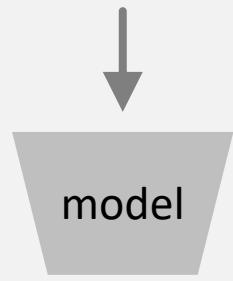
- “sample”  $x \Rightarrow$  “label”  $y$
- one desired output

## generative

- “label”  $y \Rightarrow$  “sample”  $x$
- many possible outputs

## discriminative

$x$



## generative

$y$

“dog”



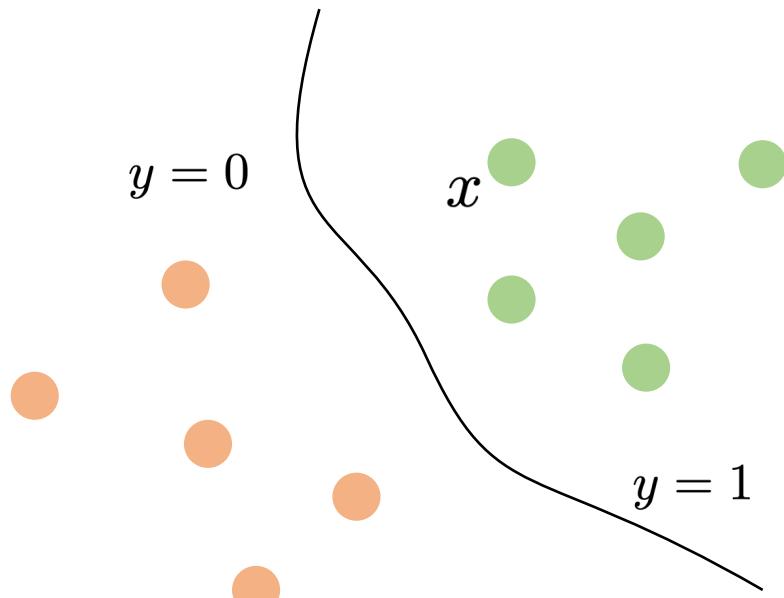
model



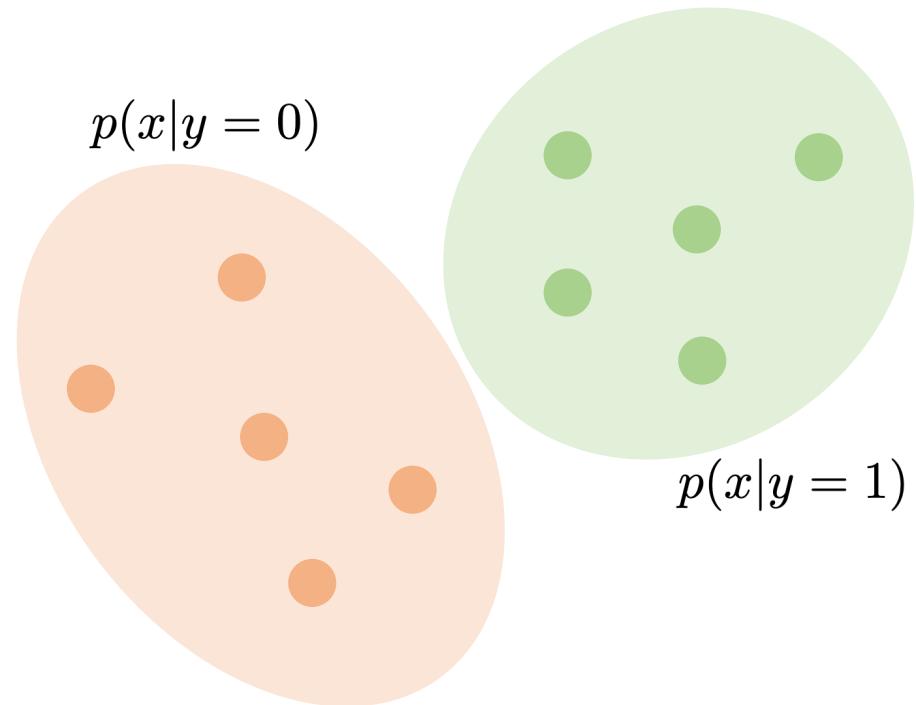
$x$

# Discriminative 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

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

discriminative      generative

assuming known prior  
constant for given x

The diagram illustrates Bayes' rule with two arrows pointing to different parts of the equation. A grey arrow points to the term  $p(y)$ , which is labeled 'assuming known prior'. Another grey arrow points to the term  $\frac{p(x)}{p(y)}$ , which is labeled 'constant for given x'. The terms  $p(y|x)$  and  $p(x|y)$  are highlighted with yellow and blue boxes respectively. Below the equation, the words 'discriminative' and 'generative' are written in orange and blue, corresponding to the highlighted terms.

- Generative models can be discriminative: Bayes' rule

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

discriminative      generative

The diagram illustrates Bayes' rule with two arrows pointing to the right from the terms  $p(x|y)$  and  $\frac{p(y)}{p(x)}$ . The arrow from  $p(x|y)$  is labeled "discriminative" below it. The arrow from  $\frac{p(y)}{p(x)}$  is labeled "generative" below it. To the right of the equation, the text "assuming known prior" is written above "constant for given  $x$ ".

- Can discriminative models be generative?

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

generative      discriminative

The diagram shows the equation  $p(x|y) = p(y|x) \frac{p(x)}{p(y)}$ . The term  $p(y|x)$  is highlighted in orange and labeled "discriminative" below it. The term  $\frac{p(x)}{p(y)}$  is highlighted in blue and labeled "generative" below it. To the right of the equation, the text "still need to model prior distribution of  $x$ " is written above "constant for given  $y$ ".

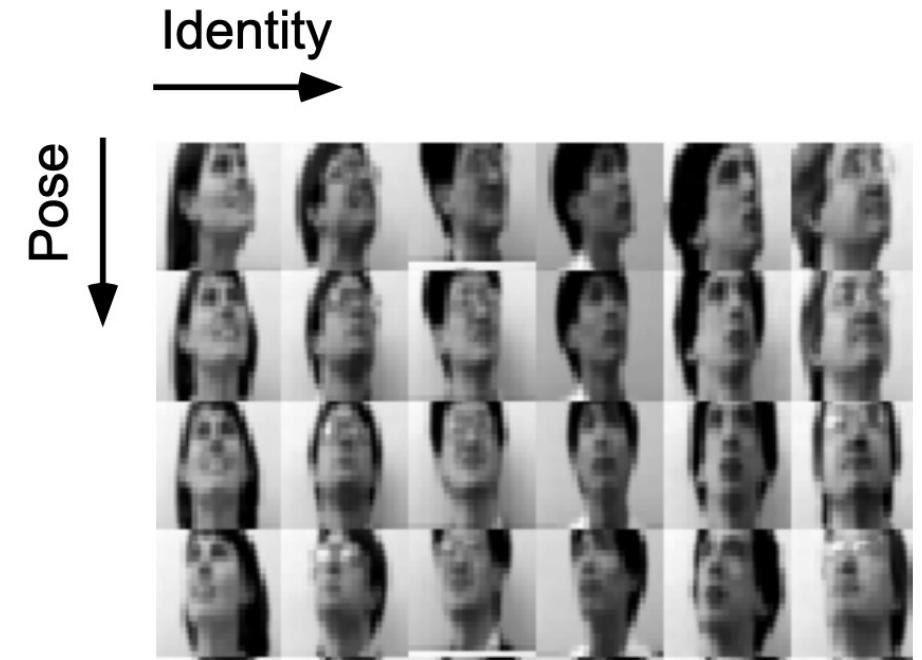
- The challenge is about representing and predicting distributions

# Probabilistic modeling

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

example:

