COMP 690-204 Deep Generative Models

Lecture 1: Introduction

Jason Ren

Fall 2026, CS, UNC

The University
of North Carolina
at Chapel Hill

Instructor: Jason

- Assistant Professor, Department of Computer Science
- 1st year at UNC, 1st time teaching DGM
- Interests: CV/ML/AI
  - Multimodal understanding and generation
  - 3D vision
  - Embodied intelligence and robotics
- Web: https://jason718.github.io/
- Contact:
  - jasonren@cs.unc.edu
  - Office hours: Tues 4-5pm (FB 240)

Course website

- All information about this course can be found on the website

https://jason718.github.io/dgm-26fall/

- We will also use Canvas

Grading Rubric

- 50% — Final project
  - 5% — Proposal
  - 20% — Presentation
  - 25% — Final artifacts
- 25% — Paper reviews
- 25% — Paper presentation
- 5% bonus for active class participation

Paper reviews (25%)

WEEK	DATE	TOPIC	COURSE MATERIAL	PRESENTER
PART I: PRINCIPLES & THEORY				
1	Thu, Aug 20	LECTURE Introduction	Slides	Jason
2	Tue, Aug 25	LECTURE Autoregressive (AR)	Slides	Jason
	Thu, Aug 27	READING Autoregressive (AR)	1. Hoffmann et al., Training Compute-Optimal Large Language Models (Chinchilla), NeurIPS 2022 2. Ouyang et al., Training Language Models to Follow Instructions with Human Feedback (InstructGPT), NeurIPS 2022 3. Tian et al., Visual Autoregressive Modeling: Scalable Image Generation via Next-Scale Prediction, NeurIPS 2024	Jason
3	Tue, Sep 1	LECTURE Variational Autoencoder (VAE)	Slides	Jason
	Thu, Sep 3	READING Variational Autoencoder (VAE)	1. Higgins et al., beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework, ICLR 2017 2. Vahdat & Kautz, NVAE: A Deep Hierarchical Variational Autoencoder, NeurIPS 2020 3. Child, Very Deep VAEs Generalize Autoregressive Models and Can Outperform Them on Images, ICLR 2021	TBD
4	Tue, Sep 8	LECTURE Normalizing flows	Slides 1. Lilian Weng, Flow-based Deep Generative Models (blog post)	Jason
	Thu, Sep 10	READING Normalizing flows	1. Kingma & Dhariwal, Glow: Generative Flow with Invertible 1x1 Convolutions, NeurIPS 2018 2. Zhai et al., Normalizing Flows are Capable Generative Models (TarFlow), ICML 2025 3. Gu et al., STARFlow: Scaling Latent Normalizing Flows for High-resolution Image Synthesis, NeurIPS 2025	TBD

- Lecture will be taught by Jason
- Reading:
  - You need to submit paper review for 1 paper before class
  - Due 11:59 PM ET the day before
  - 2 missed reviews forgiven

Paper reviews (25%)

- Before each **READING** class, every student must submit a short review of one of the suggested papers via Google Form
- Reviews are due 11:59 PM ET the day before class
- We don't accept late submissions, but everyone gets two missed reviews forgiven over the semester
- You may skip submitting a review for the class in which you are presenting a paper

Paper presentation (25%)

5	Tue, Sep 15	LECTURE Generative adversarial network (GAN)	Slides	Jason
	Thu, Sep 17	READING Generative adversarial network (GAN)	1. Zhu et al., Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (CycleGAN), ICCV 2017 2. Sauer et al., StyleGAN-T: Unlocking the Power of GANs for Fast Large-Scale Text-to-Image Synthesis, ICML 2023 3. Kang et al., Scaling up GANs for Text-to-Image Synthesis, CVPR 2023	TBD
6	Tue, Sep 22	LECTURE Discrete generative models (AR, VAE, GAN)	Slides 1. van den Oord et al., Neural Discrete Representation Learning (VQ-VAE), NeurIPS 2017 2. Esser et al., Taming Transformers for High-Resolution Image Synthesis (VQGAN), CVPR 2021 3. Chang et al., MaskGIT: Masked Generative Image Transformer, CVPR 2022	Jason
	Thu, Sep 24	LECTURE Energy-based model (EBM)	Slides 1. LeCun et al., A Tutorial on Energy-Based Learning, 2006 2. Du & Mordatch, Implicit Generation and Generalization in Energy-Based Models, NeurIPS 2019	Jason
7	Tue, Sep 29	LECTURE Score-based models	Slides	Jason
	Thu, Oct 1	PRESENTATION Proposal presentation & discussion	–	All students
8	Tue, Oct 6	No class — Well-Being Day	–	–
	Thu, Oct 8	LECTURE Diffusion models	Slides 1. Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015 2. Song & Ermon, Generative Modeling by Estimating Gradients of the Data Distribution, NeurIPS 2019 3. Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020	Jason

Paper presentation (25%)

- One person per paper
- Sign up for papers here: Google Sheet
  - Each student will present **2** papers over the semester
  - The instructor will present any papers that remain unclaimed
- Aim for roughly a 20-minute talk
- [Required!] Email the instructor a draft of your slides two days before your presentation to get feedback
- [Required!] Email your final slides to the instructor before class

Final project (50%)

- Group projects
  - Up to 3 students
  - Topic of your choosing—get creative!

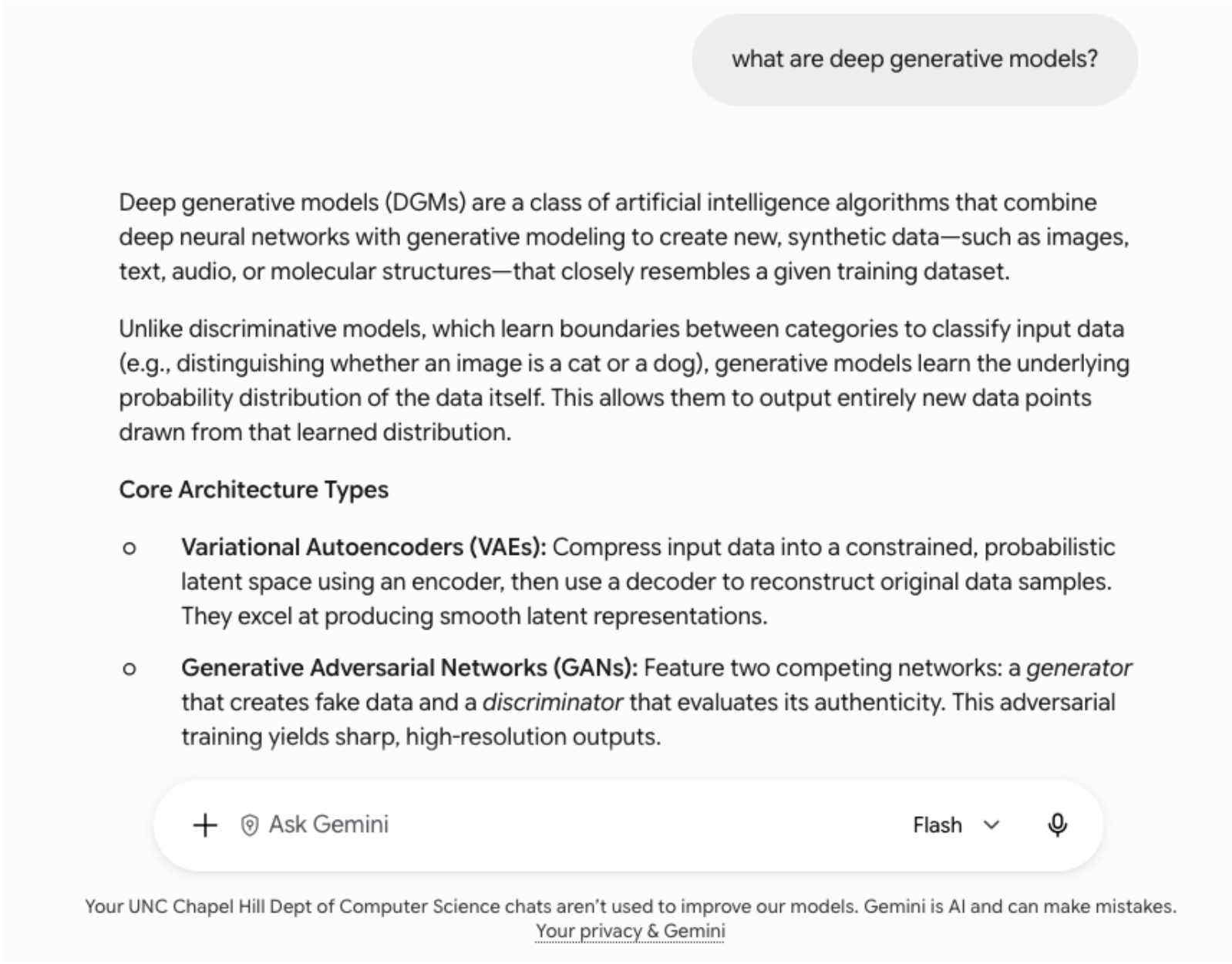
- Proposal (due 10/01)
  - 5-minute in-class presentation, plus a written proposal (up to 2 pages)
  - Explain the problem formulation, excitation plan, and any initial results

