10-301/10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Generative Models for Vision

+

Significance Testing for ML

Matt Gormley &Henry Chai
Lecture 26
Dec. 4, 2024

Reminders

- Homework 9: Learning Paradigms
  - Out: Mon, Nov. 25
  - Due: Thu, Dec. 5 at 11:59pm
    (only two grace/late days permitted)
- Exam 3 Practice Problems
  - Out: Mon, Dec. 2
- Exam 3
  - Tue, Dec 10 (9:30am – 11:30am)
- Final Exit Poll (after Exam 3)

EXAM LOGISTICS

Exam 3

- **Time / Location**
  - **Time:** **Tue, Dec 10 (9:30am – 11:30am)**
  - **Location & Seats:** You have all been split across multiple rooms. Everyone has an assigned seat in one of these room.
  - Please watch Piazza carefully for announcements.
- **Logistics**
  - Covered material: Lectures 17 – 25
  - Format of questions:
    - Multiple choice
    - True / False (with justification)
    - Derivations
    - Short answers
    - Interpreting figures
    - Implementing algorithms on paper
  - No electronic devices
  - You are allowed to **bring** one 8½ x 11 sheet of notes (front and back)

Exam 3

- **How to Prepare**
  - Attend (or watch) this exam review session
  - Review **practice problems**
  - Review **homework problems**
  - Review the **poll questions** from each lecture
  - Consider whether you have achieved the **learning objectives** for each lecture / section
  - Write your cheat sheets

Topics for Exam 1

- Foundations
  - Probability, Linear Algebra, Geometry, Calculus
  - Optimization
- Important Concepts
  - Overfitting
  - Experimental Design
- Classification
  - Decision Tree
  - KNN
  - Perceptron
- Regression
  - KNN Regression
  - Decision Tree Regression
  - Linear Regression

Topics for Exam 2

- Classification
  - Binary Logistic Regression
- Important Concepts
  - Stochastic Gradient Descent
  - Regularization
  - Feature Engineering
- Feature Learning
  - Neural Networks
  - Basic NN Architectures
  - Backpropagation
- Learning Theory
  - PAC Learning
  - MLE / MAP
- Societal Impacts of ML
- Regression
  - Linear Regression

Topics for Exam 3

- Deep Learning
  - Convolutional Neural Networks (CNNs)
  - Recurrent Neural Networks (RNNs)
  - Transformers
  - Automatic differentiation
- Reinforcement Learning
  - Value Iteration
  - Policy Iteration
  - Q-Learning
  - Deep Q-Learning
- Other Learning Paradigms
  - K-Means
  - PCA
  - Ensemble Methods
  - Recommender Systems

Classification and Regression: The Big Picture

Recipe for Machine Learning

1. Given data $\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$
2. (a) Choose a decision function $h_{\boldsymbol{\theta}}(\mathbf{x}) = \cdots$ (parameterized by $\boldsymbol{\theta}$)
   (b) Choose an objective function $J_{\mathcal{D}}(\boldsymbol{\theta}) = \cdots$ (relies on data)
3. Learn by choosing parameters that optimize the objective $J_{\mathcal{D}}(\boldsymbol{\theta})$

$$\hat{\boldsymbol{\theta}} \approx \operatorname*{argmin}_{\boldsymbol{\theta}} J_{\mathcal{D}}(\boldsymbol{\theta})$$

4. Predict on new test example $\mathbf{x}_{\text{new}}$ using $h_{\boldsymbol{\theta}}(\cdot)$

$$\hat{y} = h_{\boldsymbol{\theta}}(\mathbf{x}_{\text{new}})$$

Optimization Method

- Gradient Descent: $\boldsymbol{\theta} \to \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
- SGD: $\boldsymbol{\theta} \to \boldsymbol{\theta} - \gamma \nabla_{\boldsymbol{\theta}} J^{(i)}(\boldsymbol{\theta})$
  for $i \sim$ Uniform$(1, \ldots, N)$
  where $J(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N J^{(i)}(\boldsymbol{\theta})$
- mini-batch SGD
- closed form
  1. compute partial derivatives
  2. set equal to zero and solve

Decision Functions

- Perceptron: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T \mathbf{x})$
- Linear Regression: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x}$
- Discriminative Models: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \operatorname*{argmax}_y p_{\boldsymbol{\theta}}(y \mid \mathbf{x})$
  - Logistic Regression: $p_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \sigma(\boldsymbol{\theta}^T \mathbf{x})$
  - Neural Net (classification):
    $p_{\boldsymbol{\theta}}(y = 1 \mid \mathbf{x}) = \sigma((\mathbf{W}^{(2)})^T \sigma((\mathbf{W}^{(1)})^T \mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)})$
- Generative Models: $h_{\boldsymbol{\theta}}(\mathbf{x}) = \operatorname*{argmax}_y p_{\boldsymbol{\theta}}(\mathbf{x}, y)$
  - Naive Bayes: $p_{\boldsymbol{\theta}}(\mathbf{x}, y) = p_{\boldsymbol{\theta}}(y) \prod_{m=1}^M p_{\boldsymbol{\theta}}(x_m \mid y)$

Objective Function

- MLE: $J(\boldsymbol{\theta}) = -\sum_{i=1}^N \log p(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$
- MCLE: $J(\boldsymbol{\theta}) = -\sum_{i=1}^N \log p(\mathbf{y}^{(i)} \mid \mathbf{x}^{(i)})$
- L2 Regularized: $J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda ||\boldsymbol{\theta}||_2^2$
  (same as Gaussian prior $p(\boldsymbol{\theta})$ over parameters)
- L1 Regularized: $J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda ||\boldsymbol{\theta}||_1$
  (same as Laplace prior $p(\boldsymbol{\theta})$ over parameters)

Learning Paradigms

Paradigm	**Data**
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^N$ $\quad\quad$ $\mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ $\quad\quad$ $\mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$
Active Learning	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^N$ and can query $y^{(i)} = c^*(\cdot)$ at a cost
Imitation Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots\}$
Reinforcement Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \ldots\}$

ML Big Picture

Learning Paradigms:

What data is available and when? What form of prediction?

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Problem Formulation:

What is the structure of our output prediction?

boolean	Binary Classification
categorical	Multiclass Classification
ordinal	Ordinal Classification
real	Regression
ordering	Ranking
multiple discrete	Structured Prediction
multiple continuous	(e.g. dynamical systems)
both discrete & cont.	(e.g. mixed graphical models)

Application Areas

Key challenges?

NLP, Speech, Computer Vision, Robotics, Medicine, Search

Theoretical Foundations:

What principles guide learning?

- ❑ probabilistic
- ❑ information theoretic
- ❑ evolutionary search
- ❑ ML as optimization

Facets of Building ML Systems:

How to build systems that are robust, efficient, adaptive, effective?

1. *Data prep*
2. *Model selection*
3. *Training (optimization / search)*
4. *Hyperparameter tuning on validation data*
5. *(Blind) Assessment on test data*

Big Ideas in ML:

Which are the ideas driving development of the field?

- *inductive bias*
- *generalization / overfitting*
- *bias-variance decomposition*
- *generative vs. discriminative*
- *deep nets, graphical models*
- *PAC learning*
- *distant rewards*

Course Level Objectives

You should be able to…

1. Implement and analyze existing learning algorithms, including well-studied methods for classification, regression, structured prediction, clustering, and representation learning
2. Integrate multiple facets of practical machine learning in a single system: data preprocessing, learning, regularization and model selection
3. Describe the the formal properties of models and algorithms for learning and explain the practical implications of those results
4. Compare and contrast different paradigms for learning (supervised, unsupervised, etc.)
5. Design experiments to evaluate and compare different machine learning techniques on real-world problems
6. Employ probability, statistics, calculus, linear algebra, and optimization in order to develop new predictive models or learning methods
7. Given a description of a ML technique, analyze it to identify (1) the expressive power of the formalism; (2) the inductive bias implicit in the algorithm; (3) the size and complexity of the search space; (4) the computational properties of the algorithm: (5) any guarantees (or lack thereof) regarding termination, convergence, correctness, accuracy or generalization power.

Course Staff

Instructors


Henry Chai


Matt Gormley

Education Associate


Daniel Bird

Course Admin


Kimber Lovell

HW2 / HW6


Hailey Xia


Kushagra Agarwal

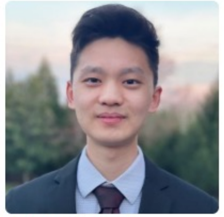

Mihir Mathur


Rohini Banerjee


Joaquin Wang

HW3 / HW7


Bhargav Hadya

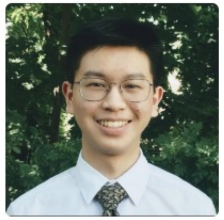

Sebastian Lu

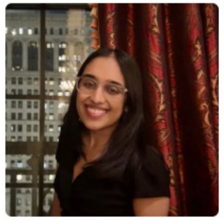

Varsha Reddy Redla


Zachary Gelman

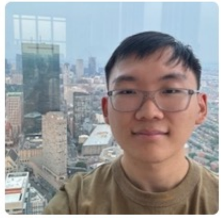

Maxwell Chien

HW4 / HW8


Max Tang


Shivi Jindal


Andra Liu


Albert Zhang


Vianna Seifi

HW5 / HW9


Zoe Xu


Rohan Chawla


Doris Gao

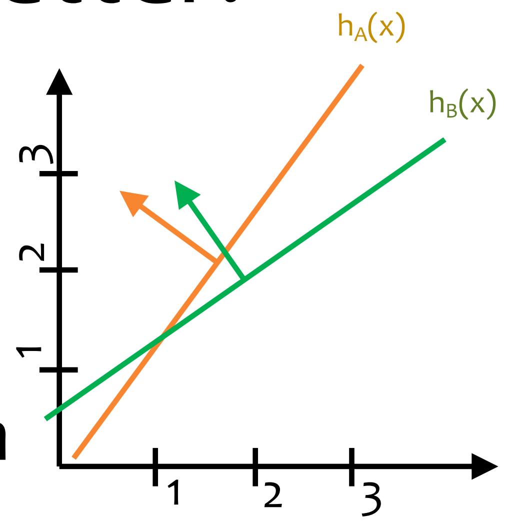

Jenny Yang


Siyan Chen

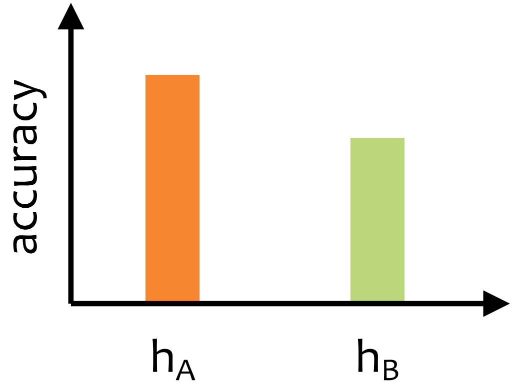
SIGNIFICANCE TESTING

Which classifier is better?

Goal: Given two classifiers: $h_A(x)$ and $h_B(x)$ which is better?



Common Approach: Evaluate each classifier on a test set and report which has higher accuracy.



Two Sources of Variance

1. Randomness in training
2. Randomness in our test data

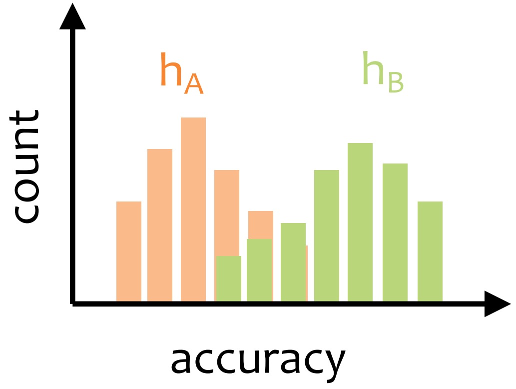
1. Randomness in training

Example: Assume we are training a **deep neural network** with a **nonconvex objective** function via **random restarts**

We collect a sequence of classifiers for R random restarts:

- $h_B(x)^{(1)} \longleftarrow \text{train}(D, \text{seed} = \text{time in ms})$
- $h_B(x)^{(2)} \longleftarrow \text{train}(D, \text{seed} = \text{time in ms})$
- …
- $h_B(x)^{(R)} \longleftarrow \text{train}(D, \text{seed} = \text{time in ms})$

Solution: histogram



Solution: confidence interval

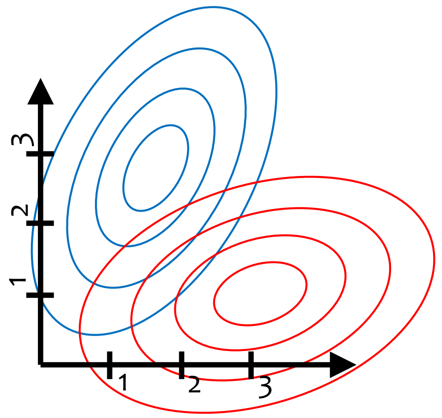
report variance of h_A and h_B

Ex:

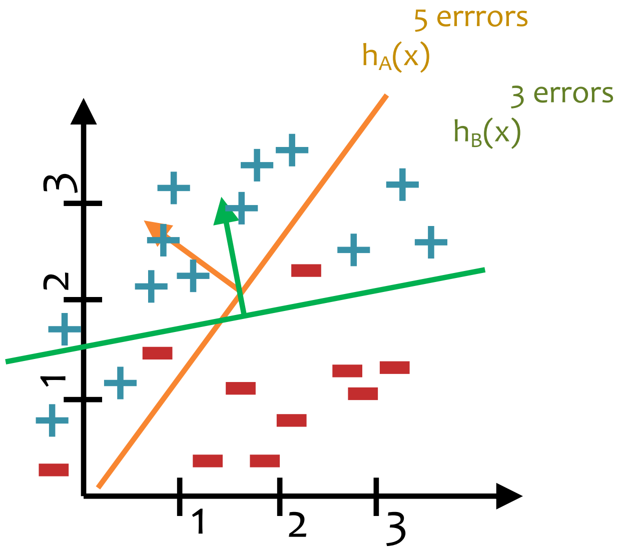
- h_A 45% +/- 5%
- h_B 47% +/- 8%

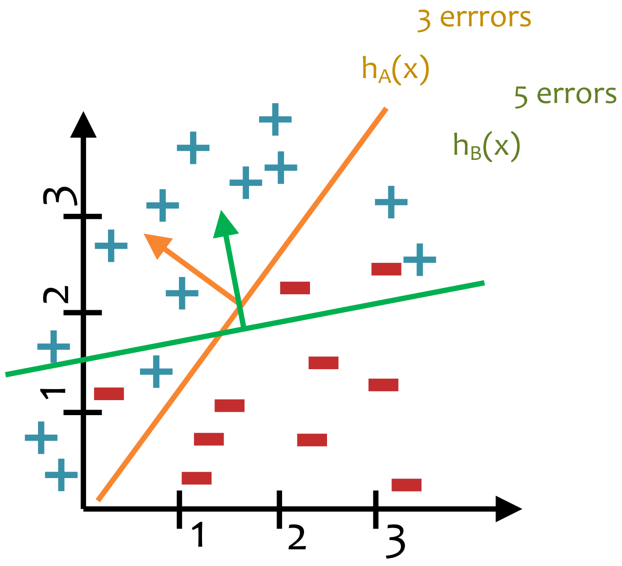
2. Randomness in our test data

Recall: we assume $x^{(i)} \sim p^*(\cdot)$ and $y^{(i)} = c^*(\mathbf{x}^{(i)})$
or $(x^{(i)}, y^{(i)}) \sim p^*(\cdot, \cdot)$



Data: Assume the data is drawn from a generative distribution $p^*(x|y)p^*(y)$ where $p^*(y)$ is an even coin flip and $p^*(x|y\text{=red})$ is the red Gaussian and $p^*(x|y\text{=blue})$ is the blue Gaussian.



