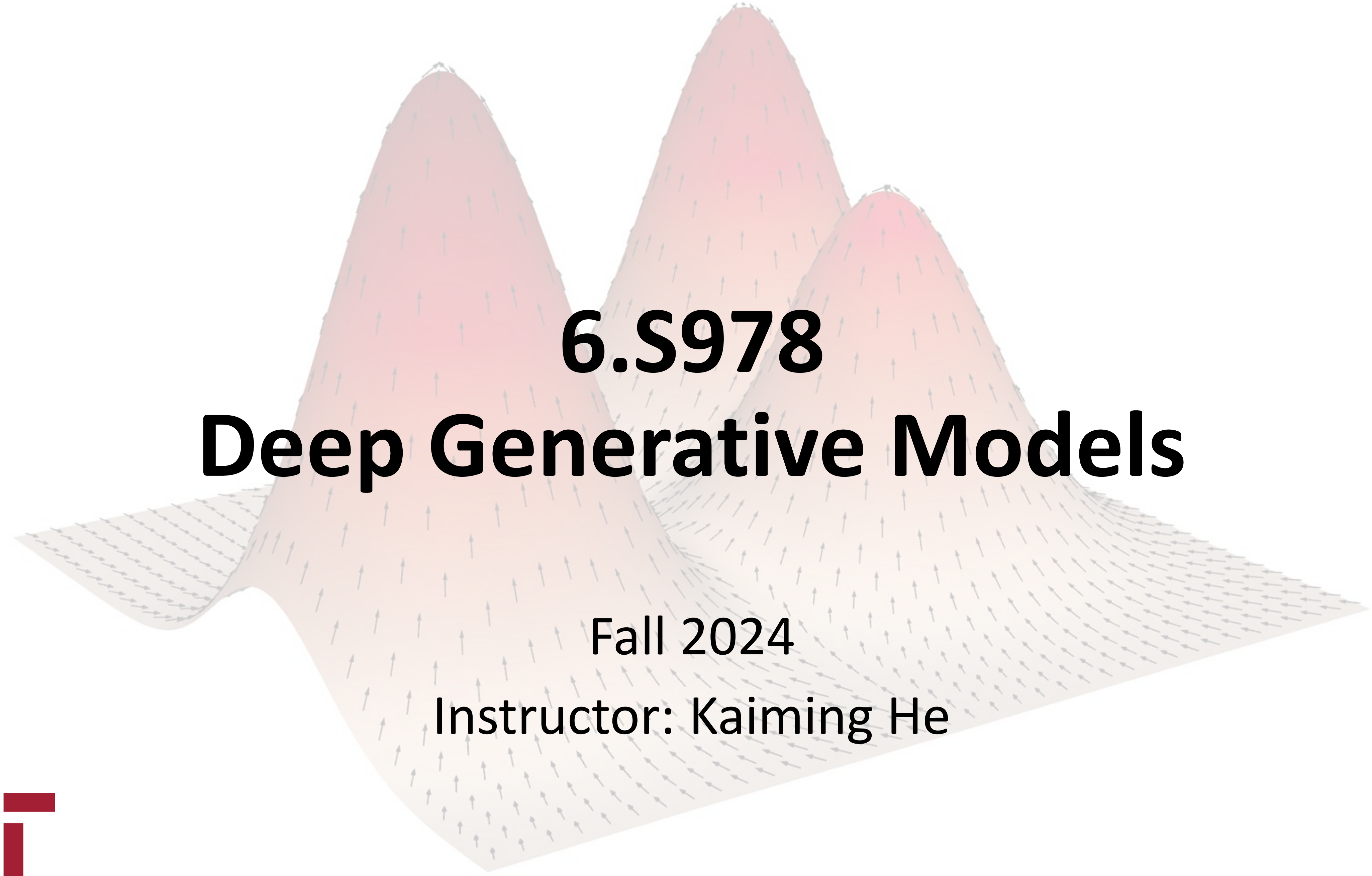


6.S978
Deep Generative Models

Fall 2024

Instructor: Kaiming He

Lecture 1

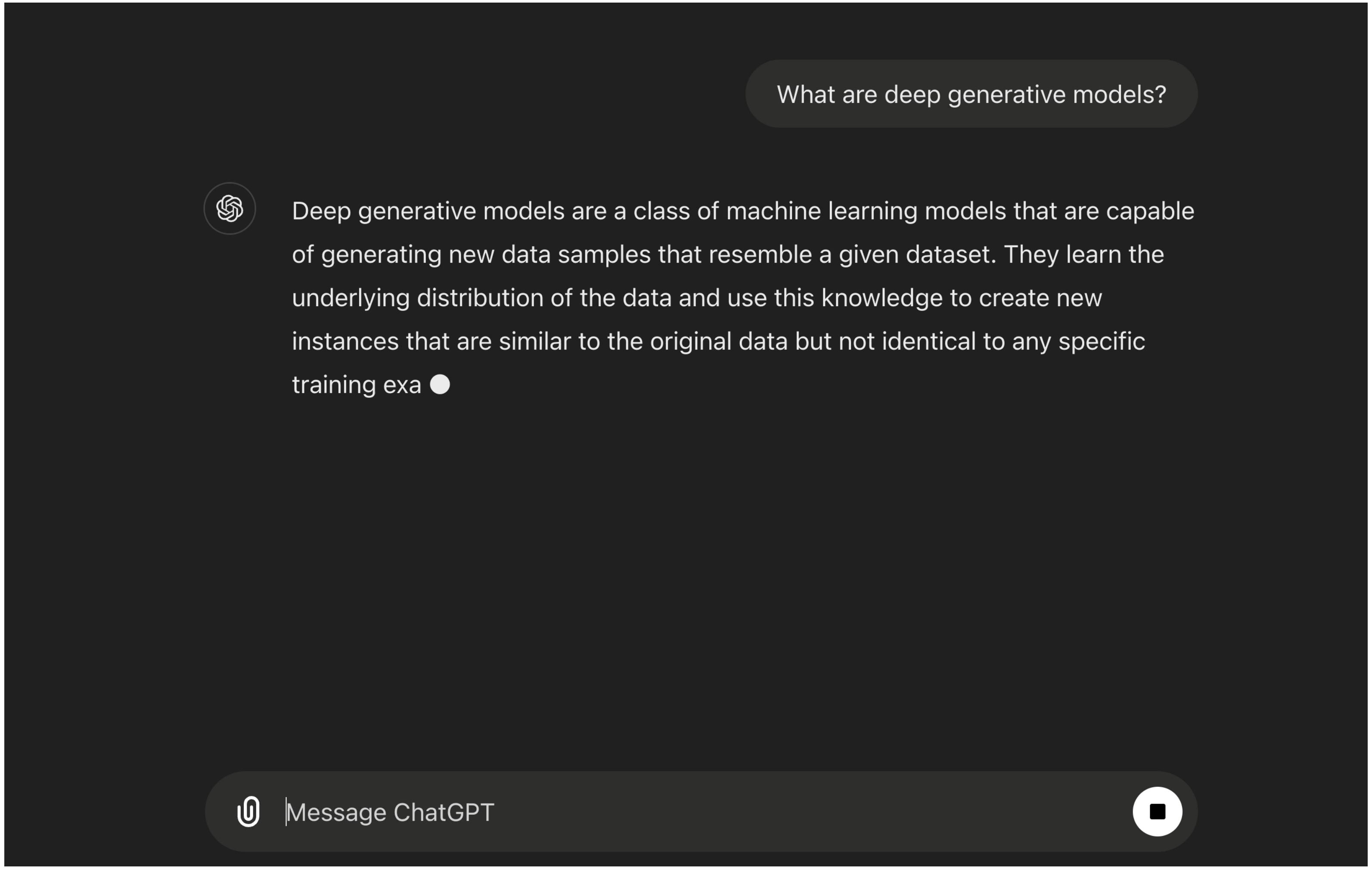
Introduction

6.S978 Deep Generative Models

Kaiming He
EECS, MIT

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

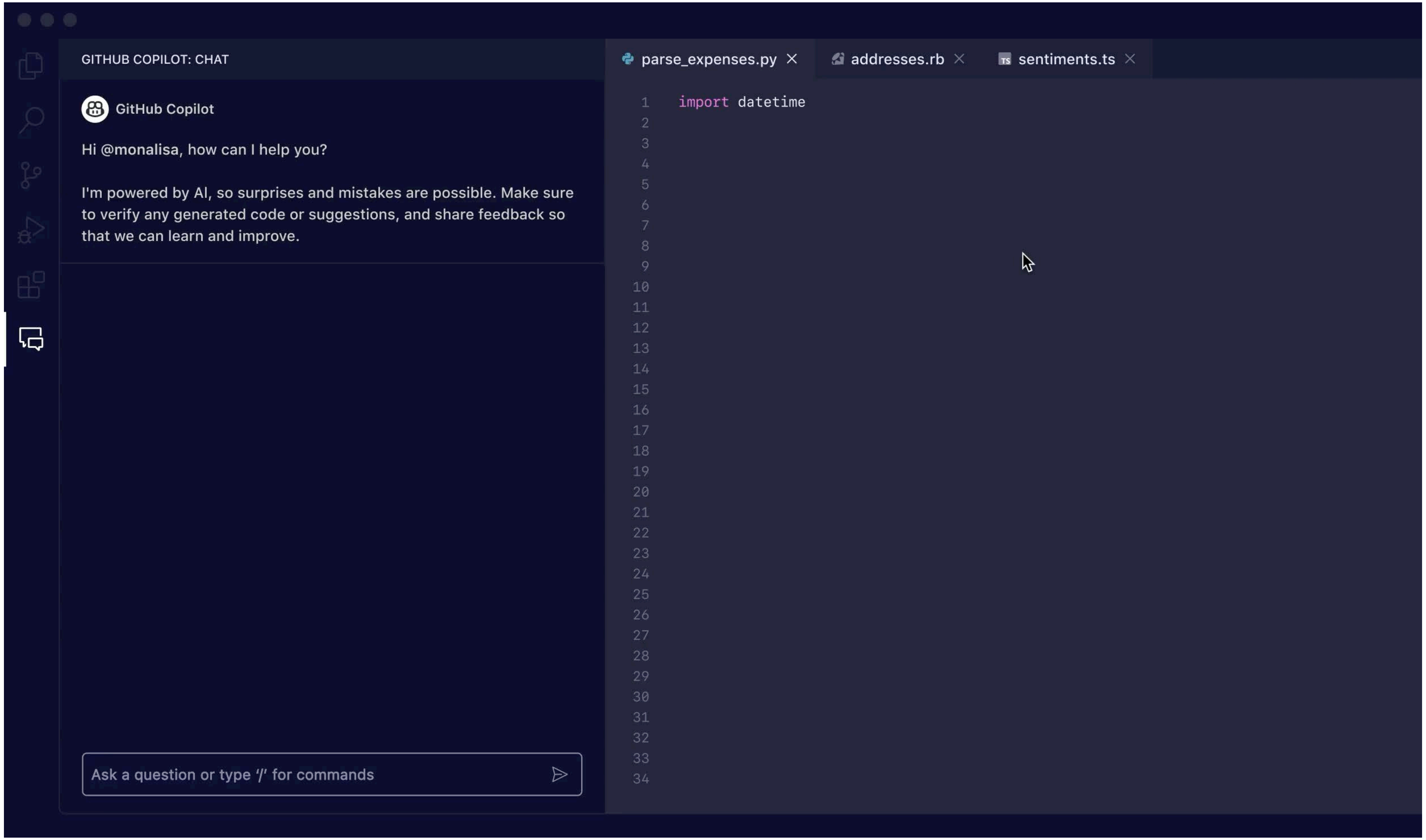
The "GenAI" Era

Text-to-video generation

Generated by Sora

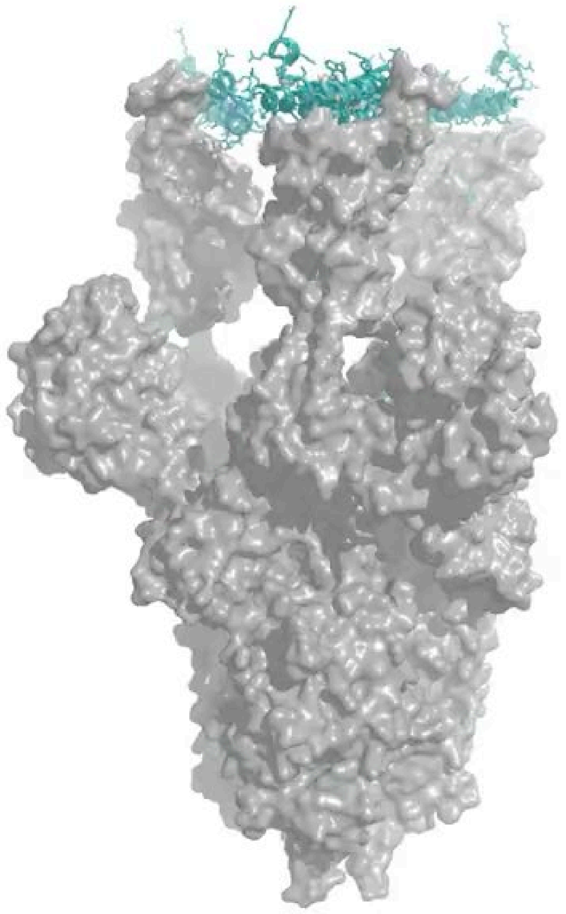
The “GenAI” Era

AI assistant for code generation



The "GenAI" Era

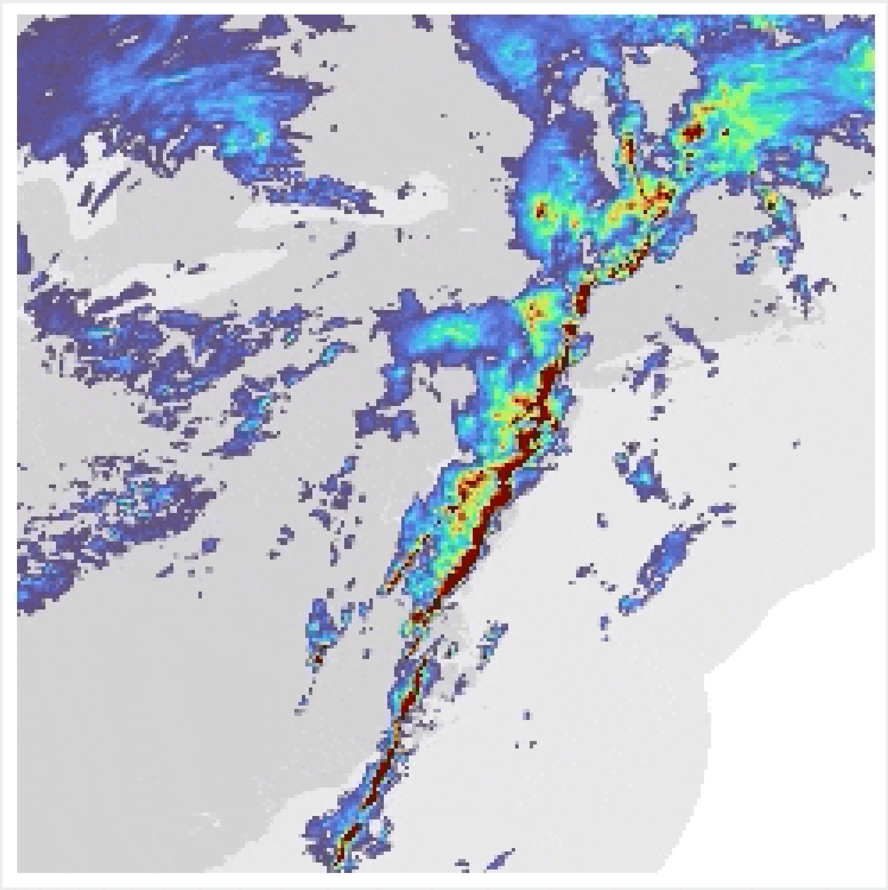
Protein design and generation