- Final artifacts
  - Final presentations (in class: 12/03, 12/08): 10-minute talk + 5-minute Q&A
  - Final report (due 12/10)
  - Results webpage, source code, etc. (due 12/10)

“GenAI” Era

*Acknowledgement: Many of the following slides are modified from Beidi Chen, Xun Huang

Chatbot


what are deep generative models?
Deep generative models (DGMs) are a class of artificial intelligence algorithms that combine deep neural networks with generative modeling to create new, synthetic data—such as images, text, audio, or molecular structures—that closely resembles a given training dataset.
Unlike discriminative models, which learn boundaries between categories to classify input data (e.g., distinguishing whether an image is a cat or a dog), generative models learn the underlying probability distribution of the data itself. This allows them to output entirely new data points drawn from that learned distribution.
Core Architecture Types
Variational Autoencoders (VAEs): Compress input data into a constrained, probabilistic latent space using an encoder, then use a decoder to reconstruct original data samples. They excel at producing smooth latent representations.
Generative Adversarial Networks (GANs): Feature two competing networks: a generator that creates fake data and a discriminator that evaluates its authenticity. This adversarial training yields sharp, high-resolution outputs.
Ask Gemini
Flash
Your UNC Chapel Hill Dept of Computer Science chats aren't used to improve our models. Gemini is AI and can make mistakes.
Your privacy & Gemini

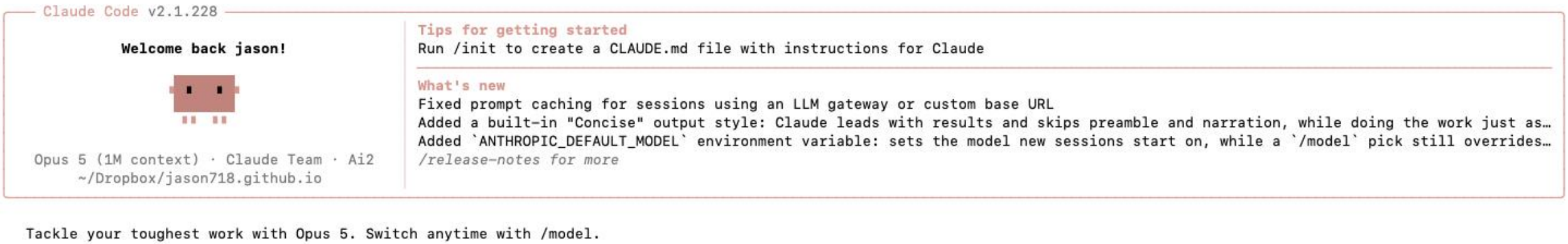
Text-to-image

- A group of students are sitting in the classroom. The lecturer writes "Deep Generative Models" on the blackboard

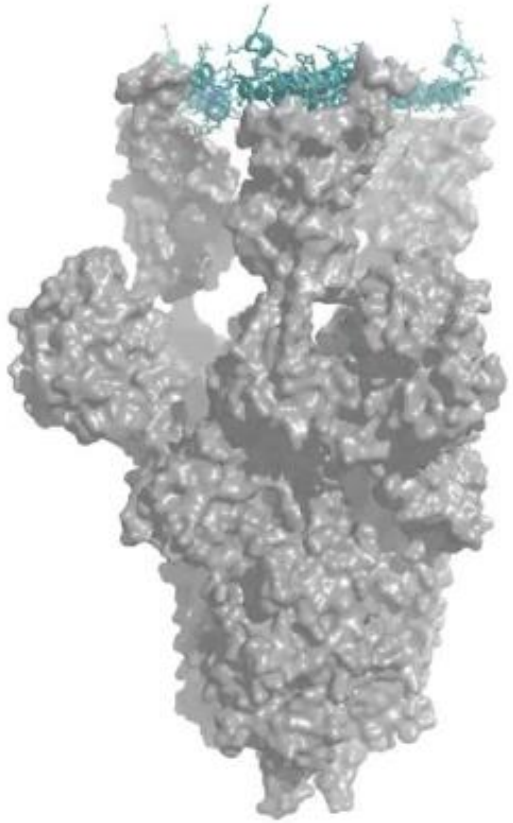
Text-to-video

A pig sitting in front of a slot machine. The arm of the slot machine moves down, causing gold coins to come bursting out of the bottom of the slot machine, showering the pig with coins

Code Generation



Protein design and generation



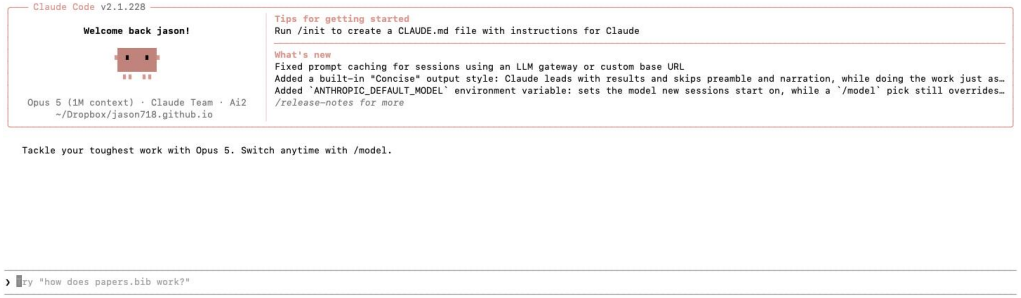
Watson, et al., Nature 2023

What are Generative Models?