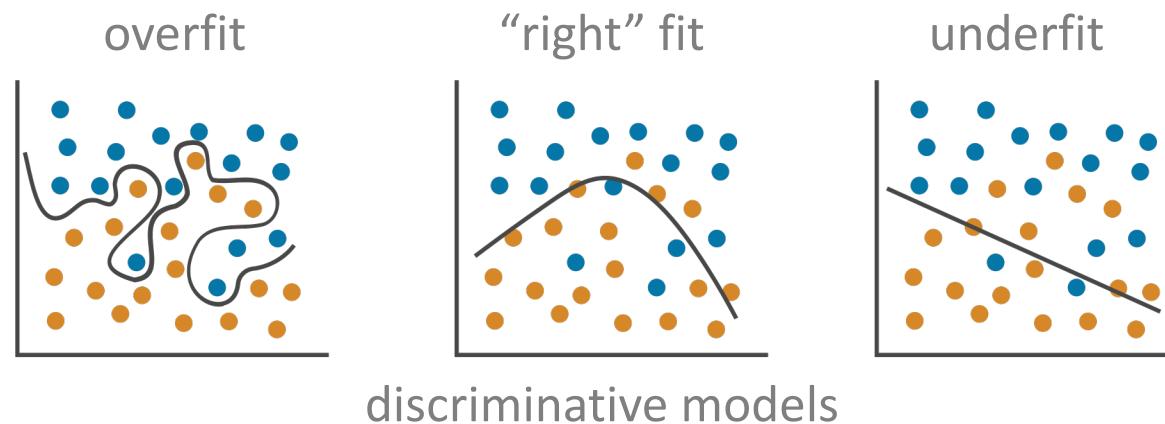
- latent factors  $z$  (pose, lighting, scale, ...)
- $z$  has simple distributions
- observations  $x$  are rendered by a “world model” that’s a function on  $z$
- observations  $x$  have complex distributions



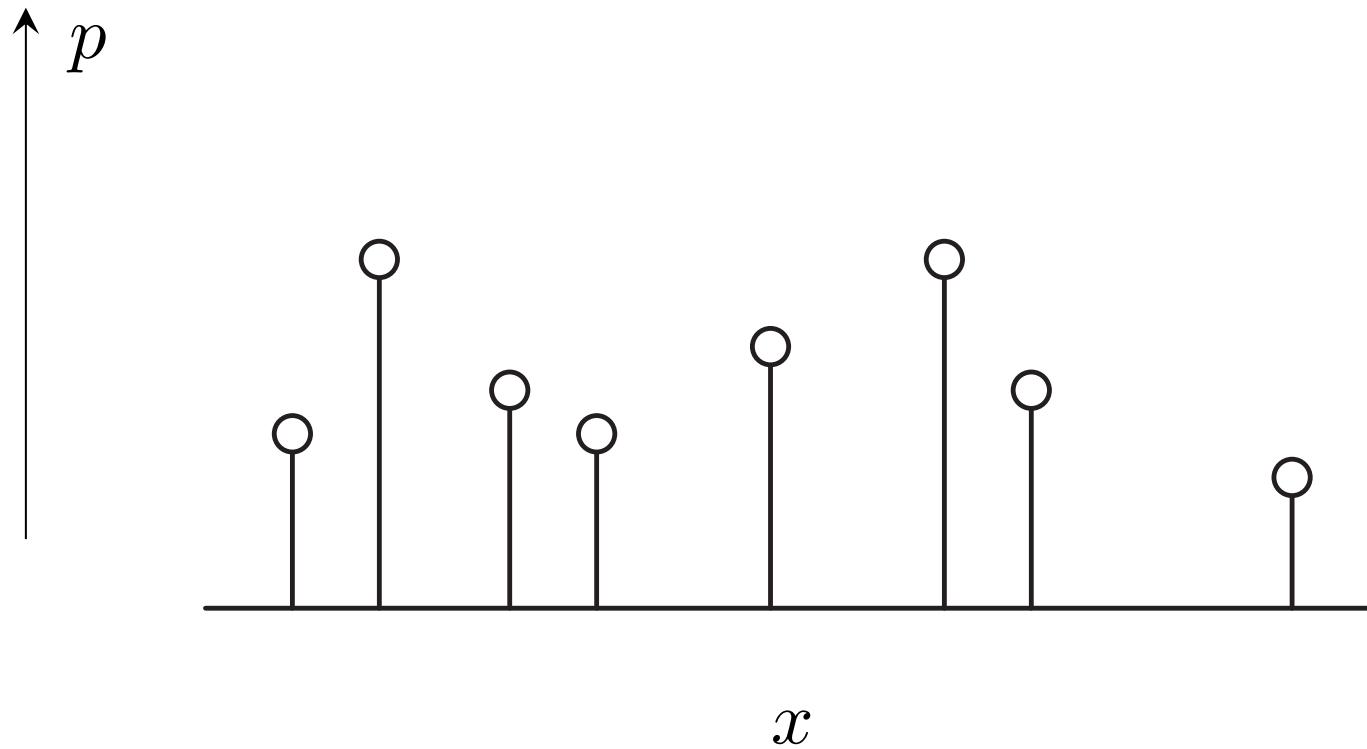
- Probability is part of the modeling.

# Probability is part of the modeling

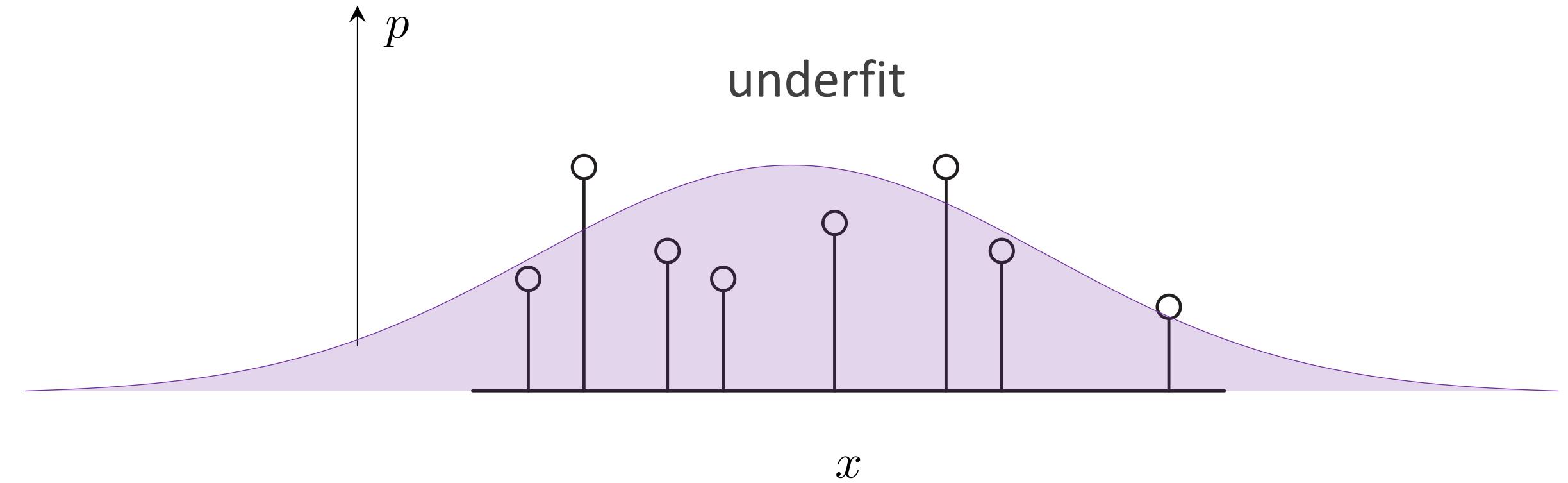
- There may not be “underlying” distributions.
- Even there are, what we can observe are a **finite** set of data points
- The models **extrapolate** the observations for modeling distributions
- Overfitting vs. underfitting: like discriminative models