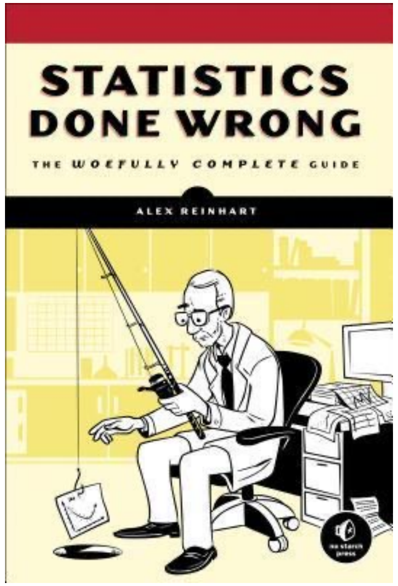


Solution:
significance testing

Significance Testing in ML

"And because any medication or intervention usually has some real effect, you can always get a statistically significant result by collecting so much data that you detect extremely tiny but relatively unimportant differences. As Bruce Thompson wrote, Statistical significance testing can involve a tautological logic in which tired researchers, having collected data on hundreds of subjects, then conduct a statistical test to evaluate whether there were a lot of subjects, which the researchers already know, because they collected the data and know they are tired. This tautology has created considerable damage as regards the cumulation of knowledge."

— Alex Reinhart
Statistics Done Wrong: The Woefully Complete Guide



For machine learning, significance testing is usually still answering an important question:

Did we evaluate our model on enough test data to conclude that our improvement over the baseline is surprising?

Significance Testing in ML

Paired Bootstrap Test

Key Idea: simulate the resampling of many test sets

Algorithm:

1. Draw B bootstrap samples
   $S^{(b)} = \{(\mathbf{x}^{(1)}, y^{(1)}) (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(n)}, y^{(n)})\}$
   with replacement from test data D_{test}
2. Let v = 0
3. For b = 1,…,B
   if $\delta(S^{(b)}) > 2\delta(D_{test})$:
   v = v + 1
4. Return p-value as v/B

$\delta(D')$ = difference in accuracy between h_A and h_B on D'

H_0 = null hypothesis = performance of h_A and h_B is the same

Remarks:

1. Notice that $E[\delta(S^{(b)})] = \delta(D_{test})$.

   We want to estimate how often A obtains a $\delta(D_{test})$-sized advantage over B (or greater) by random chance.

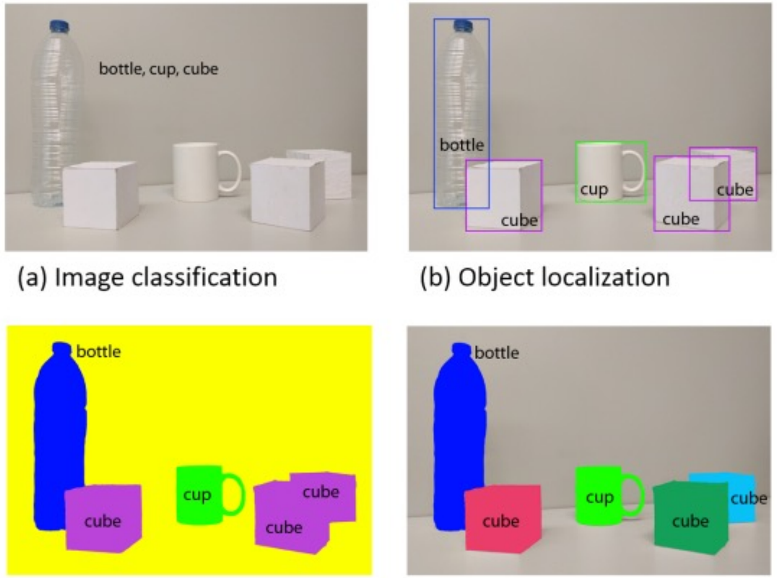
   So we check whether $\delta(S^{(b)})$ exceeds the expected value plus $\delta(D_{test}) = 2\delta(D_{test})$.

2. We needn't limit δ to the difference in *accuracy*, it can be any metric we want!

COMPUTER VISION

Common Tasks in Computer Vision

1. Image Classification
2. Image Classification + Localization
3. Human Pose Estimation
4. Semantic Segmentation
5. Object Detection
6. Instance Segmentation
7. Image Captioning
8. Image Generation

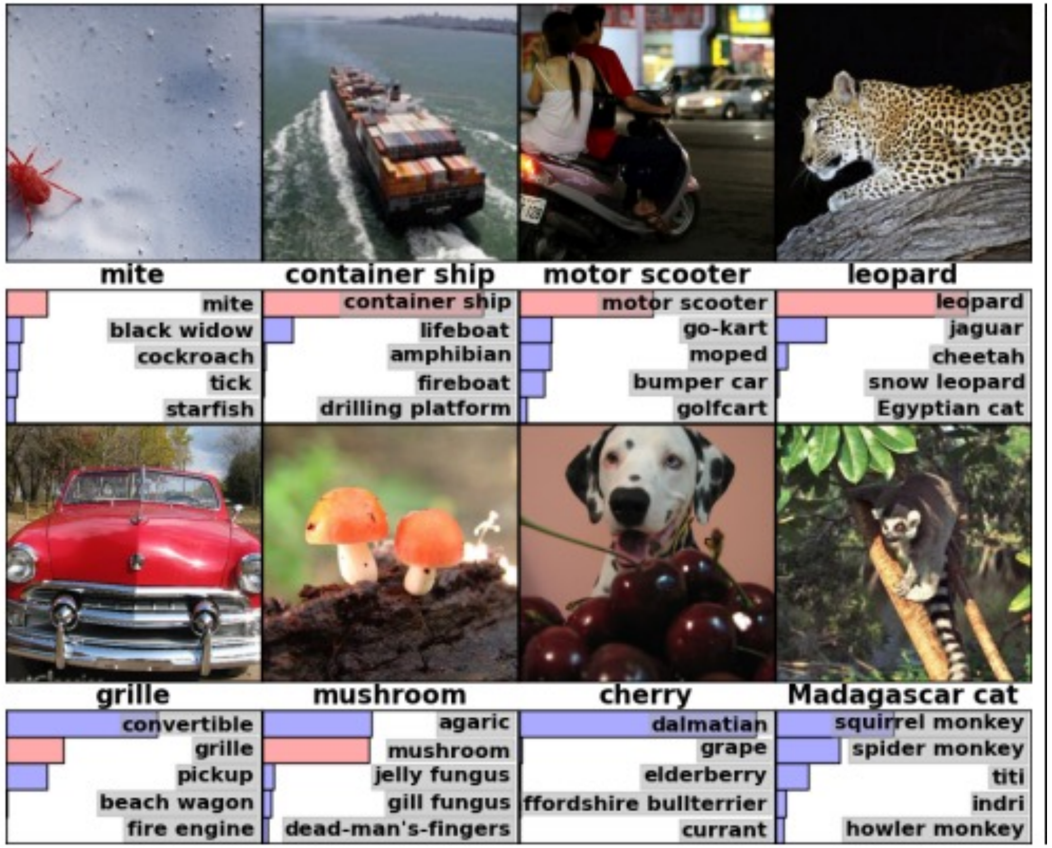


(a) Image classification (b) Object localization
(c) Semantic segmentation (d) Instance segmentation

Figure from https://arxiv.org/pdf/1704.06857.pdf

Image Classification

- Given an image, predict a single label
- A multi-class classification problem



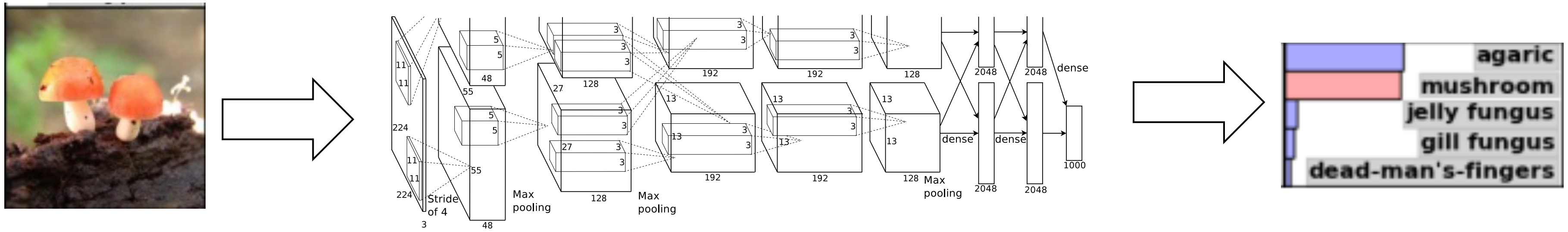


Figure from https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Image Classification + Localization

- Given an image, predict a single label and a bounding box for the object
- Bounding box is represented as (x, y, h, w), position (x,y) and height/width (h,w)

Figure from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5459257

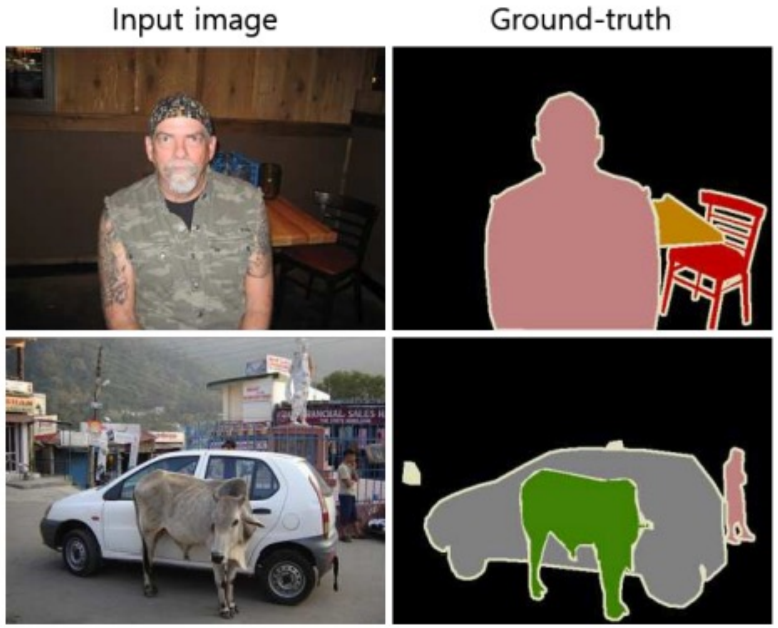
Human Pose Estimation

- Given an image of a human, predict the position of several keypoints (left hand, right hand, left elbow, … , right foot)
- This is a multiple regression problem, where each keypoint has a corresponding position (x_i, y_i)

Figure from
https://openaccess.thecvf.com/content_cvpr_2014/papers/Toshev_DeepPose_Human_Pose_2014_CVPR_paper.pdf

Semantic Segmentation

- Given an image, predict a label for every pixel in the image
- Not merely a classification problem, because there are strong correlations between pixel-specific labels



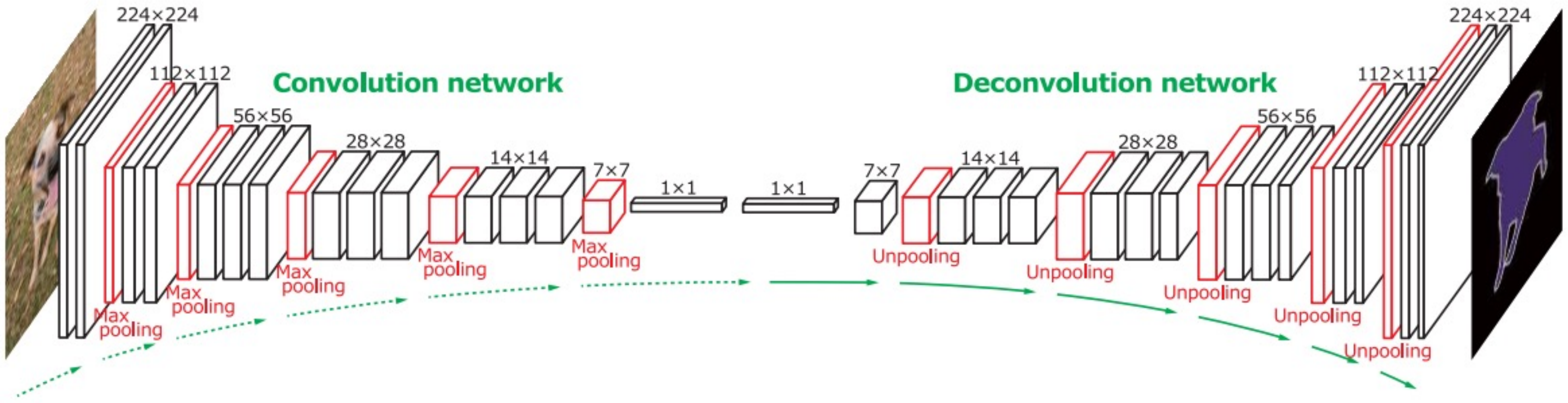


Figure from https://openaccess.thecvf.com/content_iccv_2015/papers/Noh_Learning_Deconvolution_Network_ICCV_2015_paper.pdf

Object Detection

- Given an image, for each object predict a bounding box and a label (x,y,w,h,l)
- Example: R-CNN
  - (x=110, y=13, w=50, h=72, l=person)
  - (x=90, y=55, w=81, h=87, l=horse)
  - (x=421, y=533, w=24, h=30, l=chair)
  - (x=2, y=25, w=51, h=121, l=gate)

