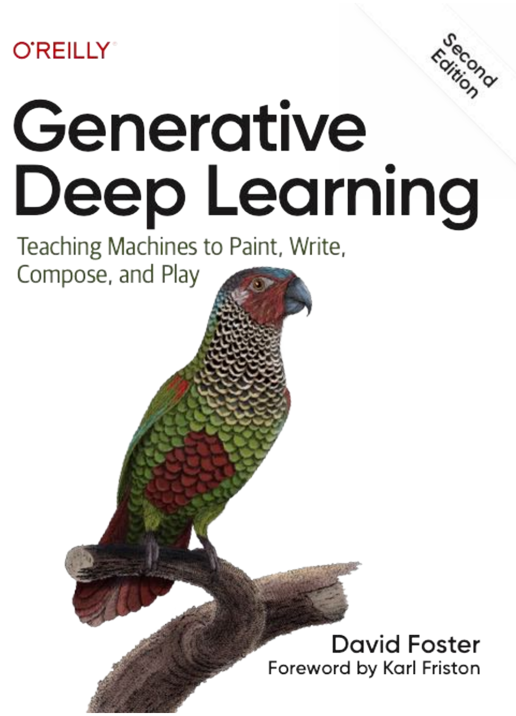
# Deep Generative Models

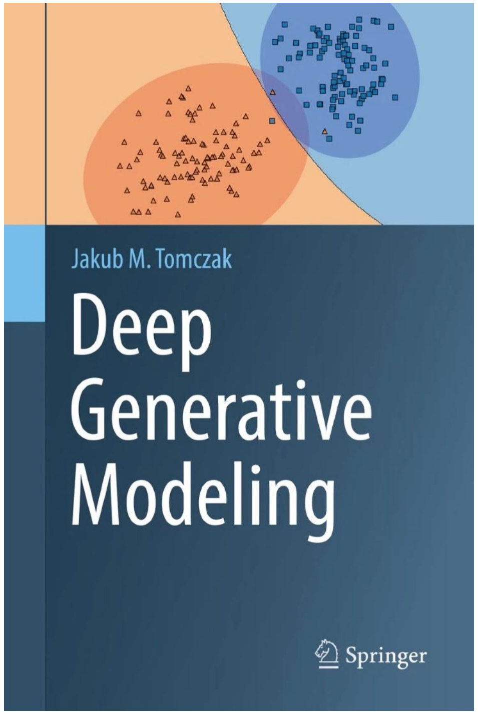
## 1. Introduction and evaluation


- 국가수리과학연구소 산업수학혁신센터 김민중

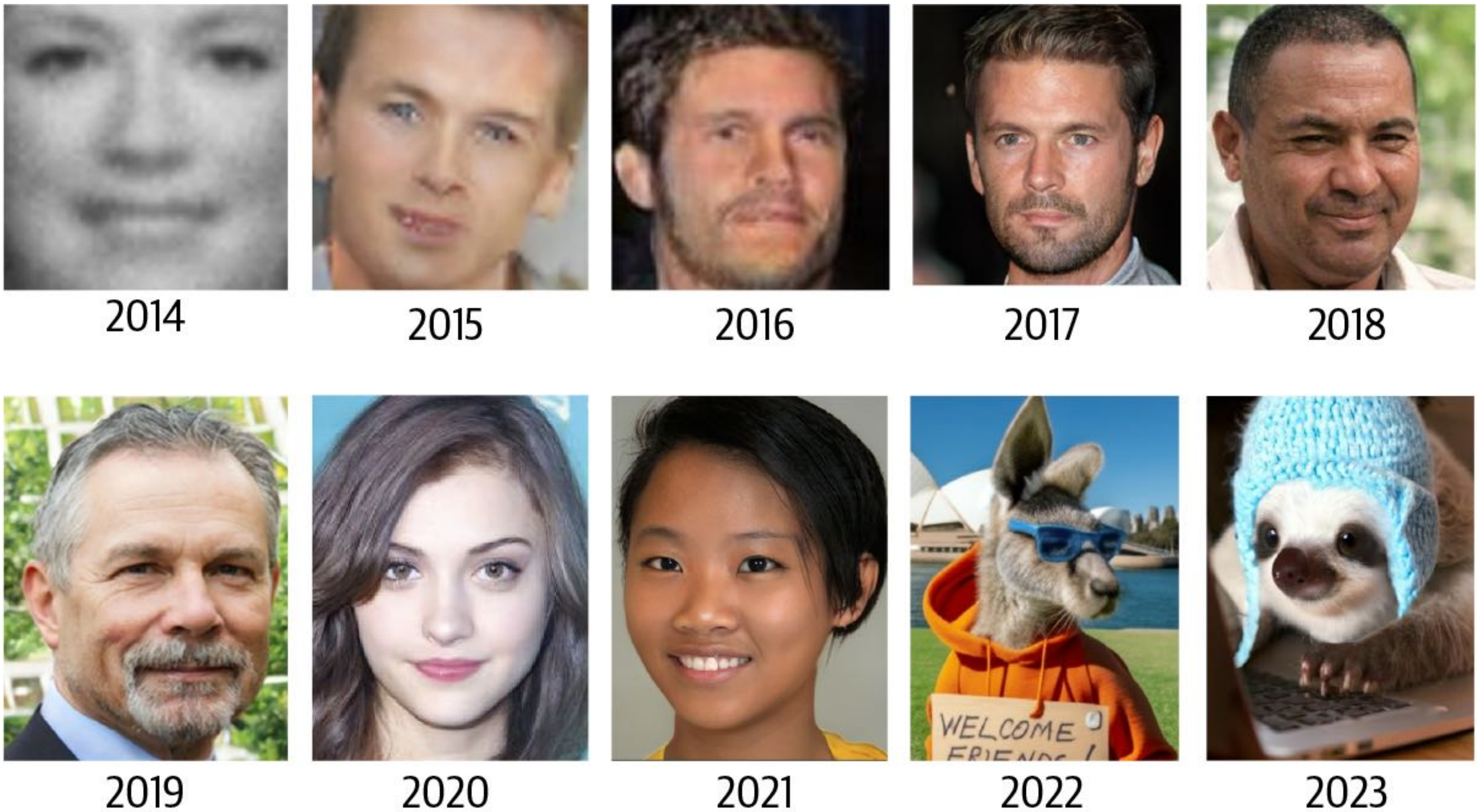
# Reference

- Stanford CS236 lecture: Deep Generative Models
- Generative Deep Learning 2nd(David Foster)
- Deep Generative Modeling(Jakub M. Tomczak)