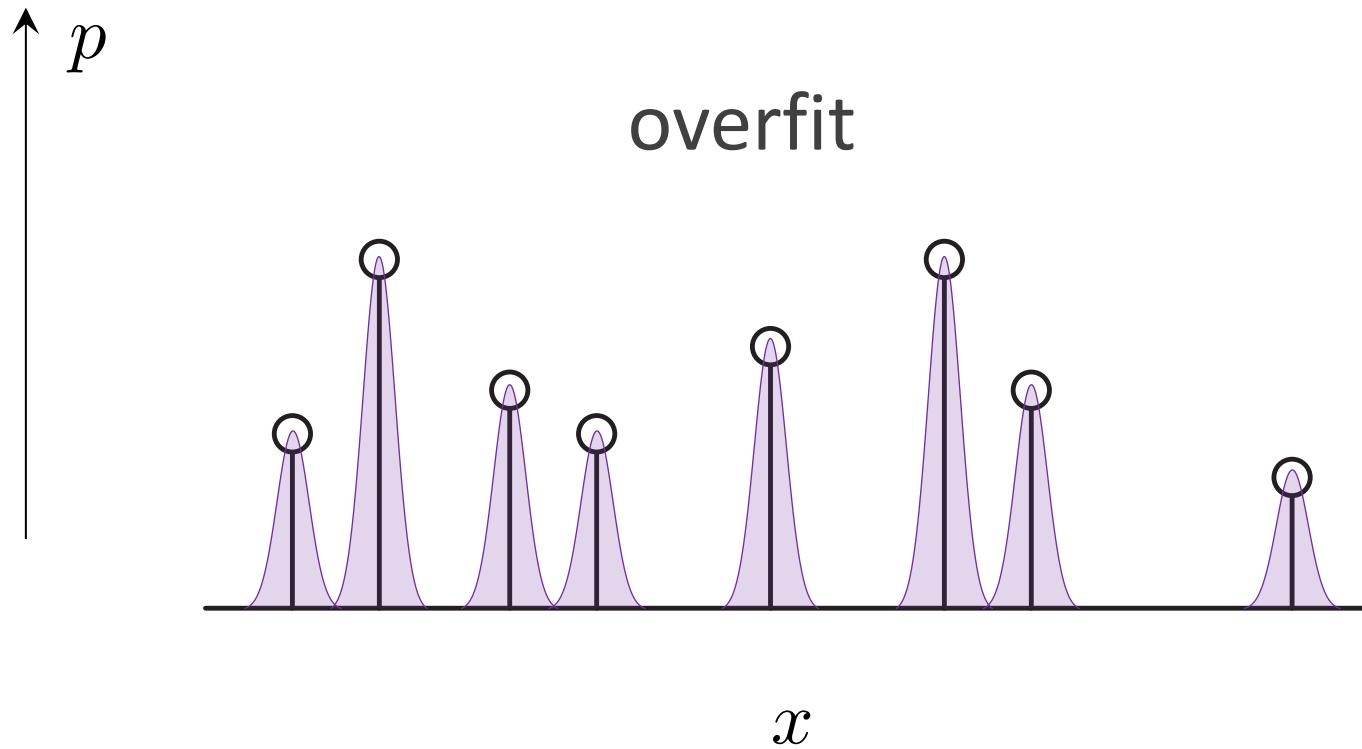
# Probability is part of the modeling



# Probability is part of the modeling

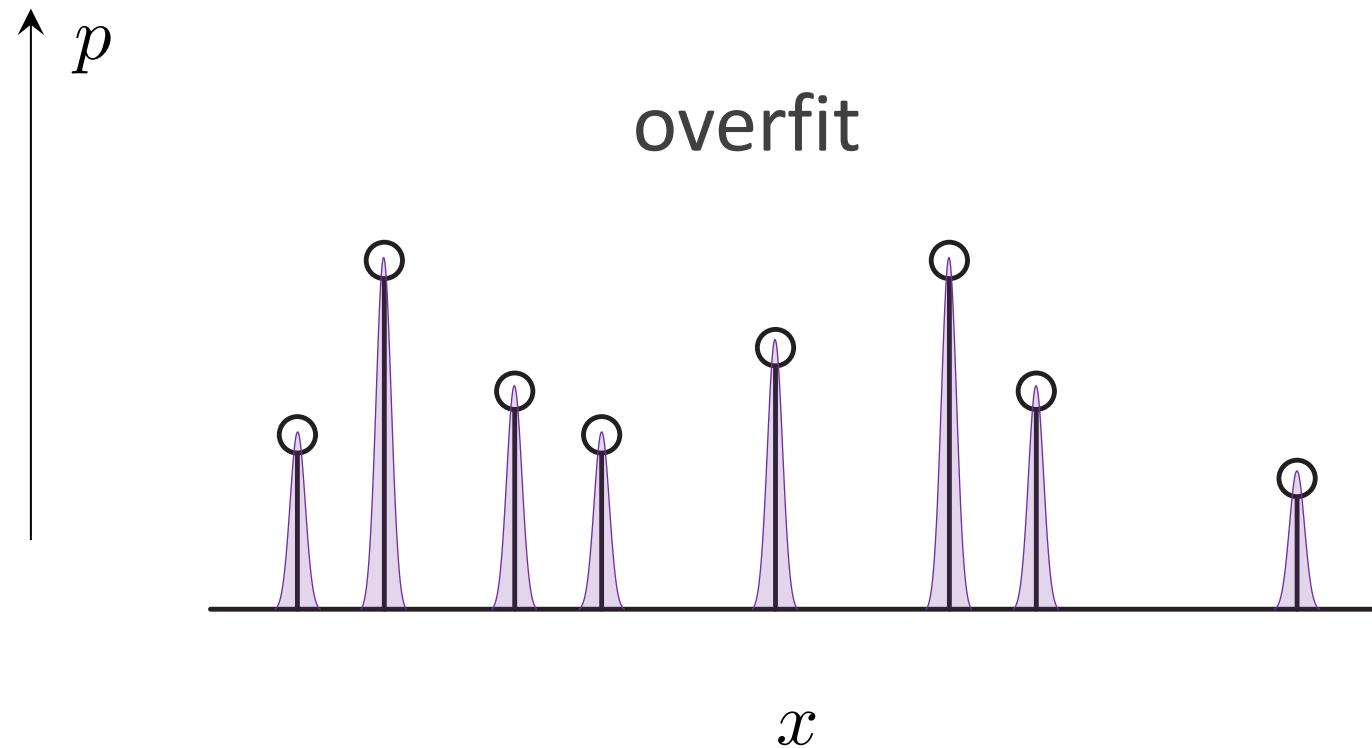


# Probability is part of the modeling



# Probability is part of the modeling

- To the extreme, using delta