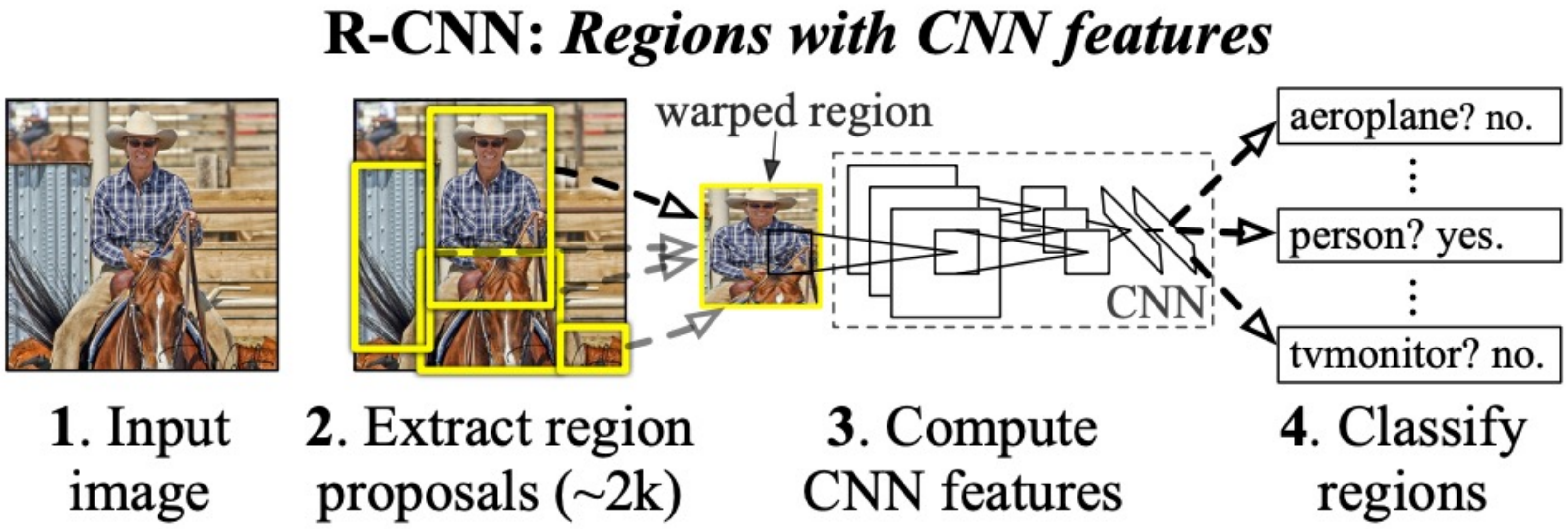


Figure from https://openaccess.thecvf.com/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf

Instance Segmentation

- Predict per-pixel labels as in semantic segmentation, but differentiate between different instances of the same label
- *Example*: if there are two people in the image, one person should be labeled **person-1** and one should be labeled **person-2**

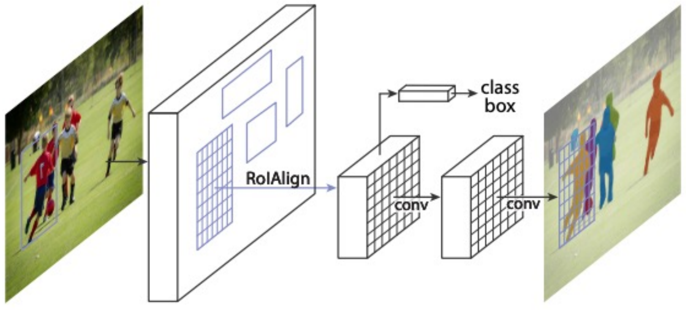
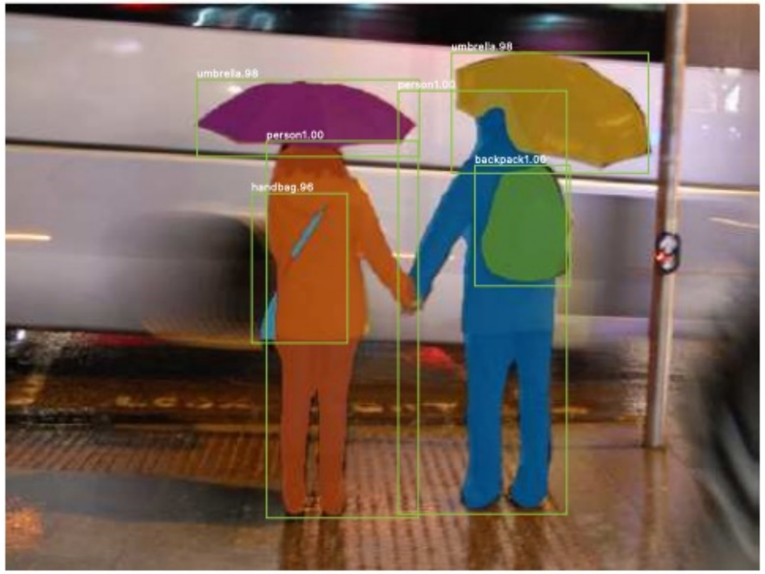


Figure 1. The **Mask R-CNN** framework for instance segmentation.



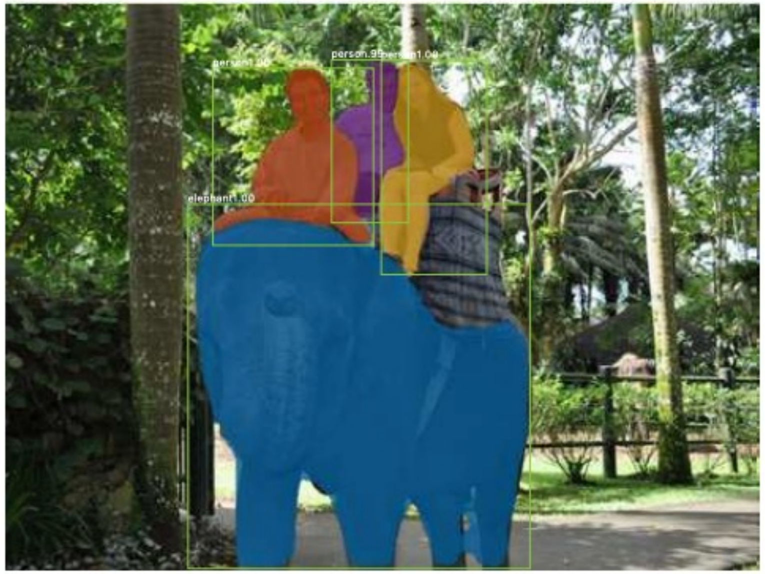
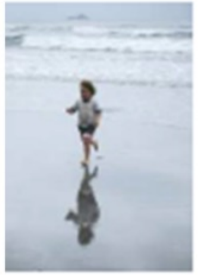


Figure from https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf

Image Captioning



Ground Truth Caption: A little boy runs away from the approaching waves of the ocean.

Generated Caption: A young boy is running on the beach.

Ground Truth Caption: A brunette girl wearing sunglasses and a yellow shirt.

Generated Caption: A woman in a black shirt and sunglasses smiles.

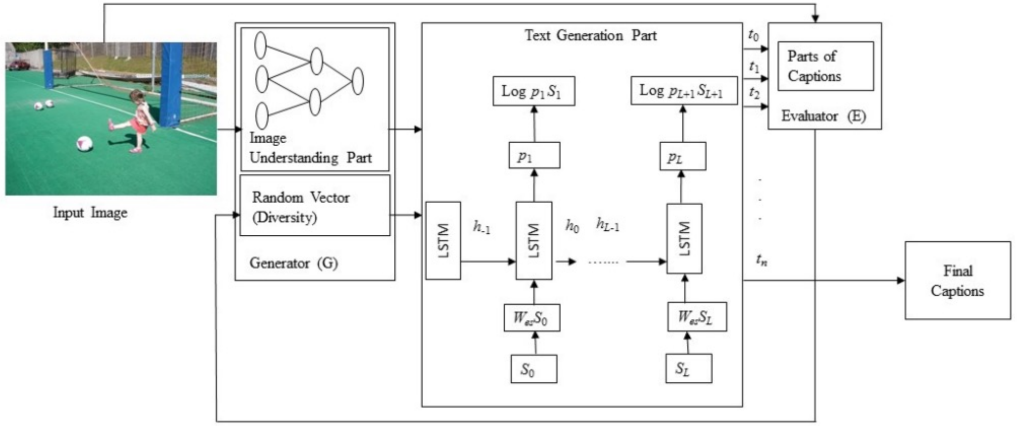


Fig. 3. A block diagram of other deep-learning-based captioning.

- Take an image as input, and generate a sentence describing it as output (i.e. the caption)
- Typical methods include a deep CNN/transformer and a RNN-like language model
- (The task of *Dense Captioning* is to generate one caption per bounding box)

Figure from https://dl.acm.org/doi/pdf/10.1145/3295748

Image Captioning

Table 1. An Overview of the Deep-Learning-Based Approaches for Image Captioning

Reference	Image Encoder	Language Model	Category
Kiros et al. 2014 [69]	AlexNet	LBL	MS, SL, WS, EDA
Kiros et al. 2014 [70]	AlexNet, VGGNet	1. LSTM 2. SC-NLM	MS, SL, WS, EDA
Mao et al. 2014 [95]	AlexNet	RNN	MS, SL, WS
Karpathy et al. 2014 [66]	AlexNet	DTR	MS, SL, WS, EDA
Mao et al. 2015 [94]	AlexNet, VGGNet	RNN	MS, SL, WS
Chen et al. 2015 [23]	VGGNet	RNN	VS, SL, WS, EDA
Fang et al. 2015 [33]	AlexNet, VGGNet	MELM	VS, SL, WS, CA
Jia et al. 2015 [59]	VGGNet	LSTM	VS, SL, WS, EDA
Karpathy et al. 2015 [65]	VGGNet	RNN	MS, SL, WS, EDA
Vinyals et al. 2015 [142]	GoogLeNet	LSTM	VS, SL, WS, EDA
Xu et al. 2015 [152]	AlexNet	LSTM	VS, SL, WS, EDA, AB
Jin et al. 2015 [61]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Wu et al. 2016 [151]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Sugano et at. 2016 [129]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Mathews et al. 2016 [97]	GoogLeNet	LSTM	VS, SL, WS, EDA, SC
Wang et al. 2016 [144]	AlexNet, VGGNet	LSTM	VS, SL, WS, EDA
Johnson et al. 2016 [62]	VGGNet	LSTM	VS, SL, DC, EDA
Mao et al. 2016 [92]	VGGNet	LSTM	VS, SL, WS, EDA
Wang et al. 2016 [146]	VGGNet	LSTM	VS, SL, WS, CA
Tran et al. 2016 [135]	ResNet	MELM	VS, SL, WS, CA
Ma et al. 2016 [90]	AlexNet	LSTM	VS, SL, WS, CA
You et al. 2016 [156]	GoogLeNet	RNN	VS, SL, WS, EDA, SCB
Yang et al. 2016 [153]	VGGNet	LSTM	VS, SL, DC, EDA
Anne et al. 2016 [6]	VGGNet	LSTM	VS, SL, WS, CA, NOB
Yao et al. 2017 [155]	GoogLeNet	LSTM	VS, SL, WS, EDA, SCB
Lu et al. 2017 [88]	ResNet	LSTM	VS, SL, WS, EDA, AB
Chen et al. 2017 [21]	VGGNet, ResNet	LSTM	VS, SL, WS, EDA, AB
Gan et al. 2017 [41]	ResNet	LSTM	VS, SL, WS, CA, SCB
Pedersoli et al. 2017 [112]	VGGNet	RNN	VS, SL, WS, EDA, AB
Ren et al. 2017 [119]	VGGNet	LSTM	VS, ODL, WS, EDA
Park et al. 2017 [111]	ResNet	LSTM	VS, SL, WS, EDA, AB
Wang et al. 2017 [148]	ResNet	LSTM	VS, SL, WS, EDA
Tavakoli et al. 2017 [134]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Liu et al. 2017 [84]	VGGNet	LSTM	VS, SL, WS, EDA, AB
Gan et al. 2017 [39]	ResNet	LSTM	VS, SL, WS, EDA, SC
Dai et al. 2017 [26]	VGGNet	LSTM	VS, ODL, WS, EDA
Shetty et al. 2017 [126]	GoogLeNet	LSTM	VS, ODL, WS, EDA
Liu et al. 2017 [85]	Inception-V3	LSTM	VS, ODL, WS, EDA
Gu et al. 2017 [51]	VGGNet	1. Language CNN 2. LSTM	VS, SL, WS, EDA
Yao et al. 2017 [154]	VGGNet	LSTM	VS, SL, WS, CA, NOB

(Continued)

- Take an image as input, and generate a sentence describing it as output (i.e. the caption)
- Typical methods include a deep CNN/transformer and a RNN-like language model
- (The task of *Dense Captioning* is to generate one caption per bounding box)

Table from https://dl.acm.org/doi/pdf/10.1145/3295748

Medical Image Analysis

Notice that **most** of these tasks are structured prediction problems, not merely classification

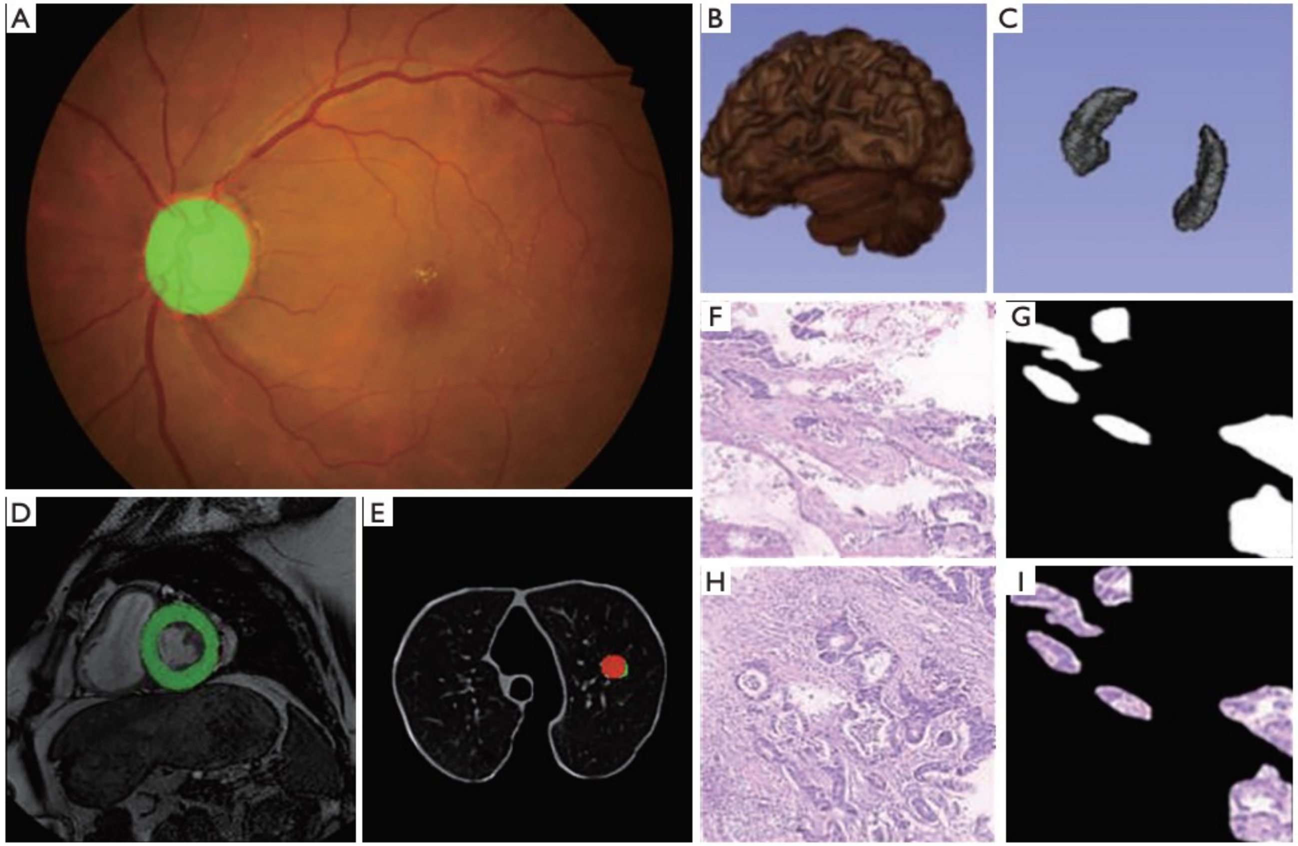


Figure 2 Deep learning application in medical image analysis. (A) Fundus detection; (B,C) hippocampus segmentation; (D) left ventricular segmentation; (E) pulmonary nodule classification; (F,G,H,I) gastric cancer pathology segmentation. The staining method is H&E, and the magnification is ×40.