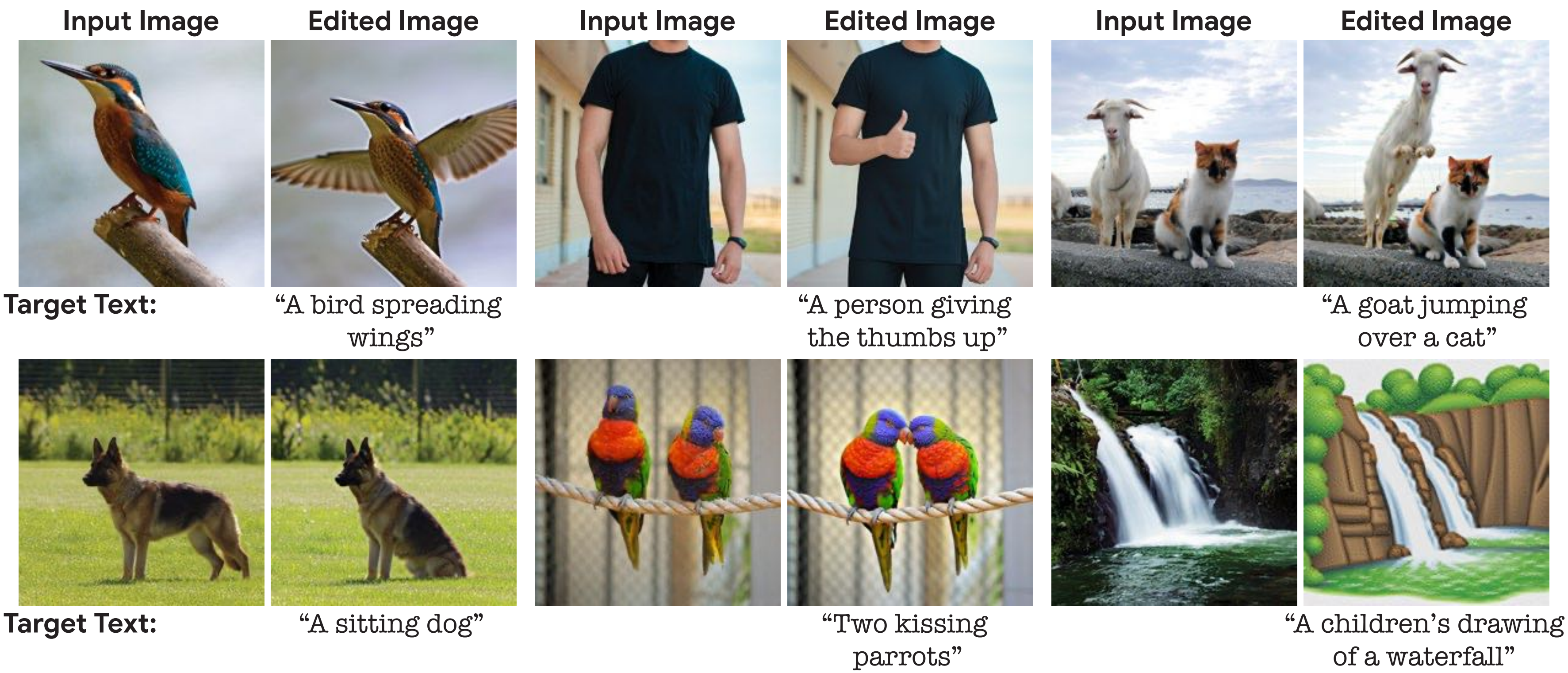



# Face generation




(adapted from Brundage et al., 2018)

# Progress in Inverse Problems




(Kawar et al., 2023)

# Code Generation

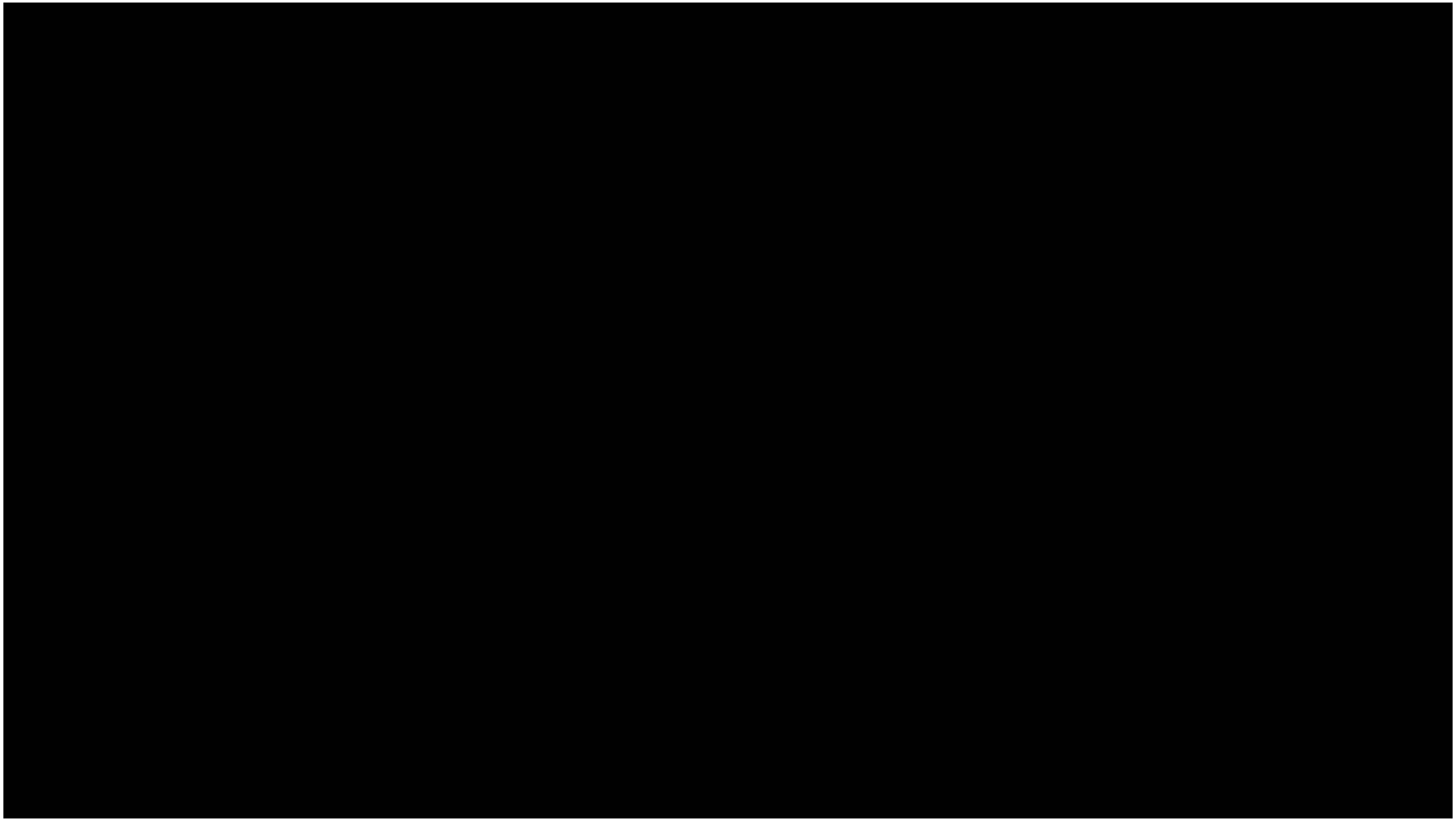
parse_expenses.py | write_sql.go | sentiment.ts | addresses.rb

```python
import datetime

def parse_expenses(expenses_string):
    """Parse the list of expenses and return the list of triples (date, va
```