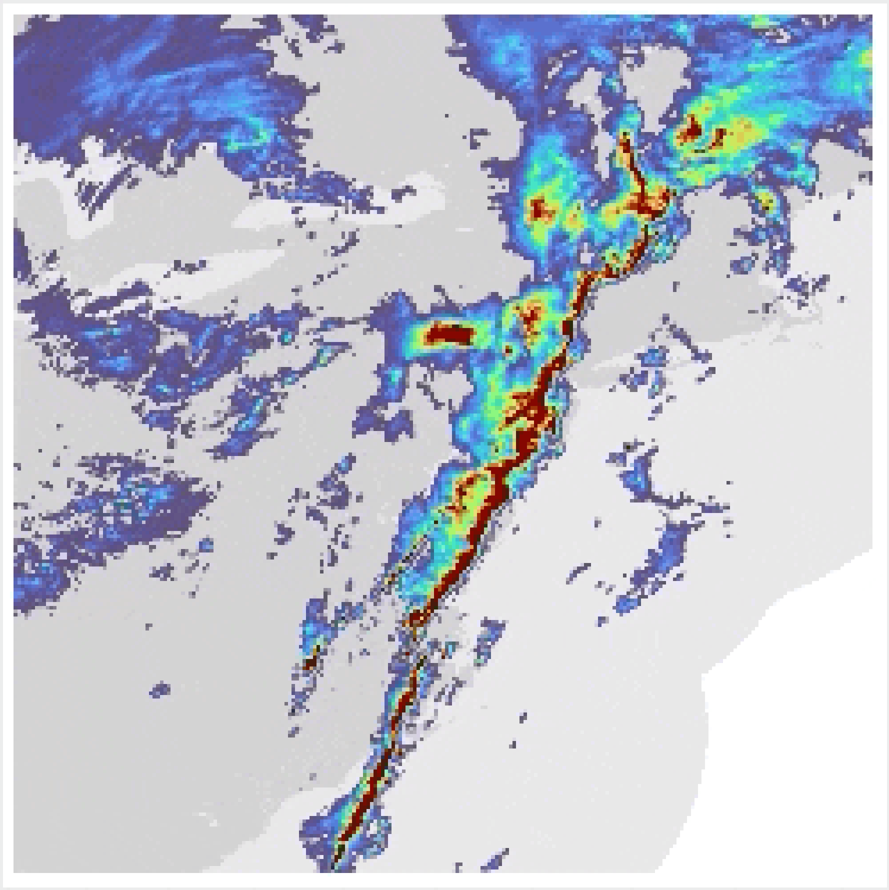
Watson, et al. De novo design of protein structure and function with RFdiffusion, Nature 2023

The "GenAI" Era

Weather forecasting



Target



DGMR

Skilful precipitation nowcasting using deep generative models of radar, Nature 2021

Generative Models before the “GenAI” Era

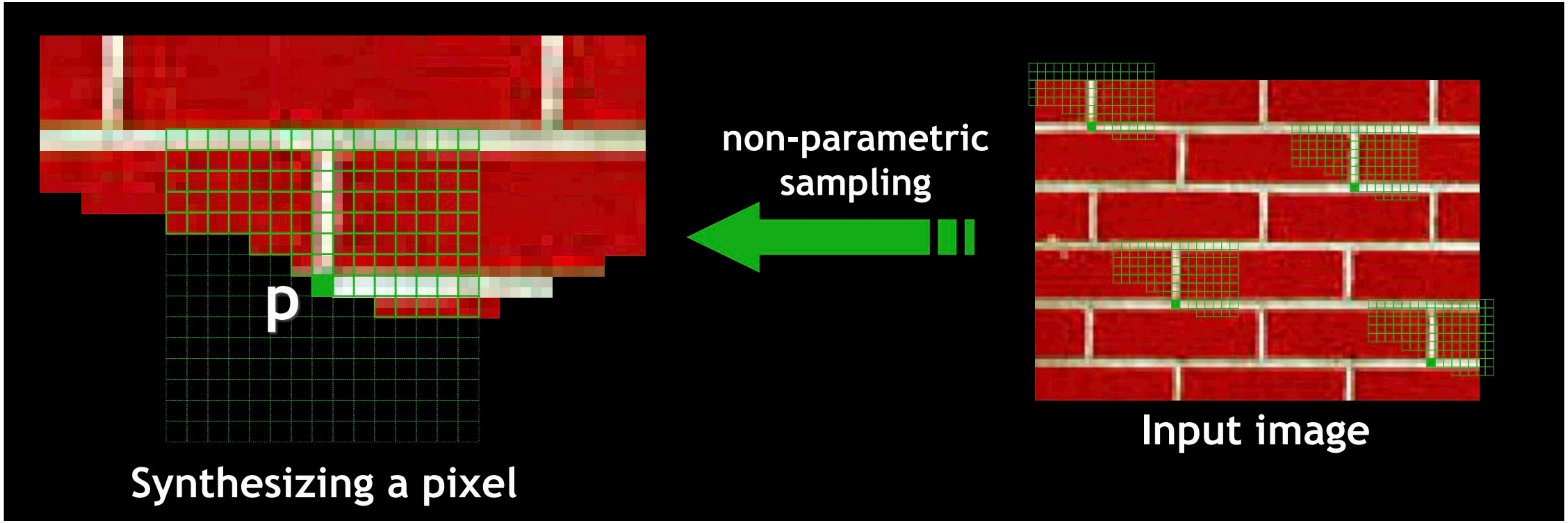
2009, PatchMatch: Photoshop’s Content-aware Fill

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

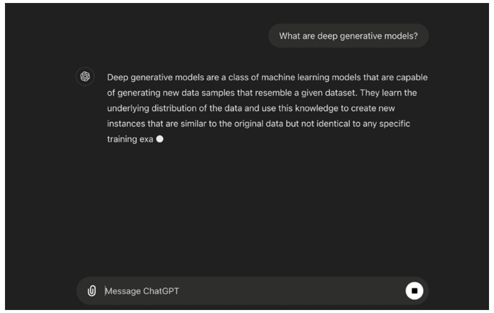


Texture Synthesis by Non-parametric Sampling, ICCV 1999

What are Generative Models?