Figure from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7327346/

TASK: IMAGE GENERATION

Image Generation

- Class-conditional generation
- Super resolution
- Image Editing
- Style transfer
- Text-to-image (TTI) generation

Figure from Razavi et al. (2019)

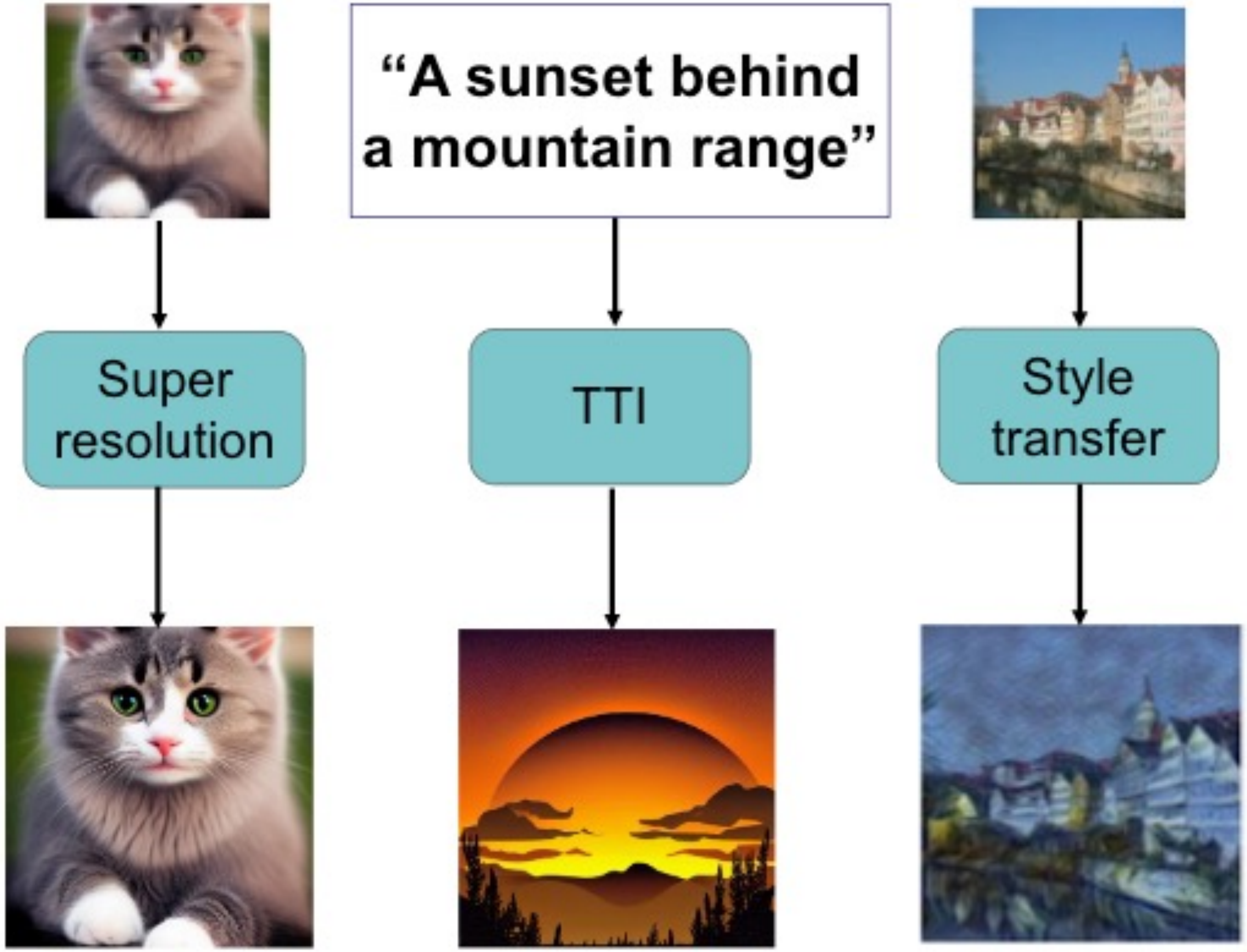


Figure from Bie et al. (2023)

Class Conditional Generation

- **Task**: Given a class label indicating the image type, sample a new image from the model with that type
- Image classification is the problem of taking in an image and predicting its label p(y|x)
- Class conditional generation is doing this in reverse p(x|y)

sea anemone

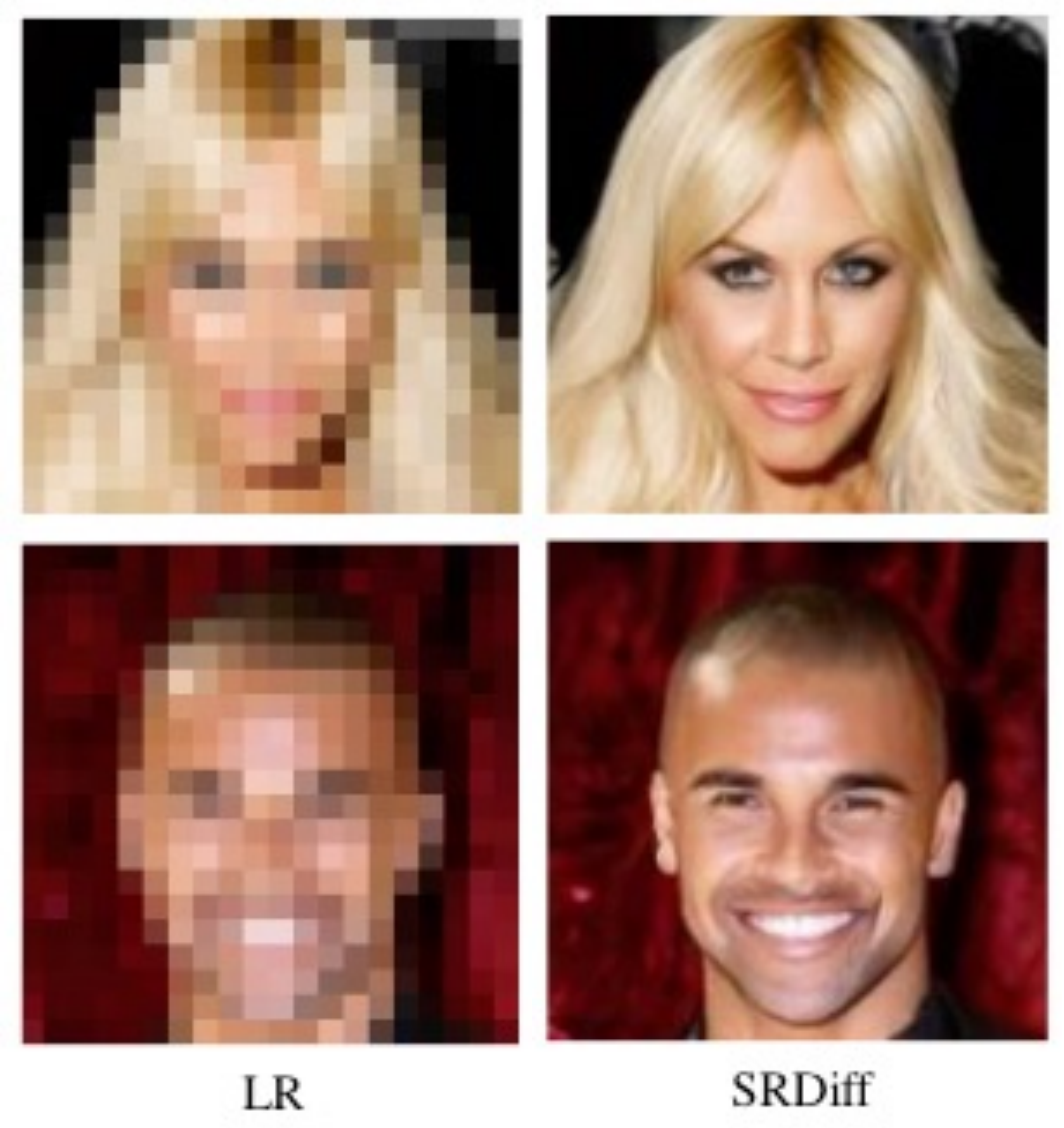
brain coral

slug

goldfinch

Figure from Razavi et al. (2019)

Super Resolution



- Given a low resolution image, generate a high resolution reconstruction of the image
- Compelling on low resolution inputs (see example to the left) but also effective on high resolution inputs

Figure from Li et al. (2021)

Image Editing

A variety of tasks involve automatic editing of an image:

- **Inpainting** fills in the (pre-specified) missing pixels
- **Colorization** restores color to a greyscale image
- **Uncropping** creates a photo-realistic reconstruction of a missing side of an image

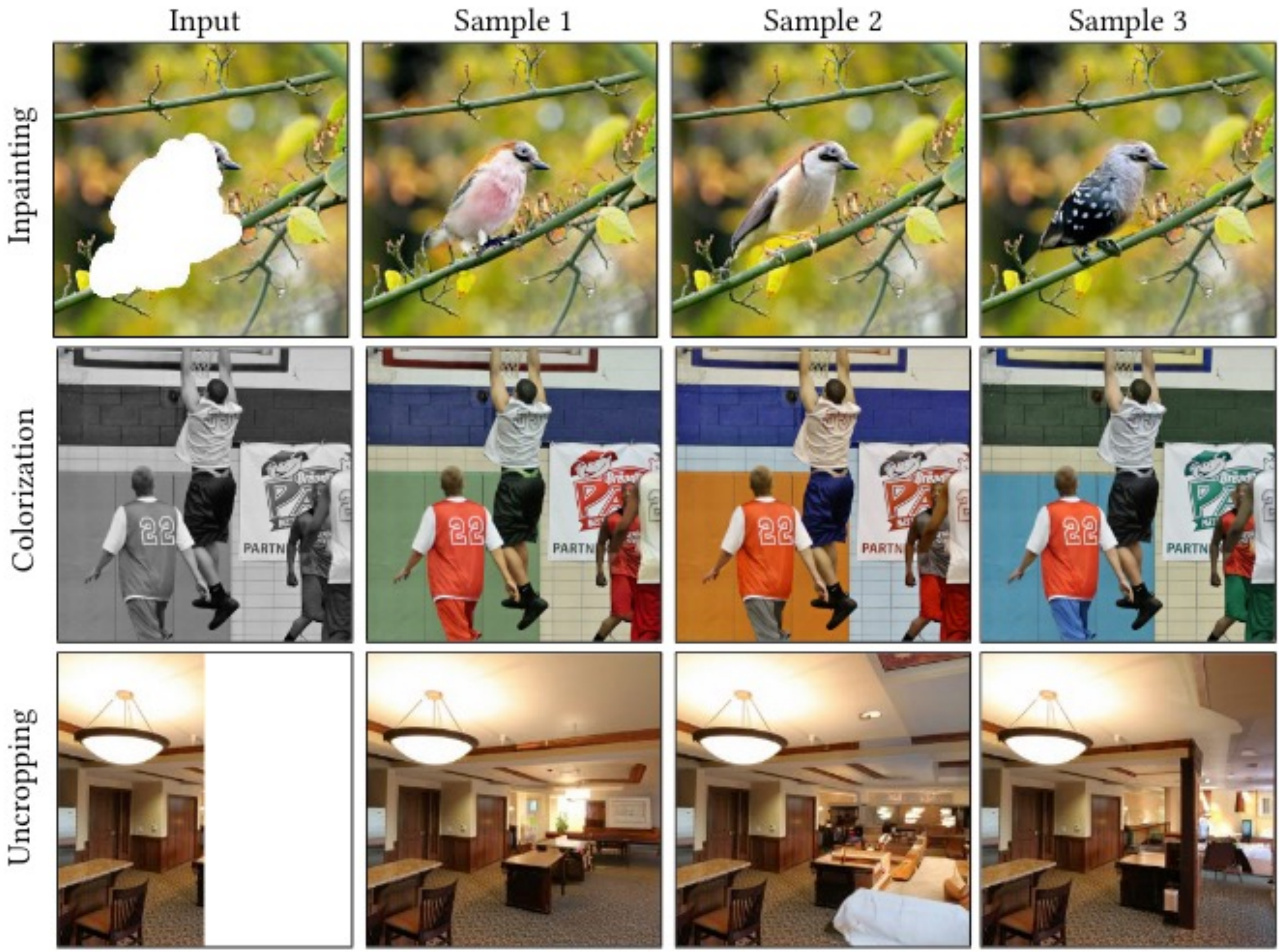


Figure from Saharia et al. (2022)

Style Transfer

- The goal of style transfer is to blend two images
- Yet, the blend should retain the semantic content of the source image presented in the style of another image

Figure 3. Images that combine the content of a photograph with the style of several well-known artworks. The images were created by finding an image that simultaneously matches the content representation of the photograph and the style representation of the artwork. The original photograph depicting the Neckarfront in Tübingen, Germany, is shown in **A** (Photo: Andreas Praefcke). The painting that provided the style for the respective generated image is shown in the bottom left corner of each panel. **B** *The Shipwreck of the Minotaur* by J.M.W. Turner, 1805. **C** *The Starry Night* by Vincent van Gogh, 1889. **D** *Der Schrei* by Edvard Munch, 1893. **E** *Femme nue assise* by Pablo Picasso, 1910. **F** *Composition VII* by Wassily Kandinsky, 1913.

Figure from Gatys et al. (2016)

Text-to-Image Generation

- Given a text description, sample an image that depicts the prompt
- The following images are samples from SDXL with refinement

Prompt: A propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese.