(Codex, OpenAI)

# Video Generation



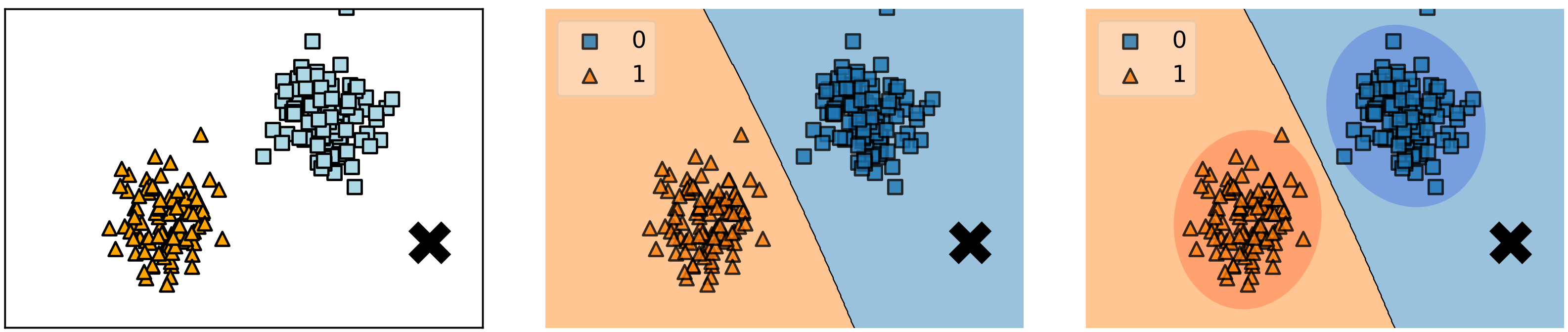
(Sora, OpenAI, 2024)

# What is Generative modeling

- A branch of machine learning that involves training a model to produce new data that is like a given dataset

# Generative vs Discriminative modeling

- $\boldsymbol{x}$: input data(e.g. image sample), $y$: label
- Discriminative modeling estimates $p(y|\boldsymbol{x})$
- Generative modeling estimates $p(\boldsymbol{x})$




Data

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

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

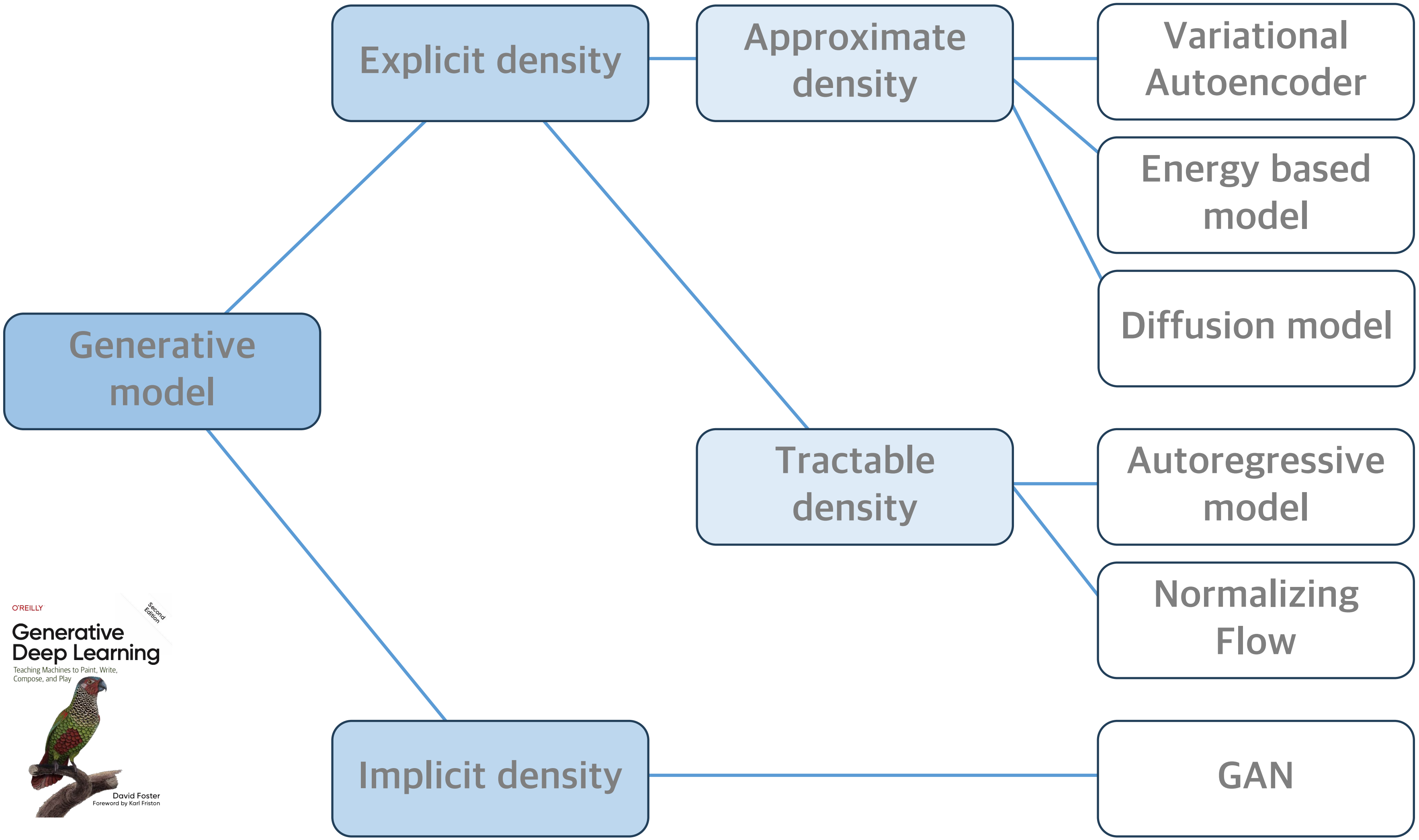
$p(blue|\mathbf{x})$ is high
= certain decision!

