

# AI 518: Deep Generative Models

Jaejun Yoo

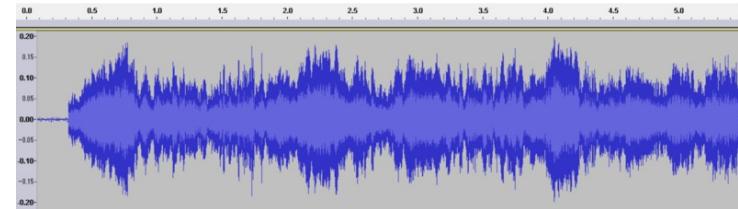
Graduate School of AI & Dept. of EE  
**UNIST**

# The Era of Deep Learning: Toward Human-like AI

**Challenge:** understand complex, unstructured inputs



Computer Vision



Computational Speech



Natural Language Processing



Robotics

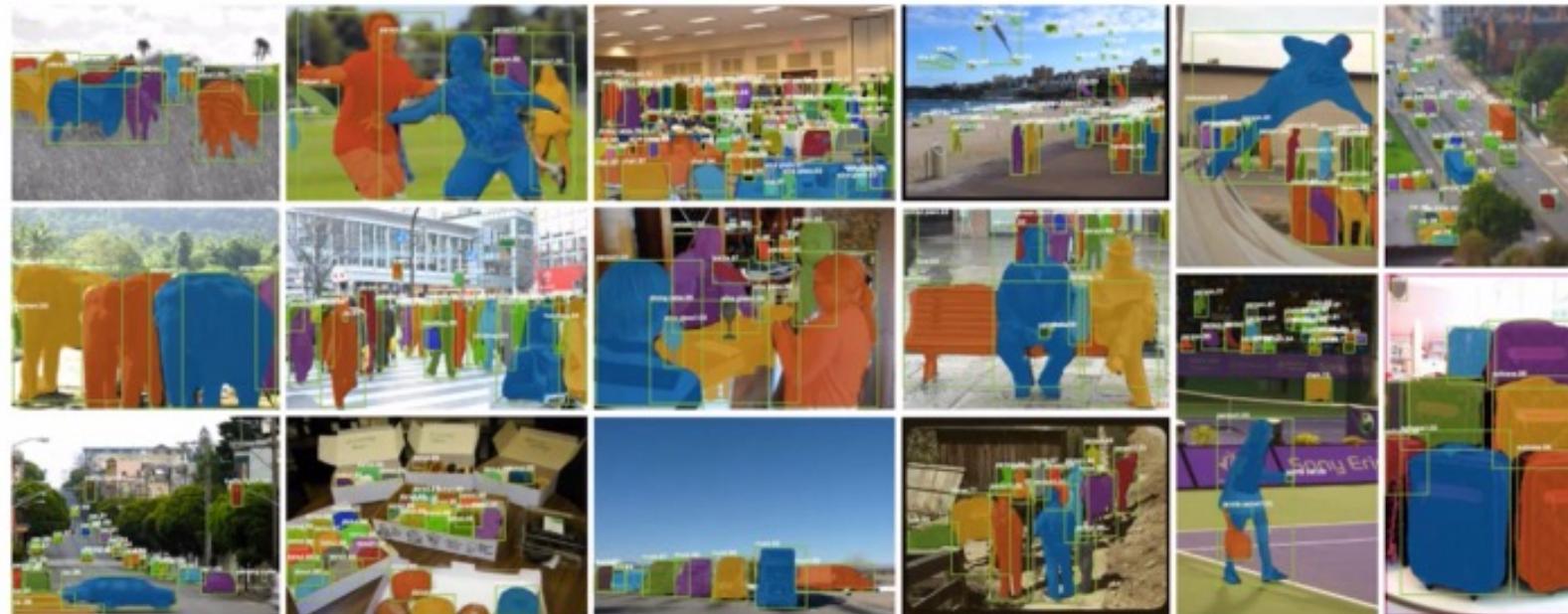
# What does it mean to “see the human way”?

One possibility: Extract from images or video everything that humans can easily, intuitively label.

- Starting ~20 years ago, with Caltech-101, Pascal-VOC, LabelMe, ImageNet, COCO, and many other datasets and benchmarks, computer vision effectively adopted this as its most ambitious goal.
- Driven by a range of factors ...
  - An increasing focus on metrics and measurable progress.
  - The internet as a data resource, and a primary application domain.
  - Ever-expanding compute resources.
  - The rise of machine learning, and then deep learning specifically.
- A good goal: coherent, practical, and scalable – but also limiting.