*Acknowledgement: Many of the following slides are modified from Kaiming He

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



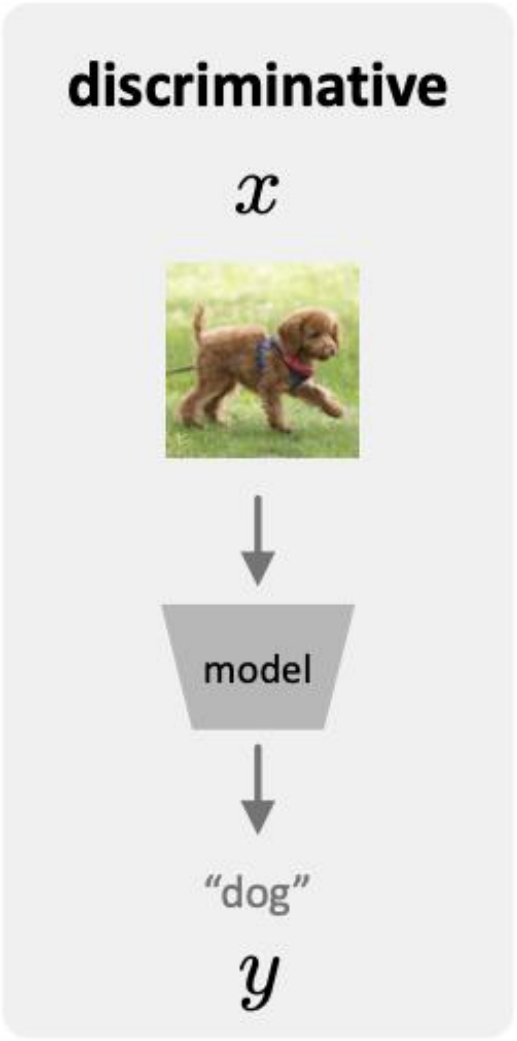
Discriminative vs. Generative models

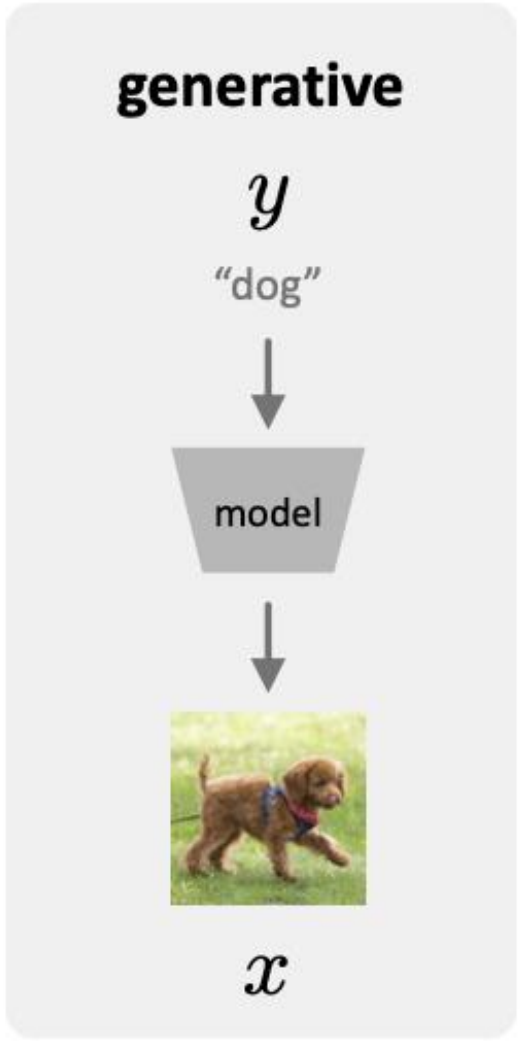
discriminative

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

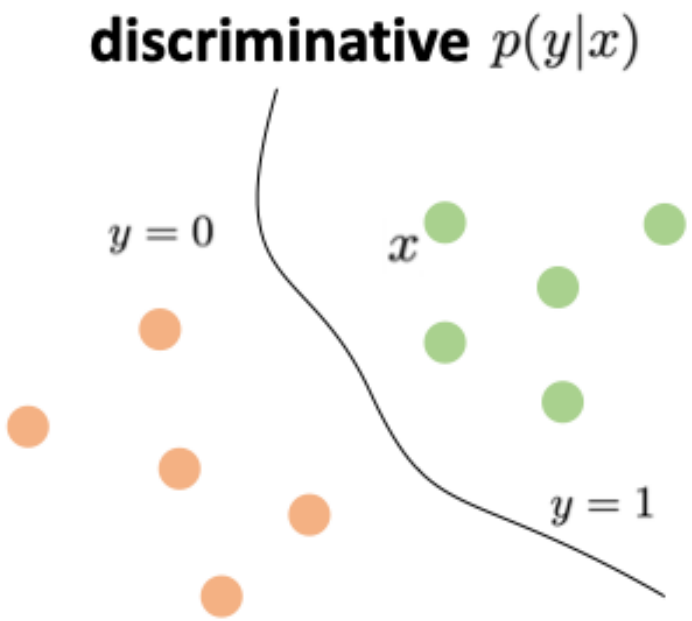
generative

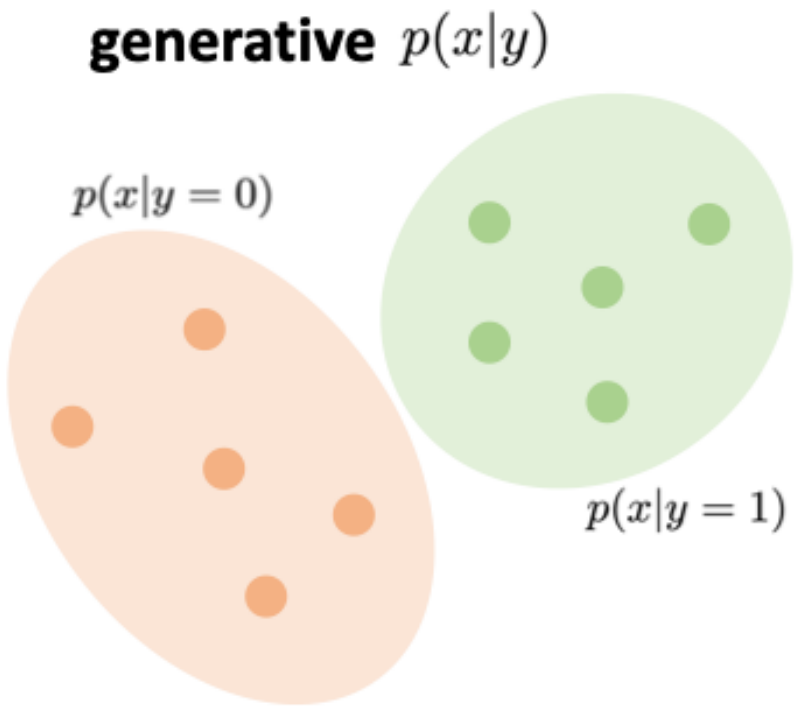
- "label" y ⇨ "sample" x
- many possible outputs





Discriminative vs. Generative models





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

Discriminative vs. Generative models

- 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

Discriminative vs. Generative models

- 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

- Can discriminative models be generative?

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

still need to model prior distribution of x

generative discriminative

constant for given y

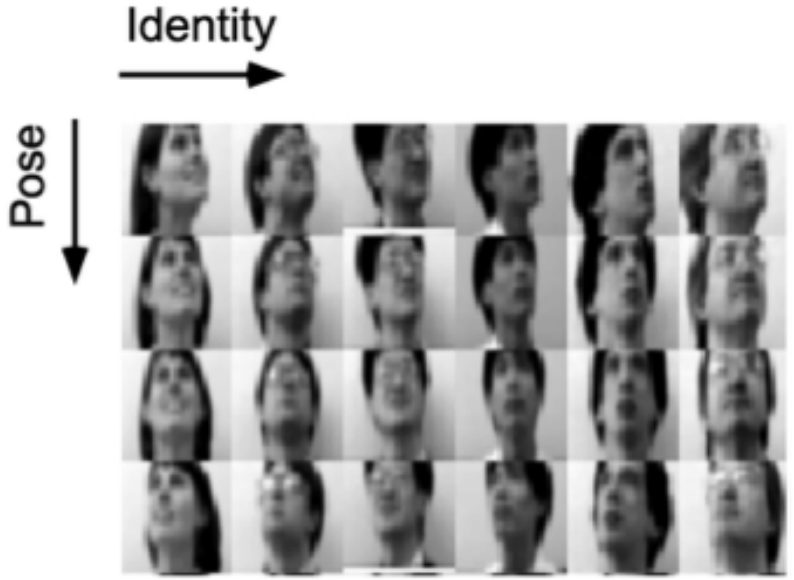
- The challenge is about representing and predicting distributions

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.

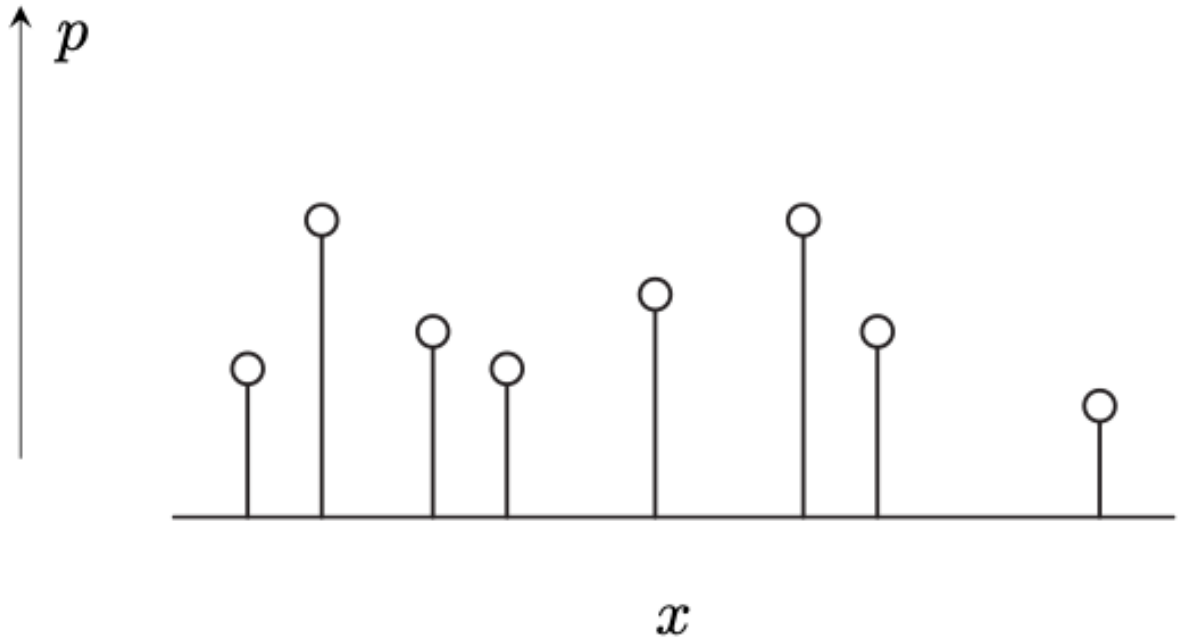
Figure from: W. T. Freeman, J. B. Tenenbaum, "Learning Bilinear Models for Two-Factor Problems in Vision", 1996