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

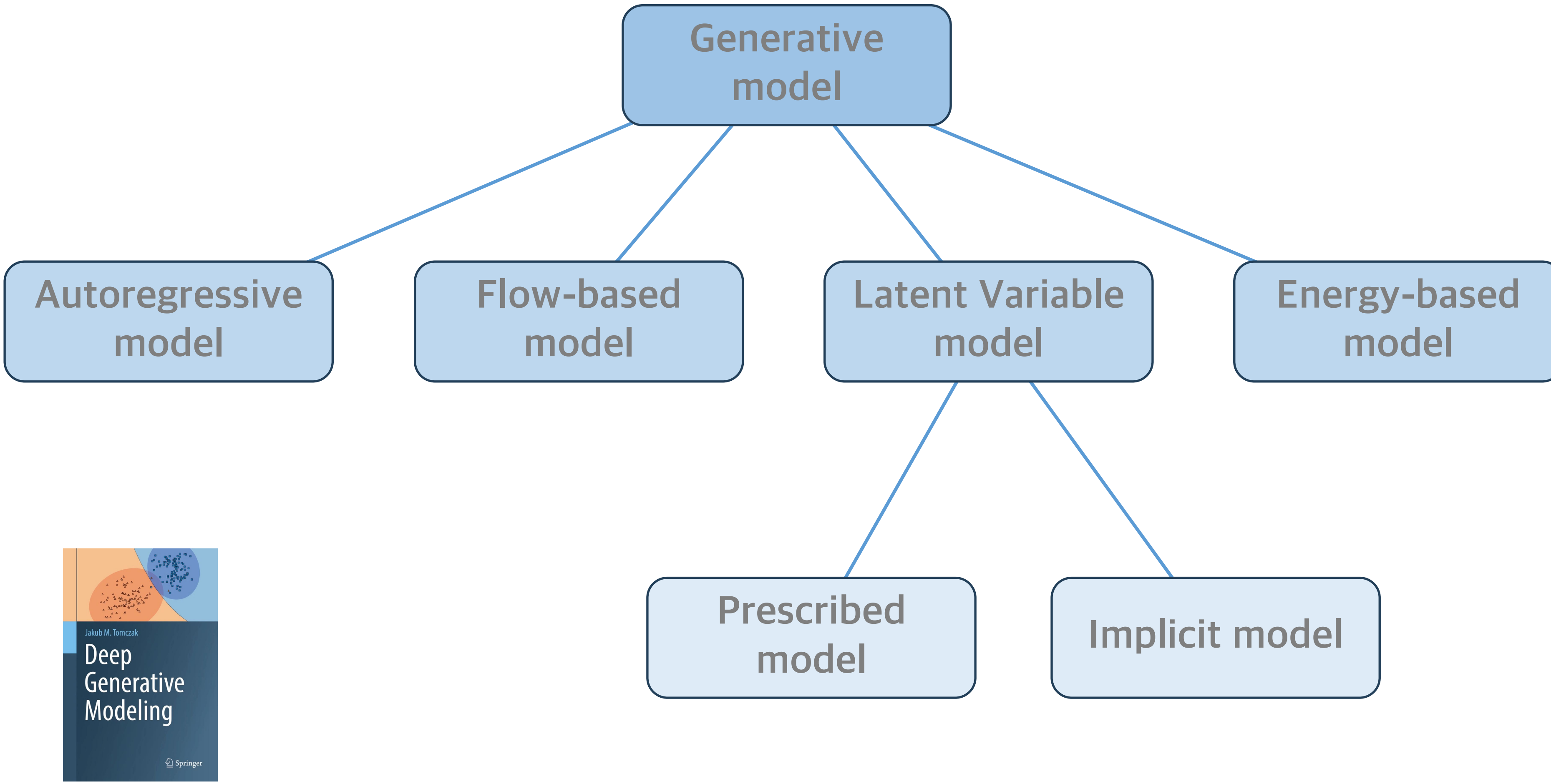
# The purpose of generative model

- **Generation**: sample $x_{new}$ should look like training set(sampling)
- **Density estimation**
- **Unsupervised representation learning**: learn what these images have in common features