functions is like sampling from training data

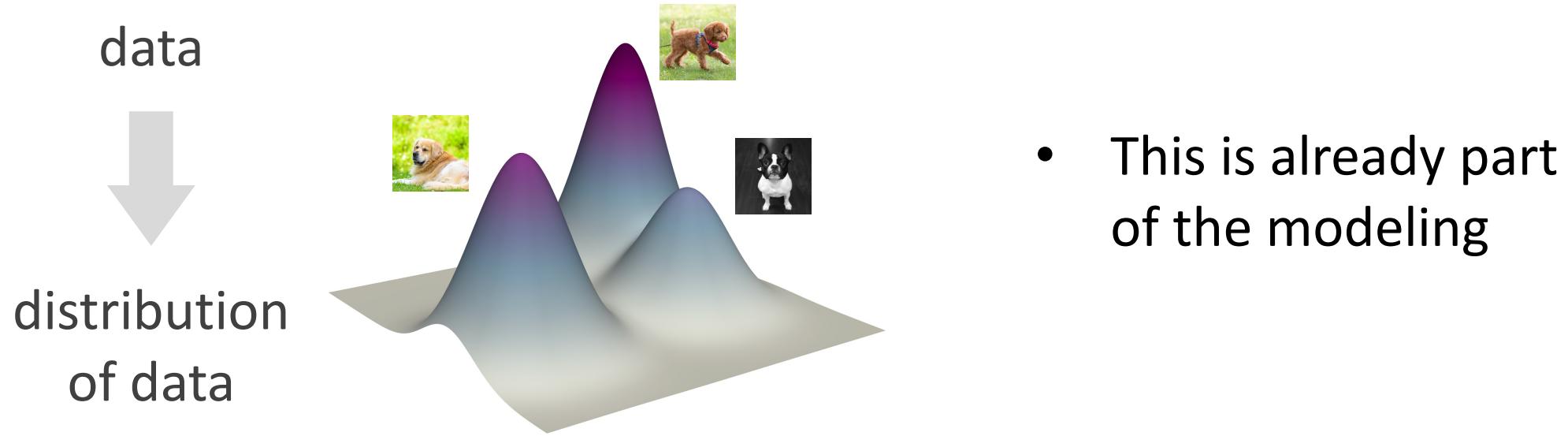


# Generative models w/ probabilistic modeling

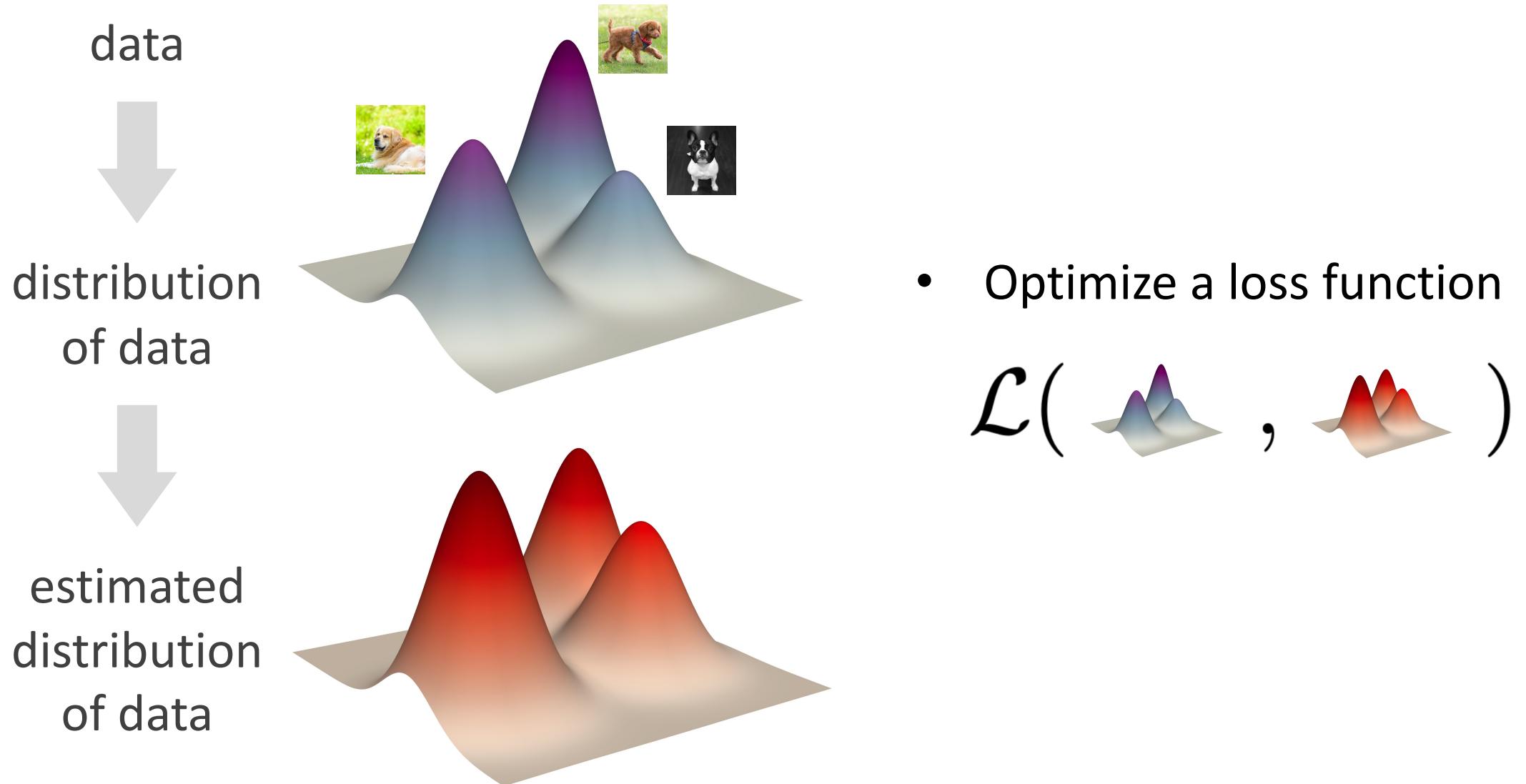
data



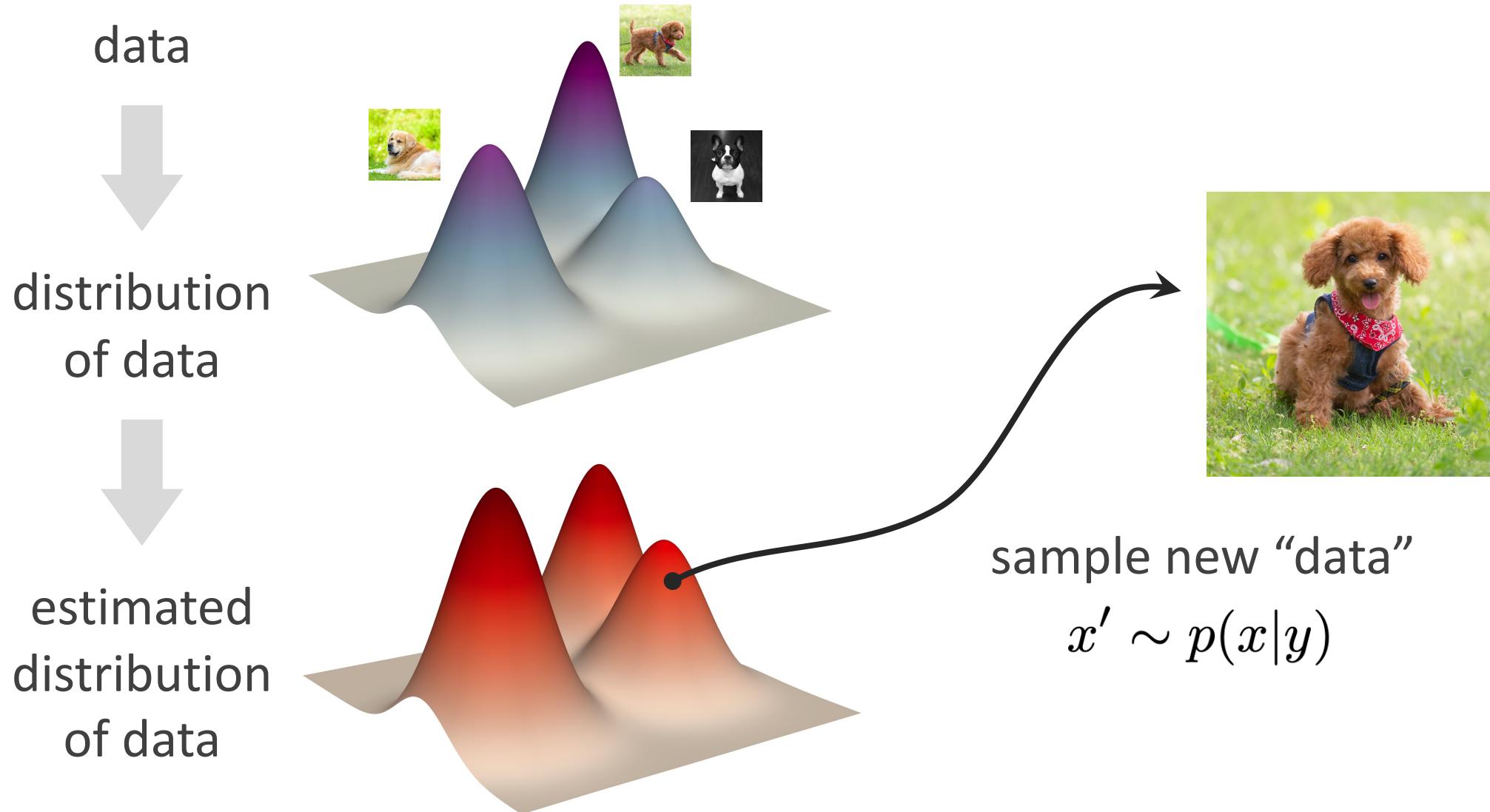
# Generative models w/ probabilistic modeling