# Object detection: a success story

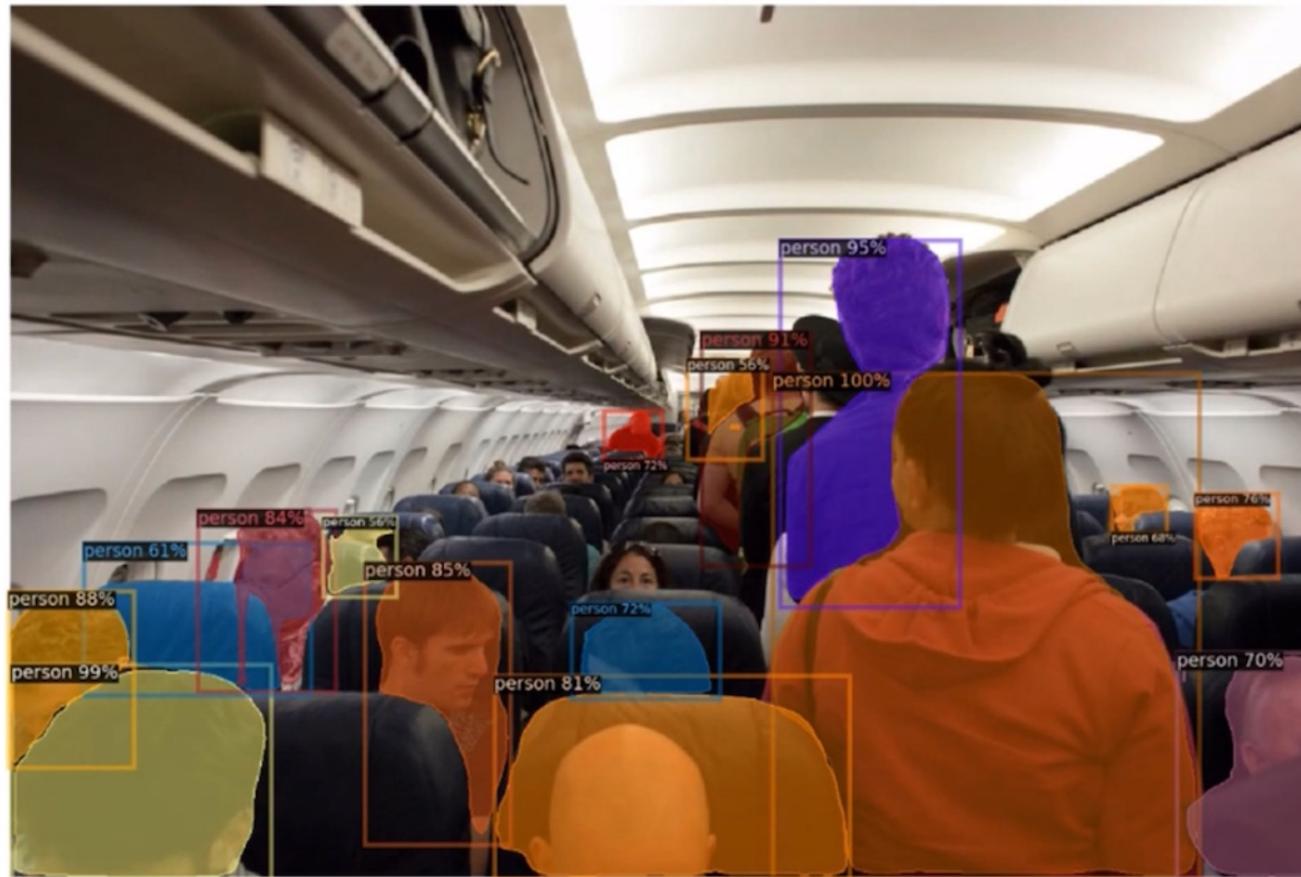
- Mask R-CNN, 2017



# Where are the people?



# Where are the people?



# Where are the people?



# Where are the people?



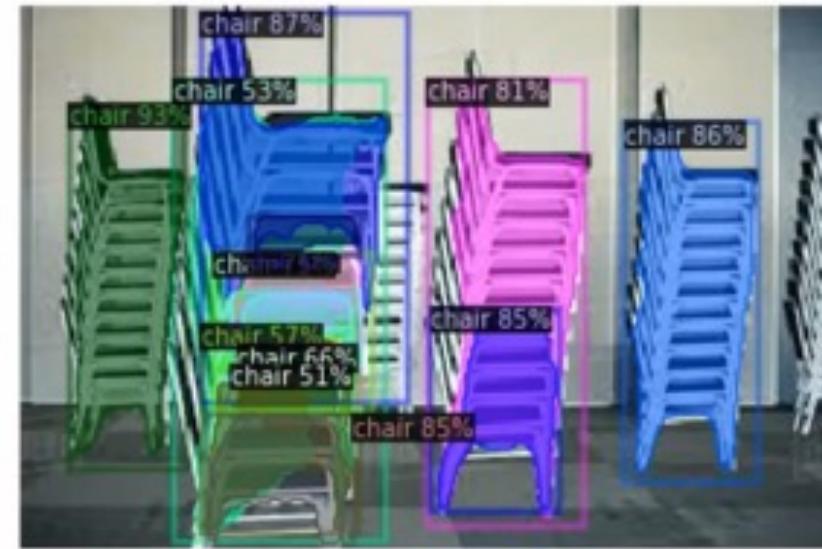
Courtesy of Seojun Kim, 2023

# Where are the people?



# Where are the people?

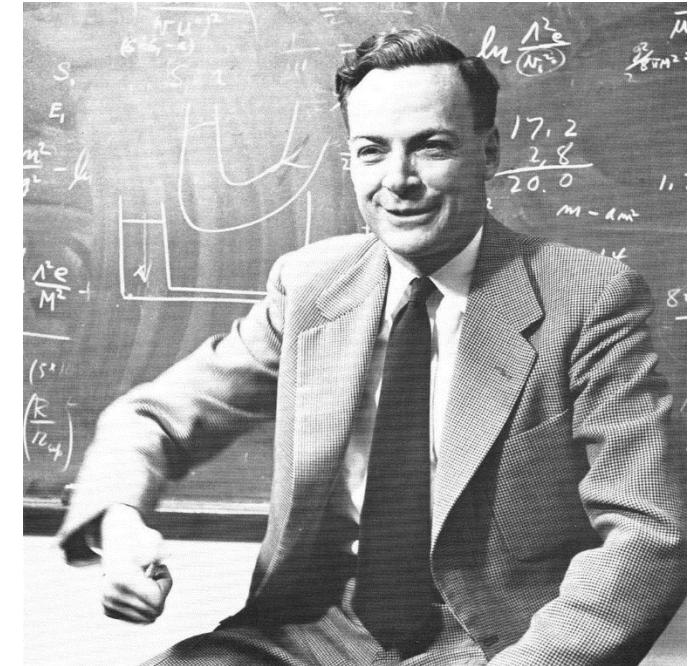




# Where's the gap?

- Intelligence is not just about pattern recognition and decision boundary – *modeling the world (that is observed through data)*.
- It is about *modeling the world* and ourselves in it, and using these models to support inference and action in service of our goals.
  - Explaining and understanding what we see.
  - Imagining things we could see but haven't yet.
  - Planning actions and solving problems to make these things real.
  - Building new models as we learn more about the world.
  - Modeling other minds, to cooperate and learn socially and culturally.

# Motivation: Old fantasy



Richard Feynman: “*What I cannot create, I do not understand*”

Generative modeling: “*What I understand, I can **create***”

# Motivation: (Very) Old fantasy



# Motivation: Old fantasy



# Motivation: ~~Old~~ fantasy



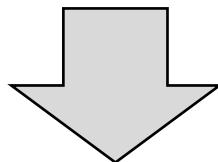
# Generative Modeling: Computer Graphics

How to generate natural images with a computer?

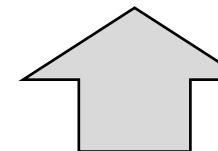
High level  
description

Cube(color=blue, position=, size=, ...)  
Cylinder(color=red, position=, size=,..)

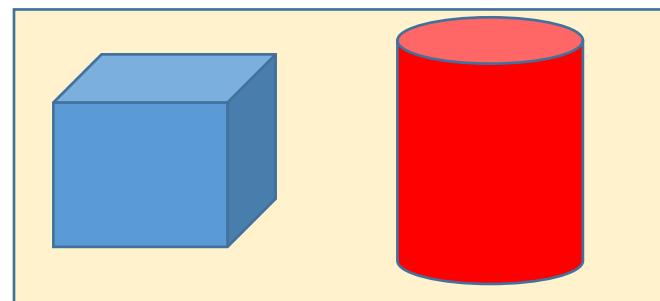
**Generation** (graphics)



**Inference** (vision)



Raw sensory  
outputs



Our models will have **similar structure (generation + inference)**

# Statistical Generative Models

Statistical generative models are **learned from data**



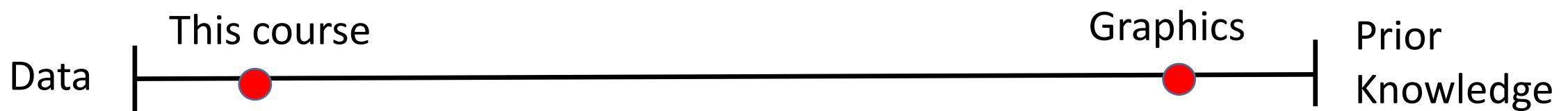
...



Data  
(e.g., images of bedrooms)

Prior Knowledge  
(e.g., physics, materials, ..)

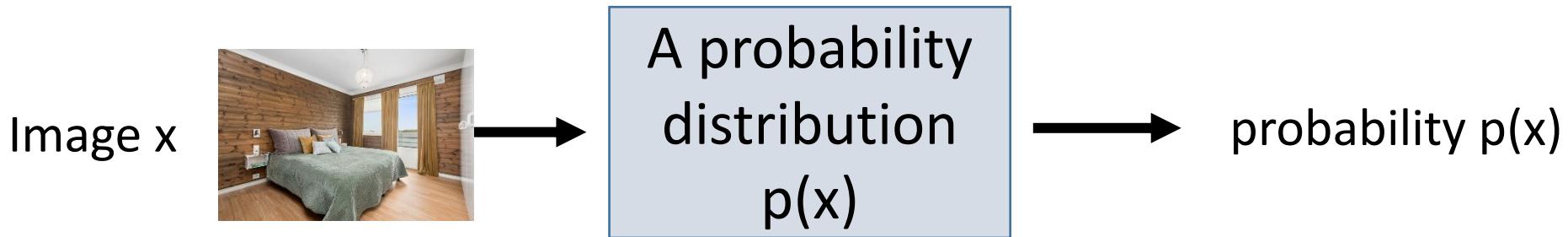
Priors are **always necessary**, but there is a spectrum



# Statistical Generative Models

A statistical generative model is a **probability distribution**  $p(x)$

- **Data:** samples (e.g., images of bedrooms)
- **Prior knowledge:** parametric form (e.g., Gaussian?), loss function (e.g., maximum likelihood?), optimization algorithm, etc.