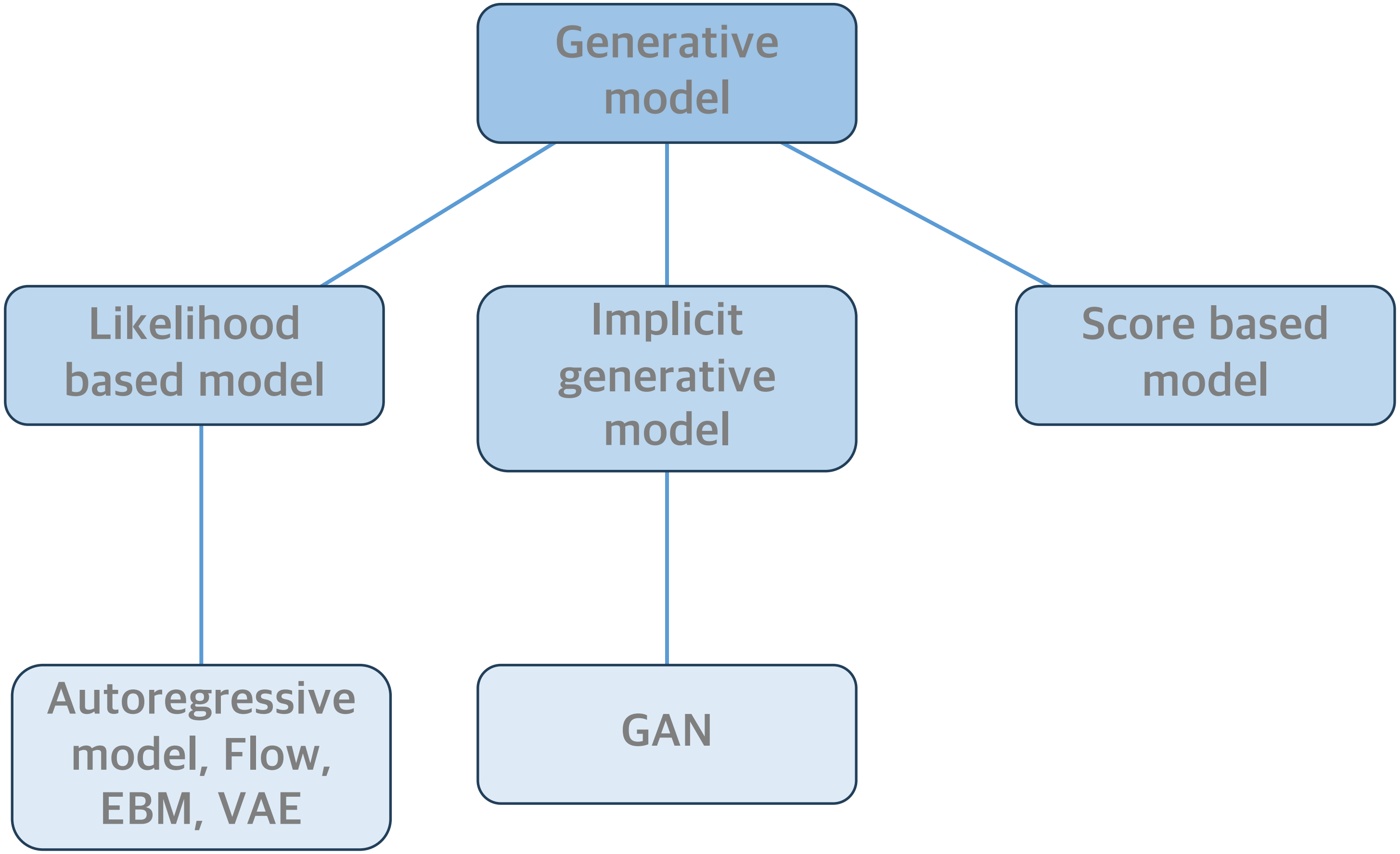
# Taxonomy of Generative model approaches



# Taxonomy of Generative model approaches



# Taxonomy of Generative model approaches




(Yang Song)

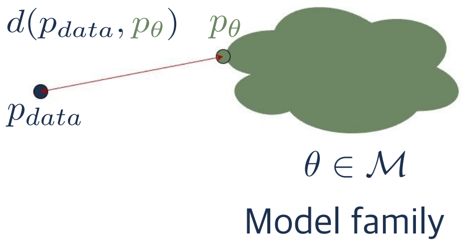
# Goal of Lecture

- We will study Generative models that view the world under the lens of probability
- In such a worldview, we can think of any kind of observed data, say $D$, as a finite set of samples from an underlying distribution, say $p_{data}$
- The goal of any generative model is to approximate this data distribution given access to the dataset $D$
- The hope is that if we can learn a good generative model, we can use the learned model for downstream inference
- Basic Probability Theory, Linear Algebra and techniques of Neural Network(e.g. CNN, RNN, Transformers, U-net etc.) are left as take-home work
- We will follow the Stanford CS236 lecture

# Road map and 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?

$\mathbf{x}_i \sim p_{data}$
$i = 1, 2, ..., N$




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

# Overview

- What is Generative modeling?
- Generative vs Discriminative models
- Evaluating Generative models
  - Density estimation
  - Sampling/generation
    - Inception scores
    - Fréchet Inception Distance
    - Kernel Inception Distance

# Evaluation

- How do we evaluate generative models?
- Evaluation of discriminative models (e.g., a classifier) is well understood compare task-specific loss(e.g., top-1 accuracy or AUROC) on unseen test data
- Evaluating generative models is highly non-trivial
- **Key question:** What is the task that you care about?
  - Density estimation
  - Sampling/generation

# Evaluation - Density Estimation

- Likelihood as a metric for density estimation
  - Split dataset into train, validation and test sets
  - Learn model $p_\theta(\boldsymbol{x})$ using the train set
  - Tune hyperparameters on validation set
  - Evaluate generalization with likelihoods on test sets

$$E_{\boldsymbol{x} \sim p_{data}}[\log p_\theta(\boldsymbol{x})]$$

- Remark: Not all models have tractable likelihoods e.g., VAE, GAN and EBM
  - For VAE, we can compare evidence lower bounds (ELBO) to log-likelihoods. How about GAN?
- Approximation methods are necessary. We can use kernel density estimates via samples alone.

# Kernel Density Estimation

- Given: A trained model $p_\theta(x)$ with an intractable/ill-defined density
- Let $S = \{x^{(1)}, x^{(2)}, \cdots, x^{(6)}\}$ be 6 data points drawn from $p_\theta(x)$

| $x^{(1)}$ | $x^{(2)}$ | $x^{(3)}$ | $x^{(4)}$ | $x^{(5)}$ | $x^{(6)}$ |
|---|---|---|---|---|---|
| $-2.1$ | $-1.3$ | $-0.4$ | $1.9$ | $5.1$ | $6.2$ |