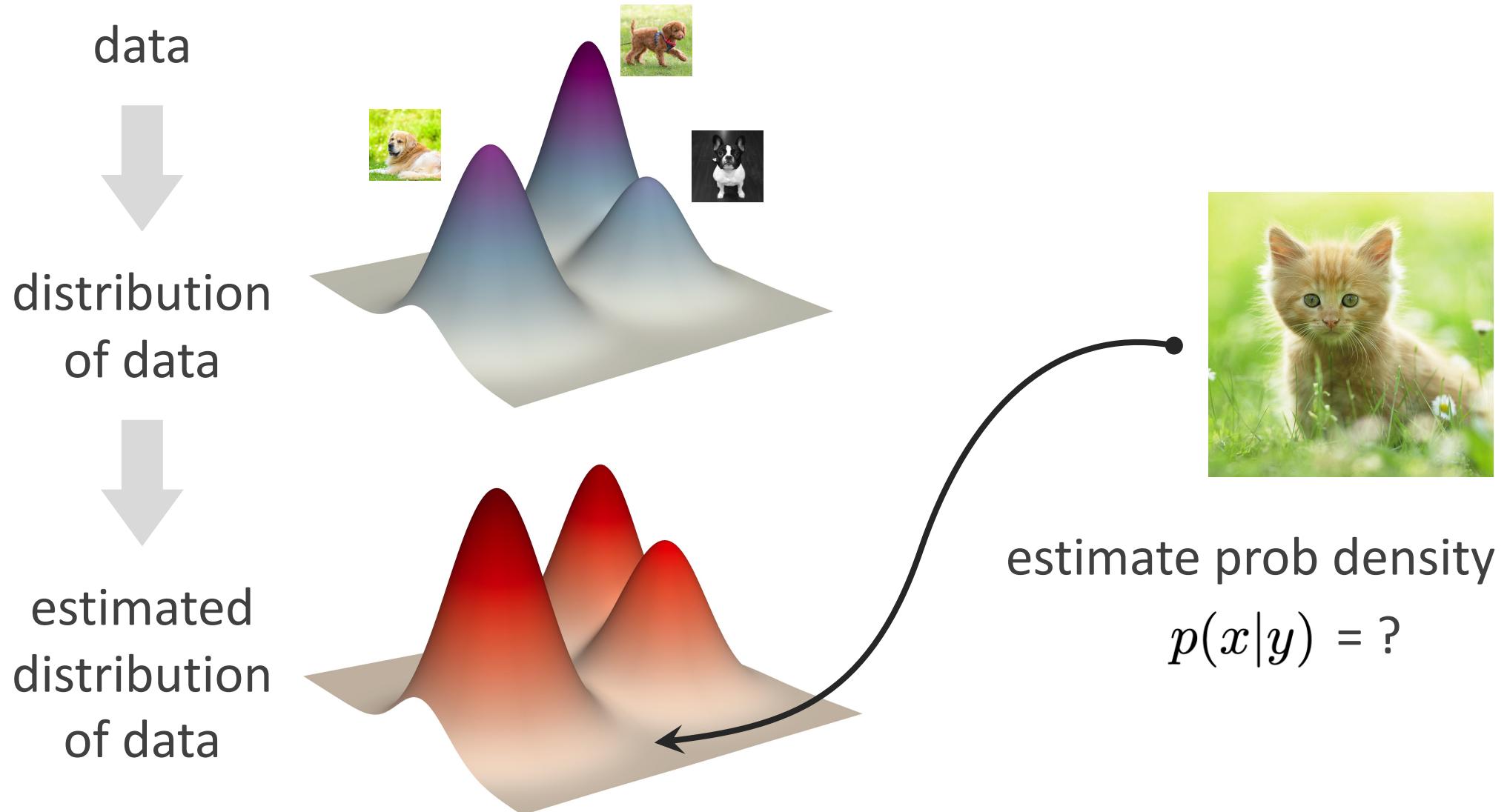
# Generative models w/ probabilistic modeling



# Generative models w/ probabilistic modeling



# Generative models w/ probabilistic modeling



# Generative models w/ probabilistic modeling

## Notes:

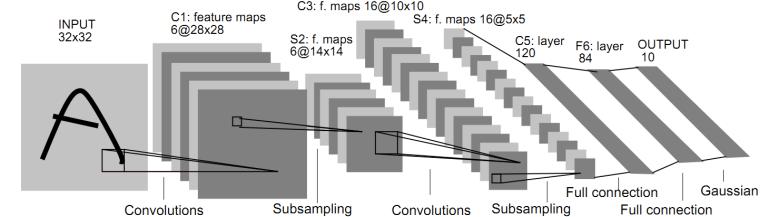
- Generative models involve statistical models which are often designed and derived by humans.
- Probabilistic modeling is not just the work of neural nets.
- Probabilistic modeling is a popular way, but not the only way.
- "*All models are wrong, but some are useful.*" - George Box

**What are Deep Generative Models?**

# Deep Generative Models

- Deep learning is **representation learning**
- Learning to represent data instances
  - map data to feature:  $x \rightarrow f(x)$
  - minimize loss w/ target:  $\mathcal{L}(y, f(x))$

$$x \longrightarrow f(x)$$



# Deep Generative Models

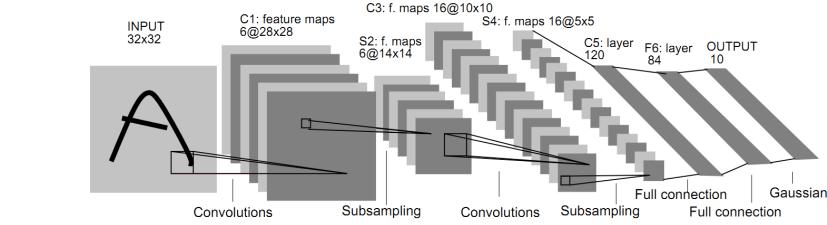
- Deep learning is **representation learning**

$$x \rightarrow f(x)$$

- Learning to represent data instances

- map data to feature:  $x \rightarrow f(x)$

- minimize loss w/ target:  $\mathcal{L}(y, f(x))$



$$\pi \leftarrow g(\pi)$$

- Learning to **represent probability distributions**

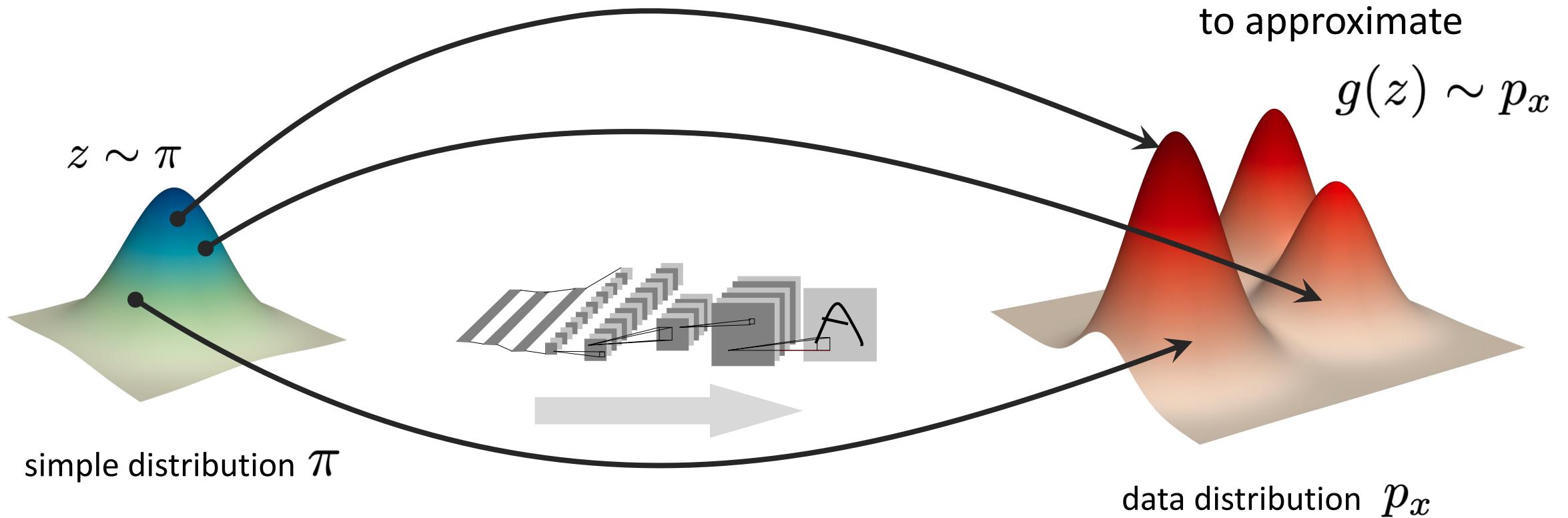
- map a simple distribution (Gaussian/uniform) to a complex one:  $\pi \rightarrow g(\pi)$

- minimize loss w/ data distribution:  $\mathcal{L}(p_x, g(\pi))$

- Often perform both together

# Learning to represent probability distributions

- From simple to complex distributions

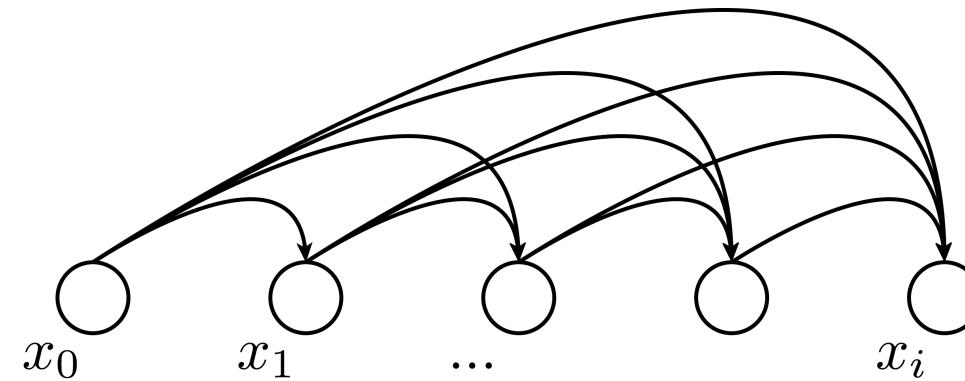


# Learning to represent probability distributions

- Not all parts of distribution modeling is done by learning

**Case study:**  
**Autoregressive model**

This dependency graph is  
designed (not learned).

