

Introduction to Machine Learning

CS 4780 / 5780 Fall 2023

Administrative stuff...
(To get it out of the way)

Course Overview

- Instructor:
 - Wen Sun
- Homepage:
 - <https://www.cs.cornell.edu/courses/cs4780/> (links from here to what you need)
- TAs:
 - Many (more and better than you think)
- Office Hours / Recitations:
 - TA Office Hours: **Every day** (Details will be posted on course webpage.)
 - Leave Feedback: contact Wen
 - Prof. Office Hours: Every Thursday 2-3 (starting this week)
- Questions:
 - Post all questions on ED (you can make them private)
 - **Do not email directly (except in an emergency or need for privacy)**

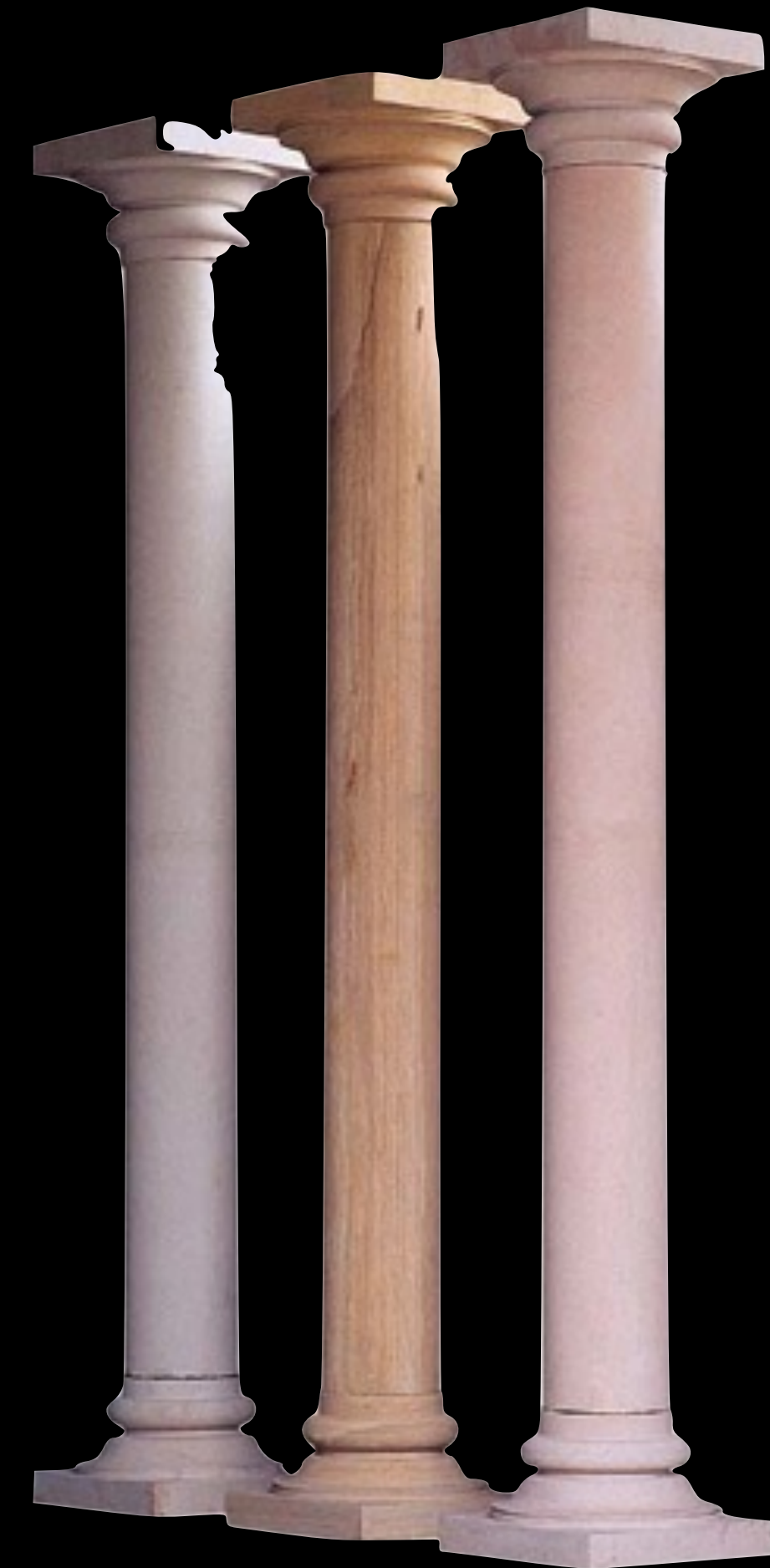
Course resources

- Primary texts
 - *Probabilistic Machine Learning: An Introduction* by K.P. Murphy
 - *The Elements of Statistical Learning* by Hastie, Tibshirani, and Friedman
- Additional texts
 - *An Introduction to Statistical Learning* by James, Witten, Hastie, and Tibshirani
 - *Patterns, Predictions, and Actions* by Hardt and Recht
 - *Fairness and Machine Learning* by Barocas, Hardt, and Narayanan
- Background and programming resources on the website



Prerequisites

- Three pillars of ML:
 - **Statistics / Probability**
 - **Linear Algebra**
 - **Multivariate Calculus**
- Should be confident in at least 1, ideally 2 or 3



Warm up quiz

- Due August 29!!!
- Available later today on Canvas
- **Not** a placement exam; it's designed to help you assess your comfort with prerequisite material
- If you find aspects of the quiz challenging/unfamiliar please use the background resources on the website
- Not part of your final grade, but you **must** complete it

Homework

- Roughly 8 assignments (subject to change)
- Due every 1-2 weeks
- Can work in groups of up to 3
- Submitted via Gradescope
- Primarily theoretical, they reinforce concepts from class and provide practice for the exams

Exams

- Midterm: October 17
- Final: TBA
- Closed book, no notes, no cheat sheet

Projects

- vocareum.com
- Enrollment details coming soon