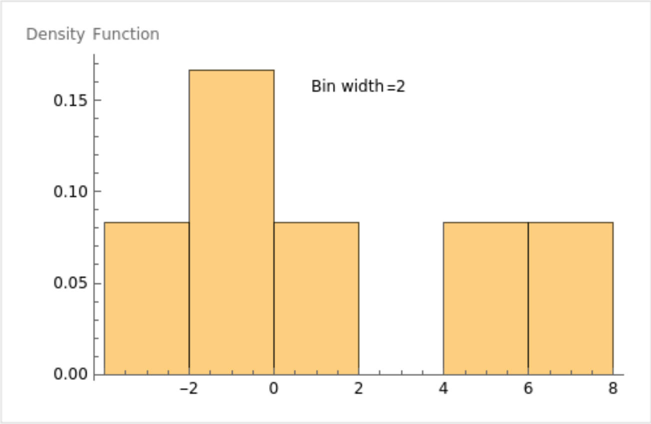
- What is $p_\theta(-0.5)$? for $-0.5 \in$ test set

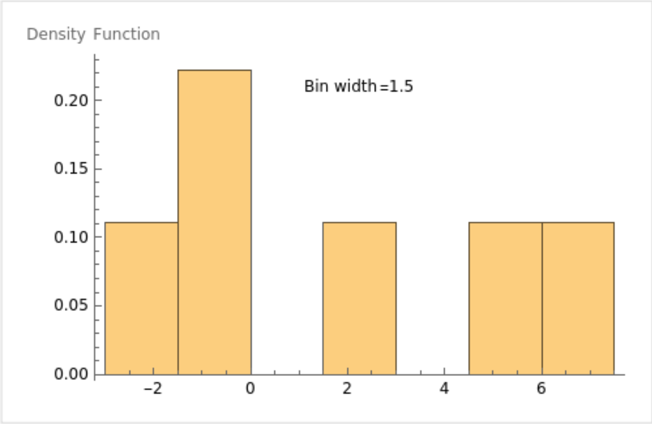
# Kernel Density Estimation

- Let $S = \{x^{(1)}, x^{(2)}, \cdots, x^{(6)}\}$ be 6 data points drawn from $p_\theta(\boldsymbol{x})$

| $x^{(1)}$ | $x^{(2)}$ | $x^{(3)}$ | $x^{(4)}$ | $x^{(5)}$ | $x^{(6)}$ |
|---|---|---|---|---|---|
| $-2.1$ | $-1.3$ | $-0.4$ | $1.9$ | $5.1$ | $6.2$ |

- What is $p_\theta(-0.5)$?
- Answer 1: Since $0.5 \notin S$, $p_\theta(-0.5) = 0$
- Answer 2: Compute a histogram



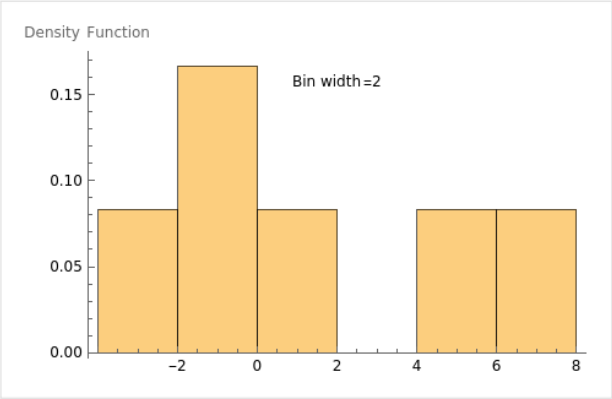



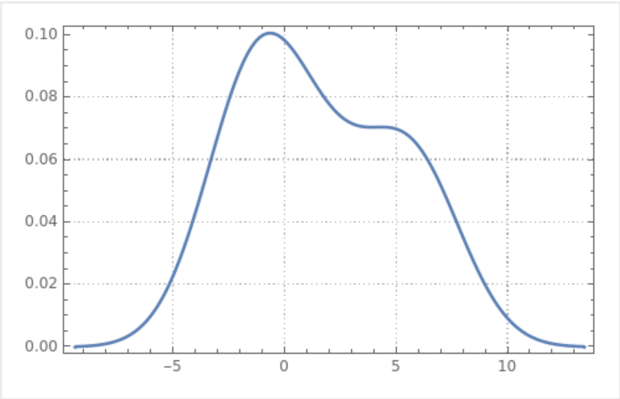
# Kernel Density Estimation

- Answer 3: Compute kernel density estimate (KDE) over $\mathcal{S}$

$$\hat{p}(x) := \frac{1}{N} \sum_{x^{(i)} \in S} K\left(\frac{x - x^{(i)}}{\sigma}\right)$$

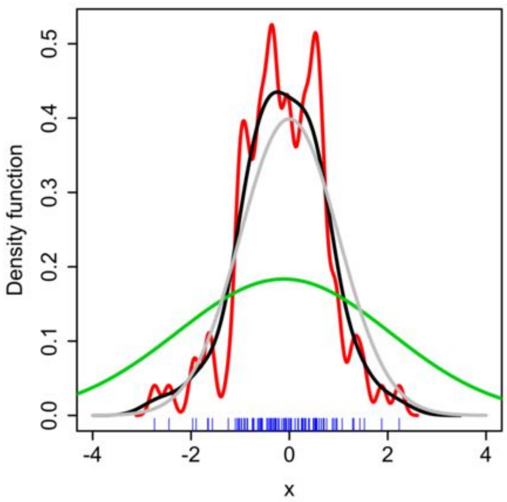
- where $N = |S|$, $\sigma$ is called the bandwidth parameter and $K$ is a kernel function
- Example: Gaussian kernel, $K(u) := \frac{1}{\sqrt{2\pi}} exp\left(-\frac{1}{2}u^2\right)$
- Histogram density estimate vs KDE estimate with Gaussian kernel