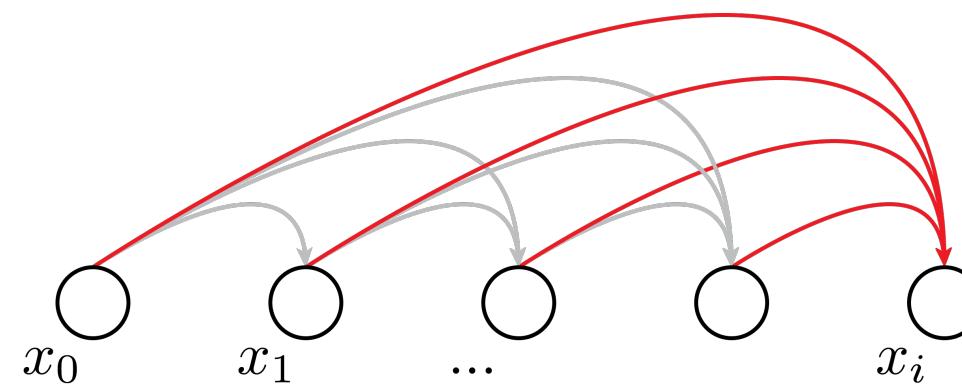


# Learning to represent probability distributions

- Not all parts of distribution modeling is done by learning

**Case study:**  
**Autoregressive model**

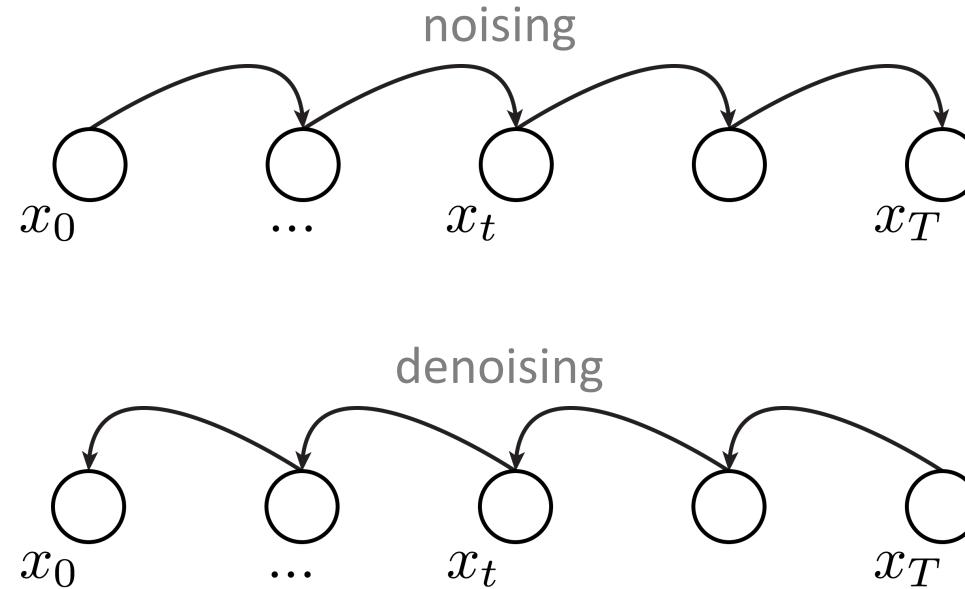
The mapping function is learned  
(e.g., Transformer)



# Learning to represent probability distributions

- Not all parts of distribution modeling is done by learning

## Case study: Diffusion model

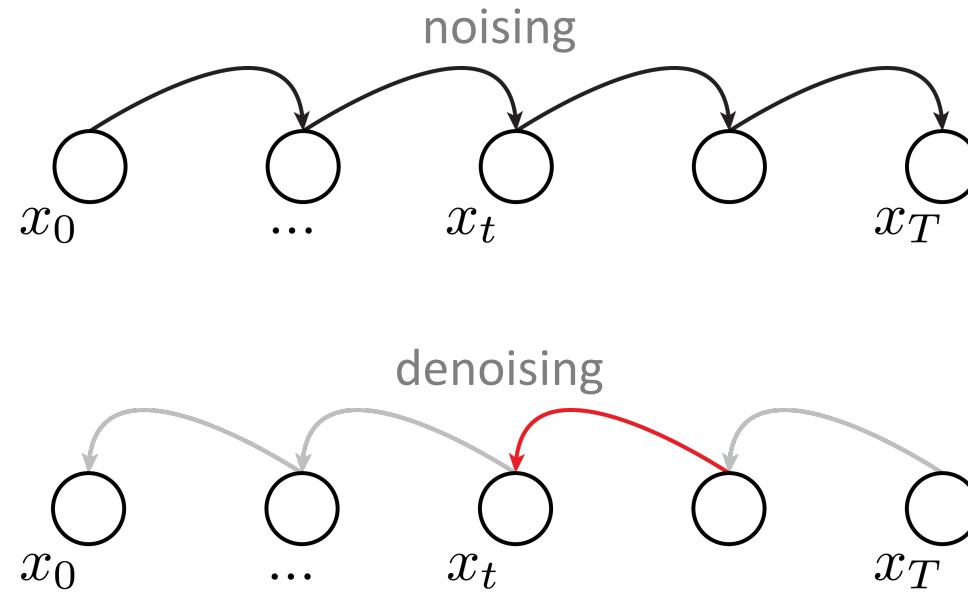


This dependency graph is designed (not learned).

# Learning to represent probability distributions

- Not all parts of distribution modeling is done by learning

## Case study: Diffusion model



The mapping function is learned  
(e.g., Unet)

# Deep Generative Models may involve:

- **Formulation:**
  - formulate a problem as probabilistic modeling
  - decompose complex distributions into simple and tractable ones
- **Representation:** deep neural networks to represent data and their distributions
- **Objective function:** to measure how good the predicted distribution is
- **Optimization:** optimize the networks and/or the decomposition
- **Inference:**
  - sampler: to produce new samples
  - probability density estimator (optional)

# **Formulating Real-world Problems as Generative Models**

# Formulating Real-world Problems as Generative Models

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

**What can be  $y$ ?**

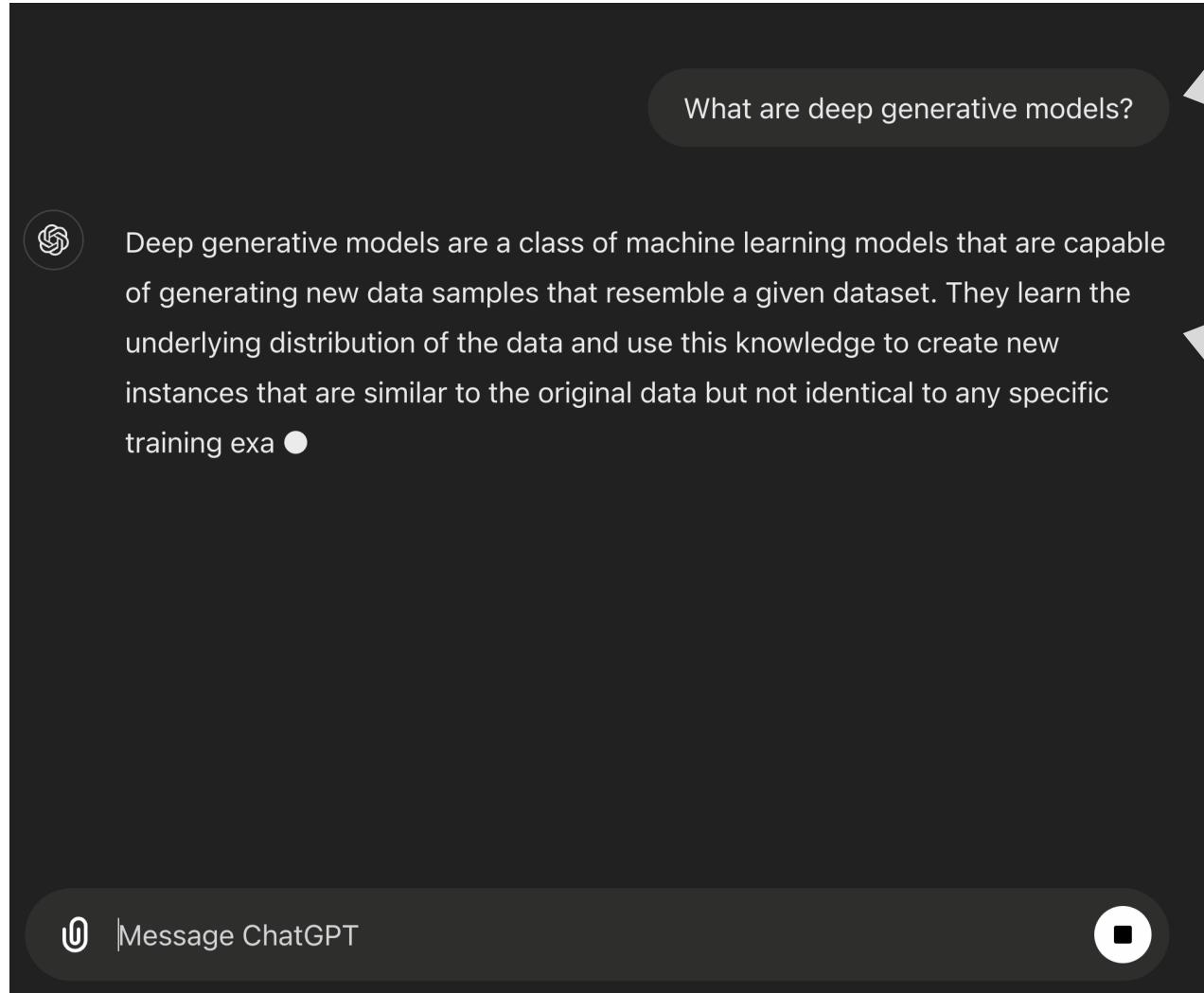
- condition
- constraint
- labels
- attributes
  