Figure from Podell et al. (2023)

Text-to-Image Generation

- Given a text description, sample an image that depicts the prompt
- The following images are samples from SDXL with refinement

Prompt: Epic long distance cityscape photo of New York City flooded by the ocean and overgrown buildings and jungle ruins in rainforest, at sunset, cinematic shot, highly detailed, 8k, golden light

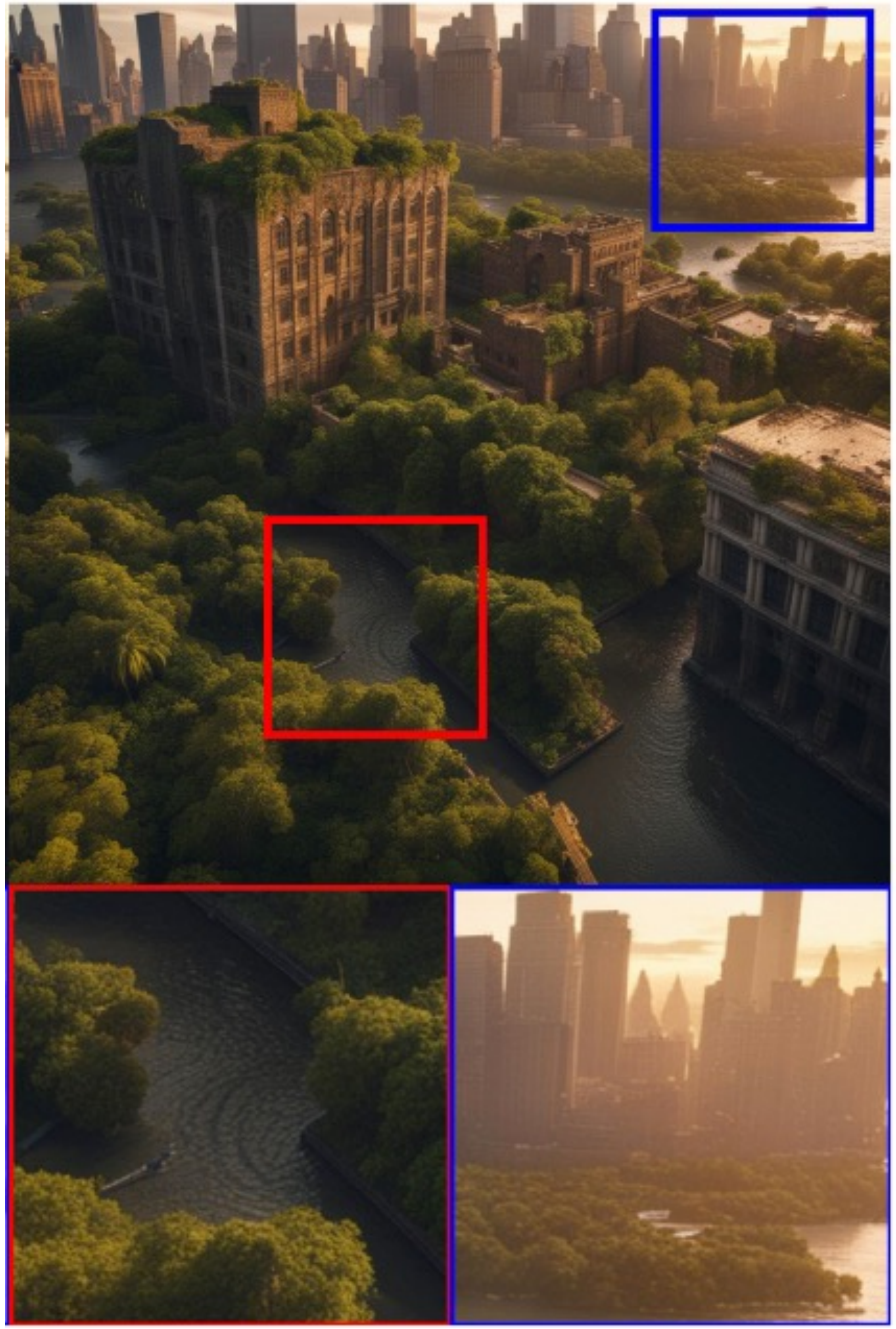


Figure from Podell et al. (2023)

Text-to-Image Generation

- Given a text description, sample an image that depicts the prompt
- The following images are samples from SDXL with refinement

Prompt: close up headshot, futuristic young woman, wild hair sly smile in front of gigantic UFO, dslr, sharp focus, dynamic composition

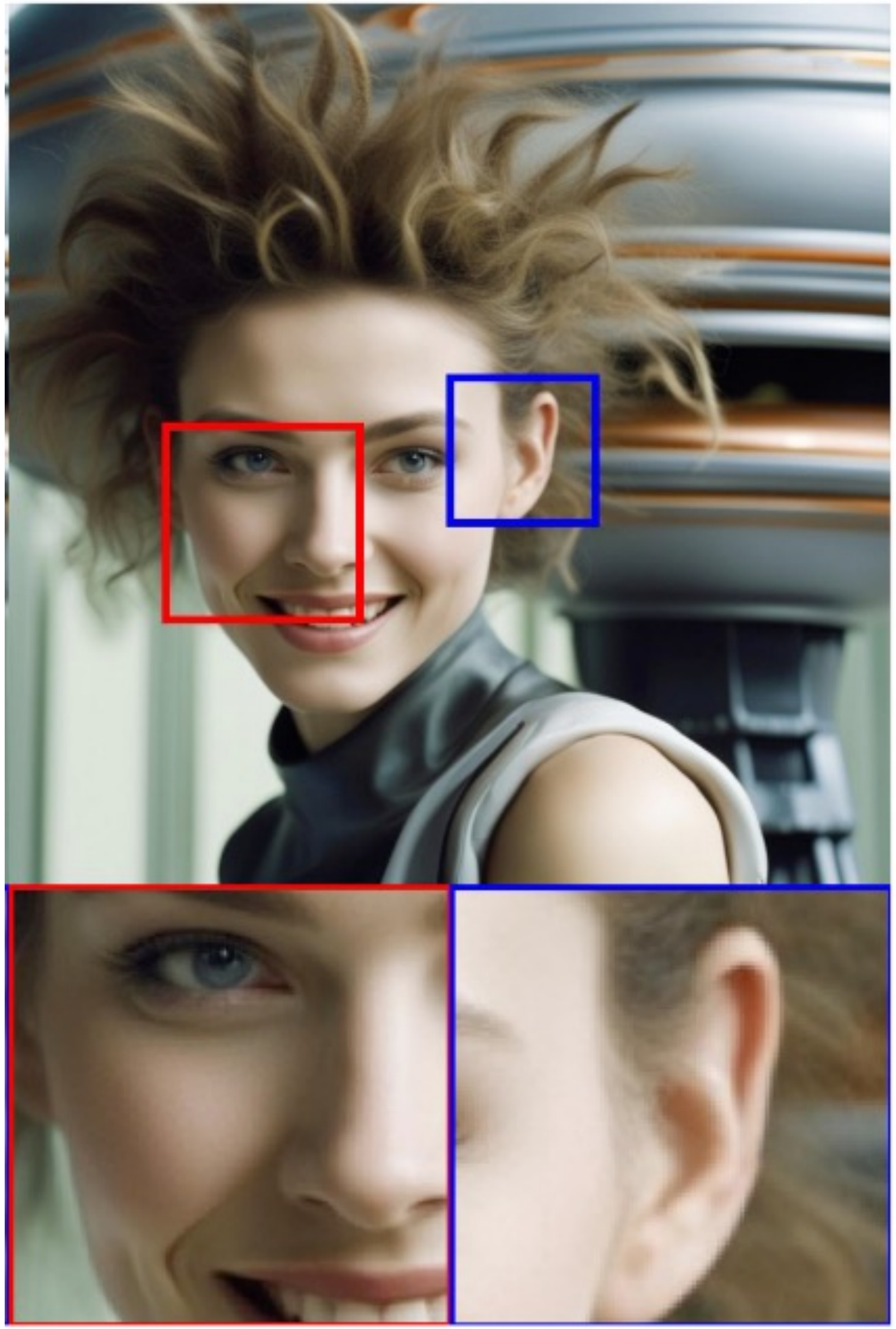


Figure from Podell et al. (2023)

Text-to-Image Generation

- Given a text description, sample an image that depicts the prompt
- The following images are samples from SDXL with refinement

Prompt: close up headshot, futuristic **old man**, wild hair sly smile in front of gigantic UFO, dslr, sharp focus, dynamic composition, **rule of thirds**

Figure from https://stablediffusionweb.com/

In-Class Poll

Question:

What are the potential societal impacts of image generation?

Answer:

Summary

- Computer Vision
- Task: Image Generation
- Model: Generative Adversarial Network (GAN)
- Learning for GANs
- Scaling Up the Model Size
- Societal Impacts of Image Generation

MODEL: GENERATIVE ADVERSARIAL NETWORK (GAN)

Stable Diffusion still can't explain GANs

Prompt: slide explaining Generative Adversarial Networks (GANs) for Intro to Machine Learning course, carefully designed, easy to follow

Negative Prompt: boring, unclear, nontechnical

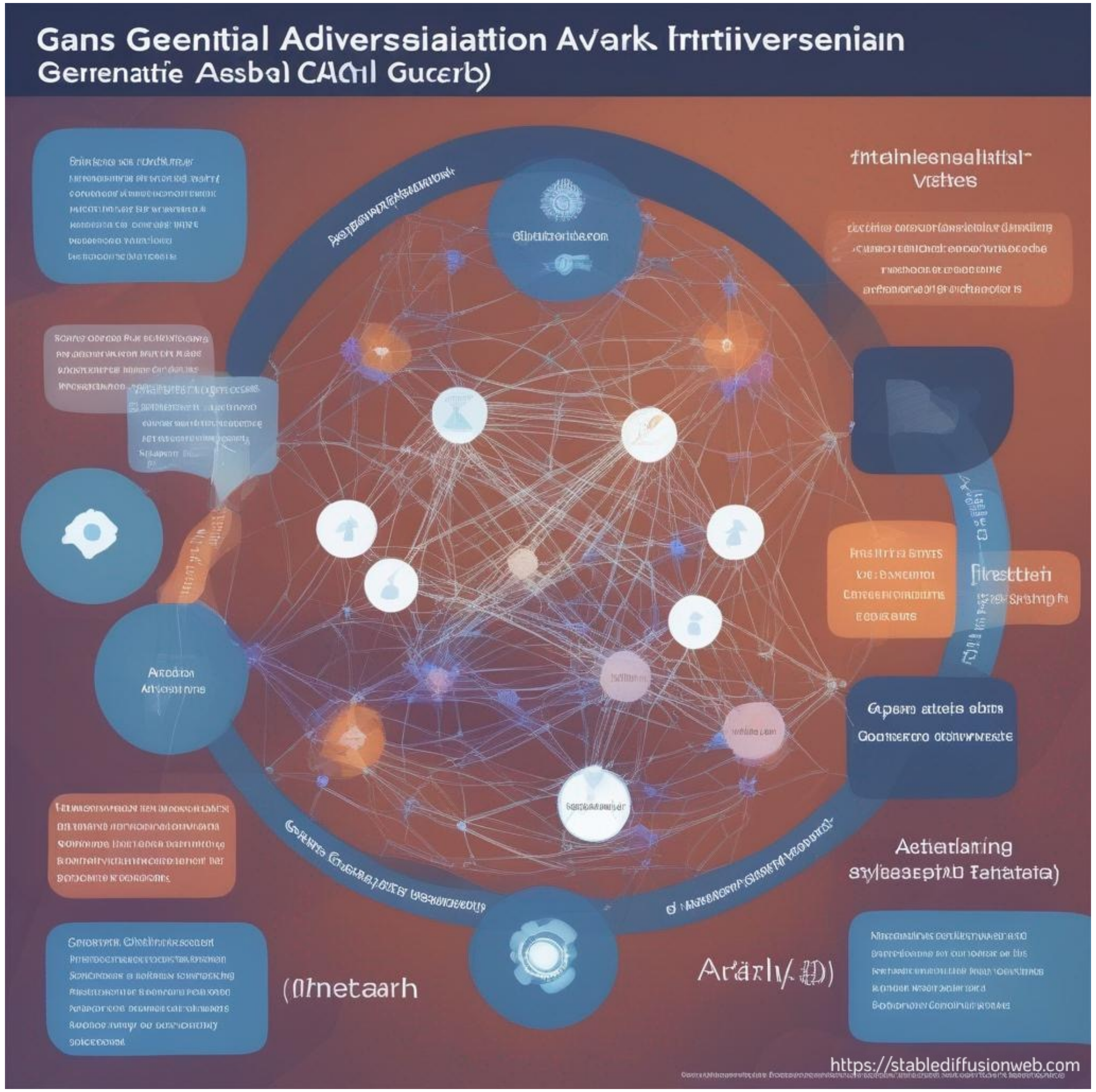


Figure from https://stablediffusionweb.com/

DALL-E isn’t much better

Prompt: a lemming from the classic computer game Lemmings explaining GANs

Figure from https://www.bing.com/images/create

Generative Adversarial Networks (GANs)

A GAN consists of two deterministic neural network models:

1) the Generator

takes a vector of random noise as input, and generates an image

2) the Discriminator

takes in an image classifies whether it is real (label 1) or fake (label 0)

Generator Model

1) the Generator

takes a vector of random noise as input, and generates an image

Example Generator: DCGAN

- An inverted CNN with four **fractionally-strided** convolution layers (not deconvolution)
- These fractional strides grow the size of the image from layer to layer
- The final layer has three channels for red/green/blue

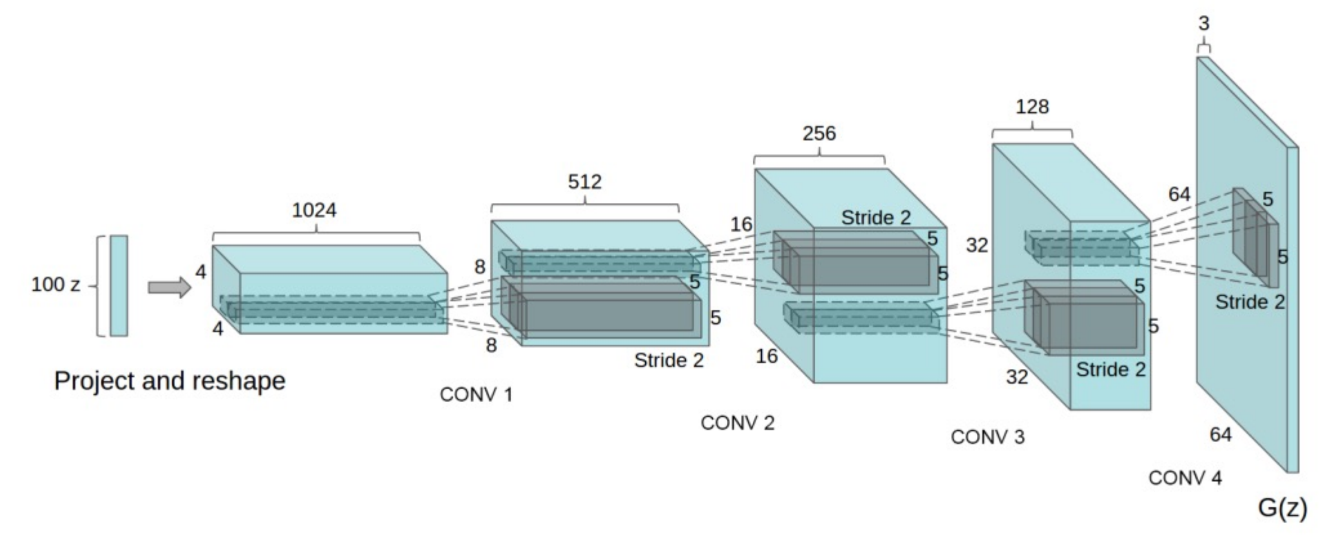


Figure from Radford et al. (2016)