What do these scenarios have in common?

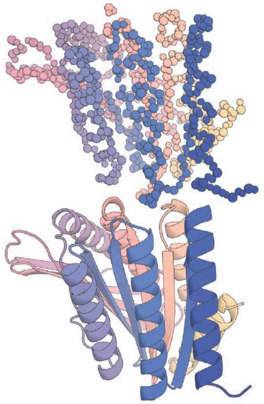
- There are **multiple** or infinite predictions to one input.
- Some predictions are more "**plausible**" than some others.
- Training data may contain **no exact solution**.
- Predictions may be **more complex**, more informative, and higher-dimensional than input.



Chatbot

Image generation

Video generation



Protein generation

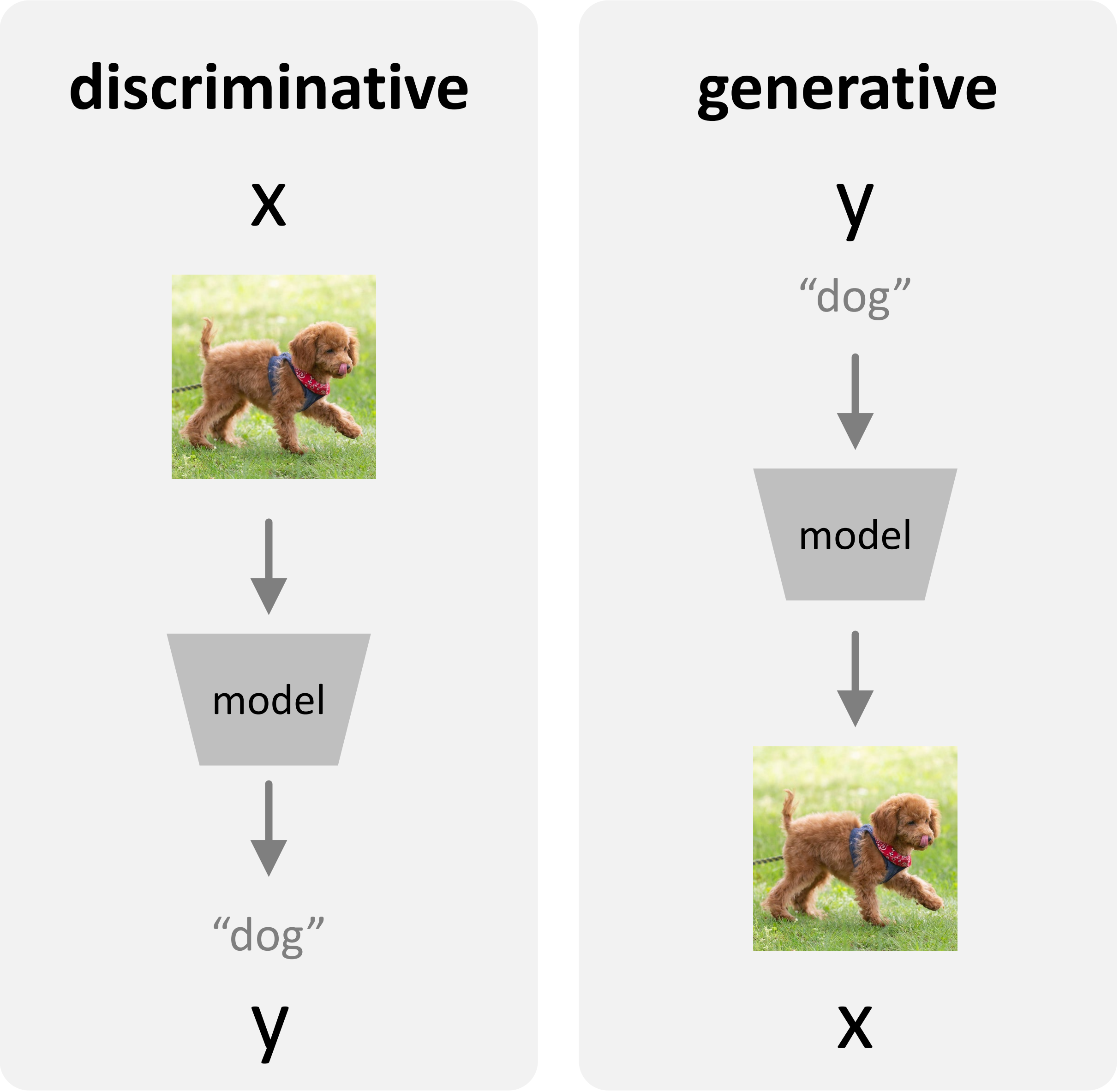
Discriminative vs. Generative models

discriminative

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

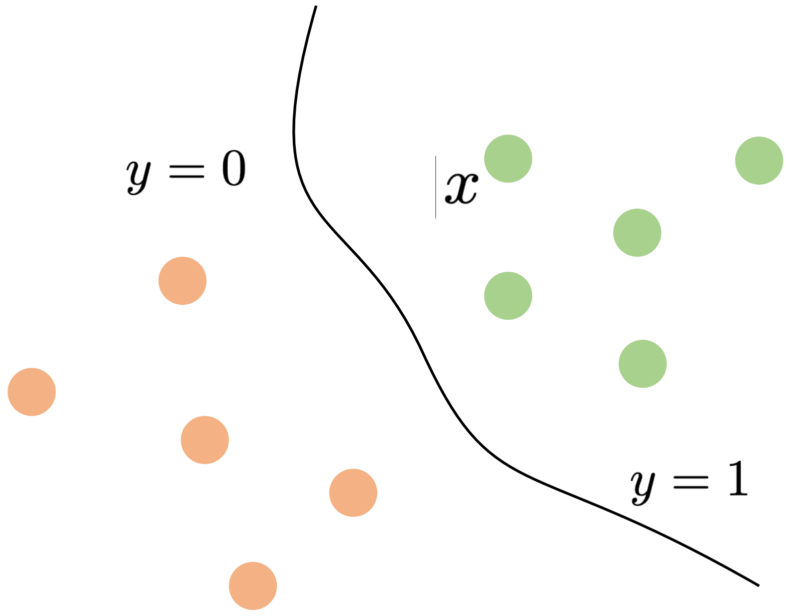
generative

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

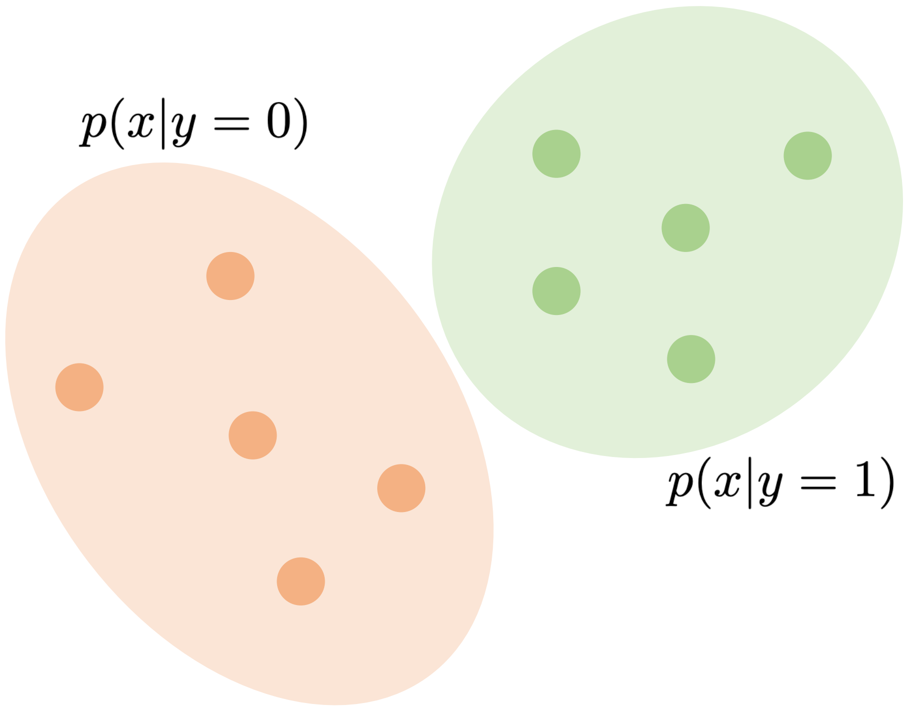


Discriminative vs. Generative models

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

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

assuming known prior

discriminative

generative

constant for given x

- 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

constant for given x

- Can discriminative models be generative?