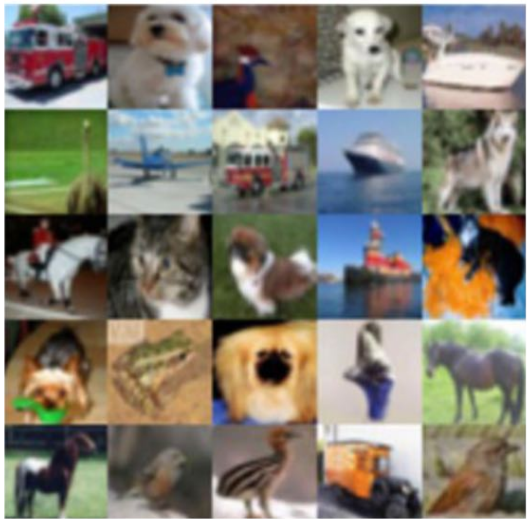



# Kernel Density Estimation

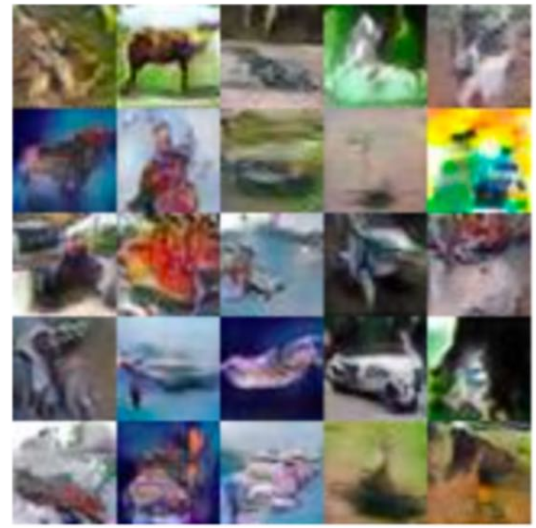
- A kernel $K(\cdot)$ is any non-negative function satisfying two properties
  - Normalization: $\int_{-\infty}^{\infty} K(u)\, du = 1$ (ensures KDE is normalized)
  - Symmetric: $K(u) = K(-u)$ for all $u$
- Intuitively, a kernel is a measure of similarity between pairs of points
- Bandwidth parameter $\sigma$ controls the smoothness
  - Optimal sigma (black) is such that KDE is closed to true density (grey)
  - Low sigma (red): under smoothed
  - High sigma (green): over smoothed
  - Tuned via cross validation
- Con: KDE is very unreliable in high dimension



# Evaluation - Sample quality



vs



- Which of these two sets of generated samples look better?
- Human evaluation (e.g., Mechanical Turk) is the gold standard

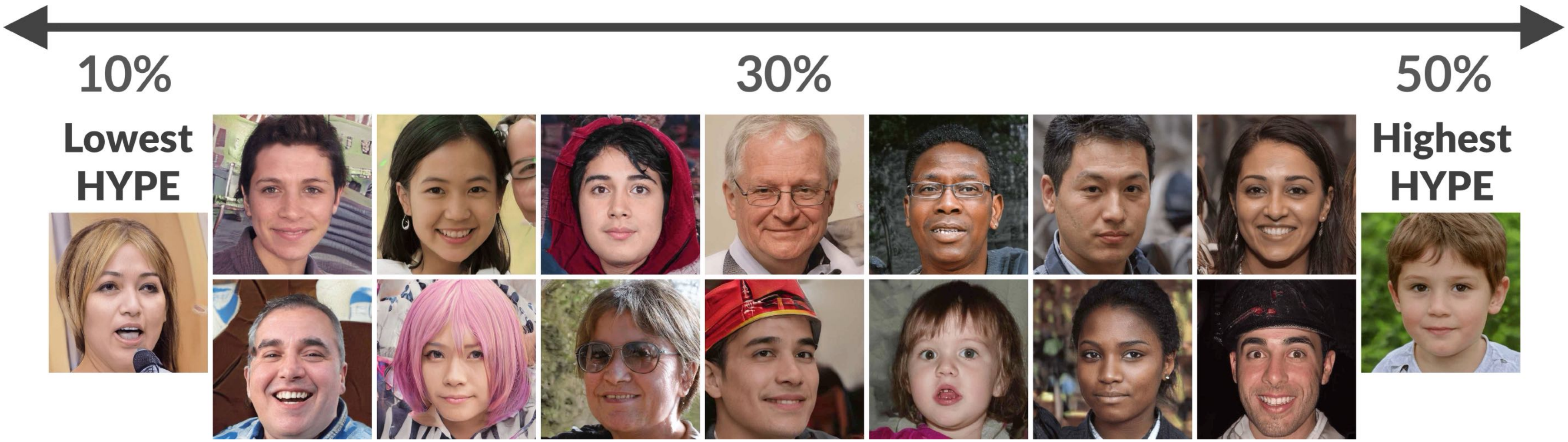
# Evaluation - HYPE

**HYPE: A Benchmark for Human eYe Perceptual Evaluation of Generative Models**

**Sharon Zhou*, Mitchell L. Gordon*, Ranjay Krishna,**
**Austin Narcomey, Li Fei-Fei, Michael S. Bernstein**
Stanford University
{sharonz, mgord, ranjaykrishna, aon2, feifeili, msb}@cs.stanford.edu

- HYPE: Human eYe Perceptual Evaluation (Zhou et al., 2019)
  - $\text{HYPE}_{\text{time}}$: the minimum time human needed to decide a classification. The larger, the better
  - $\text{HYPE}_{\infty}$: The percentage of samples the deceive human under unlimited time. The larger, the better
  - https://stanfordhci.github.io/gen-eval

# Evaluation - HYPE




- Generalization is hard to define and assess. Memorizing the training set would give excellent samples but clearly undesirable
- Quantitative evaluation of a qualitative task can have many answers
- Popular metrics: Inception Scores, Fréchet Inception Distance Scores, Kernel Inception Distance

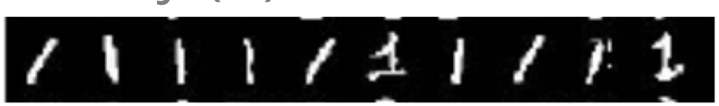
# Inception Scores

- **Assumption 1**: We are evaluating sample quality for generative models trained on labelled datasets
- **Assumption 2**: We have a good probabilistic classifier $c(y|\boldsymbol{x})$ for predicting the label $y$ for any point(image) $\boldsymbol{x}$
- We want samples from a good generative model to satisfy two criteria: sharpness and diversity (Salimans et al. 2016)
- Sharpness (S)

Low sharpness

High sharpness

- Diversity (D)



Low diversity

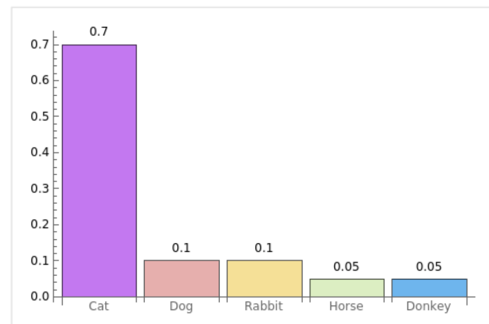
High diversity

# Inception Scores

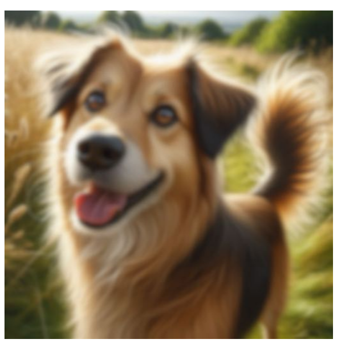
- Sharpness (S)