Probability is part of the modeling

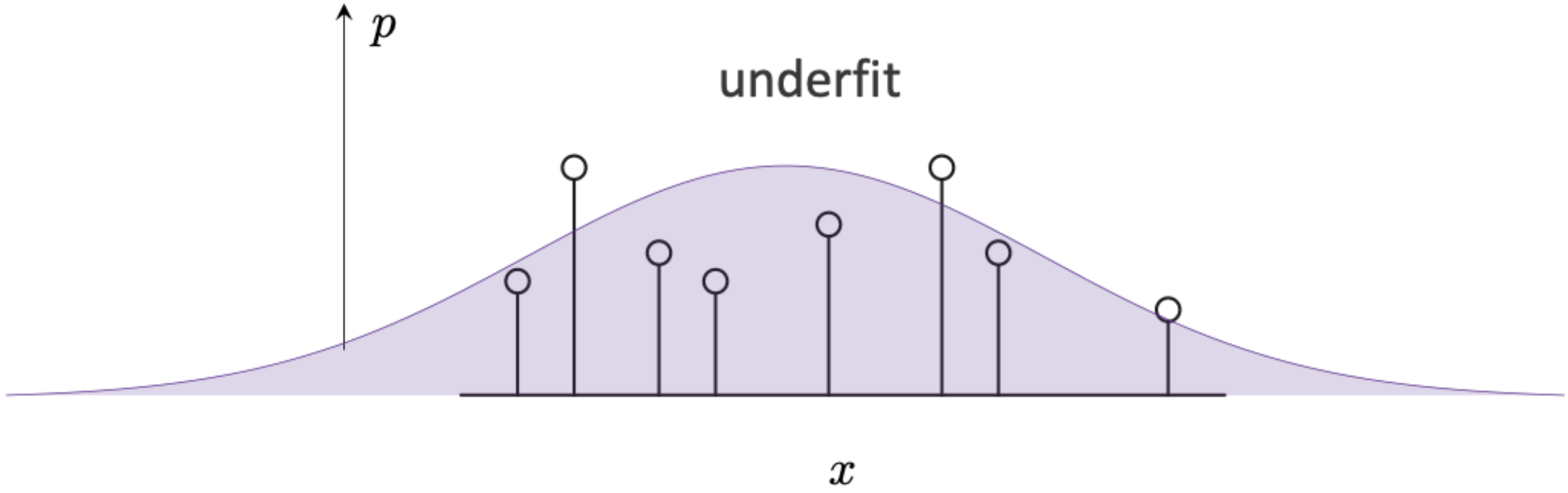
- There may not be “underlying” distributions.
- Even there are, what we can observe are a **finite** set of data points
- The models **extrapolate** the observations for modeling distributions
- Overfitting vs. underfitting: like discriminative models