It is generative because **sampling from  $p(x)$  generates new images**

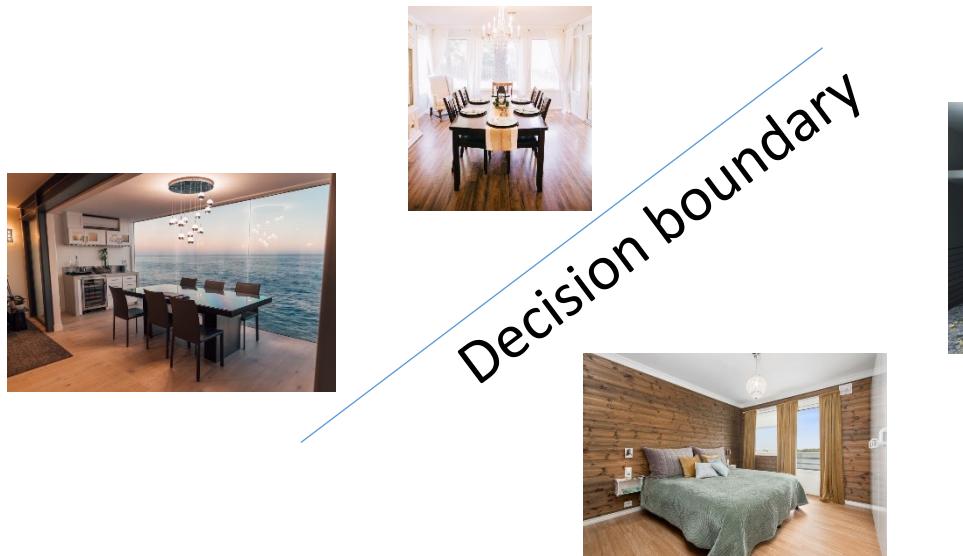


...



# Discriminative vs. generative

**Discriminative:** classify bedroom vs. dining room

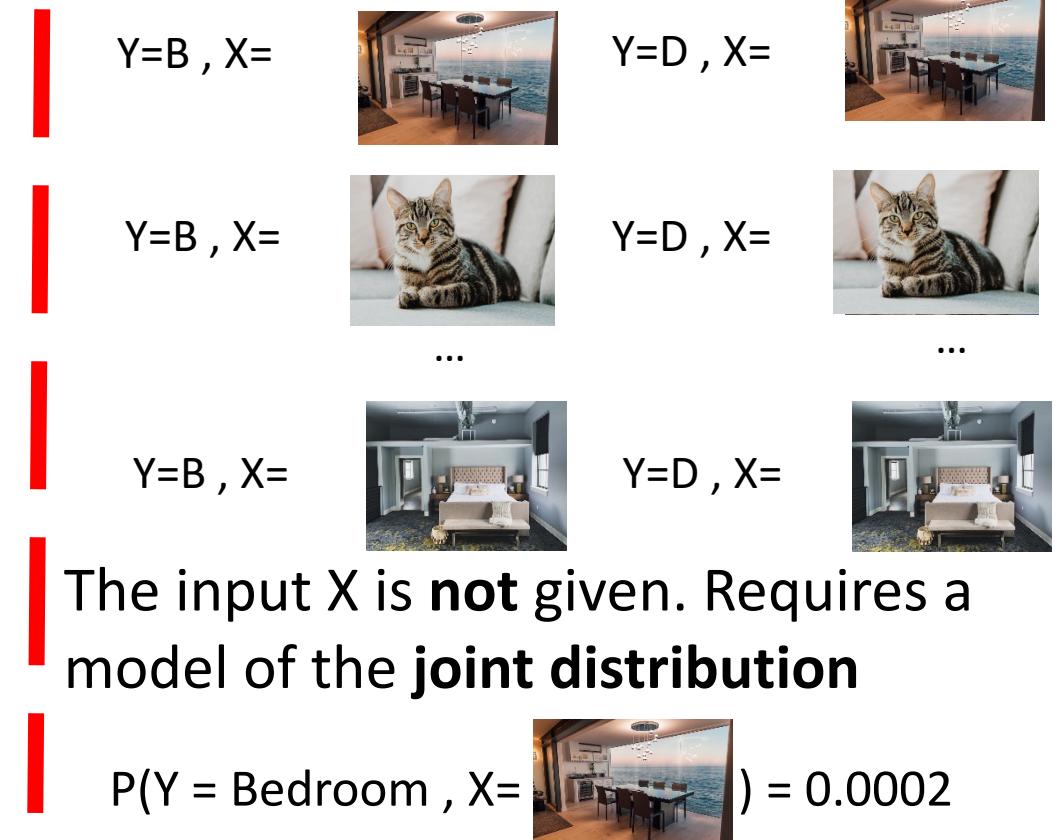


The input image  $X$  is always given. **Goal:** a good decision boundary, via **conditional distribution**

$$P(Y = \text{Bedroom} \mid X= \text{[dining room image]}) = 0.0001$$

Ex: logistic regression, convolutional net, etc.

**Generative:** generate  $X$



The input  $X$  is **not** given. Requires a model of the **joint distribution**

$$P(Y = \text{Bedroom} , X= \text{[dining room image]}) = 0.0002$$

# Discriminative vs. generative

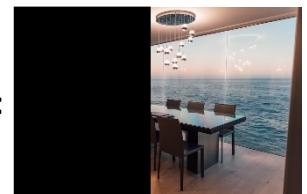
Joint and conditional are related via **Bayes Rule**:

$$P(Y = \text{Bedroom} | X = \text{ }) = \frac{P(Y = \text{Bedroom}, X = \text{ })}{P(X = \text{ })}$$


**Discriminative:**  $X$  is always given, does not need to model

$$P(X = \text{ })$$

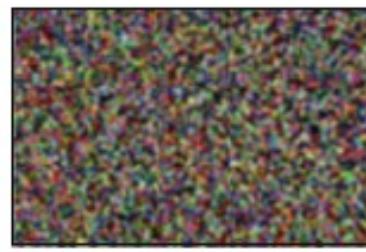

Therefore it cannot handle missing data

$$P(Y = \text{Bedroom} | X = \text{ })$$


# Discriminative vs. generative



+



=

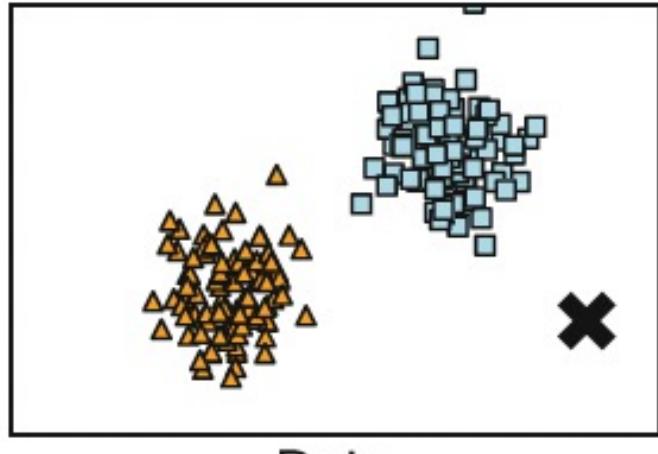


$$\begin{aligned} p(y = \text{cat}|\mathbf{x}) &= 0.90 \\ p(y = \text{dog}|\mathbf{x}) &= 0.05 \\ p(y = \text{horse}|\mathbf{x}) &= 0.05 \end{aligned}$$

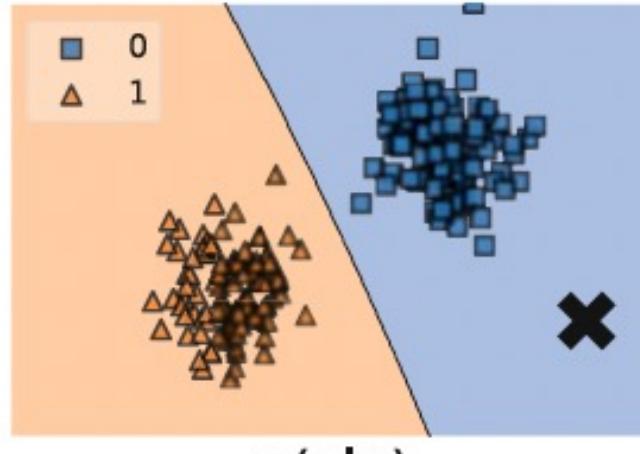
$$\begin{aligned} p(y = \text{cat}|\mathbf{x}) &= 0.05 \\ p(y = \text{dog}|\mathbf{x}) &= 0.05 \\ p(y = \text{horse}|\mathbf{x}) &= 0.90 \end{aligned}$$