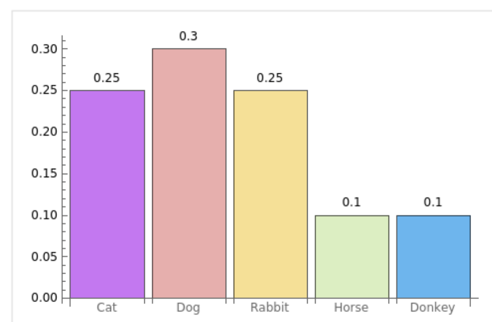
$x \sim p_\theta$

$c(y|x)$




Highly confident






Lowly confident

# Inception Scores

- Sharpness (S)

Low sharpness

High sharpness

- Given: generated data $\boldsymbol{x}$, well trained probabilistic classifier $c(y|\boldsymbol{x})$
- High sharpness implies classifier is confident in making predictions for generated images
- I.e., classifier's predictive districution $c(y|\boldsymbol{x})$ has low entropy
- The label $y \sim$Categorical distribution

$$S := exp\left(E_{\boldsymbol{x}\sim p_\theta}\left[\int c(y|\boldsymbol{x}) \log c(y|\boldsymbol{x})\, dy\right]\right)$$

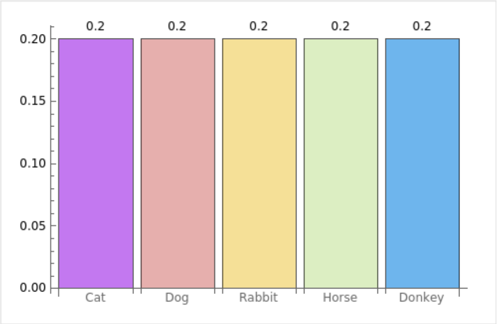
- where $p_\theta$ is generative model distribution

# Inception Scores

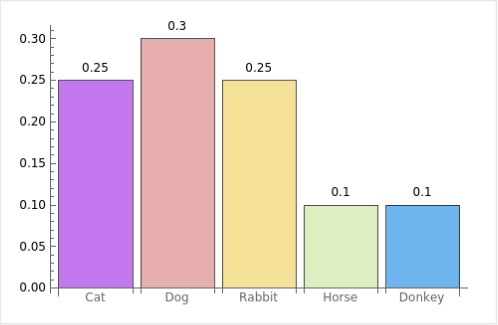
- Diversity (D)

$x \sim p_\theta$

$E_{x \sim p_\theta}[c(y|x)]$




High diversity




Low diversity

# Inception Scores

- Diversity (D)

Low diversity

High diversity

- High diversity implies $c(y)$ has high entropy

$$D := \exp\left(-\int c(y)\log c(y)\,dy\right)$$

- where $c(y) := E_{\boldsymbol{x}\sim p_\theta}[c(y|\boldsymbol{x})]$ is the classifier's marginal predictive distribution

# Inception Scores

- Inception scores (IS) combine the two criteria of sharpness and diversity into a simple metric

$$S \cdot D = \exp\left(-E_{\boldsymbol{x}\sim p_\theta}\left[\int c(y|\boldsymbol{x})(\log c(y) - \log c(y|\boldsymbol{x}))dy\right]\right)$$

- Notice that IS can be written as

$$\exp\left(E_{\boldsymbol{x}\sim p_\theta}\left[KL\left(c(y|\boldsymbol{x}) \parallel c(y)\right)\right]\right)$$

- Higher IS corresponds to better generation quality
- If classifiers are not available, we can not obtain Inception scores
- IS only requres samples from $p_\theta$ and do not consider the desired data distribution $p_{data}$

# Fréchet Inception Distance

- Fréchet Inception Distance (FID) measures similarities in the feature representations(e.g. those learned by a pretrained classifier) for datapoints sampled from $p_\theta$ and the test dataset
- Computing FID
  - Let $G$ denote the generated samples and $T$ denote the test dataset
  - Compute feature representation $F_G$ and $F_T$ for $G$ and $T$ respectively (e.g., prefinal layer of Inception Net)
  - Fit a multivariate Gaussian to each of $F_G$ and $F_T$.
  - Let $(\mu_G, \Sigma_G)$ and $(\mu_F, \Sigma_F)$ denote the mean and covariances of the two Gaussians
  - FID is defined as the 2nd Wasserstein distance between these two Gaussians(Heusel et al. 2017)

# Fréchet Inception Distance

- FID is defined as the 2nd Wasserstein distance between these two Gaussians:

$$FID = ||\mu_T - \mu_G||_2^2 + Tr\left(\Sigma_T + \Sigma_G - 2(\Sigma_T \Sigma_G)^{1/2}\right)$$

- Lower FID implies better sample quality
- Feature representations are assumed to follow Multivariate Gaussian

# Kernel Inception Distance

- Maximum Mean Discrepancy (MMD) is a two-sample test statistic that compares samples from two distributions $p$ and $q$ by computing differences in their moments (mean, variances etc.)
- **Key idea**: Use a suitable kernel e.g., Gaussian kernel to measure similarity between points

$$MMD(p,q) = E_{\boldsymbol{x},\boldsymbol{x}'\sim p}[K(x,x')] + E_{\boldsymbol{x},\boldsymbol{x}'\sim q}[K(x,x')] \\ -2E_{\boldsymbol{x}\sim p,\boldsymbol{x}'\sim q}[K(x,x')]$$

- Intuitively, MMD is comparing the "**similarity**" between samples within $p$ and $q$ individually to the samples from the mixture of $p$ and $q$
- Kernel Inception Distance (KID): compute the MMD in the feature space of a classifier (e.g., Inception Network) (Bińkowski et al., 2018)

# Summary

- How do we evaluate generative models?
- For unsupervised evaluation, metrics can significantly vary based on end goal: Density estimation, sampling, latent representations
  - Kernel density estimation
  - Inception scores
  - Fréchet inception distance
  - Kernel inception distance

# Thanks