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

still need to model prior distribution of x

constant for given y

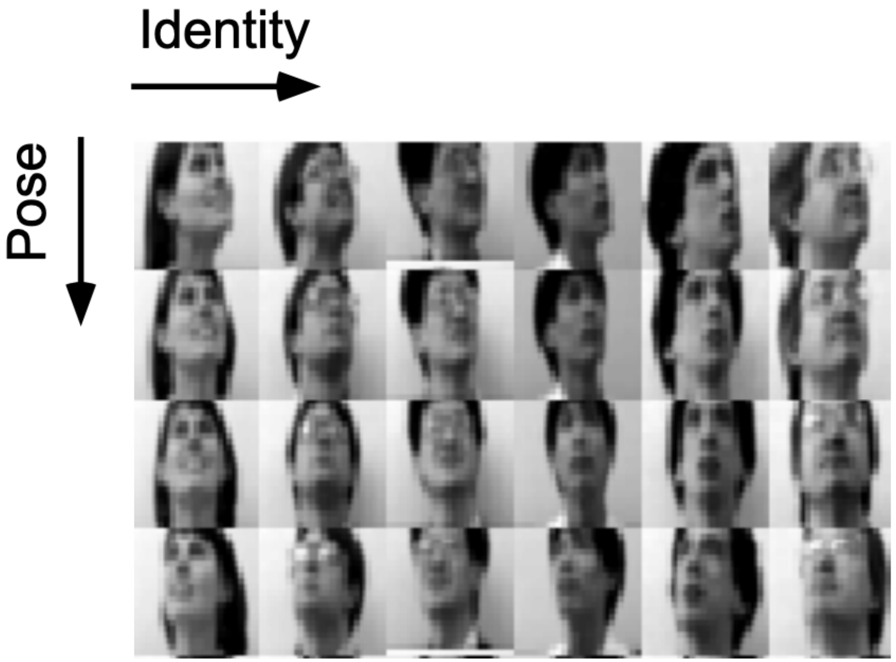
- The challenge is about representing and predicting distributions

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

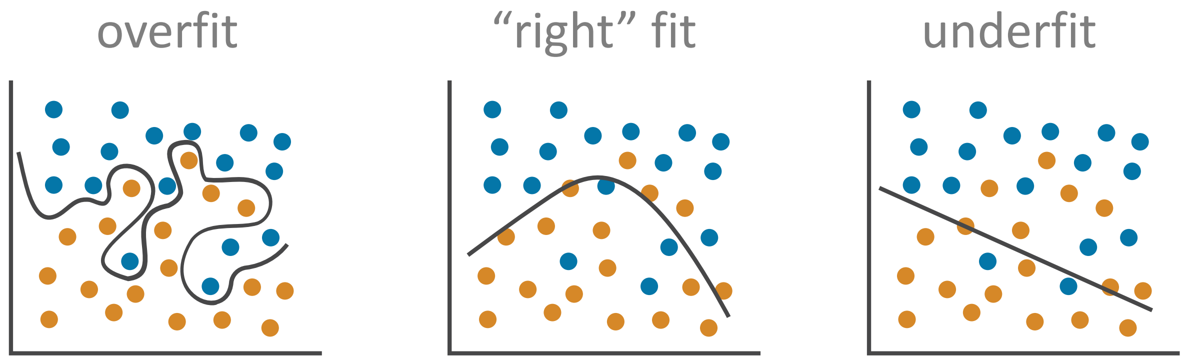


- Probability is part of the modeling.

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

Probability is part of the modeling

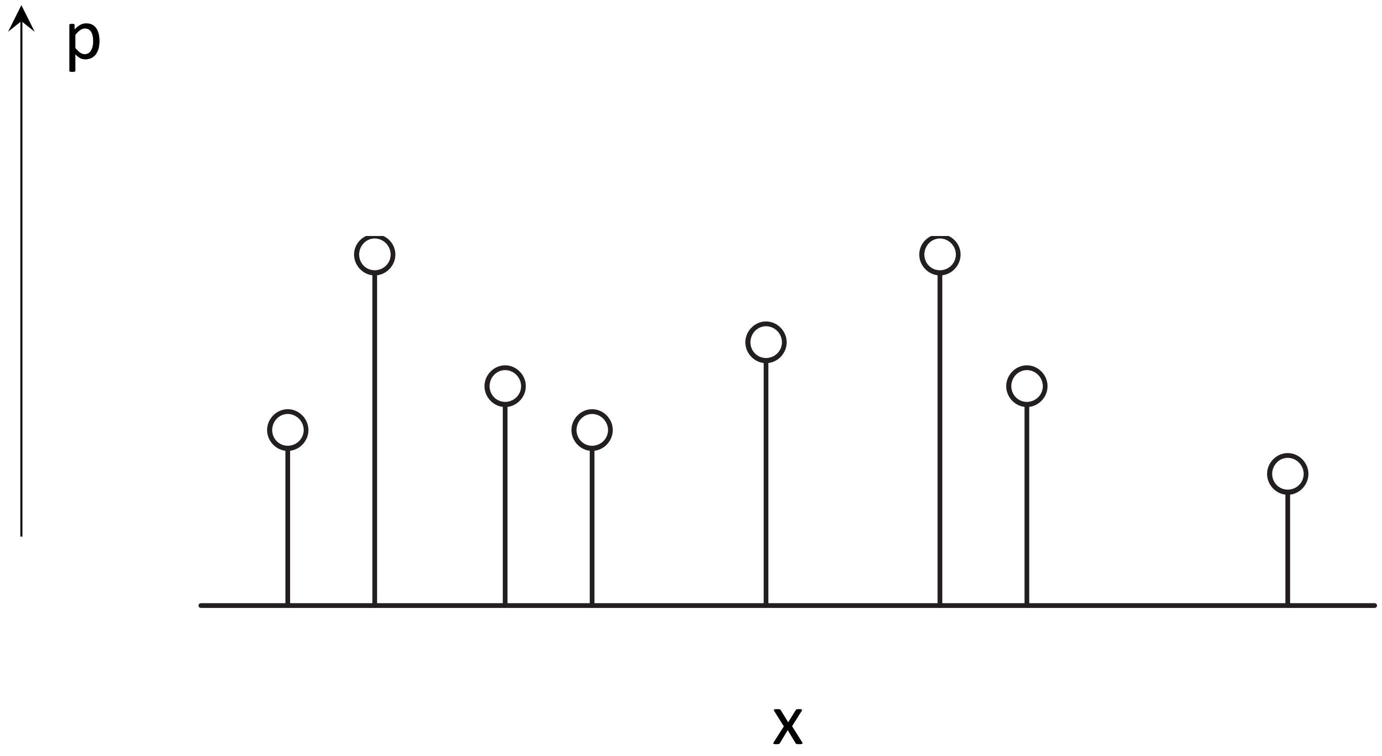
- There may not be “underlying” distributions.
- Even there are, what we can observe are a **finite** set of data points
- The models **extrapolate** the observations for modeling distributions
- Overfitting vs. underfitting: like discriminative models