Generative Adversarial Networks (GANs)

A GAN consists of two deterministic neural network models:

1) the Generator

takes a vector of random noise as input, and generates an image

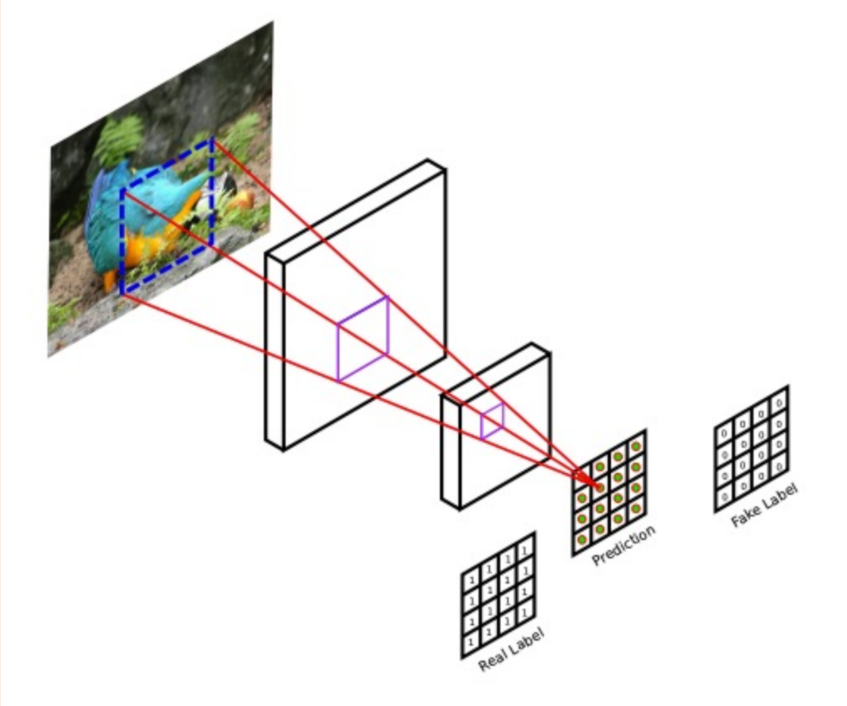
2) the Discriminator

takes in an image classifies whether it is real (label 1) or fake (label 0)

Discriminator Model

Example Discriminator: PatchGAN

- Convolutional neural network
- Looks at each patch of the image and tries to predict whether it is real or fake
- Helps avoid producing blurry images



2) the Discriminator

takes in an image classifies whether it is real (label 1) or fake (label 0)

Figure from Demir & Unal (2018)

Generative Adversarial Networks (GANs)

A GAN consists of two deterministic neural network models:

1) the Generator

takes a vector of random noise as input, and generates an image

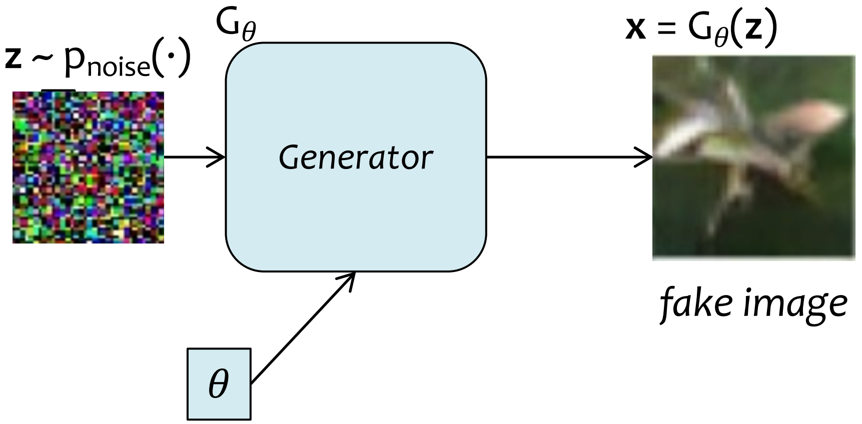
2) the Discriminator

takes in an image classifies whether it is real (label 1) or fake (label 0)

In training, the GAN plays a two player minimax game:

1. the Generator tries to create realistic images to fool the Discriminator into thinking they are real
2. the Discriminator tries to identify the real images from the fake

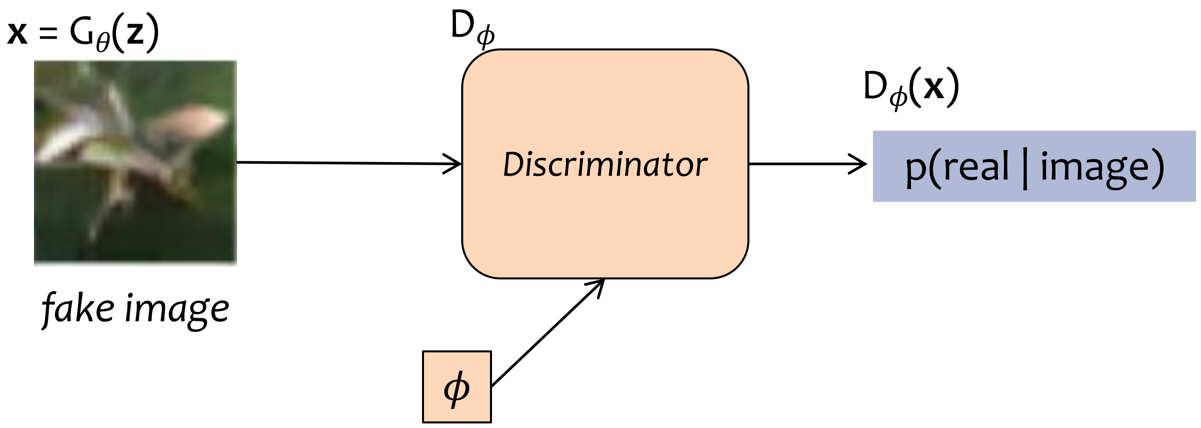
Generative Adversarial Networks (GANs)



Real/fake images from Huang et al. (2017)

Gaussian noise from https://physbam.stanford.edu/cs448x/old/Noise_Review.html

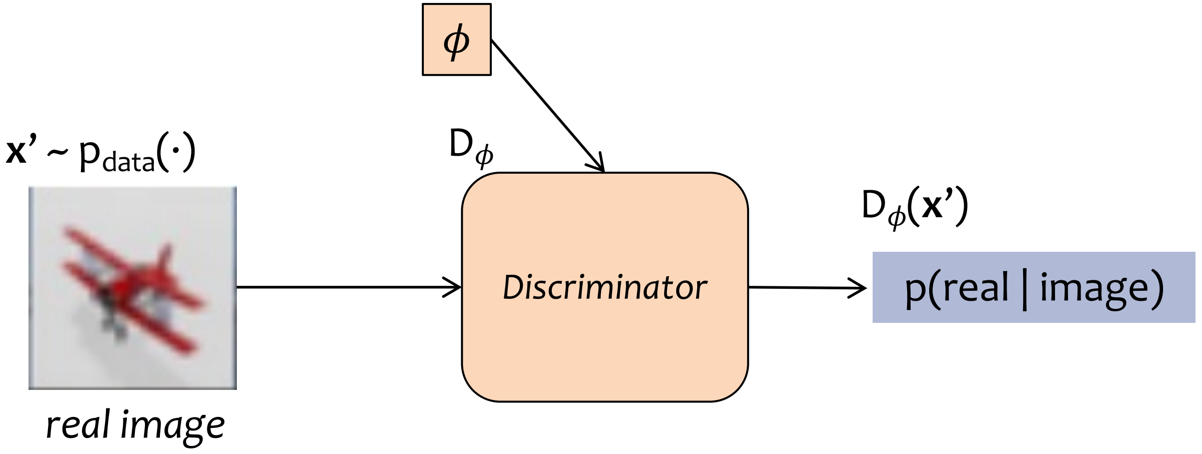
Generative Adversarial Networks (GANs)



Real/fake images from Huang et al. (2017)

Gaussian noise from https://physbam.stanford.edu/cs448x/old/Noise_Review.html

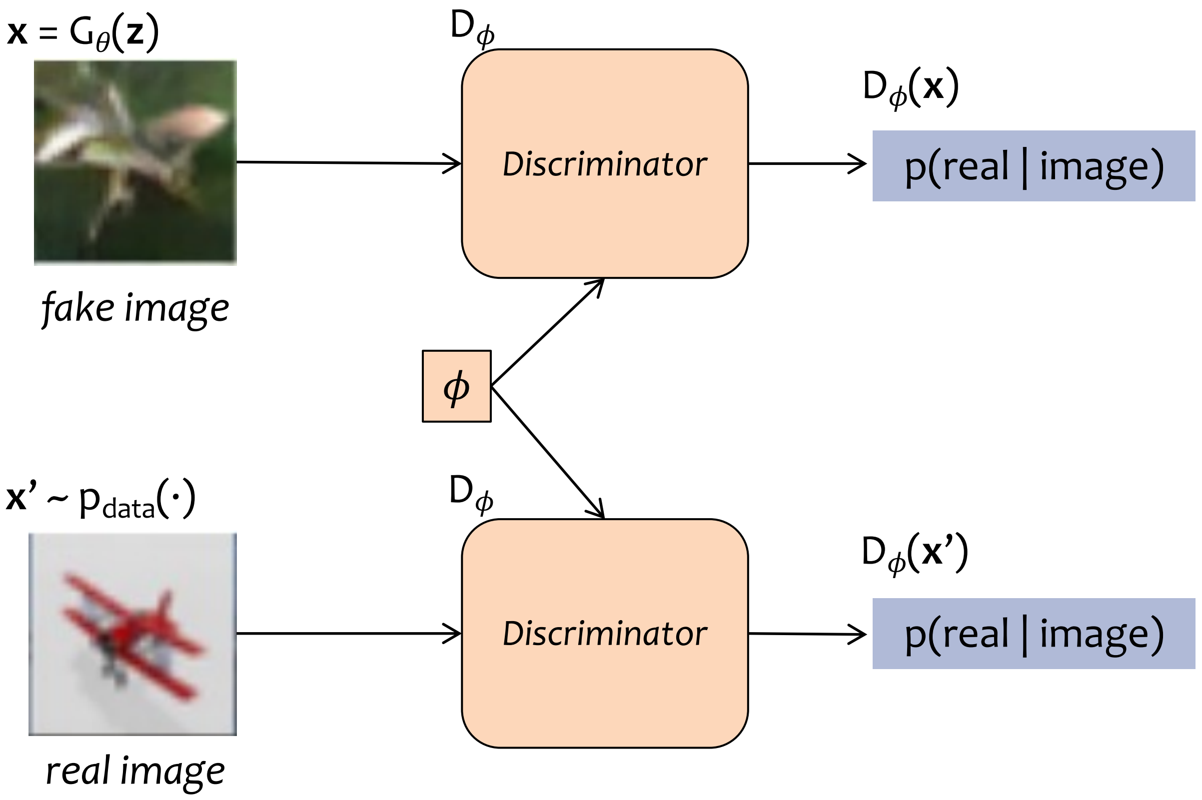
Generative Adversarial Networks (GANs)



Real/fake images from Huang et al. (2017)

Gaussian noise from https://physbam.stanford.edu/cs448x/old/Noise_Review.html

Generative Adversarial Networks (GANs)



LEARNING FOR GANS

Generative Adversarial Networks (GANs)

A GAN consists of two deterministic neural network models:

1) the Generator

takes a vector of random noise as input, and generates an image

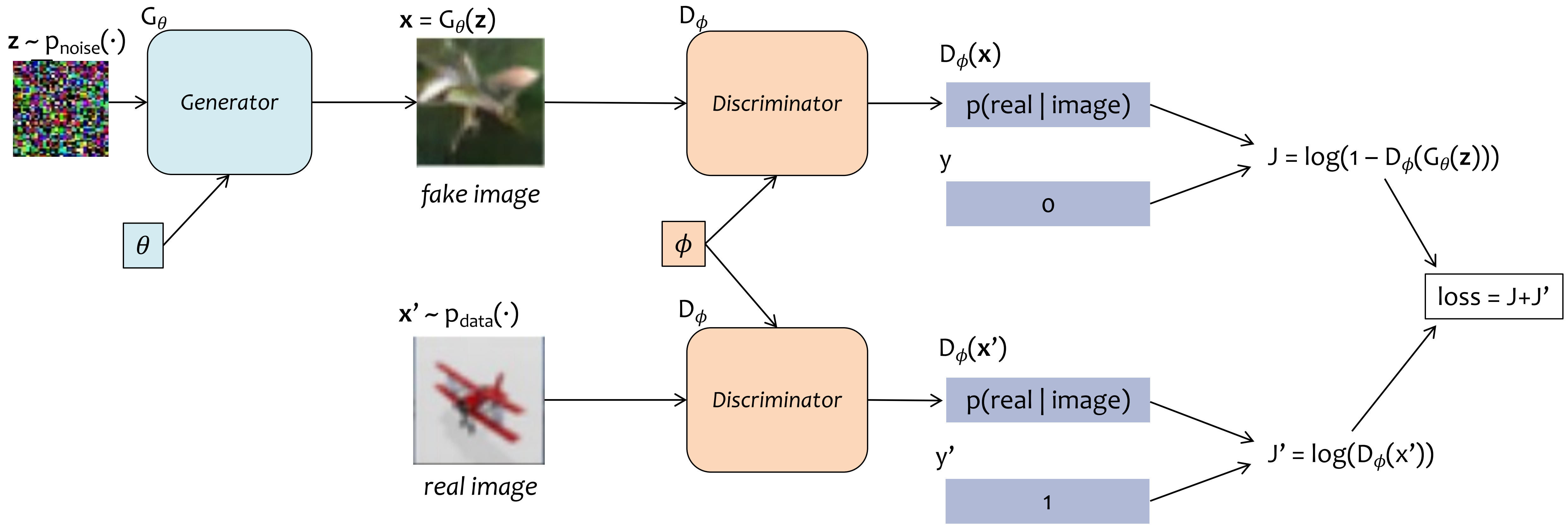
2) the Discriminator

takes in an image classifies whether it is real (label 1) or fake (label 0)