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

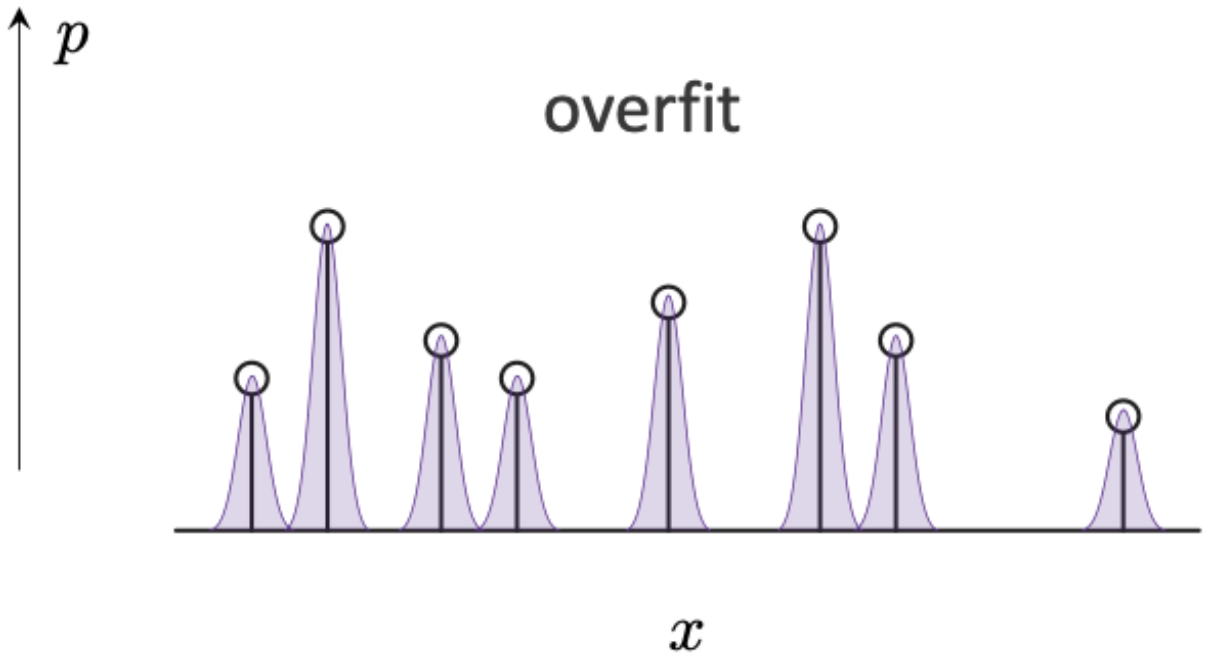
Probability is part of the modeling



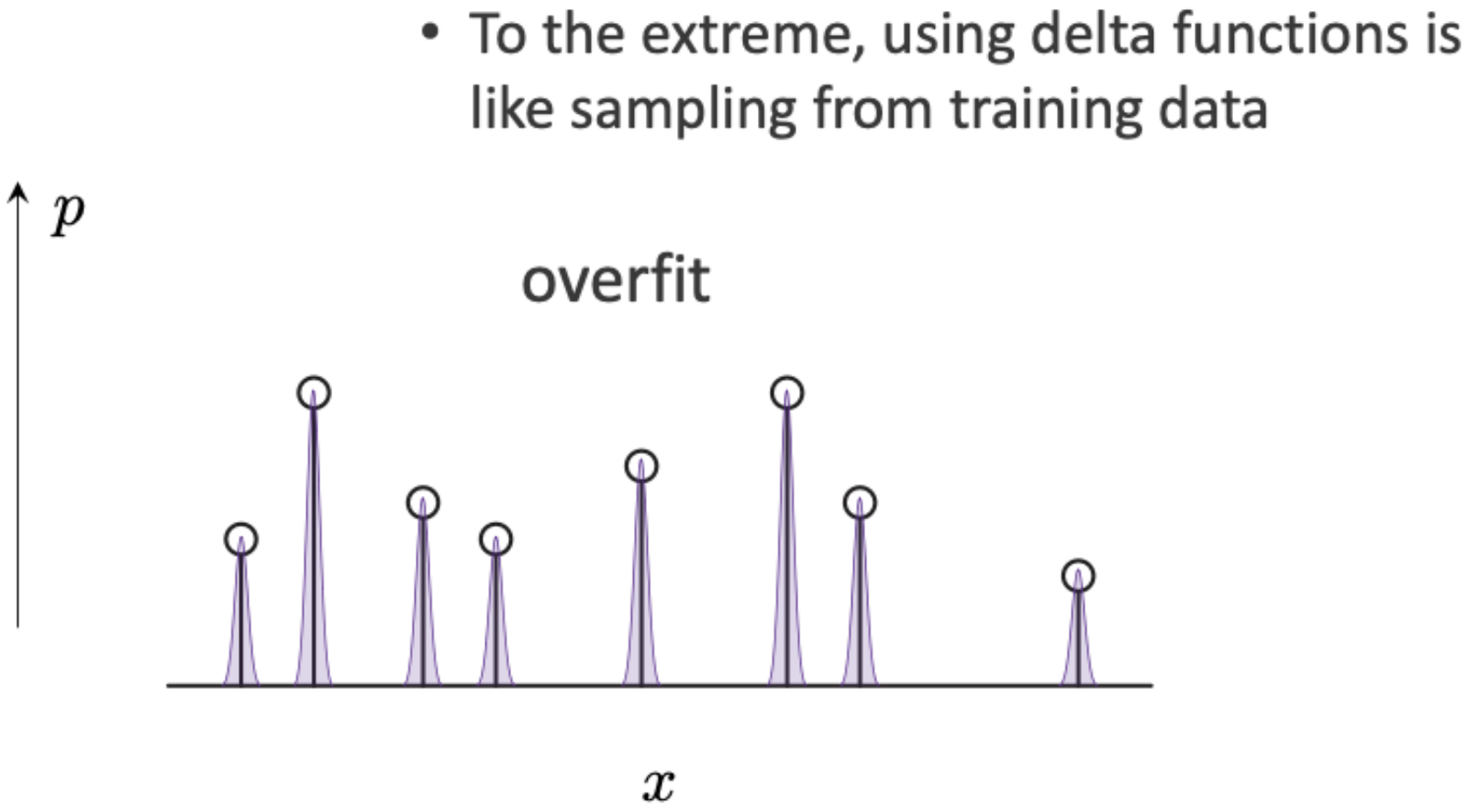
Probability is part of the modeling



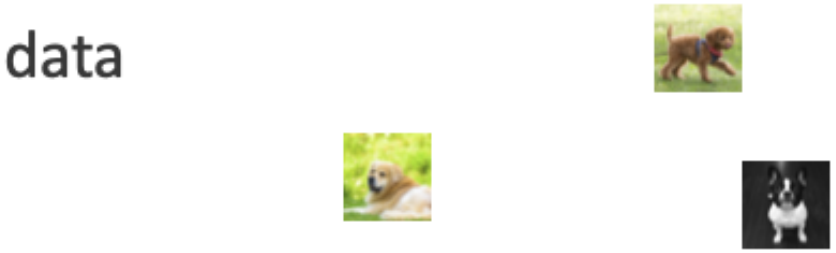
Probability is part of the modeling



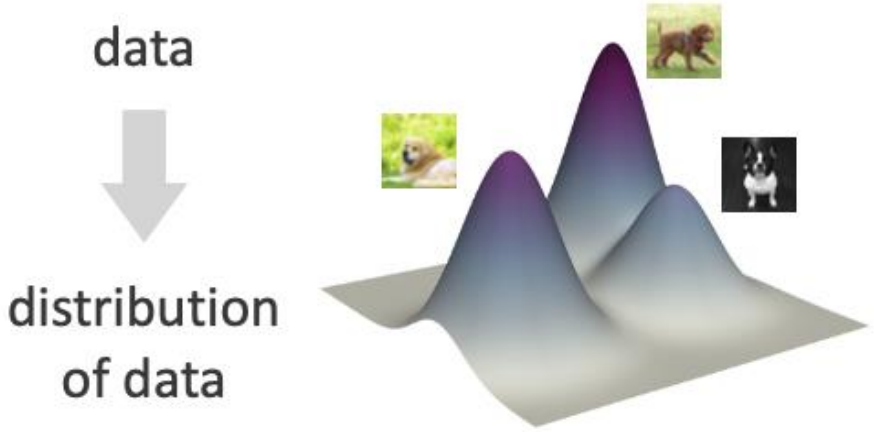
Probability is part of the modeling



Generative models w/ probabilistic modeling

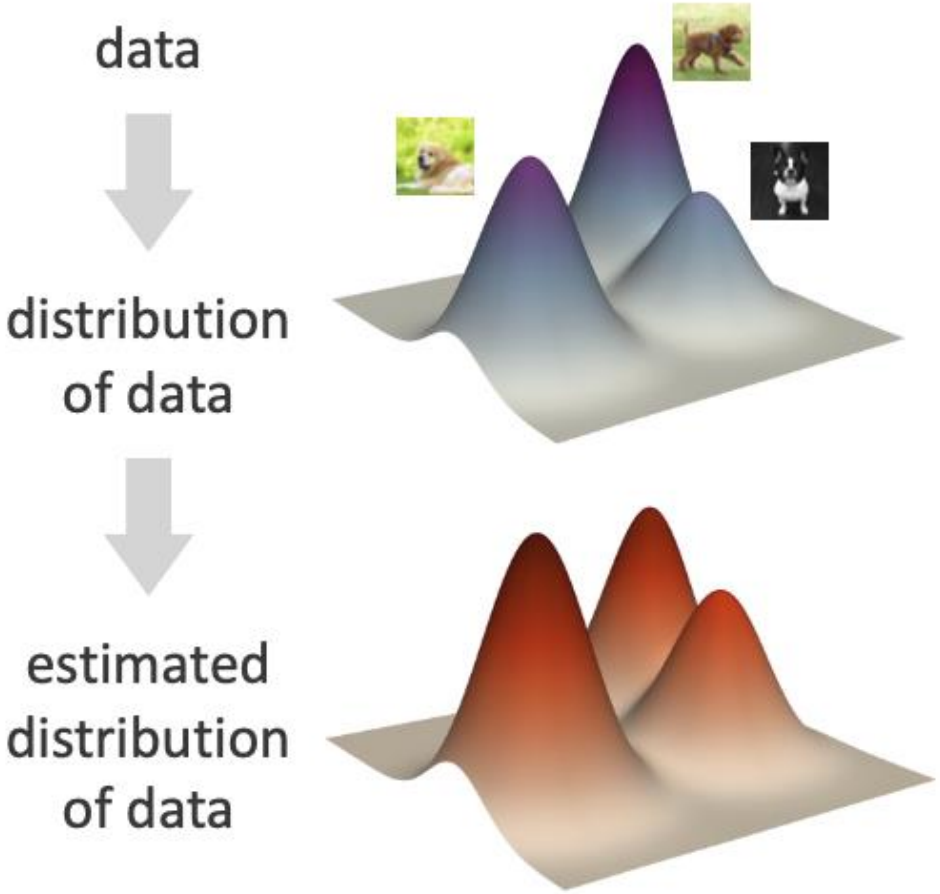


Generative models w/ probabilistic modeling



- This is already part of the modeling

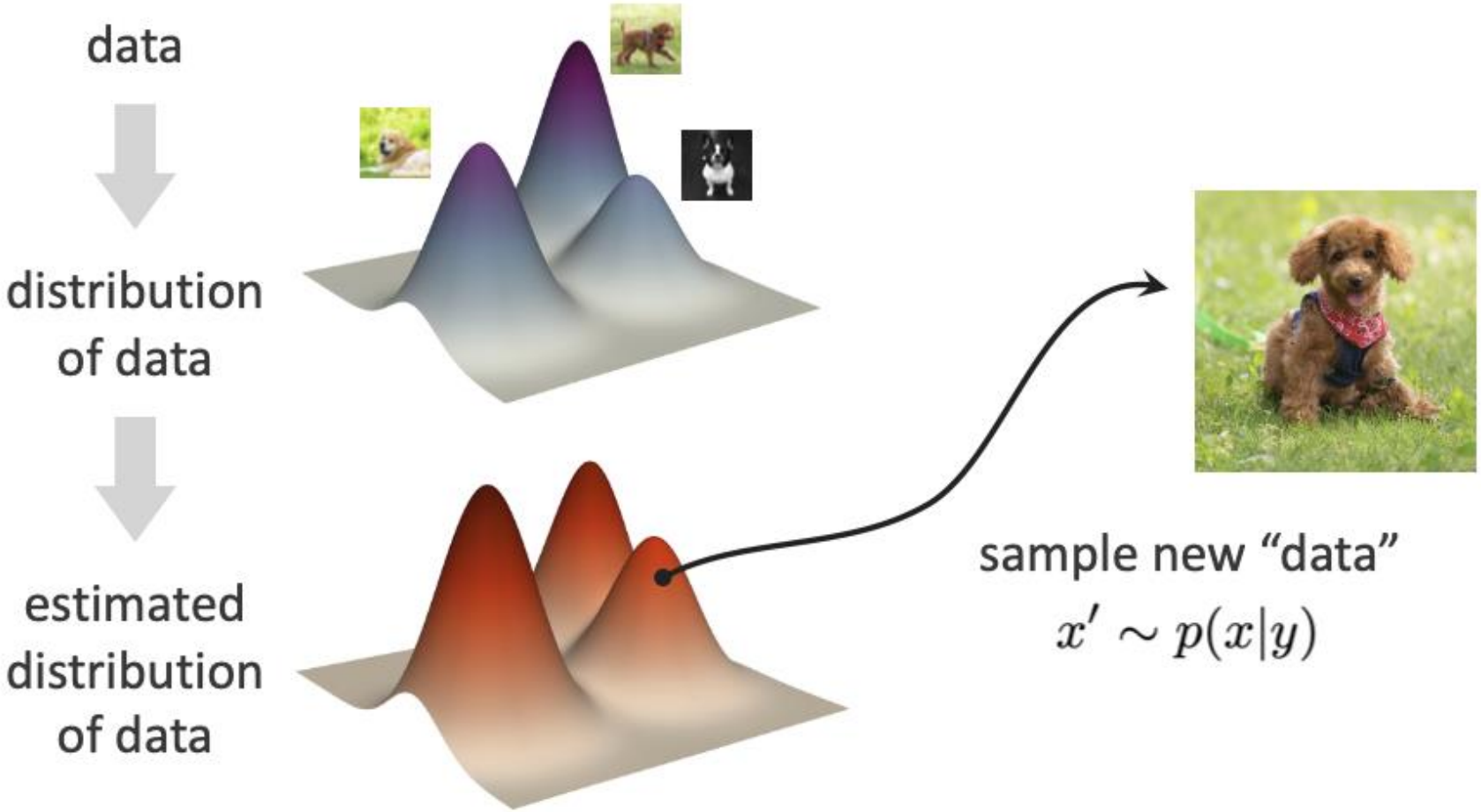
Generative models w/ probabilistic modeling