- more abstract
- less informative

**What can be  $x$ ?**

- “data”
- samples
- observations
- measurements
  
- more concrete
- more informative

# Case study: Formulating as $p(x|y)$

- Natural language conversation



$y$ : prompt

$x$ : response of the chatbot

# Case study: Formulating as $p(x|y)$

- **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

# Case study: Formulating as $p(x|y)$

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



“motorcycle”



“mech suit”



“ghost lantern”



“furry fox head”



“dresser”



“swivel chair”



“astronaut”



“mushroom house”



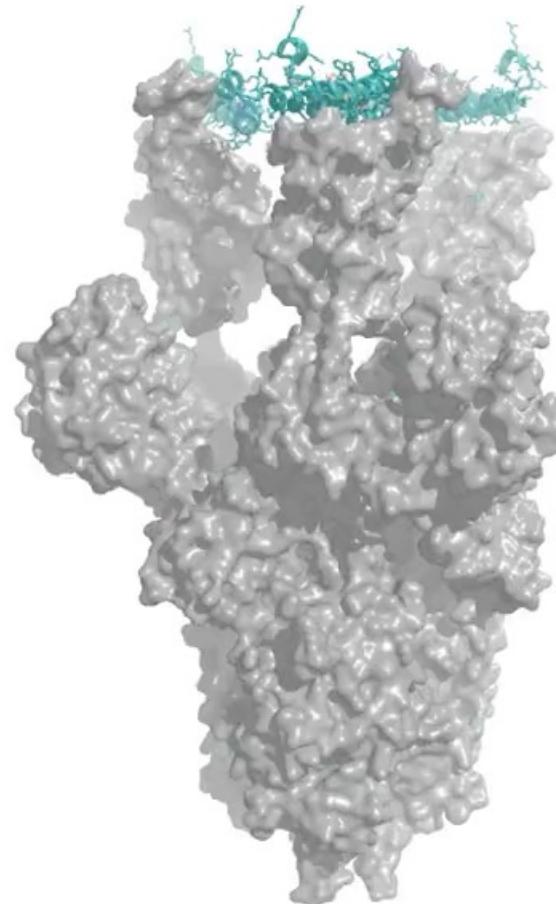
$x$ : generated  
3D structures

$y$ : text prompt

# Case study: Formulating as $p(x|y)$

- Protein structure generation

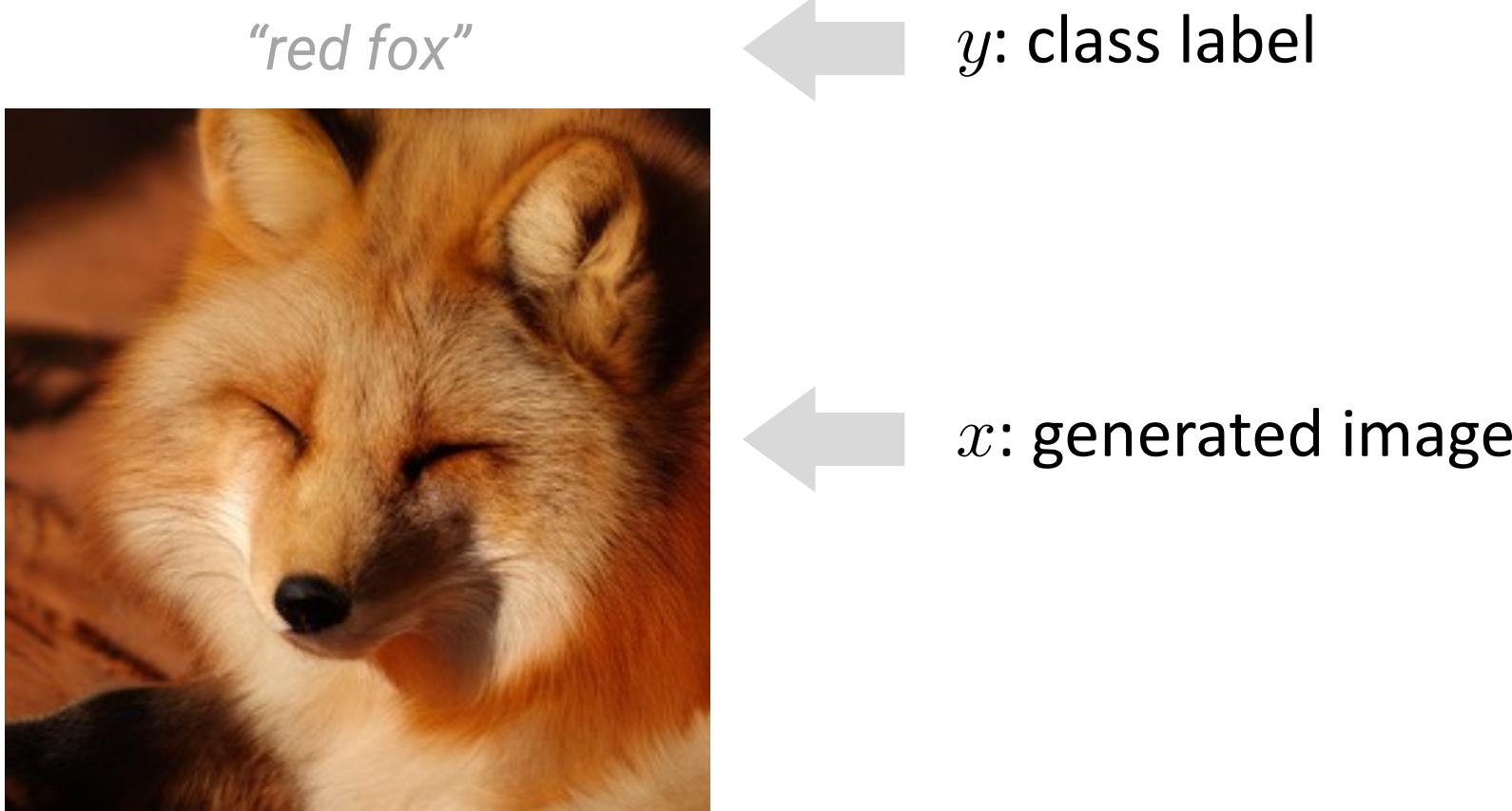
$y$ : condition/constraint  
(e.g., symmetry)