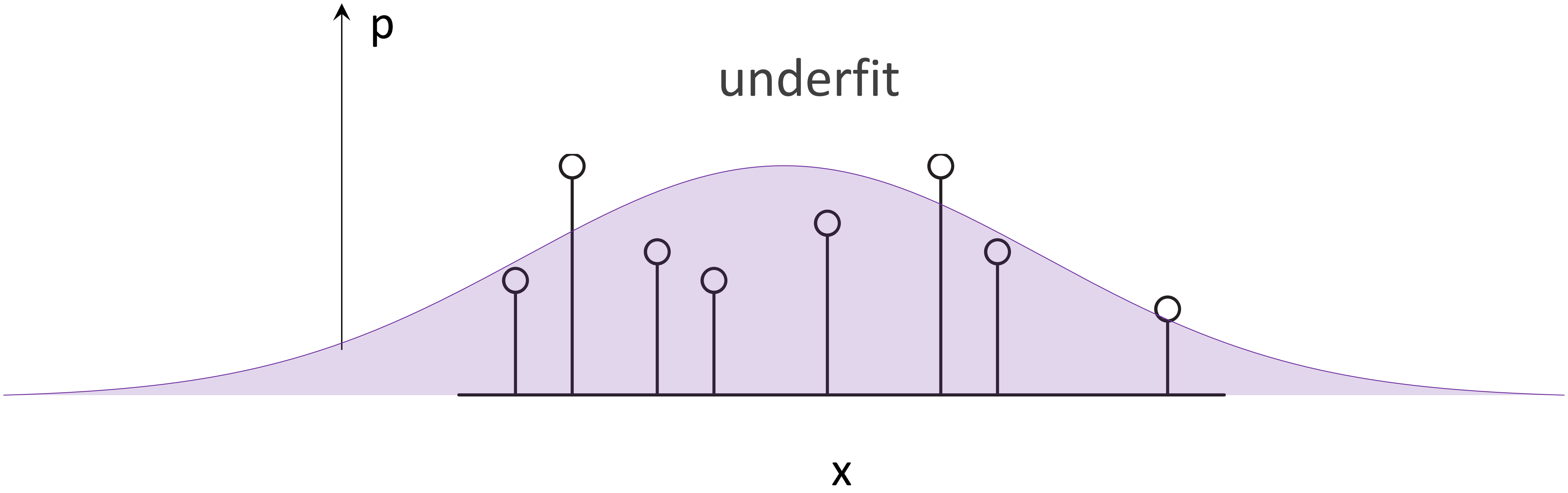
discriminative models

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

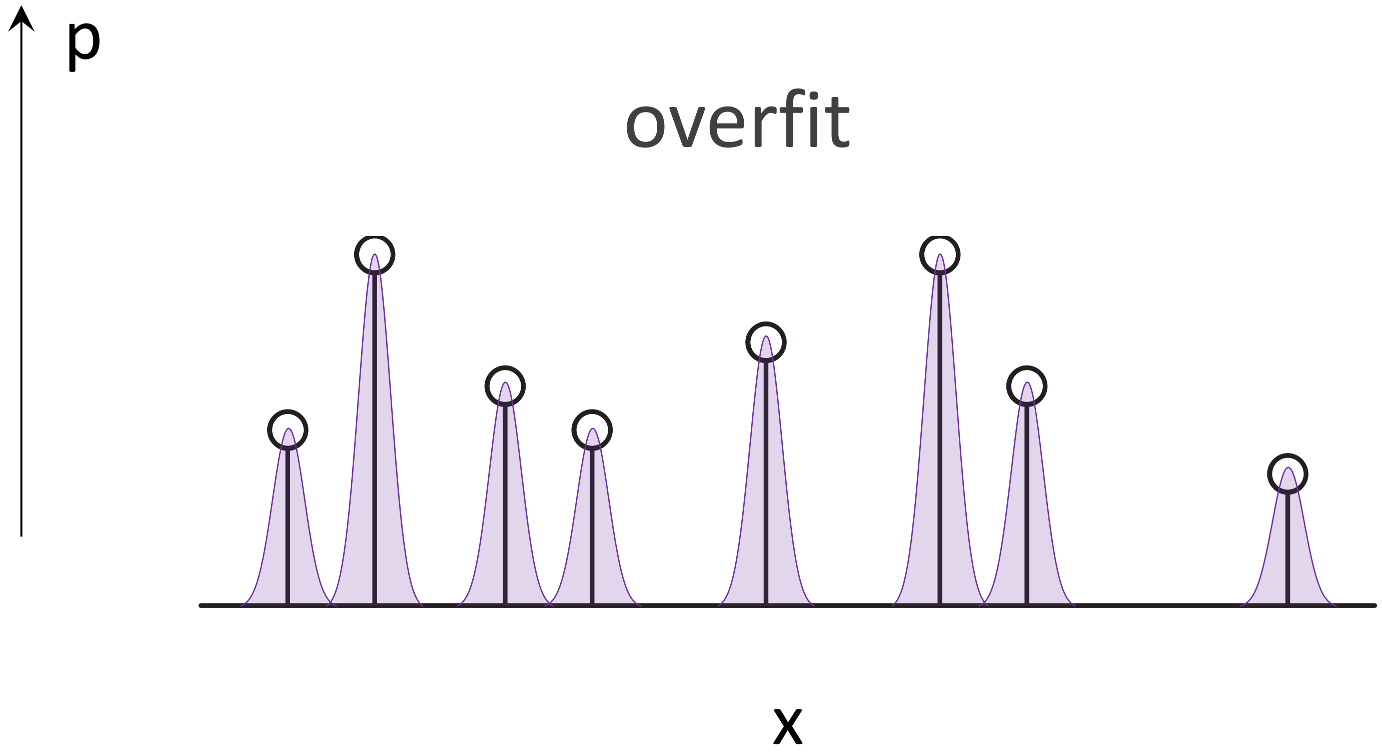
Probability is part of the modeling



Probability is part of the modeling

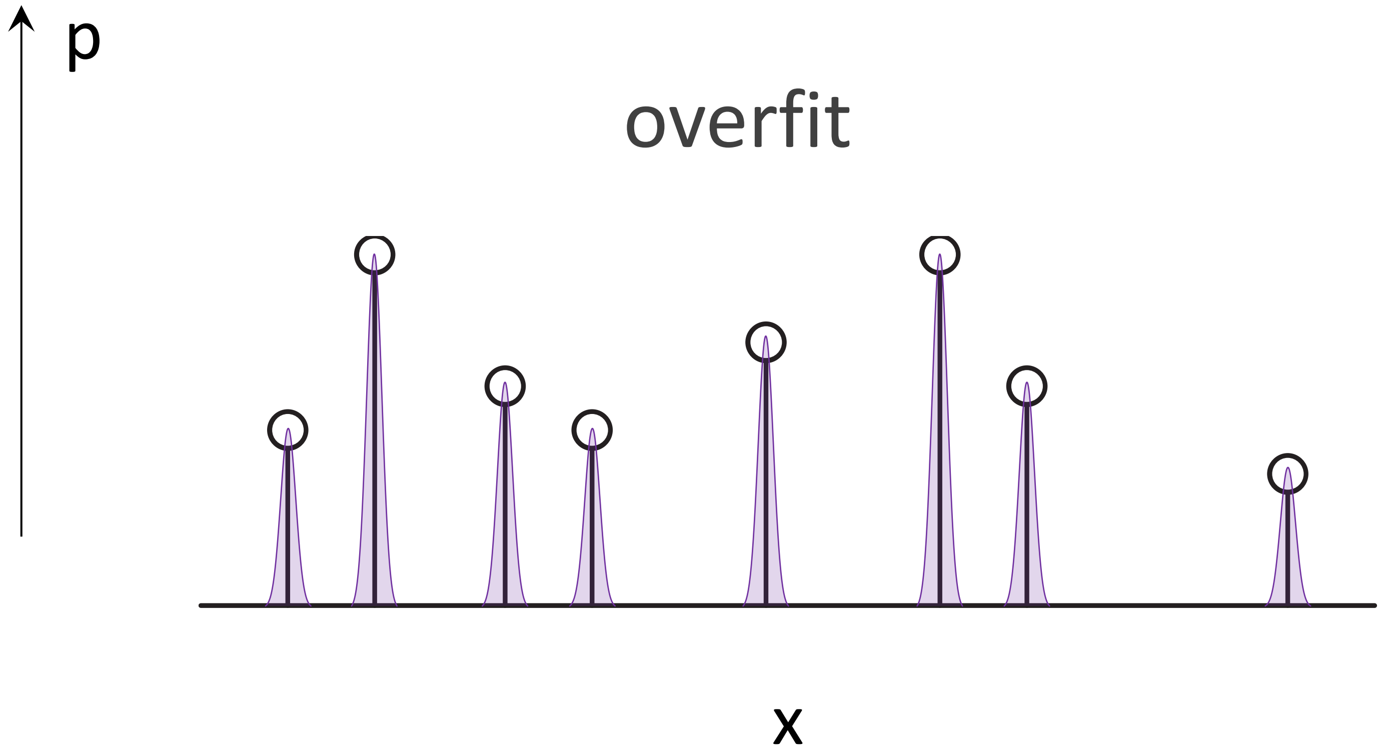


Probability is part of the modeling



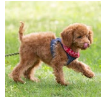
Probability is part of the modeling

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

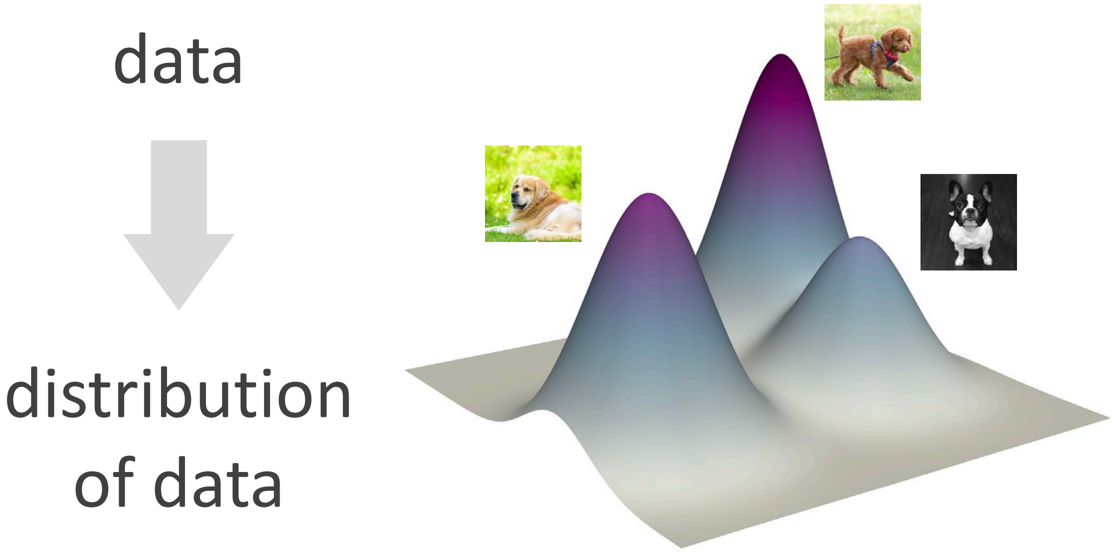


Generative models w/ probabilistic modeling

data



Generative models w/ probabilistic modeling

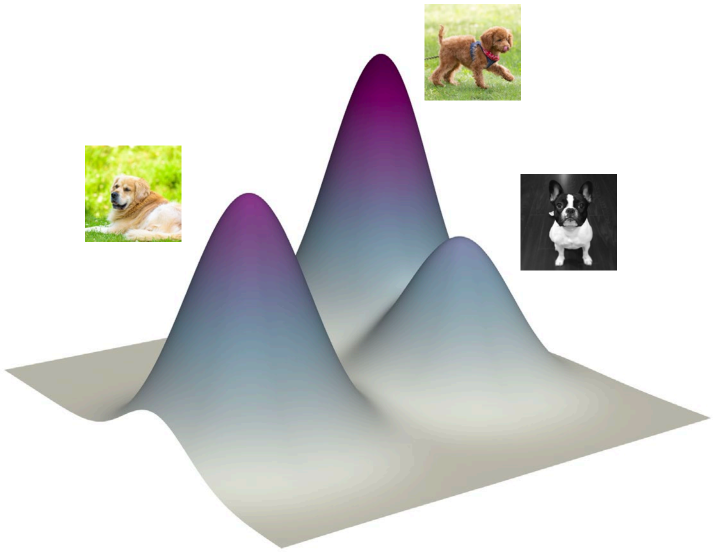


- This is already part of the modeling

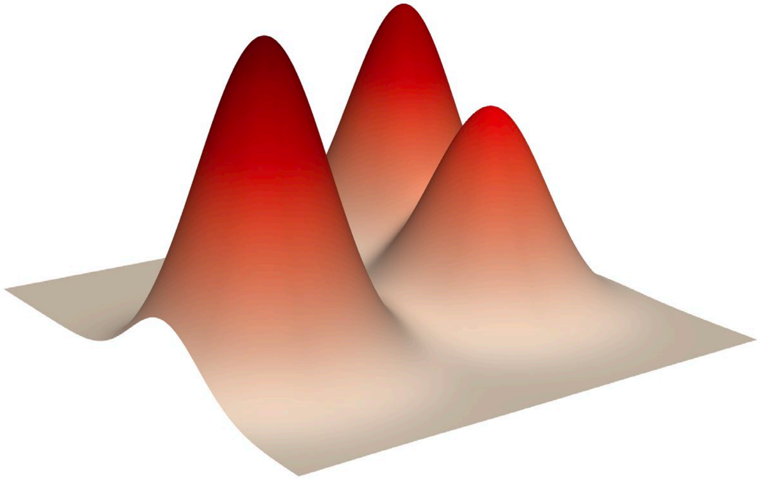
Generative models w/ probabilistic modeling

data

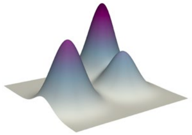
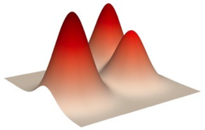
distribution
of data