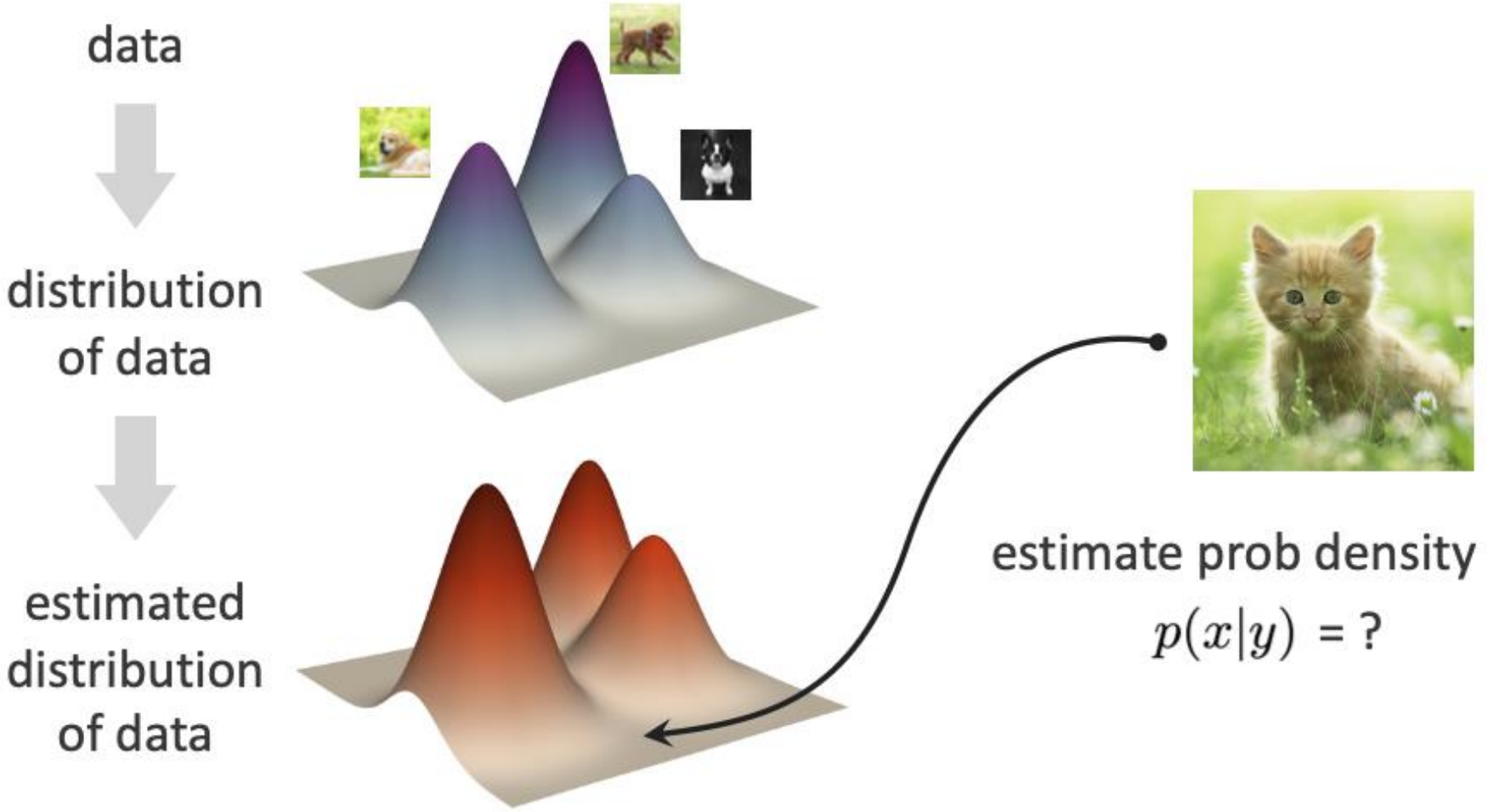
- Optimize a loss function

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

Generative models w/ probabilistic modeling



Generative models w/ probabilistic modeling



Generative models w/ probabilistic modeling

Notes:

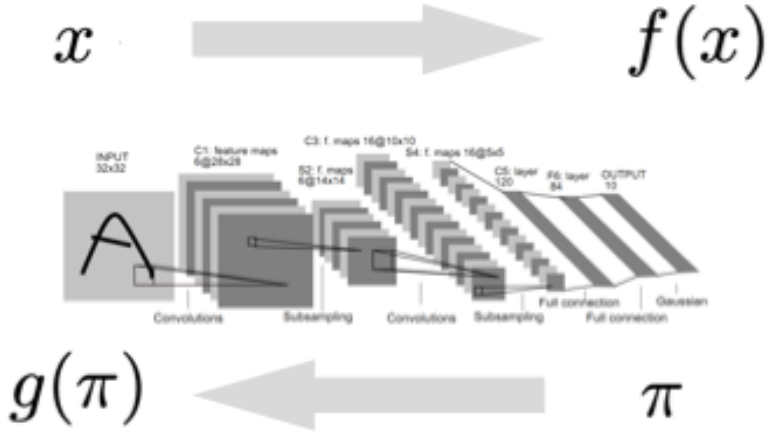
- Generative models involve statistical models which are often designed and derived by humans
- Probabilistic modeling is not just the work of neural nets
- Probabilistic modeling is a popular way, but not the only way
- "All models are wrong, but some are useful.” - George Box

What are Deep Generative Models?