**Fig. 1.1** An example of adding noise to an almost perfectly classified image that results in a shift of predicted label

# Discriminative vs. generative

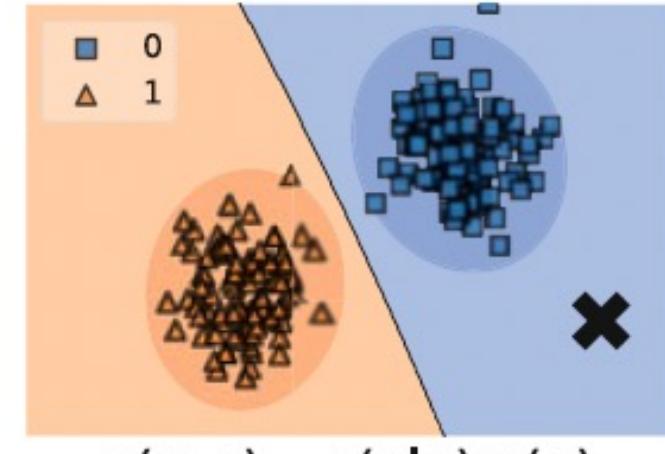


Data



$p(y|\mathbf{x})$

$p(\text{blue}|\mathbf{x})$  is high  
= certain decision!



$p(\mathbf{x}, y) = p(y|\mathbf{x}) p(\mathbf{x})$

$p(\text{blue}|\mathbf{x})$  is high  
and  $p(\mathbf{x})$  is low  
= uncertain decision!

**Fig. 1.2** An example of data (*left*) and two approaches to decision making: (*middle*) a discriminative approach and (*right*) a generative approach

# Conditional Generative Models

Class **conditional generative models** are also possible:

$$P(X = \text{[image]} | Y = \text{Bedroom})$$



It's often useful to condition on rich side information Y

$$P(X = \text{[image]} | \text{Caption} = \text{"A black table with 6 chairs"})$$

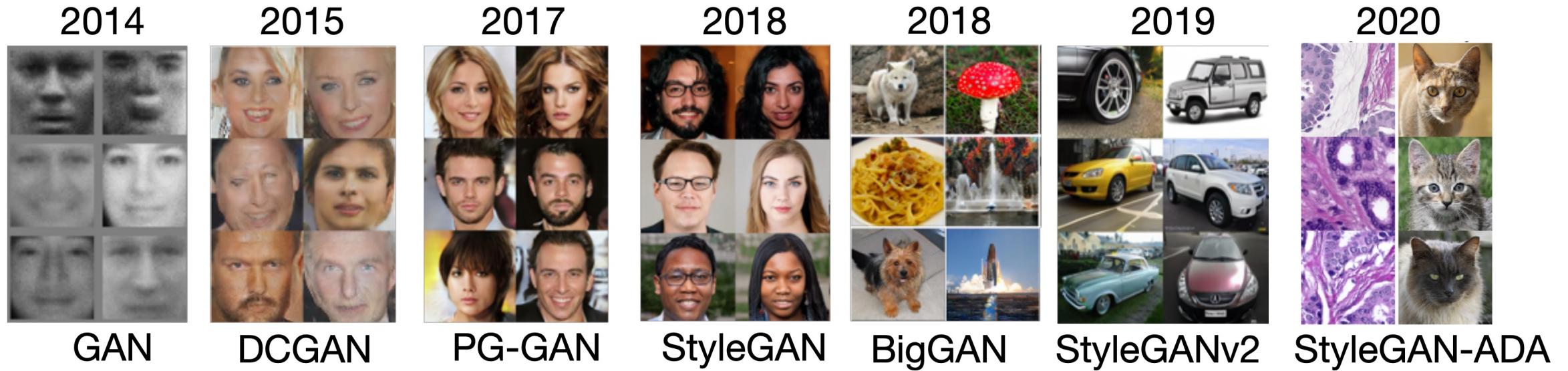


A discriminative model is a very simple conditional generative model of Y:

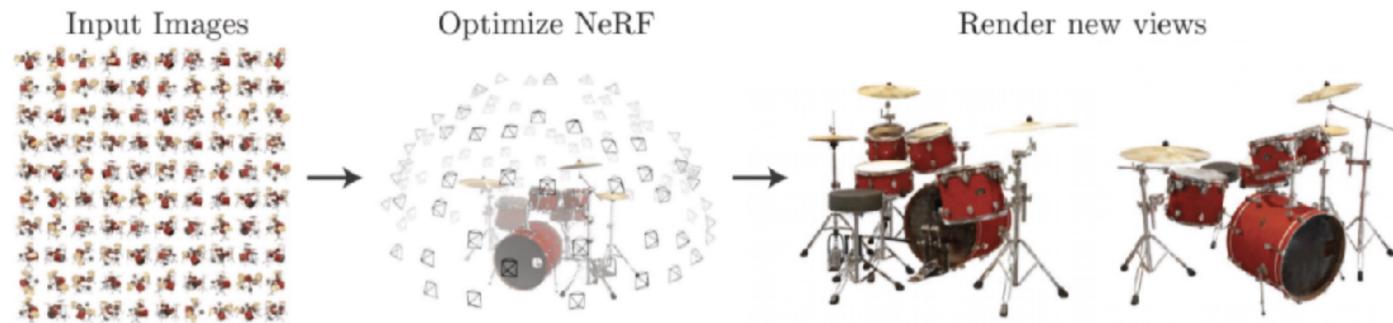
$$P(Y = \text{Bedroom} | X = \text{[image]})$$



# Image Generation (Ver. 2021)



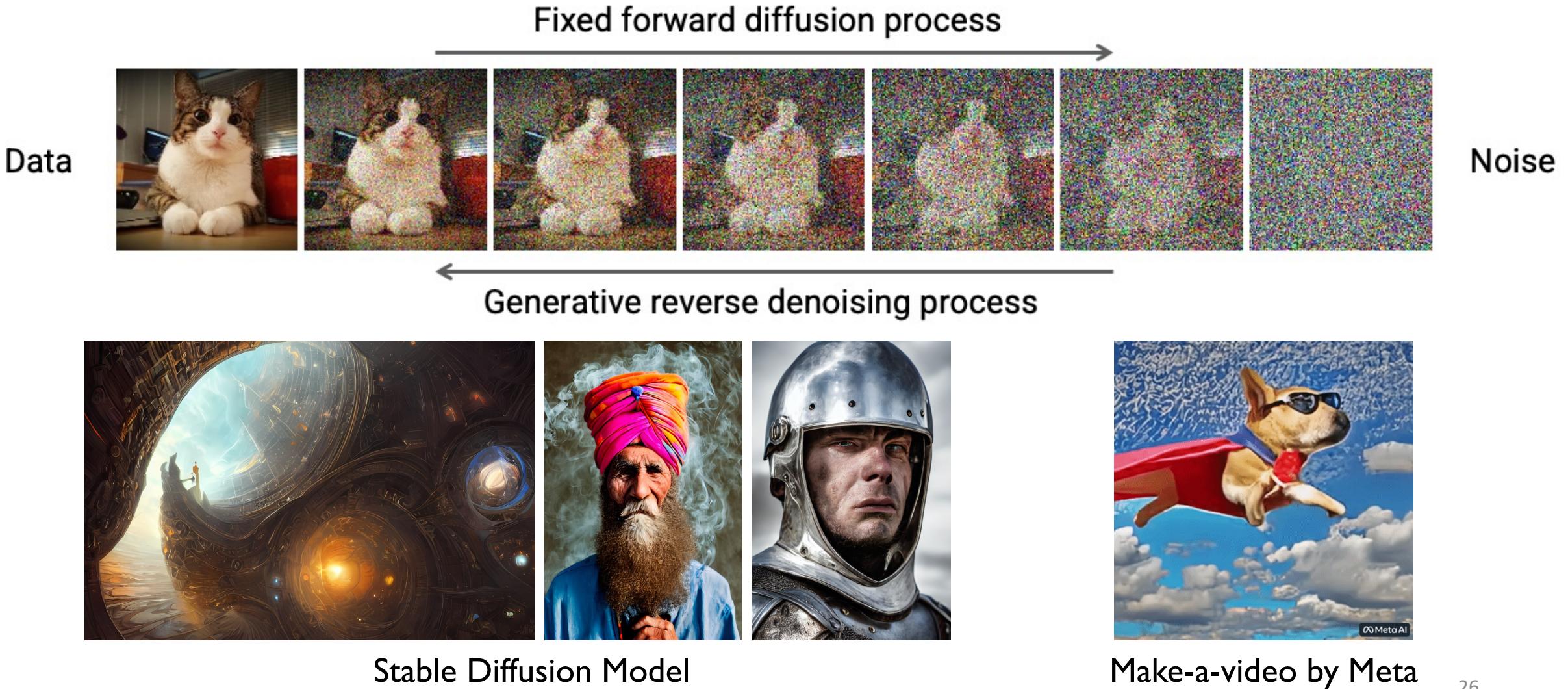
2020: NeRF (Neural Radiance Fields)