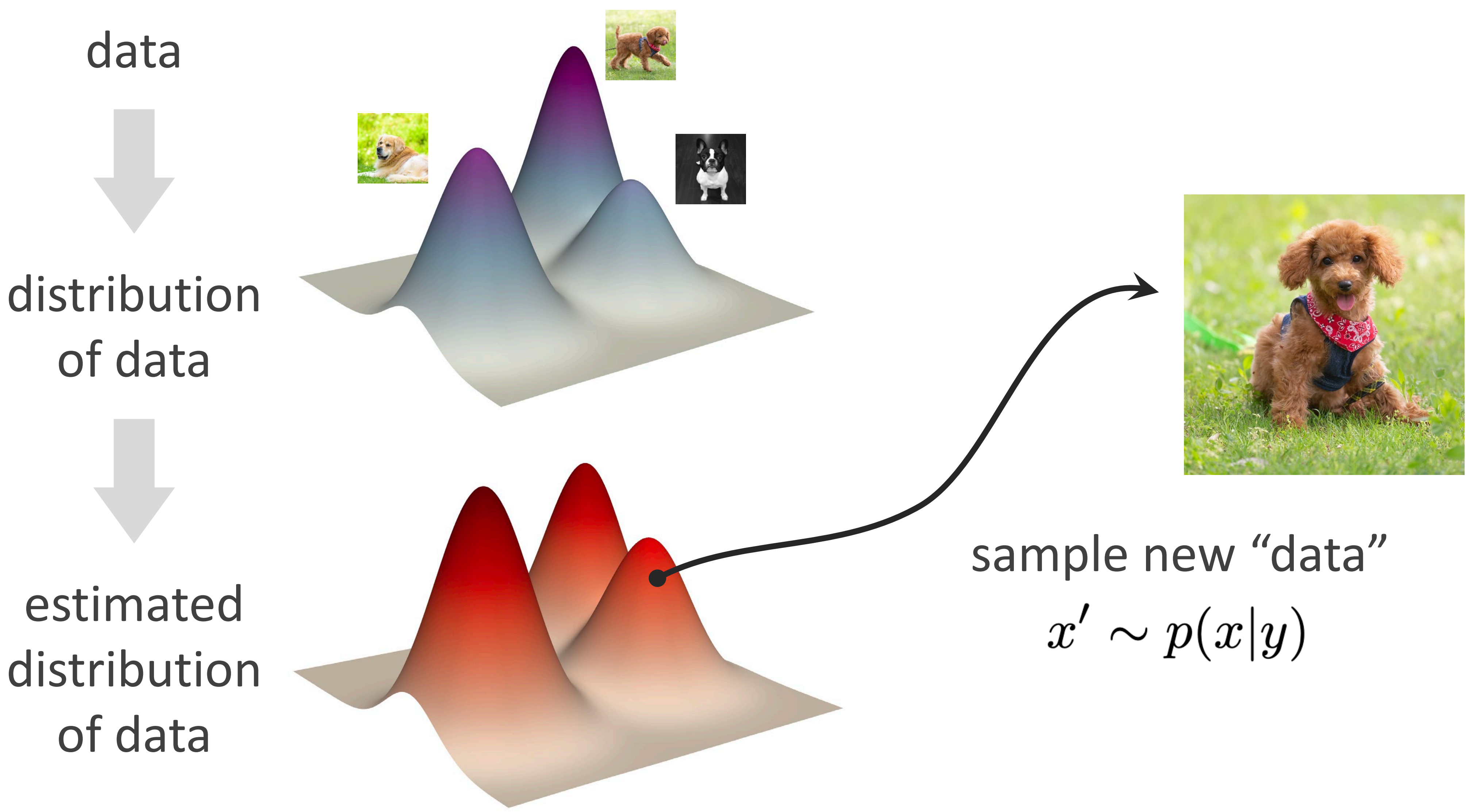
estimated
distribution
of data



- Optimize a loss function

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

Generative models w/ probabilistic modeling

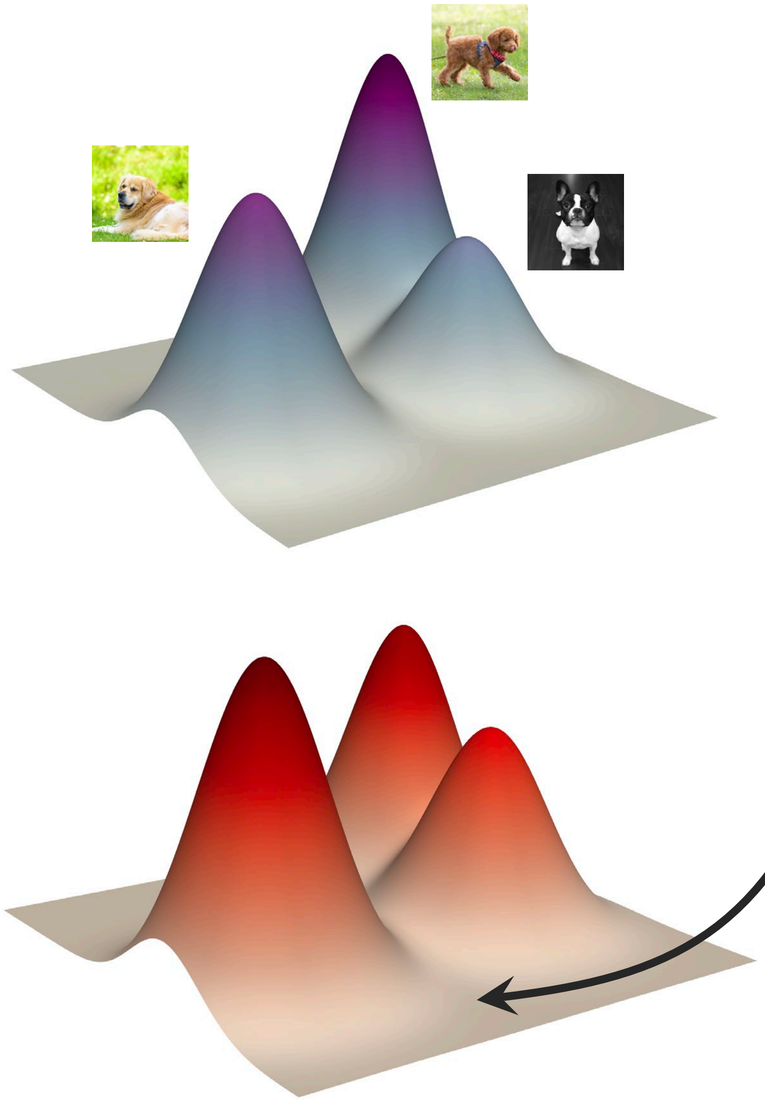


Generative models w/ probabilistic modeling

data

distribution
of data

estimated
distribution
of data



estimate prob density

$p(x|y)$ = ?

Generative models w/ probabilistic modeling

Notes:

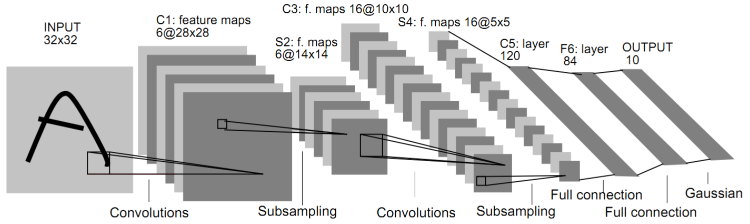
- Generative models involve statistical models which are often designed and derived by humans.
- Probabilistic modeling is not just the work of neural nets.
- Probabilistic modeling is a popular way, but not the only way.
- "*All models are wrong, but some are useful*." - George Box

What are <u>Deep</u> Generative Models?

<u>Deep</u> Generative Models

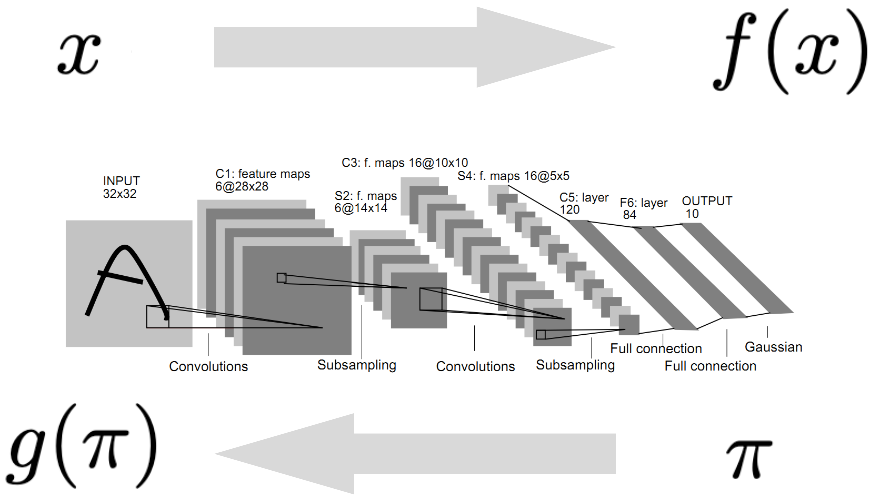
- Deep learning is **representation learning**
- Learning to represent data instances
  - map data to feature: $x \rightarrow f(x)$
  - minimize loss w/ target: $\mathcal{L}(y, f(x))$

$x \rightarrow f(x)$



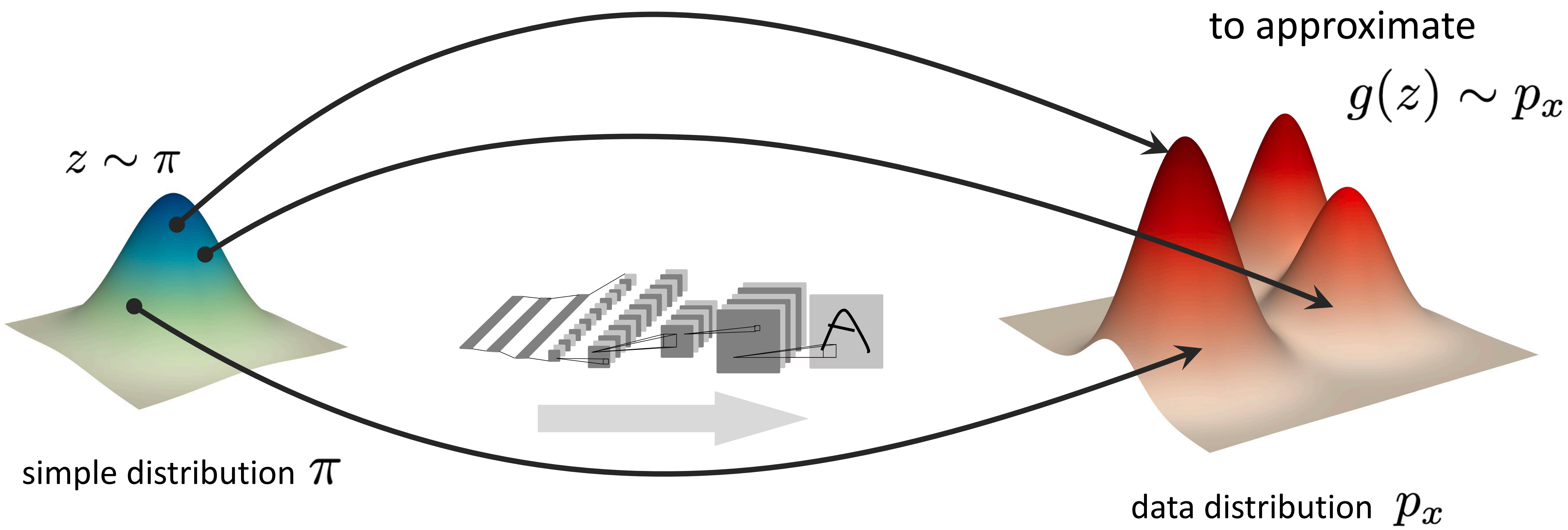
<u>Deep</u> Generative Models

- Deep learning is **representation learning**
- Learning to represent data instances
  - map data to feature: $x \to f(x)$
  - minimize loss w/ target: $\mathcal{L}(y, f(x))$
- Learning to **represent probability distributions**
  - map a simple distribution (Gaussian/uniform) to a complex one: $\pi \to g(\pi)$
  - 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

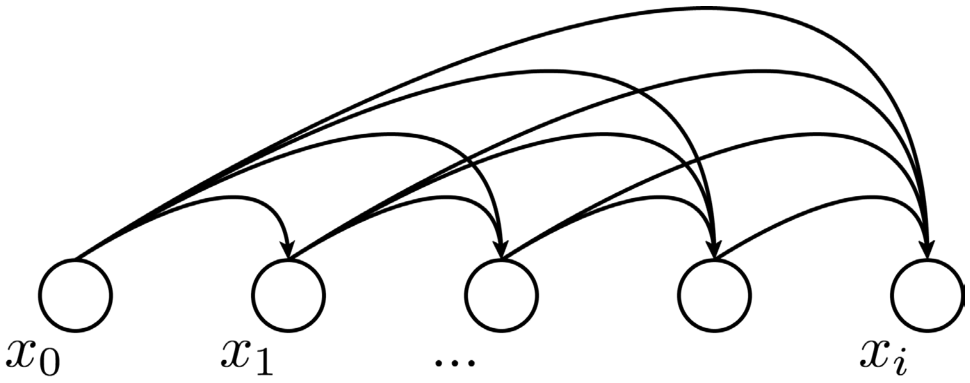


Learning to represent probability distributions

- Not all parts of distribution modeling is done by learning

Case study:
Autoregressive model

This dependency graph is designed (not learned).