*Acknowledgement: Many of the following slides are modified from Kaiming He

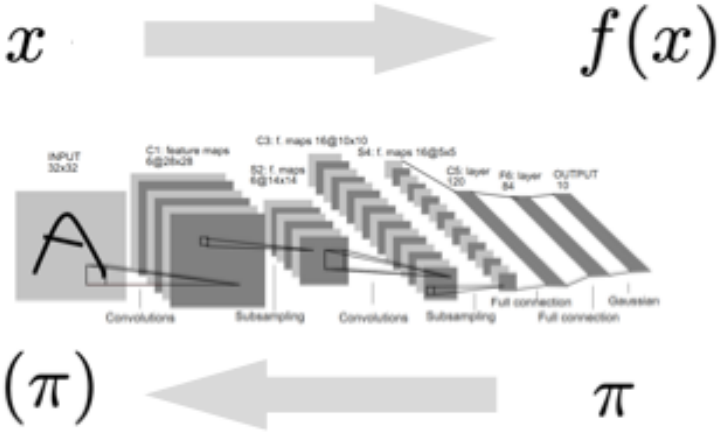
Deep Generative Models

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



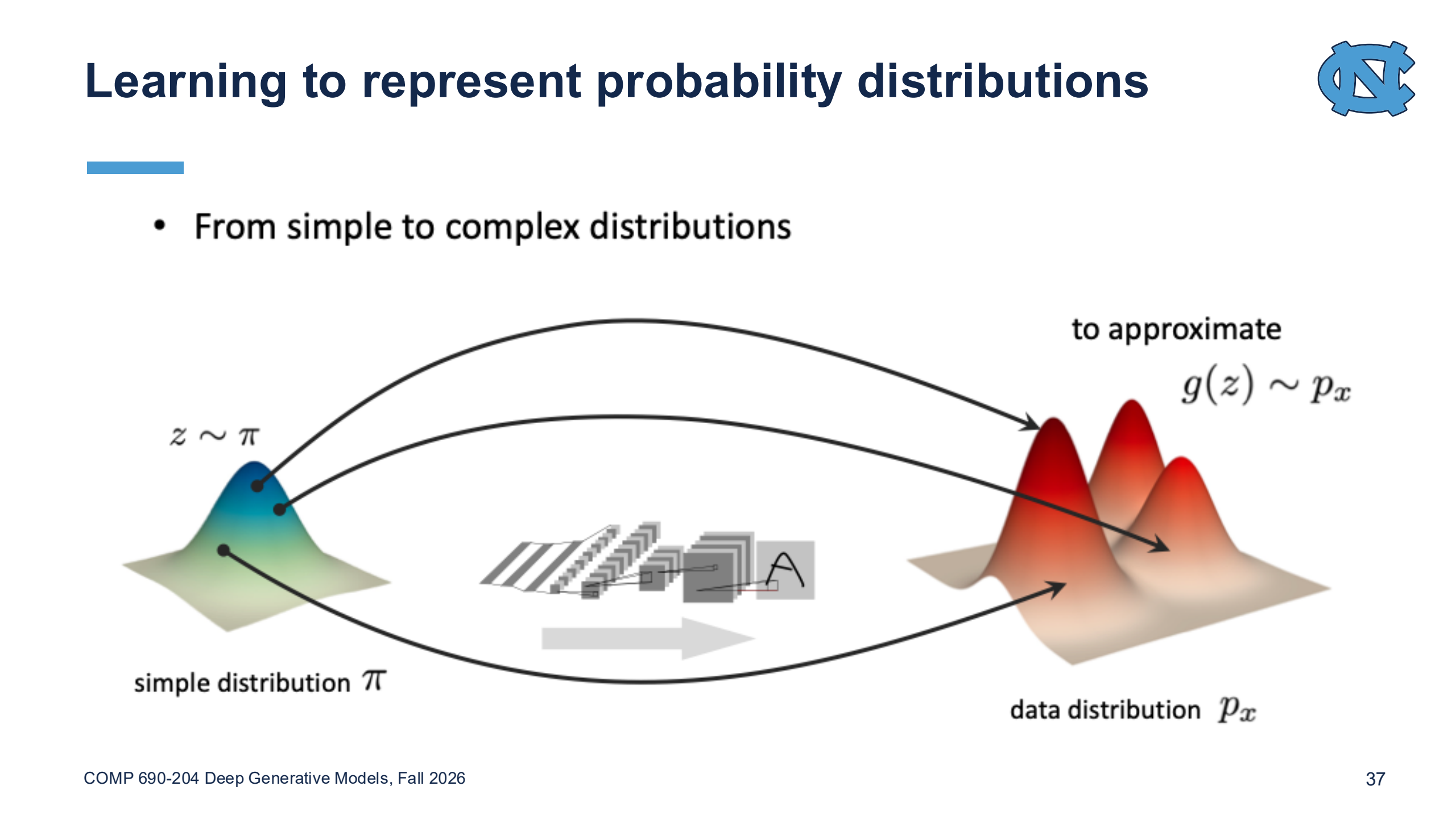
Deep Generative Models

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



- 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



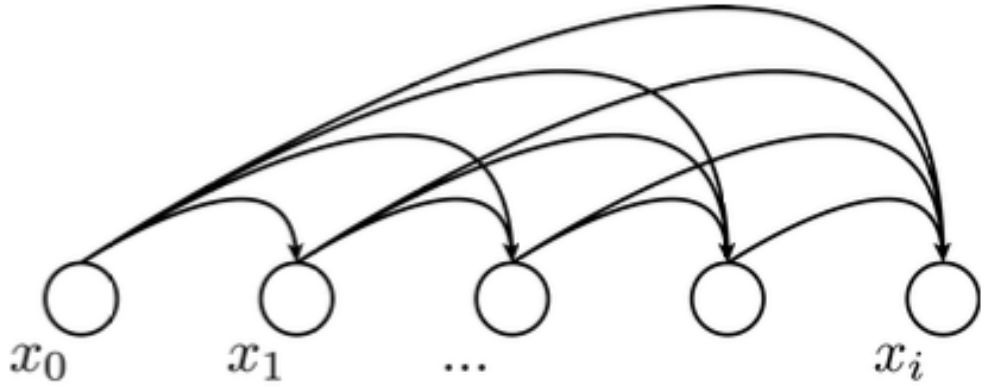
- From simple to complex distributions

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).

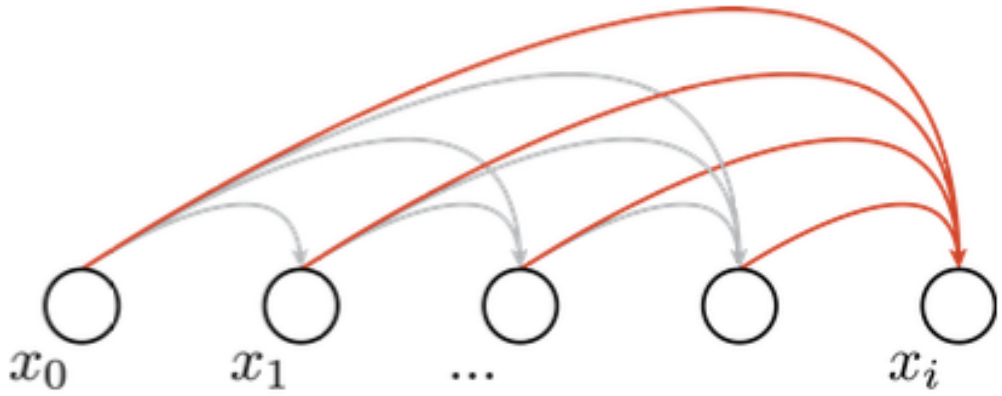


Learning to represent probability distributions

- Not all parts of distribution modeling is done by learning

Case study:
Autoregressive model

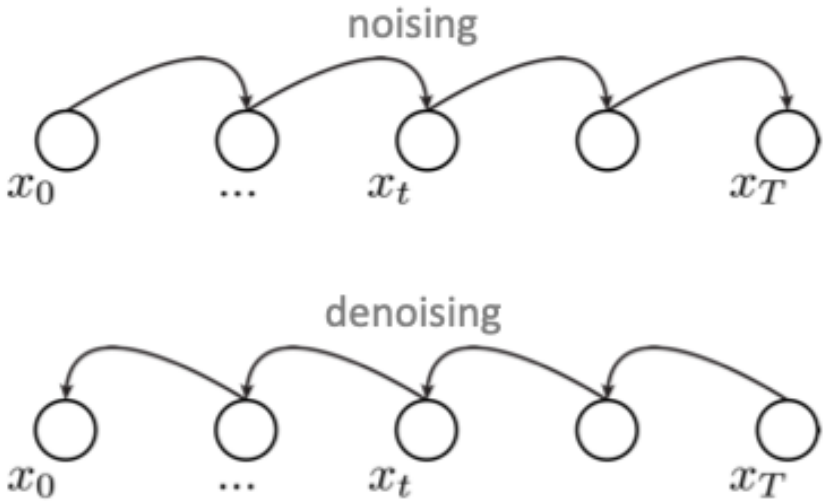
The mapping function is learned
(e.g., Transformer)



Learning to represent probability distributions

- Not all parts of distribution modeling is done by learning

Case study:
Diffusion model

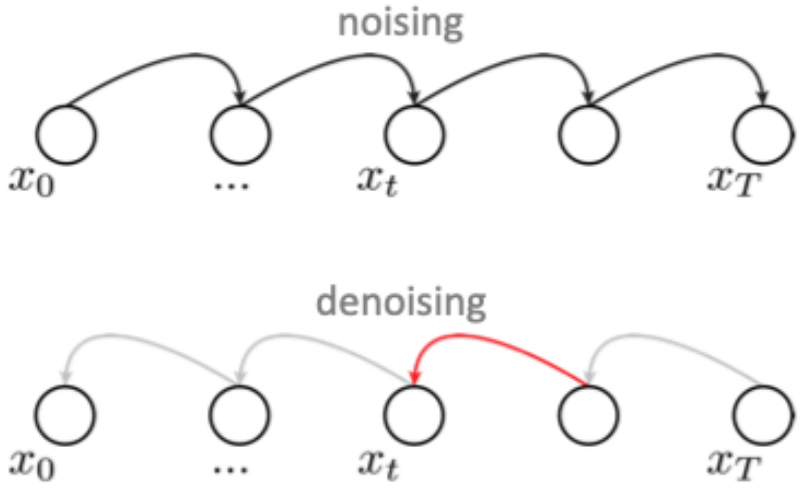


This dependency graph is designed (not learned).