2021: OpenAI DALLE (VQ-VAE)  
Arm chair in shape of avocado



# Image Generation (Ver. 2022)



# Image Generation (Ver. 2023)

# Image Generation (Ver. 2024)



# Language Generation (Ver. 2019)

## Completion

Custom prompt

To get an A+ in deep generative models, students have to

**To get an A+ in deep generative models, students have to** be willing to work with problems that are a whole lot more interesting than, say, the ones that most students work on in class. If you're a great student, the question above can be avoided and you'll be able to do great work, but if you're not, you will need to go beyond the basics before getting good.

Now to be clear, this advice is not just for the deep-learning crowd; it is good advice for any student who is taking his or her first course in machine learning.

The key point is that if you have a deep, deep brain of a computer scientist, that's just as important to you.

Radford et al., 2019  
Demo from [talktotransformer.com](http://talktotransformer.com)

# Language Generation (Ver. 2022)

JA

To get an A+ in deep generative models, student have to...?



To get an A+ in deep generative models, a student should demonstrate a thorough understanding of the following concepts and skills:

1. Fundamentals of probability theory: Deep generative models are based on probabilistic models, so a good understanding of probability theory is essential.
2. Machine learning basics: Understanding of the basics of supervised and unsupervised learning, including concepts such as optimization, gradient descent, and backpropagation.
3. Variational autoencoders (VAEs): VAEs are a popular deep generative model that uses variational inference to approximate the true posterior distribution. A student should have a good understanding of VAEs, including their architecture, training, and applications.
4. Generative adversarial networks (GANs): GANs are another popular deep generative model that involves training two neural networks against each other. A student should have a good understanding of GANs, including their architecture, training, and applications.
5. Deep belief networks (DBNs): DBNs are deep generative models that are based on layers of restricted Boltzmann machines (RBMs). A student should have a good understanding of DBNs, including their architecture, training, and applications.
6. Real-world applications: A student should be able to apply deep generative models to real-world problems, such as image generation, text generation, and speech synthesis.

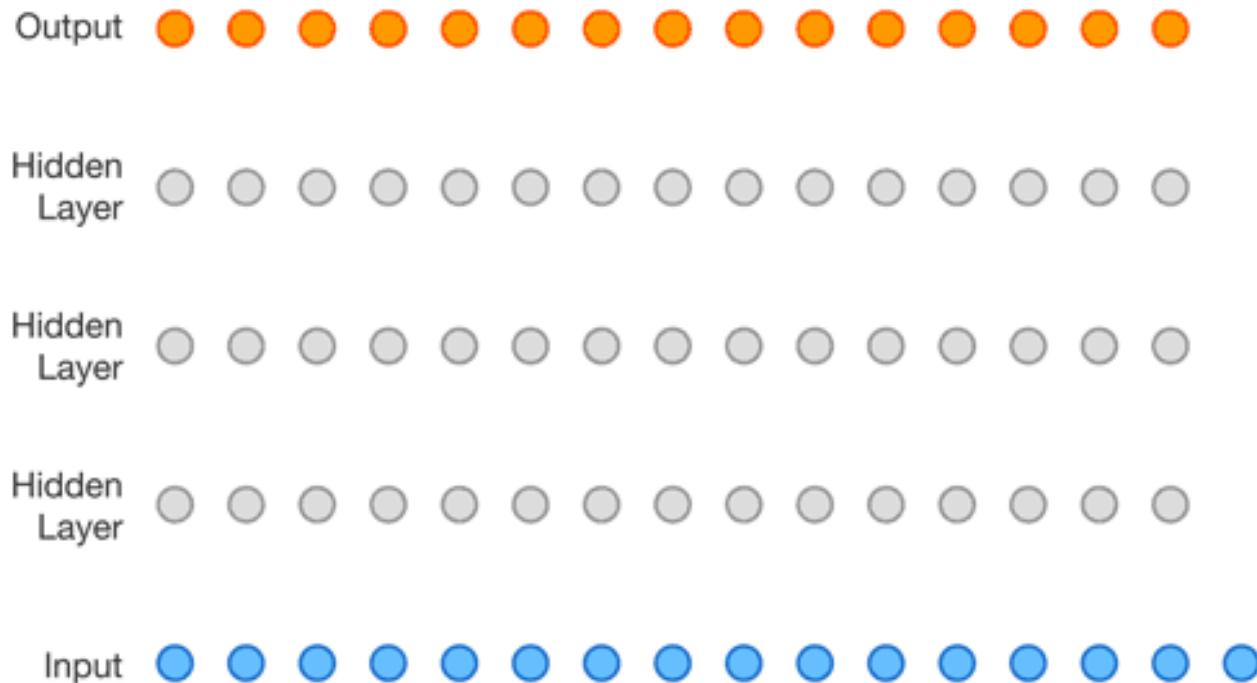
7. Programming skills: A student should be proficient in programming using popular deep learning frameworks such as TensorFlow or PyTorch.

Overall, to get an A+ in deep generative models, a student should have a deep understanding of the fundamental concepts of probability theory and machine learning, as well as the ability to apply this knowledge to real-world problems using a range of deep generative models.

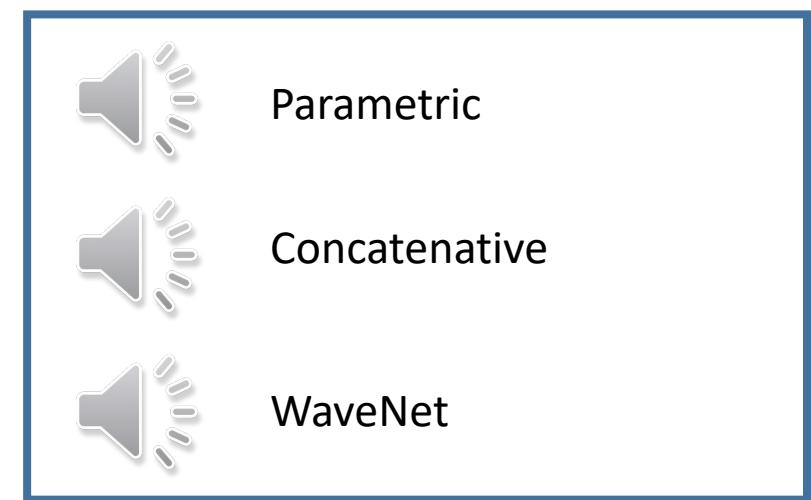
OpenAI, 2022  
Demo from <https://chat.openai.com/chat>