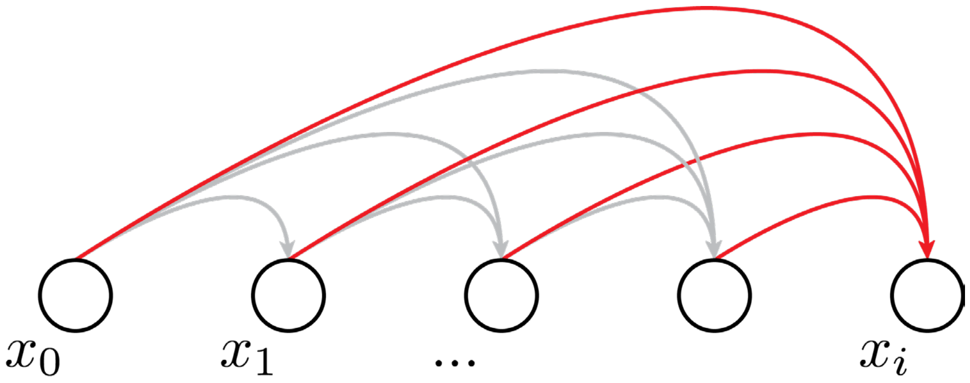


Learning to represent probability distributions

- Not all parts of distribution modeling is done by learning

Case study:
Autoregressive model

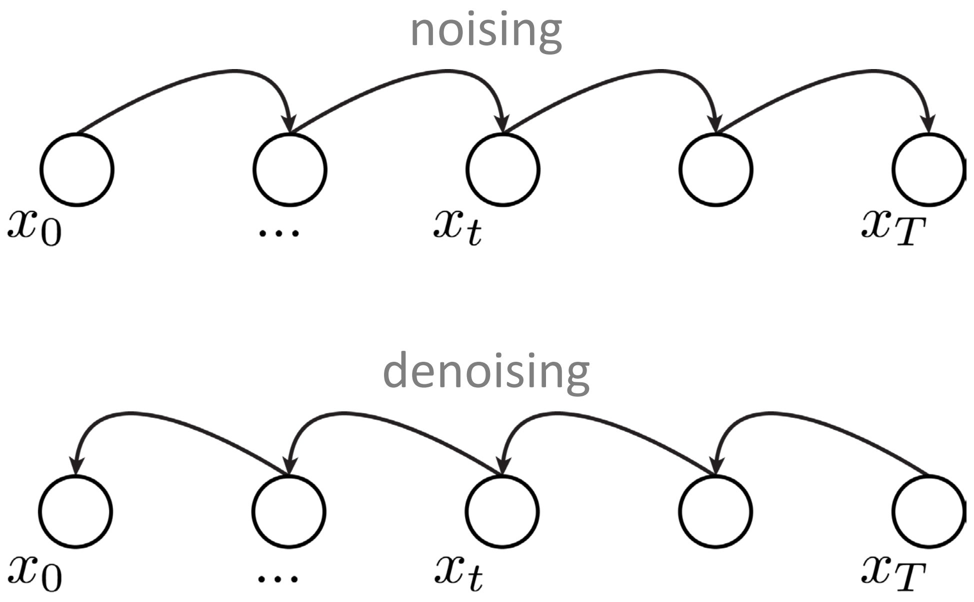
The mapping function is learned
(e.g., Transformer)



Learning to represent probability distributions

- Not all parts of distribution modeling is done by learning

Case study:
Diffusion model

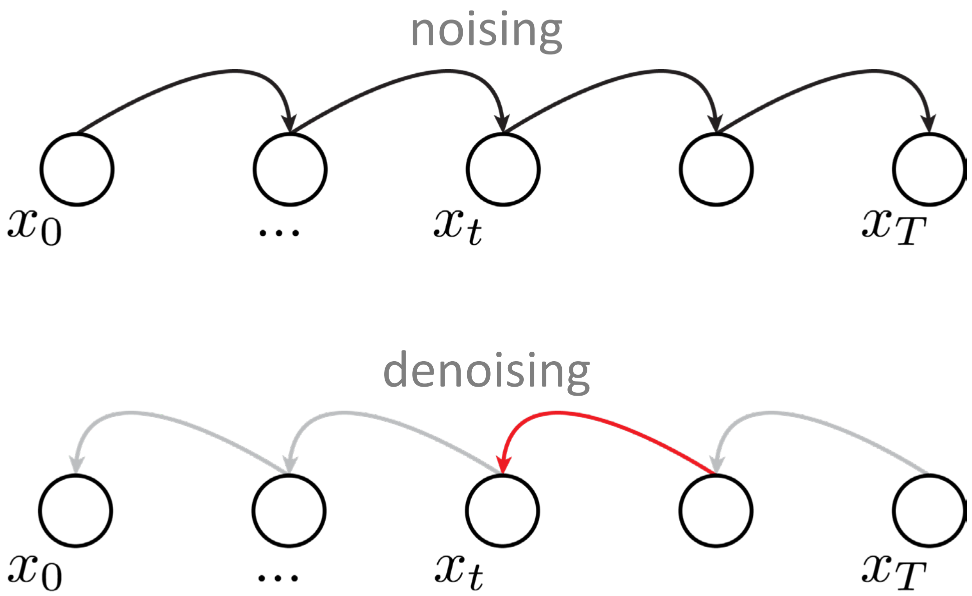


This dependency graph is designed (not learned).

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

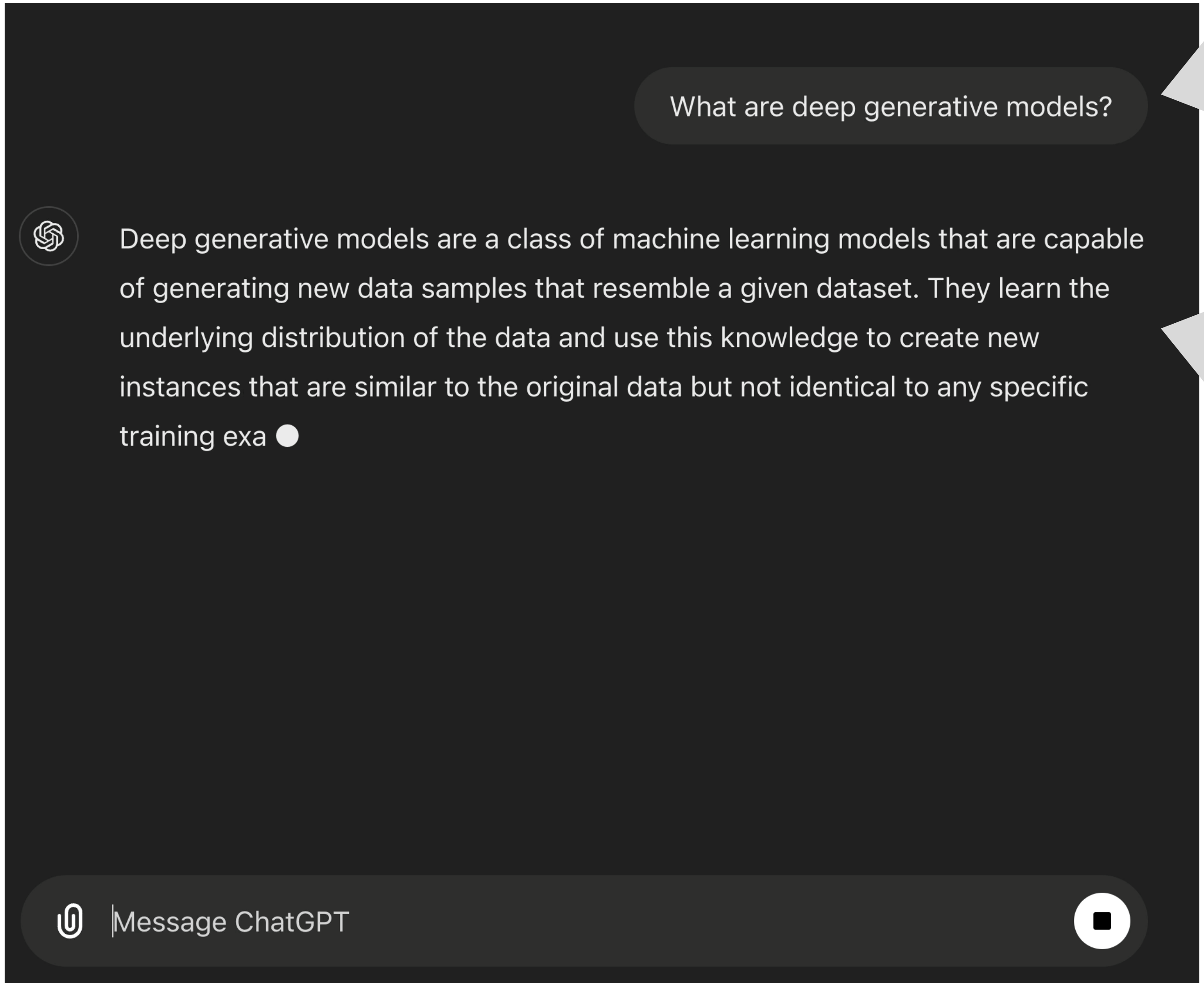
- more abstract
- less informative

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)