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

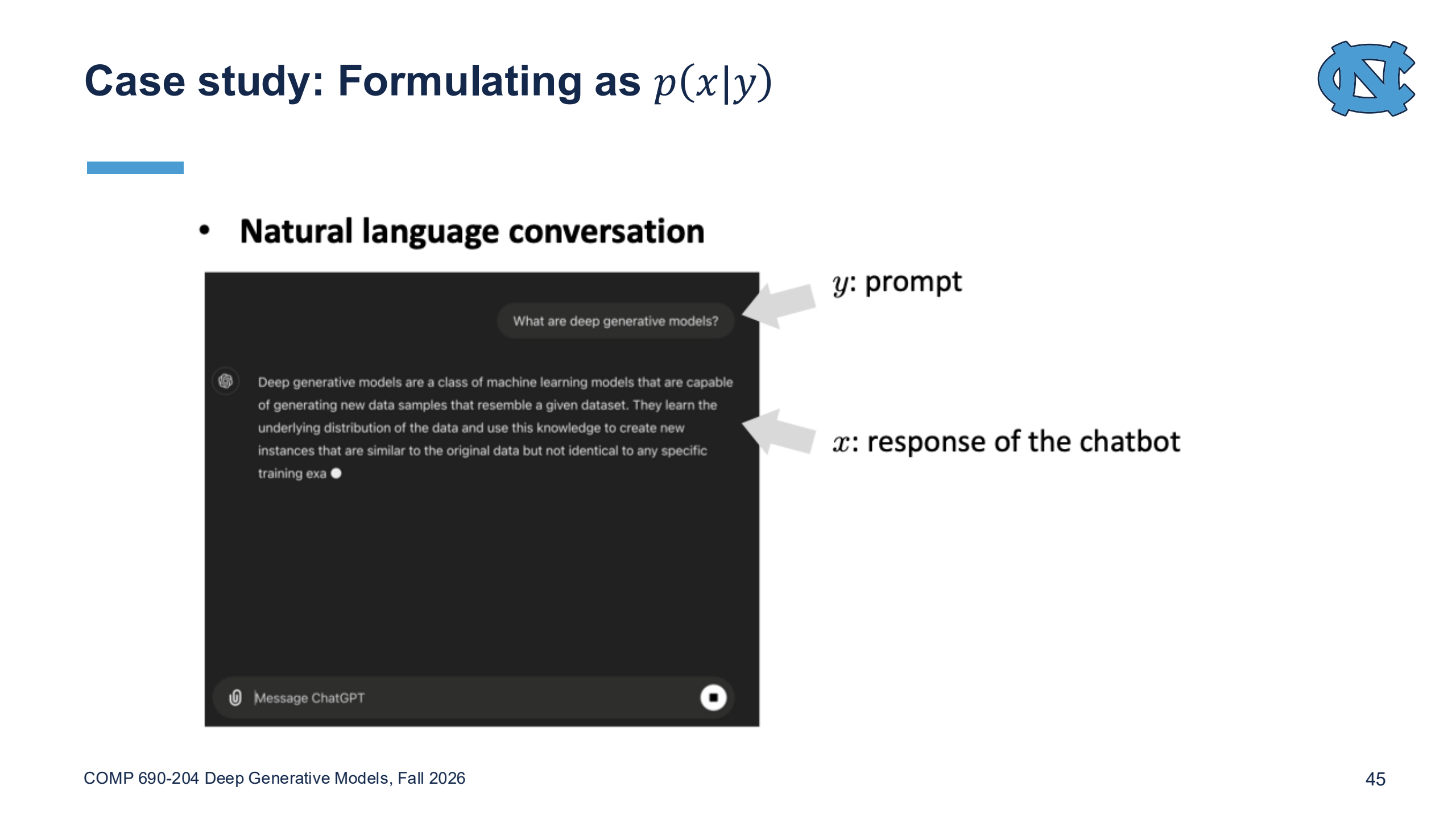
- more abstract
- less informative

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 $p(x|y)$

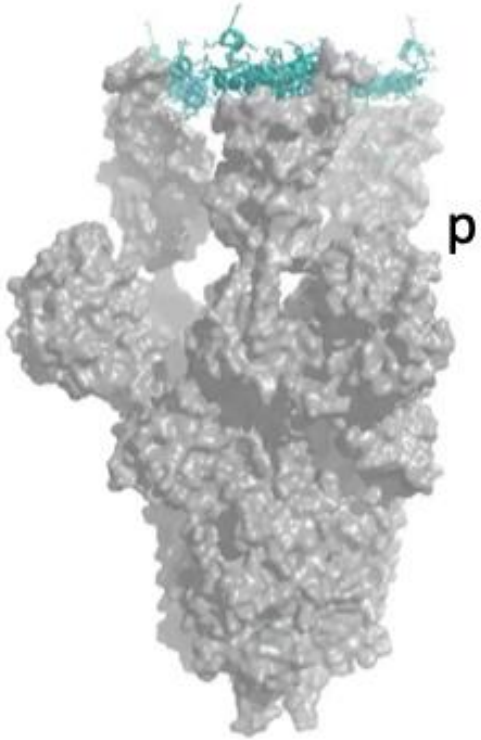
- **Text-to-3D structure generation**

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

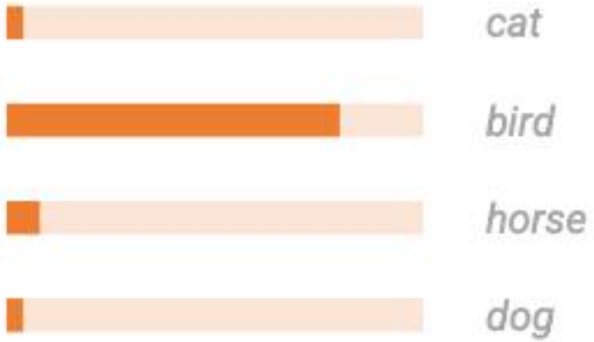
Images generated by: Karras, et al. Elucidating the Design Space of Diffusion-Based Generative Models, NeurIPS 2022

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

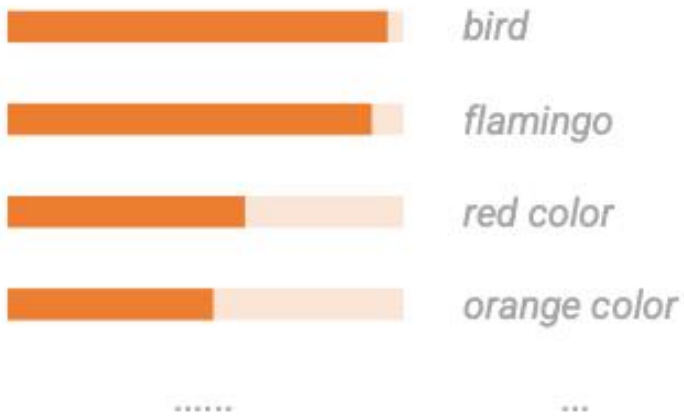


Case study: Formulating as $p(x|y)$

- **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

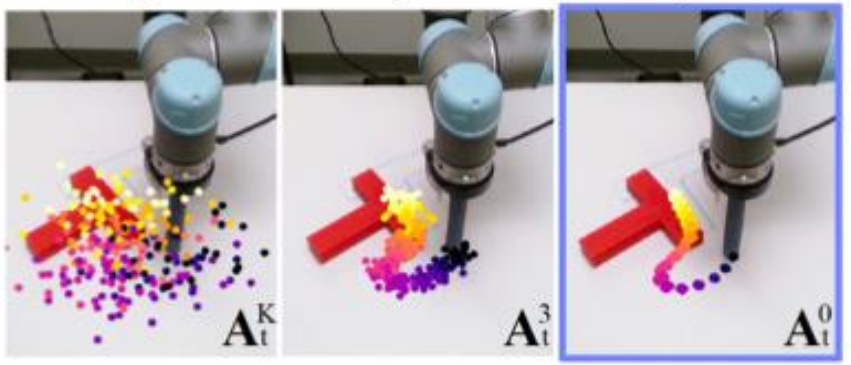
Figure from: GPT-4 Technical Report, 2023

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

- **Policy Learning in Robotics**

y: visual and other sensory observations

x: policies (probability of actions)

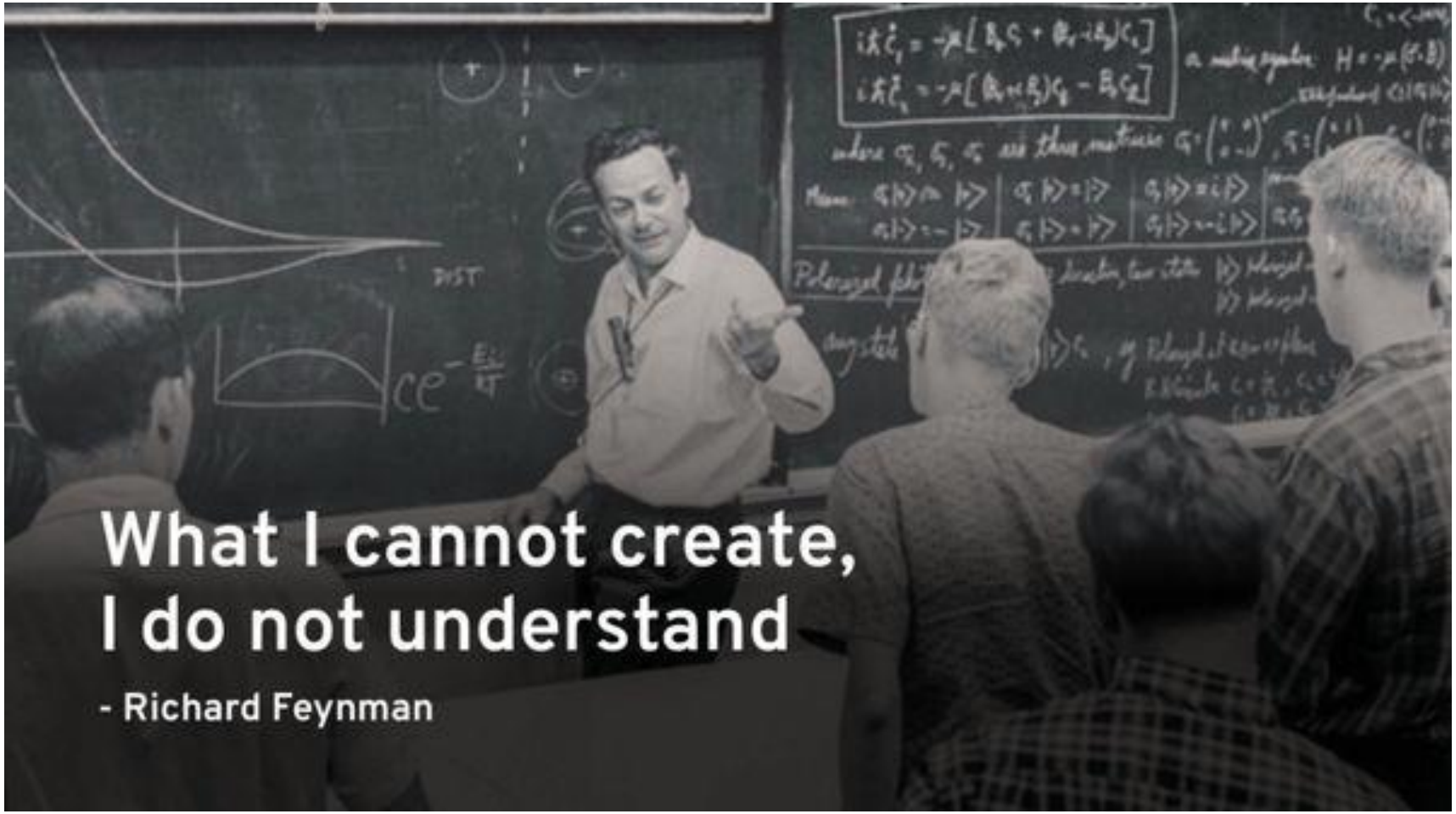


Chi, et al. Diffusion Policy: Visuomotor Policy Learning via Action Diffusion, RSS 2023

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

- 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?

Summary



The University
of North Carolina
at Chapel Hill