# WaveNet

## Generative model of speech signals



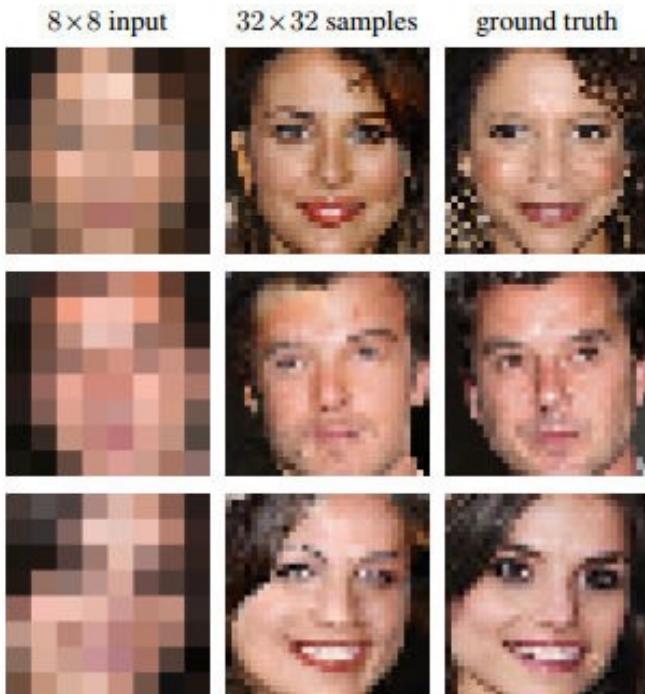
### Text to Speech



van den Oord et al, 2016c

# Image Super Resolution

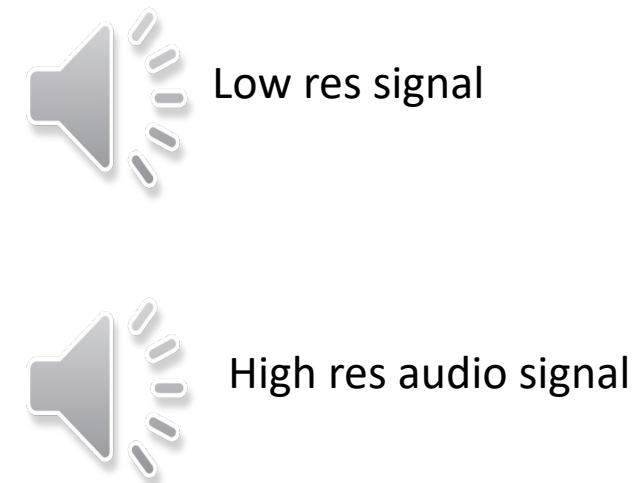
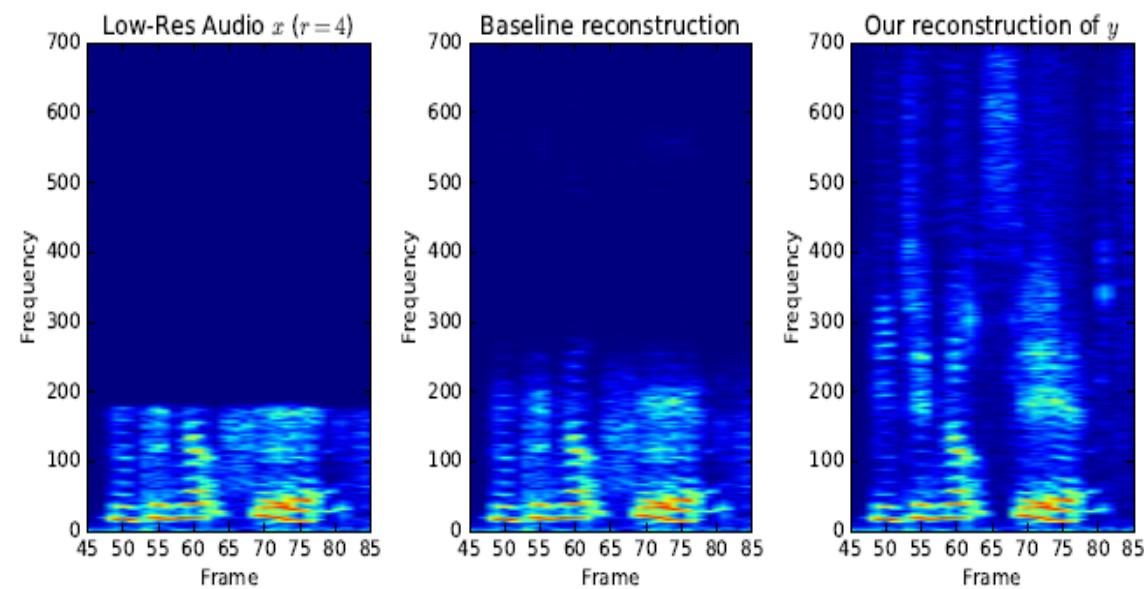
Conditional generative model  $P(\text{high res image} \mid \text{low res image})$



Ledig et al., 2017

# Audio Super Resolution

Conditional generative model  $P(\text{high-res signal} \mid \text{low-res audio signal})$



Kuleshov et al., 2017

# Machine Translation

Conditional generative model  $P(\text{ English text} | \text{ Chinese text})$

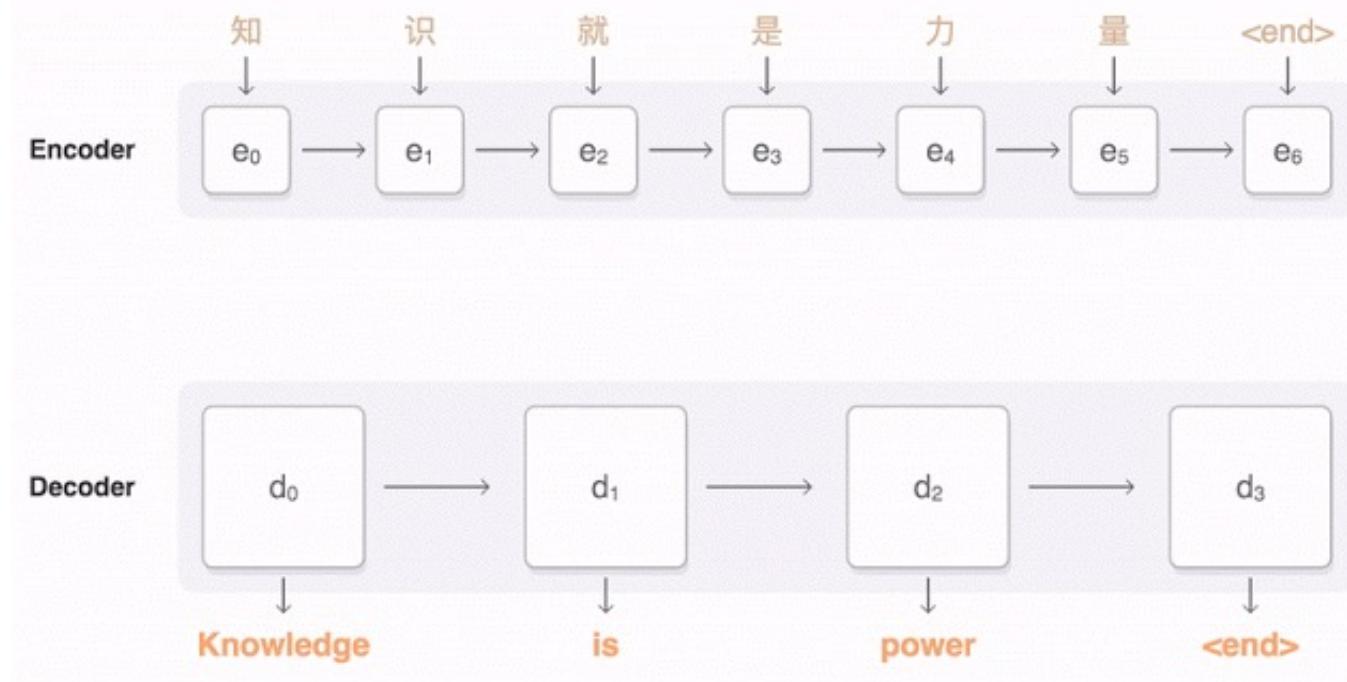


Figure from Google AI research blog.