In training, the GAN plays a two player minimax game:

1. the Generator tries to create realistic images to fool the Discriminator into thinking they are real
2. the Discriminator tries to identify the real images from the fake

Generative Adversarial Networks (GANs)



Real/fake images from Huang et al. (2017)

Gaussian noise from https://physbam.stanford.edu/cs448x/old/Noise_Review.html

Generative Adversarial Networks (GANs)

$$\max_{\phi} \log\left(D_\phi(\mathbf{x}^{(i)})\right) + \log\left(1 - D_\phi(G_\theta(\mathbf{z}^{(i)}))\right)$$

$$\min_{\theta} \log\left(1 - D_\phi(G_\theta(\mathbf{z}^{(i)}))\right)$$

The discriminator is trying to maximize the likelihood of a binary classifier with labels {real = 1, fake = 0}, on the fixed output of the generator

The generator is trying to minimize the likelihood of its generated (fake) image being classified as fake, according to a fixed discriminator

In training, the GAN plays a two player minimax game:

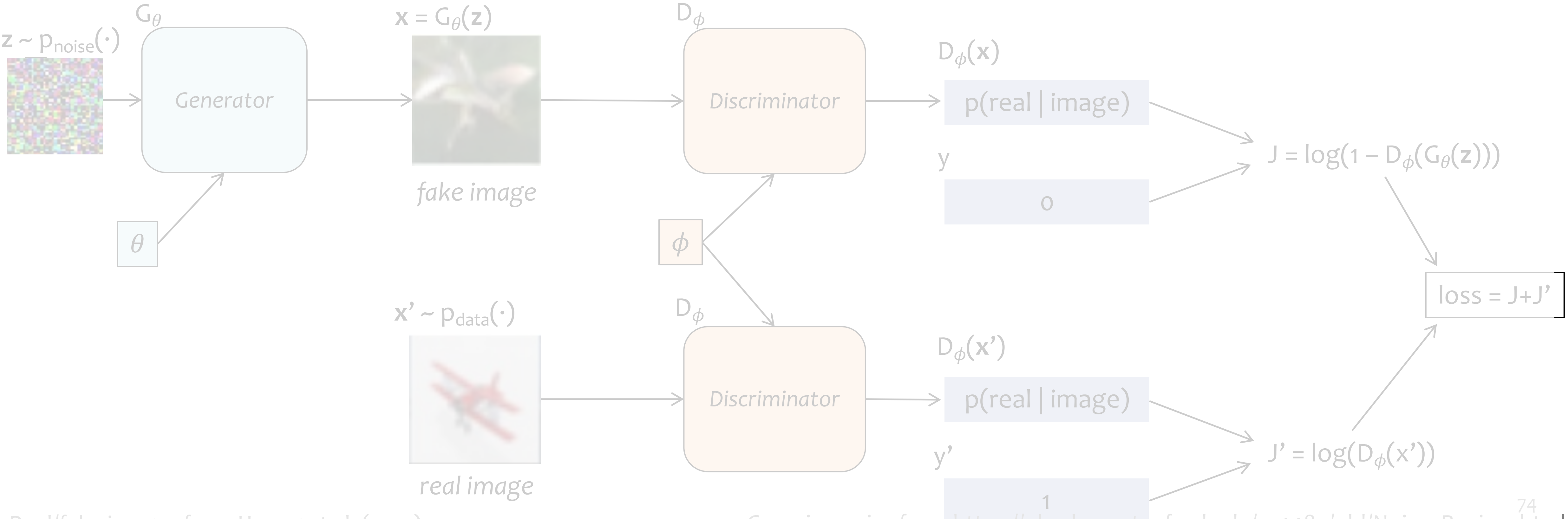
1. the Generator tries to create realistic images to fool the Discriminator into thinking they are real
2. the Discriminator tries to identify the real images from the fake

Learning a GAN

- Objective function is a simple differentiable function
- We chose G and D to be differentiable neural networks

Training alternates between:

- Keep G_θ fixed and backprop through D_ϕ
- Keep D_ϕ fixed and backprop through G_θ



Real/fake images from Huang et al. (2017)

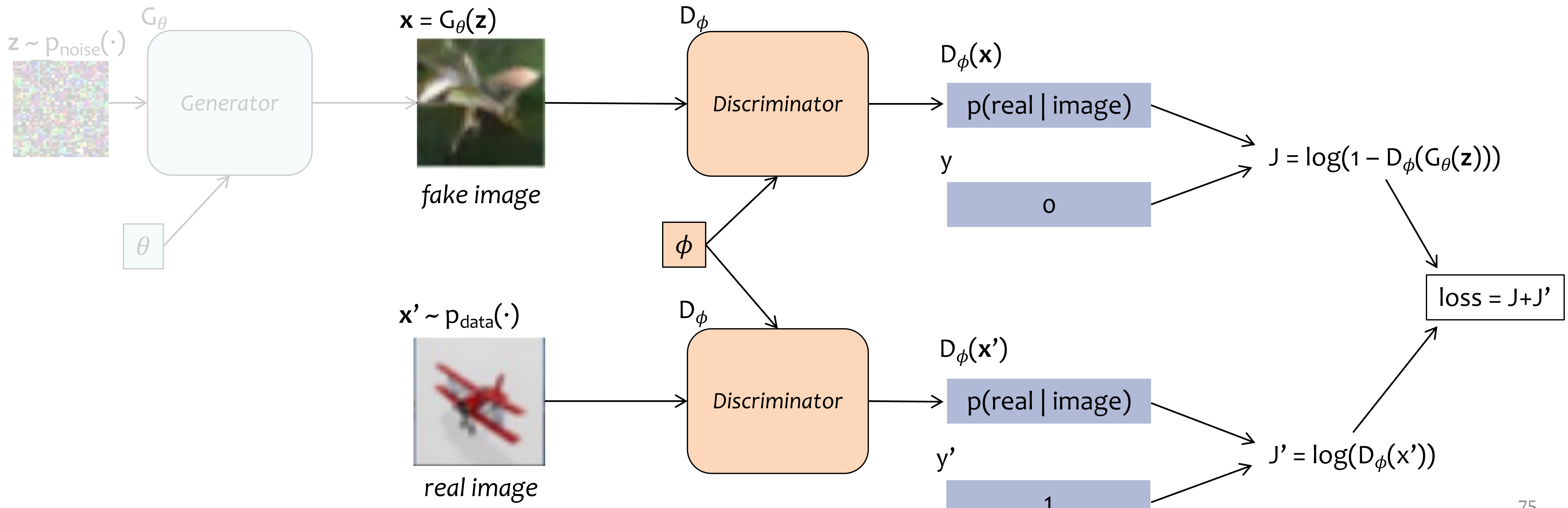
Gaussian noise from https://physbam.stanford.edu/cs448x/old/Noise_Review.html

Learning a GAN

- Objective function is a simple differentiable function
- We chose G and D to be differentiable neural networks

Training alternates between:

- **Keep G_θ fixed and backprop through D_ϕ**
- Keep D_ϕ fixed and backprop through G_θ



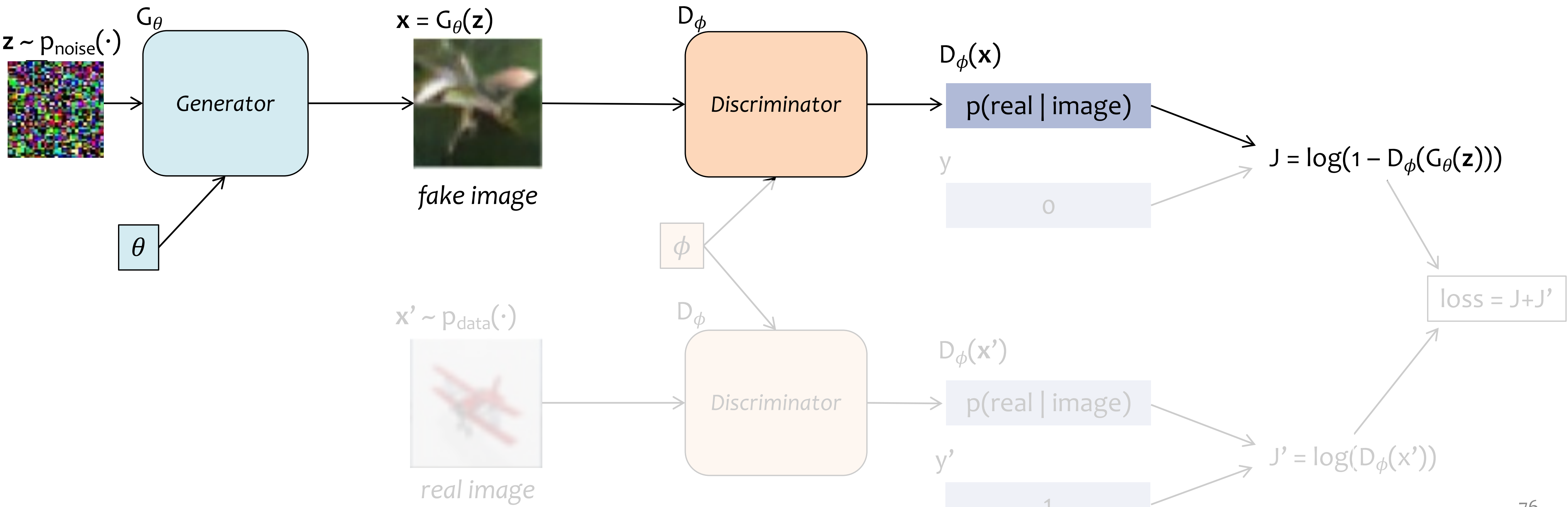
Real/fake images from Huang et al. (2017)

Learning a GAN

- Objective function is a simple differentiable function
- We chose G and D to be differentiable neural networks

Training alternates between:

- Keep G_θ fixed and backprop through D_ϕ
- **Keep D_ϕ fixed and backprop through G_θ**



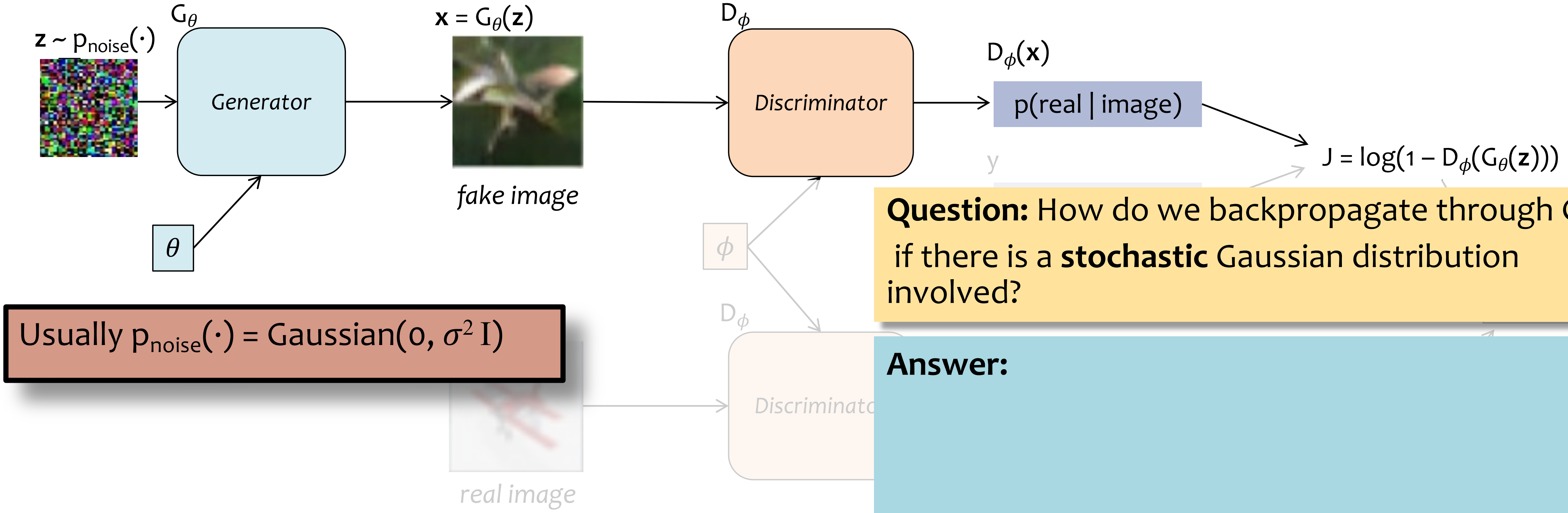
Real/fake images from Huang et al. (2017)

Learning a GAN

- Objective function is a simple differentiable function
- We chose G and D to be differentiable neural networks

Training alternates between:

- Keep G_θ fixed and backprop through D_ϕ
- **Keep D_ϕ fixed and backprop through G_θ**



Usually $p_{noise}(\cdot)$ = Gaussian(0, σ^2 I)

Question: How do we backpropagate through G_θ if there is a **stochastic** Gaussian distribution involved?