$x$ : generated  
protein structures

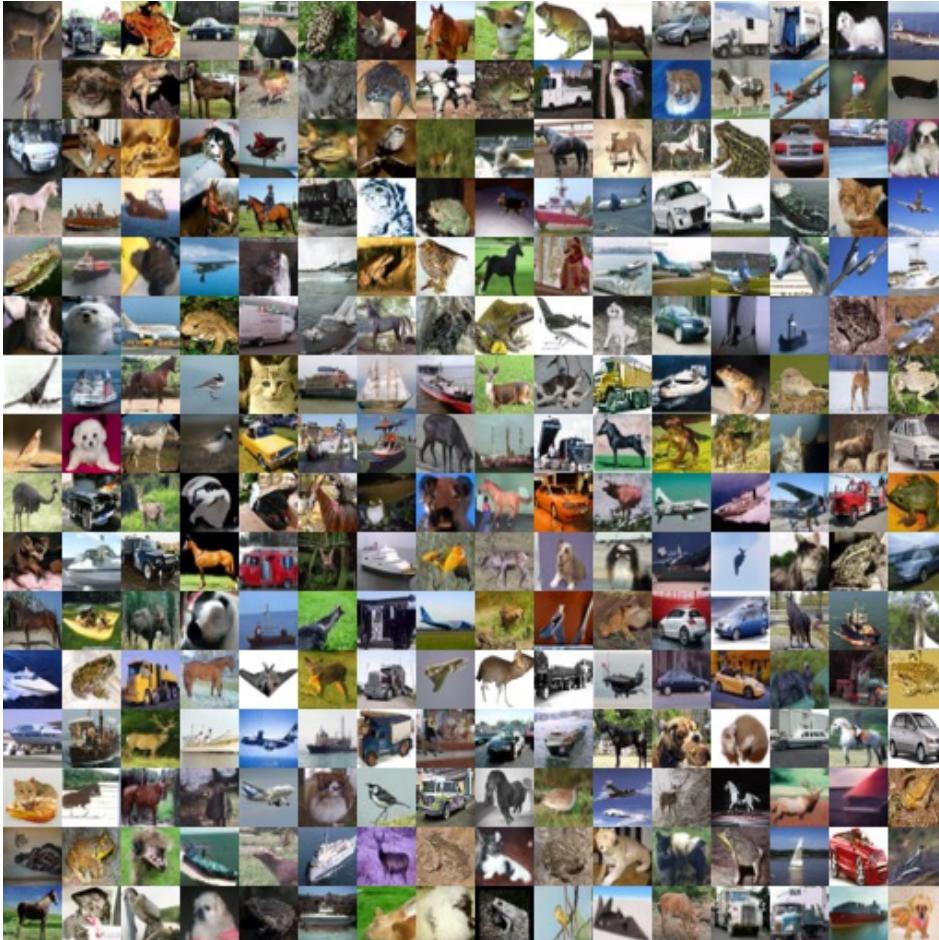
# Case study: Formulating as $p(x|y)$

- Class-conditional image generation



# Case study: Formulating as $p(x|y)$

- “Unconditional” image generation



$y$ : an implicit condition

“*images following CIFAR10 distribution*”

$x$ : generated CIFAR10-like images

- $p(x|y)$ : images  $\sim$  CIFAR10
- $p(x)$ : all images

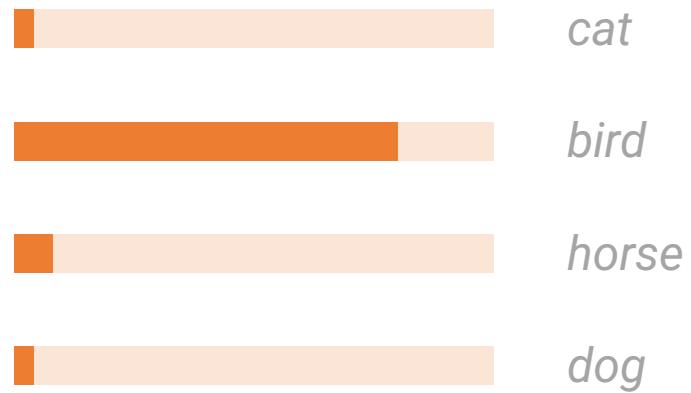
# Case study: Formulating as $p(x|y)$

- **Classification** (a generative perspective)

$y$ : an image as the “condition”



$x$ : probability of classes  
conditioned on the image



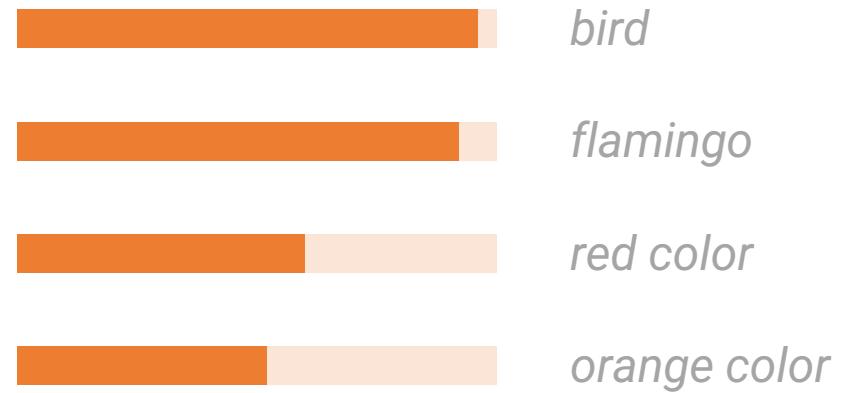
# Case study: Formulating as $p(x|y)$

- **Open-vocabulary recognition**

$y$ : an image as the “condition”



$x$ : plausible descriptions conditioned on the image



# Case study: Formulating as $p(x|y)$

- **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.

# Case study: Formulating as $p(x|y)$

- Chatbot with visual inputs

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

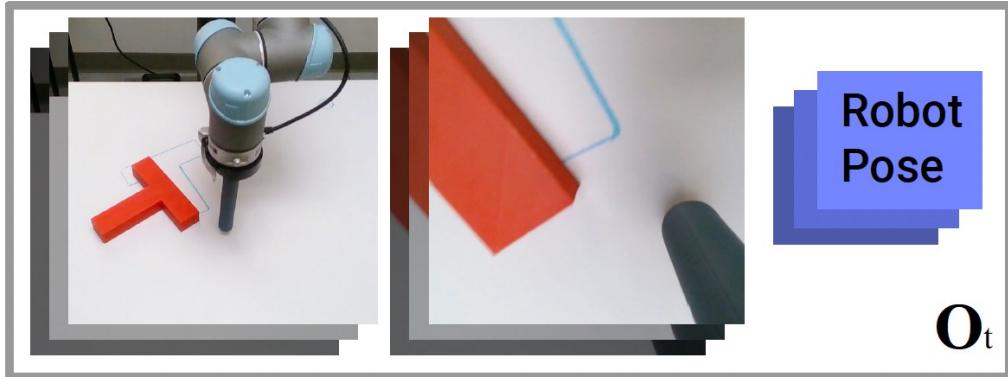
The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

$x$ : response of the chatbot

# Case study: Formulating as $p(x|y)$

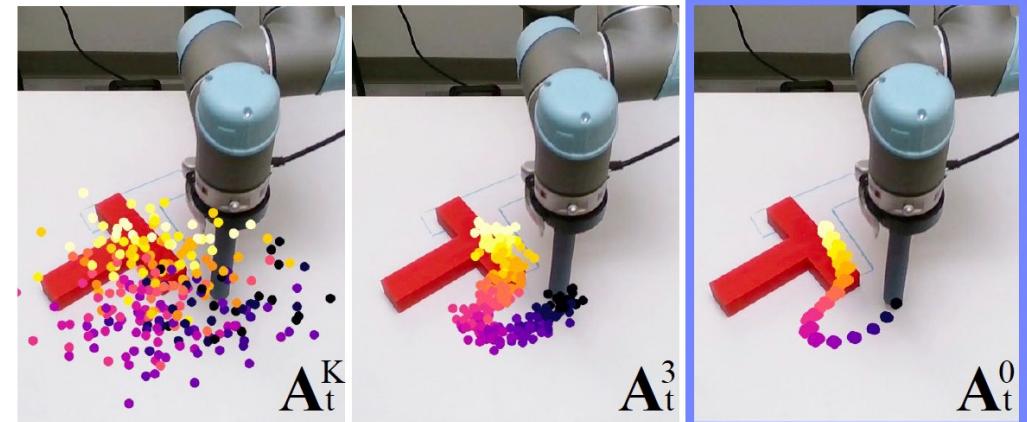
- **Policy Learning in Robotics**

$y$ : visual and other sensory observations



$x$ : policies

(probability of actions)



# Formulating Real-world Problems as Generative Models

- Generative models are about  $p(x|y)$
- Many problems can be formulated as generative models
- What's  $x$ ? What's  $y$ ?
- How to represent  $x$ ,  $y$ , and their dependence?

# About this course

This course will cover:

- How real-world problems are formulated as generative models?
- Probabilistic foundations and learning algorithms
- Challenges, opportunities, open questions