# Image Translation

Conditional generative model  $P(\text{ zebra images} | \text{ horse images})$



Zhu et al., 2017

# Imitation Learning

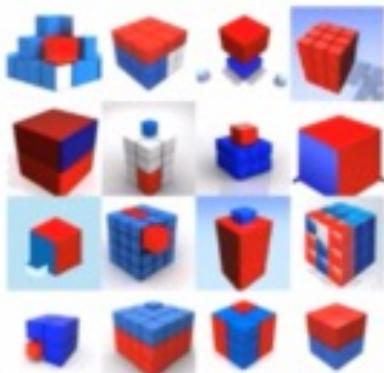
Conditional generative model  $P(\text{actions} \mid \text{past observations})$



Li et al., 2017

# World models in text -> image (DALL-E 2)?

“a red cube on a blue cube”



“the bowl has fewer tomatoes than cucumbers”



“a cup on a spoon”

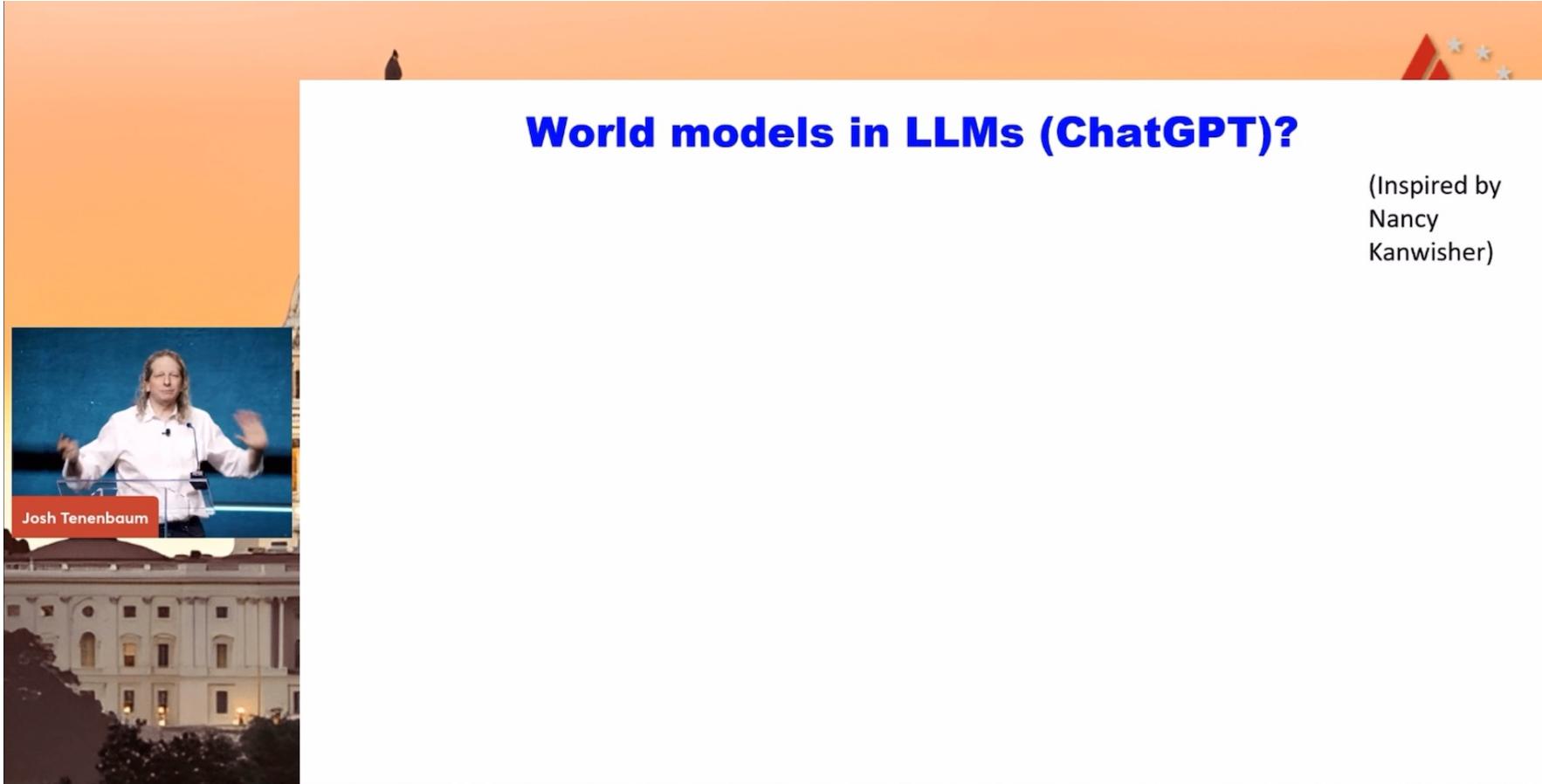


“A pear cut into seven pieces arranged in a ring.”



Gary Marcus; Colin Conwell & Tomer Ullman; Nan Liu, Shuang Li & Yilun Du; many others

# World models in LLMs (ChatGPT)?



# World models in LLMs (ChatGPT)?

JA

A ping pong ball has rolled underneath your couch. How can you retrieve it? You have a ball of string, a pair of scissors, a hockey stick, a lamp, a tennis shoe, and a jar of peanut butter.



Here's a possible solution:

1. Turn on the lamp and shine it under the couch to get a better view of where the ping pong ball is located.
2. Use the hockey stick to try and reach the ball from a distance. If the ball is too far away or the hockey stick is not long enough, move on to the next step.
3. Tie one end of the ball of string securely to the scissors, and toss the scissors under the couch. Keep holding on to the other end of the string.
4. Use the tennis shoe to nudge the ping pong ball towards you until it is close enough to grab with your fingers or the hockey stick.
5. If the ball is still out of reach, use the scissors and the string to try and hook the ball and pull it towards you. To do this, hold the string taut and use the scissors to create a small hook on the end of the string. Carefully lower the string under the couch, maneuver the hook around the ping pong ball, and pull the string back up to retrieve the ball.
6. If all else fails, apply a small amount of peanut butter to the hockey stick and use it to attract the ball towards you. Once the ball is within reach, use the tennis shoe or your fingers to pick it up.

Remember to be careful when using sharp objects like scissors and to avoid damaging the furniture or injuring yourself in the process.

# World models in LLMs (ChatGPT)?

JA

The Eiffel tower has rolled underneath your couch. How can you retrieve it? You have a ball of string, a pair of scissors, a hockey stick, a lamp, a tennis shoe, and a jar of peanut butter.



Retrieving an Eiffel Tower that has rolled underneath the couch can be quite a challenging task, but here are some steps that you can take using the items at hand:

1. Move the couch: First, you need to move the couch to get a better view of the Eiffel Tower. If the couch is too heavy to move, you can try using the hockey stick to push it away from the wall.
2. Use the lamp: Once you have a clear view of the Eiffel Tower, you can use the lamp to shine some light on it. This will help you see it more clearly, and also make it easier to grab.
3. Tie the string: Take the ball of string and tie one end of it to the hockey stick. This will allow you to extend your reach and get closer to the Eiffel Tower.
4. Use the peanut butter jar: Open the jar of peanut butter and use a spoon to scoop out some peanut butter. Then, apply a generous amount of peanut butter to the tennis shoe. This will help you create a sticky surface on the bottom of the shoe that can help you pick up the Eiffel Tower.
5. Retrieve the Eiffel Tower: Use the hockey stick to push the Eiffel Tower towards you, and use the tennis shoe to pick it up. Once you have the Eiffel Tower in your hands, you can cut the string with the scissors and remove it from the hockey stick.

With these steps, you should be able to retrieve the Eiffel Tower from underneath the couch using the items at hand.

# World models in LLMs (ChatGPT)?

JA

How tall is the Eiffel Tower?



The Eiffel Tower is a wrought-iron lattice tower located on the Champ de Mars in Paris, France. It was named after Gustave Eiffel, who designed and built it. The Eiffel Tower stands at a height of 324 meters (1,063 feet) including its antenna at the top. When it was completed in 1889, it was the tallest man-made structure in the world, a title it held until the completion of the Chrysler Building in New York City in 1930. Today, the Eiffel Tower remains an iconic symbol of Paris and a popular tourist attraction.