Answer:

Real/fake images from Huang et al. (2017)

Learning a GAN

- Training data consists of a collection of m unlabeled images $x^{(1)}, \ldots, x^{(m)}$
- Optimization is similar to block coordinate descent
- But instead of exactly solving the min/max problem, we take a step of mini-batch SGD

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**
  for k steps **do**
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

  end for
  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for
The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

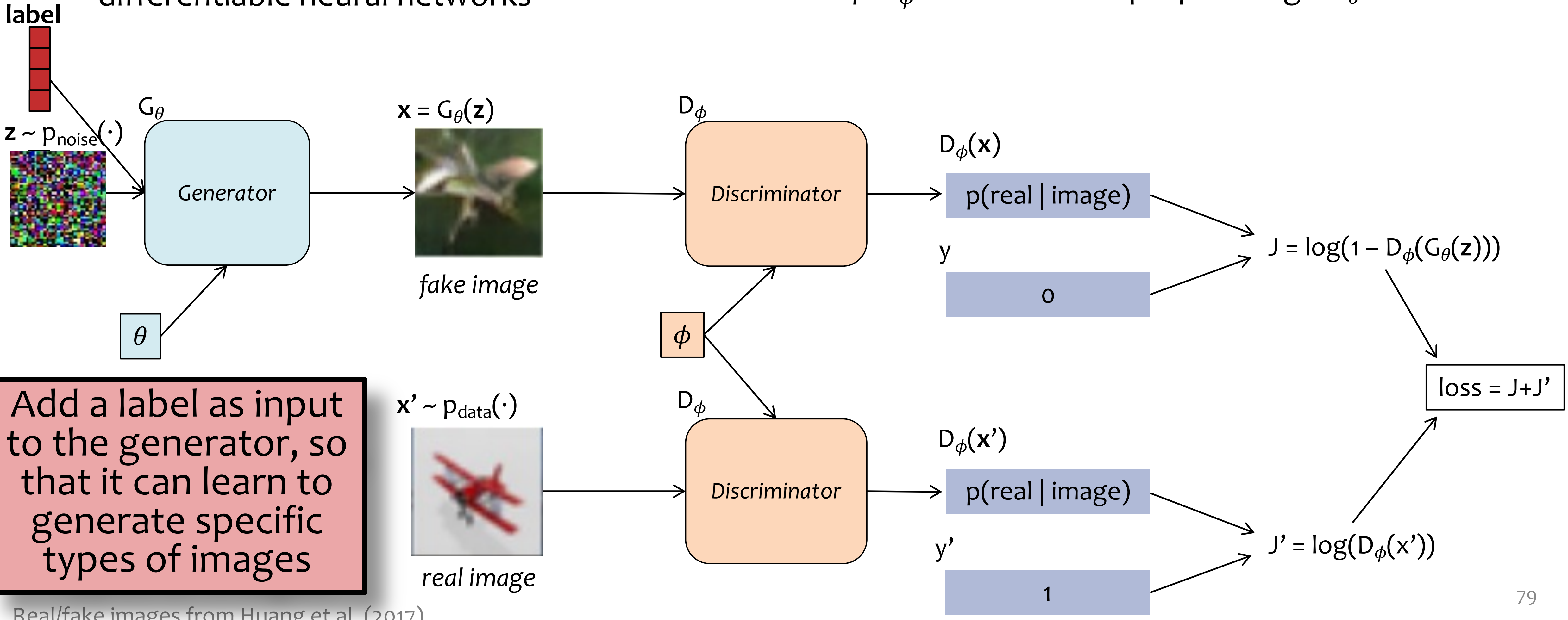
Figure from https://arxiv.org/pdf/1406.2661.pdf

Class-conditional GANs

- Objective function is a simple differentiable function
- We chose G and D to be differentiable neural networks

Training alternates between:

- Keep G_θ fixed and backprop through D_ϕ
- Keep D_ϕ fixed and backprop through G_θ



Add a label as input to the generator, so that it can learn to generate specific types of images

Real/fake images from Huang et al. (2017)

SCALING UP THE MODEL SIZE

Scaling Up the Model Size

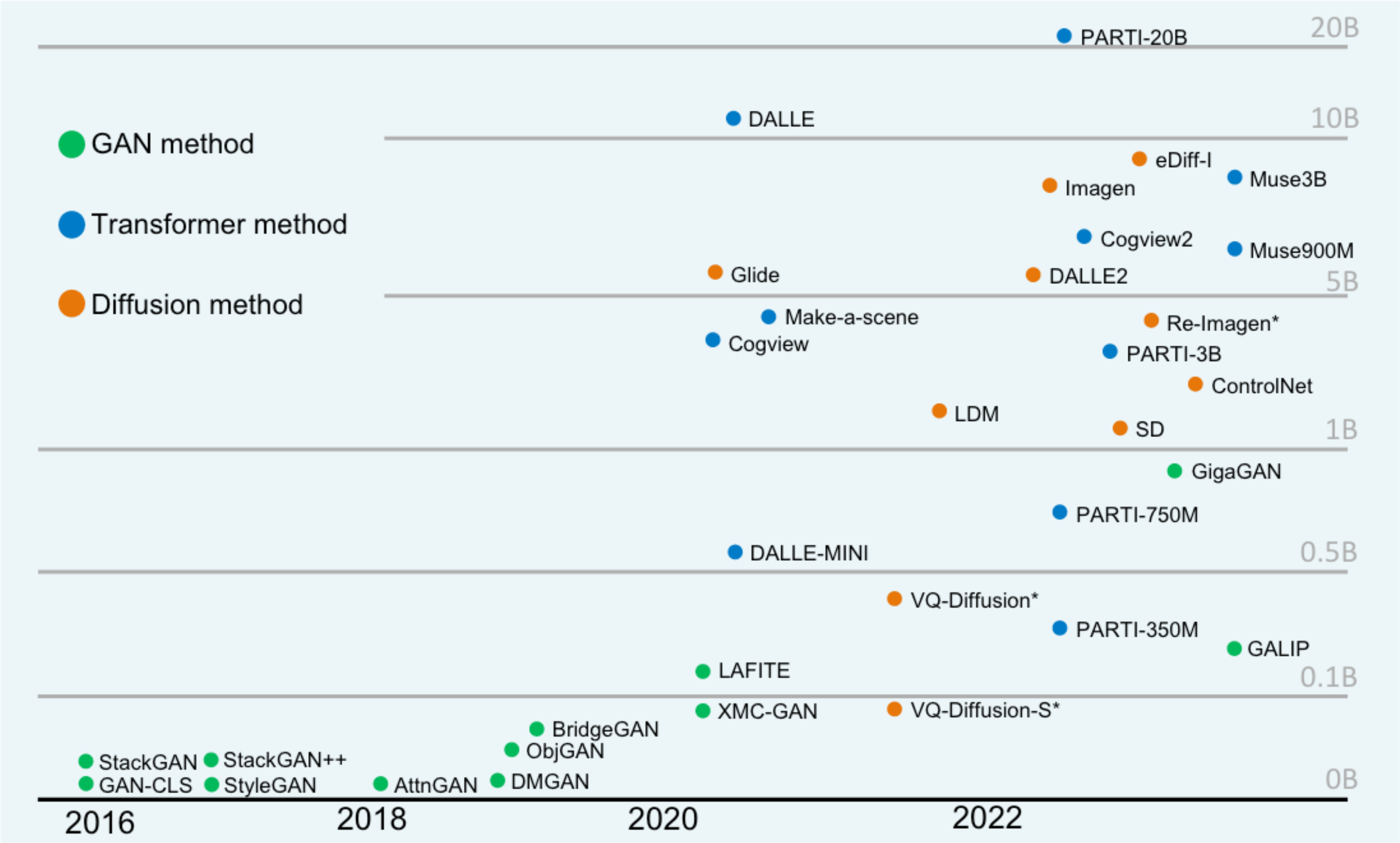


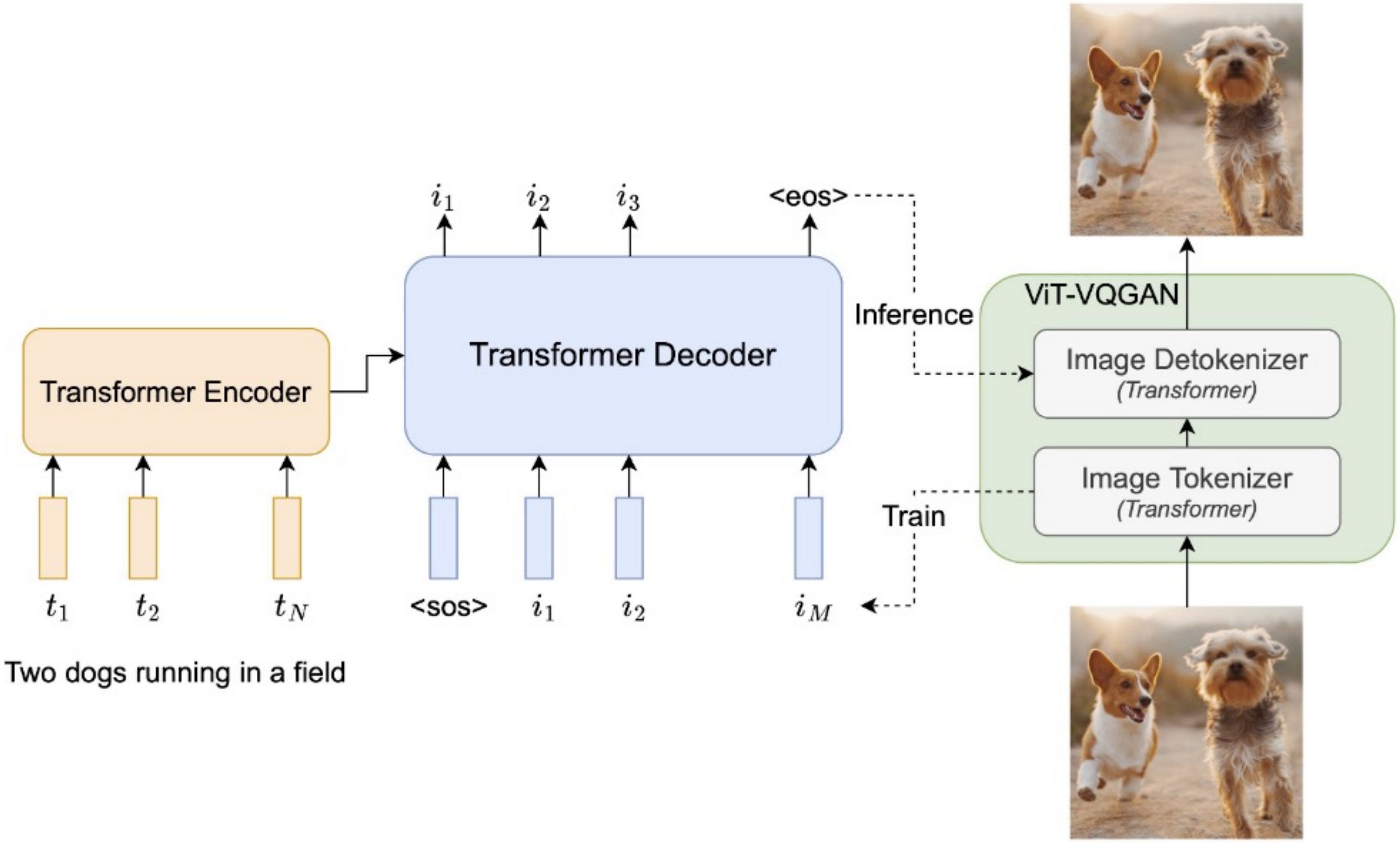
Fig. 5. Timeline of TTI model development, where green dots are GAN TTI models, blue dots are autoregressive Transformers and orange dots are Diffusion TTI models. Models are separated by their parameter, which are in general counted for all their components. Models with asterisk are calculated without the involvement of their text encoders.

Figure from Bie et al. (2023)

Scaling Up the Model Size

The Pathways Autoregressive Text-to-Image (Parti) model:

- treat image generation as a sequence-to-sequence problem
- text prompt is input to encoder
- sequence of image tokens is output of decoder
- ViT-VQGAN takes in the image tokens and generates a high-quality image



Scaling Up the Model Size

Prompt: A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!

Parti with different model sizes

Figure from https://sites.research.google/parti/

Watermarking & Attribution

- **Watermarking**
  - A digital watermark allows one to identify when an image has been created by a model
  - Most methods for image generation (GANs, VAEs, stable diffusion) can be augmented with watermarking
- **Fake-image Detection**
  - Goal: identify fakes even without a watermark
- **Model Attribution**
  - Identify which generative model created an image (e.g. Dalle-2 vs. SDXL)
  - Very successful (natural watermarks)
- **Image Attribution**
  - Goal: identify the source images that led to the generation of a new image
  - Extremely challenging

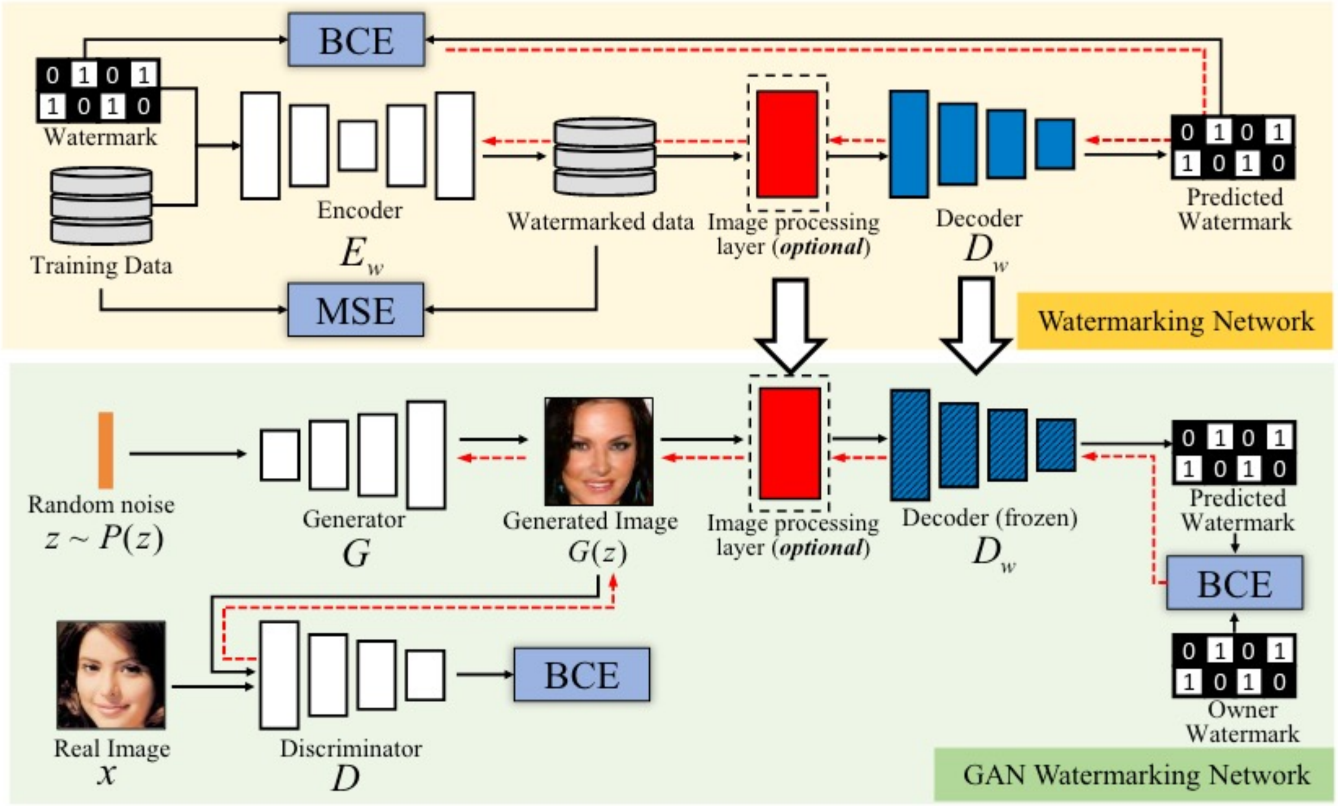


Figure from Fei et al. (2022)

SOCIETAL IMPACTS OF IMAGE GENERATION

Societal Impacts of Image Generation

Pros

- New tools for artists
- Faster creation of memes

Cons

- Copyright infringement / loss of work for artists
- Societal decrease in creativity
- Potential to create dehumanizing content
- Fake news / false realities / increased difficulty of fact checking
- Not rooted in reality
- Video generation is around the corner

https://www.bbcearth.com/flying-draco-lizard