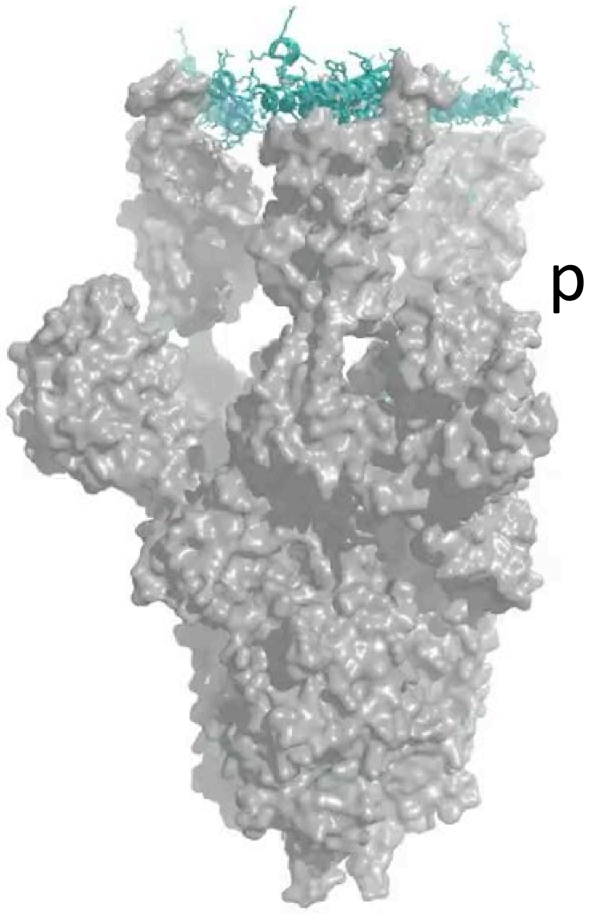
- **Text-to-3D structure generation**

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

Case study: Formulating as p(x|y)

- **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

Case study: Formulating as p(x|y)

- **Class-conditional image generation**

"red fox"

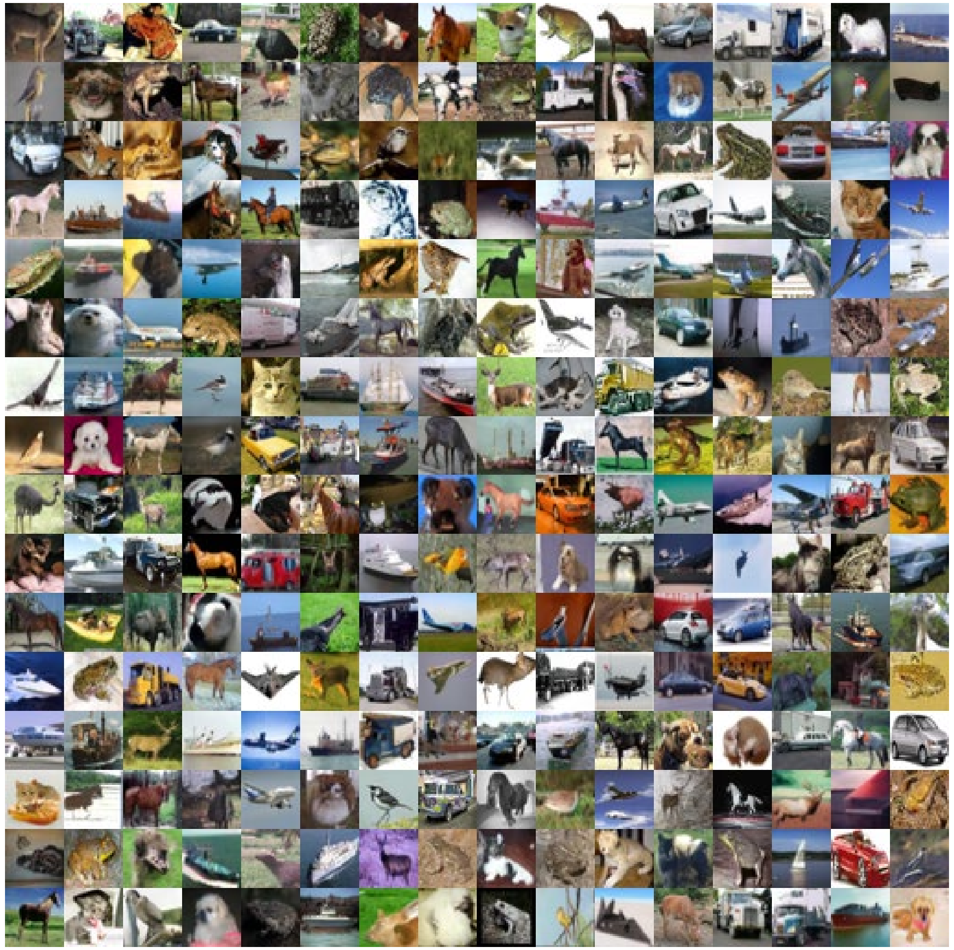
y: class label

x: generated image

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

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 ~ CIFAR10
- p(x): all images

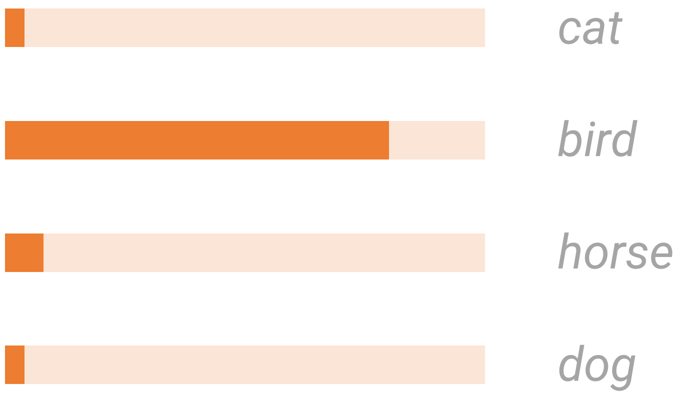
Images generated by: Karras, et al. Elucidating the Design Space of Diffusion-Based Generative Models, NeurIPS 2022

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

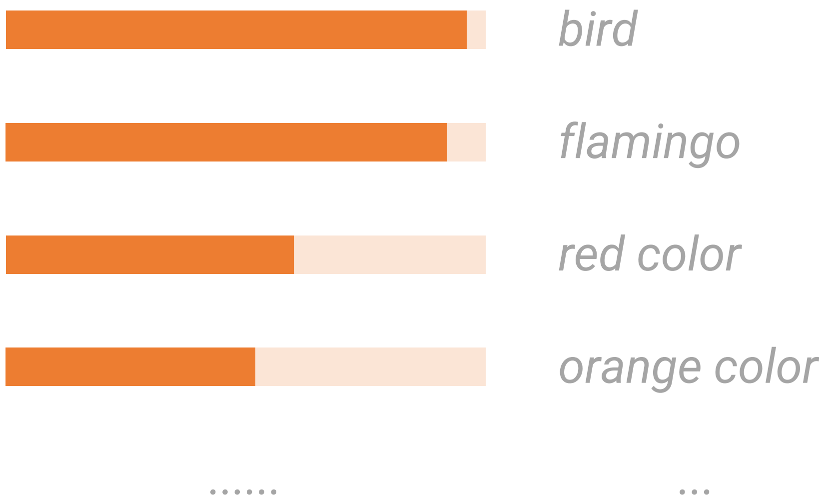


Case study: Formulating as p(x|y)

- **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.
```

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

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

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

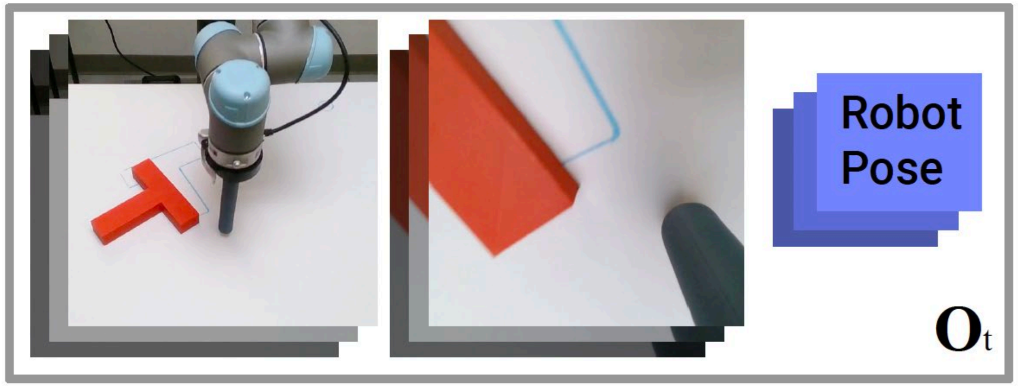
x: response of the chatbot

Figure from: GPT-4 Technical Report, 2023

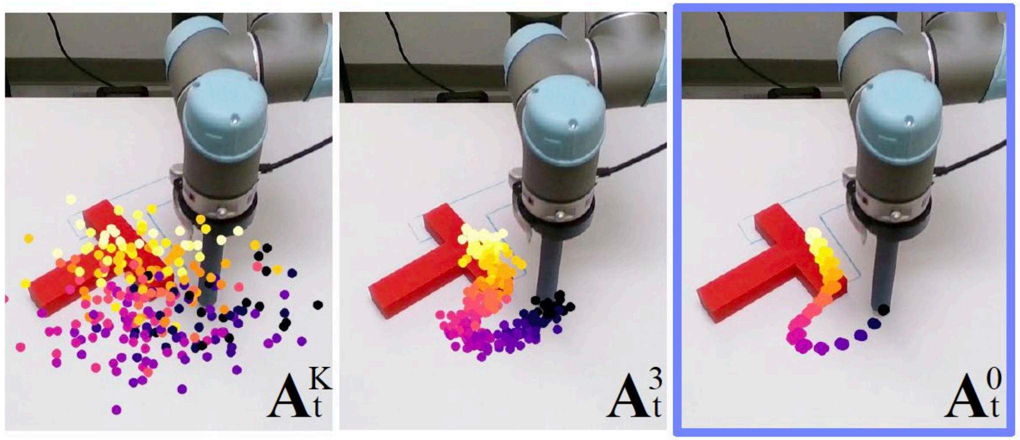
Case study: Formulating as p(x|y)

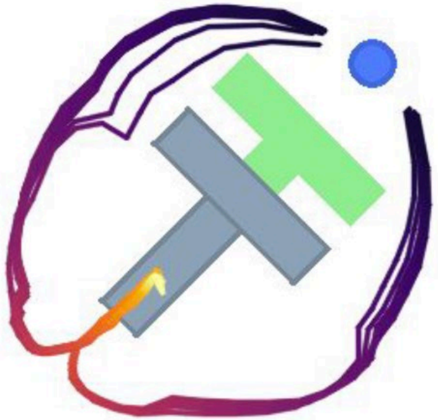
- **Policy Learning in Robotics**

y: visual and other sensory observations



x: policies (probability of actions)





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

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