## Why is this a challenging task?



Retrieving the Eiffel Tower from underneath a couch using the items provided can be a challenging task because:

1. Size and weight: The Eiffel Tower is a large and relatively heavy object, which may make it difficult to move or pick up with just a tennis shoe.
2. Limited space: The space between the couch and the floor may be small, making it hard to reach the Eiffel Tower even with the help of a hockey stick and string.
3. Slippery surface: The smooth surface of the Eiffel Tower may make it hard to grip or pick up with just a tennis shoe, especially if it has rolled to a hard-to-reach spot.
4. Lack of proper tools: The items provided may not be the best tools for the job, as they were not specifically designed for retrieving objects from underneath furniture.

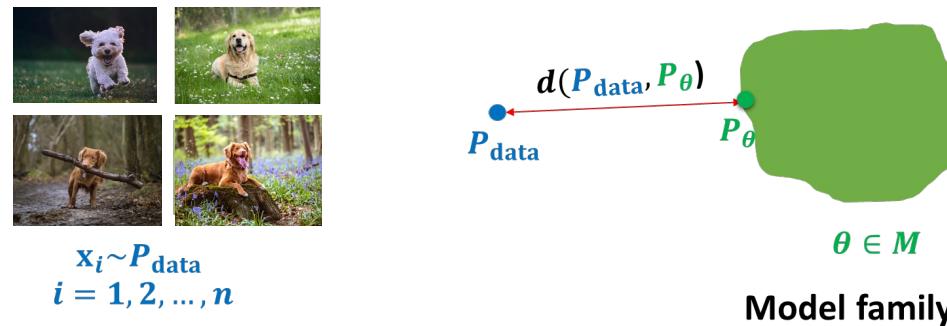
Despite these challenges, with a little creativity and resourcefulness, it is possible to retrieve the Eiffel Tower from underneath the couch using the items provided.

# World models in VGMs (SORA)?



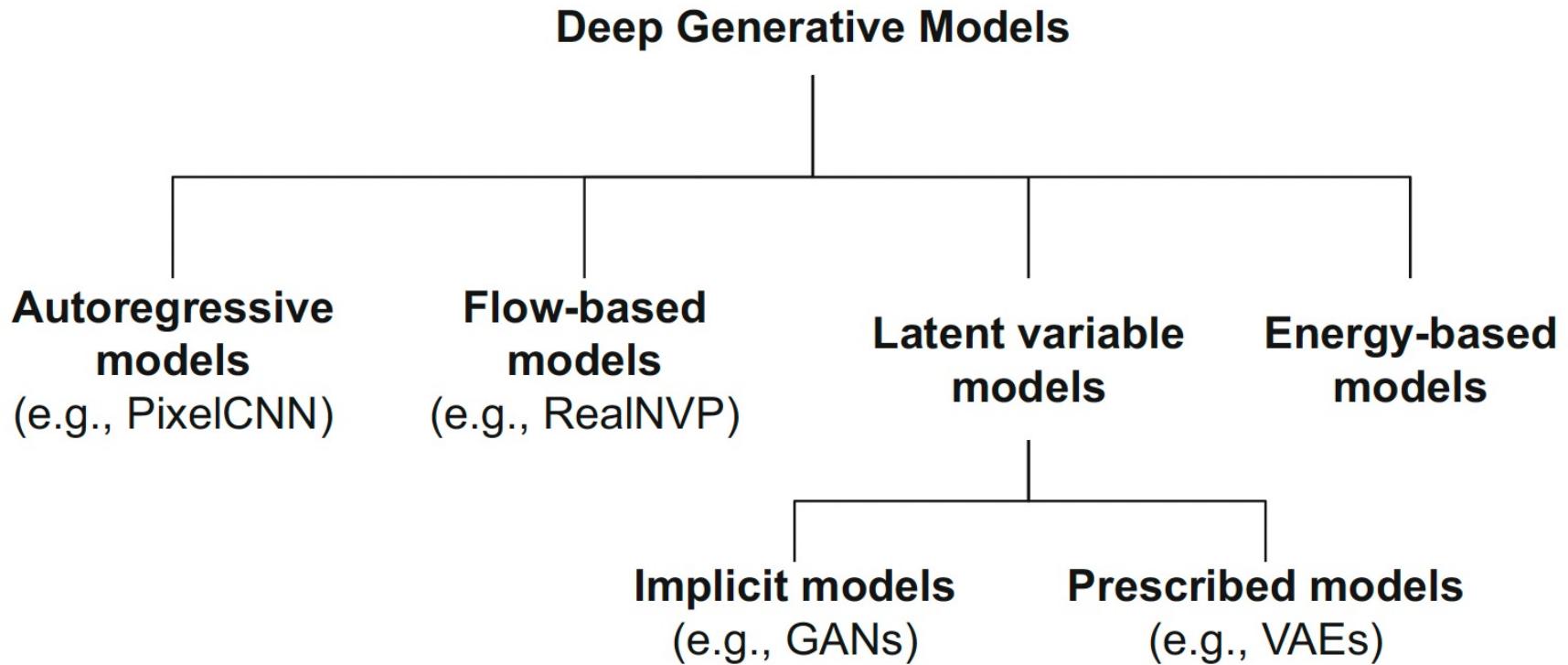
# Roadmap and Key Challenges

- **Representation:** how do we model the joint distribution of many random variables?
  - Need compact representation
- **Learning:** what is the right way to compare probability distributions?



- **Inference:** how do we invert the generation process (e.g., vision as inverse graphics)?
  - Unsupervised learning: recover high-level descriptions (features) from raw data

# How to formulate (Deep) Generative Modeling?



**Fig. 1.4** A taxonomy of deep generative models

# Autoregressive Models

- The distribution over  $x$  is represented in an autoregressive manner:

$$p(\mathbf{x}) = p(x_0) \prod_{i=1}^D p(x_i | \mathbf{x}_{<i})$$

- Modeling all conditional distributions  $p(x_i | \mathbf{x}_{<i})$  would be inefficient. However, we can take advantage of causal convolutions for audio and images.

# Flow-based Models

- The change of variables formula provides a principled manner of expressing a density of a random variable by transforming it with an invertible transformation  $f$ :

$$p(\mathbf{x}) = p(\mathbf{z} = f(\mathbf{x})) |\mathbf{J}_{f(\mathbf{x})}|$$

- All generative models that take advantage of the change of variables formula are referred to as *flow-based* models or *flows* for short.

# Latent Variable Models

- The idea behind latent variable models is to assume a lower-dimensional latent space and the following generative process:

$$\mathbf{z} \sim p(\mathbf{z})$$

$$\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$$

- In other words, the latent variables correspond to hidden factors in data, and the conditional distribution.  $p(x|z)$  could be treated as a generator.

# Energy-based Models

- Physics provide an interesting perspective on defining a group of generative models through defining an energy function,  $E(x)$ , and, eventually, the Boltzmann distribution:

$$\mathbf{x} = \frac{\exp\{-E(\mathbf{x})\}}{Z}$$

- To formulate the energy function and calculate (or rather approximate) the partition function. The largest group of EBMs consists of *Boltzmann Machines* that entangle  $\mathbf{x}$ 's through a bilinear form, i.e.,  $E(\mathbf{x}) = \mathbf{x}^T W \mathbf{x}$ . Introducing latent variables and taking  $E(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T W \mathbf{z}$  results in *Restricted Boltzmann Machines*.

# Comparison

**Table 1.1** A comparison of deep generative models

Generative models	Training	Likelihood	Sampling	Compression	Representation
Autoregressive models	Stable	Exact	Slow	Lossless	No
Flow-based models	Stable	Exact	Fast/slow	Lossless	Yes
Implicit models	Unstable	No	Fast	No	No
Prescribed models	Stable	Approximate	Fast	Lossy	Yes
Energy-based models	Stable	Unnormalized	Slow	Rather not	Yes

# Questions?