- There will be 8 (+2) projects
- Roughly 2 weeks per project
- Unlimited submits until deadline
- Costs \$30 :-)

The screenshot shows a Vocareum Jupyter Notebook interface. The page title is "P-1 Introduction to Numpy". The notebook content is as follows:

Project -1: Introduction to NumPy

Introduction

This activity aims to introduce you to Numpy - a package for scientific computing with Python that we will use extensively in this class. This activity is by no means a complete tutorial on NumPy but it should be enough for you to do most of projects and activities in this class. For more information, please see NumPy's [official tutorial](#) and [API](#). To use NumPy, first import the package as what we do in the following cell:

```
In [ ]: import numpy as np
```

Creating Vectors and Matrices

NumPy's main object is a multidimensional array, in other words, a table of the same data type. Let's see an example on how to create a NumPy array:

```
In [ ]: X = np.array([[1,2,3], [4,5,6]])
X
```

In the cell above, we created a two dimensional table, a.k.a, a matrix of size 2×3 . To create an array, what you need to do is to pass in a list of objects into the function `np.array()`. Now that we have shown you how to create a matrix, you might have wondered how we can represent a vector in NumPy. There are three ways to represent a vector in NumPy. In the cell below we are using the function `.reshape()` to specify the length of the 2-D array in each dimension.

```
In [ ]: v1=np.array([3,4,5])
print("This is a numpy vector:{}. It's shape is {}".format(v1, v1.shape))
v2=v1.reshape((3,1))
print("This is a column vector (matrix):\n{}. It's shape is {}".format(v2, v2.shape))
v3=v1.reshape((1,3))
print("This is a row vector (matrix):{}. It's shape is {}".format(v3,v3.shape))
```

These three representation are usually not compatible. Some operations will still work, but not in the way we expect. We will always prefer the vector notation. You can transform any (matrix) vector into a numpy vector with `.flatten()`.

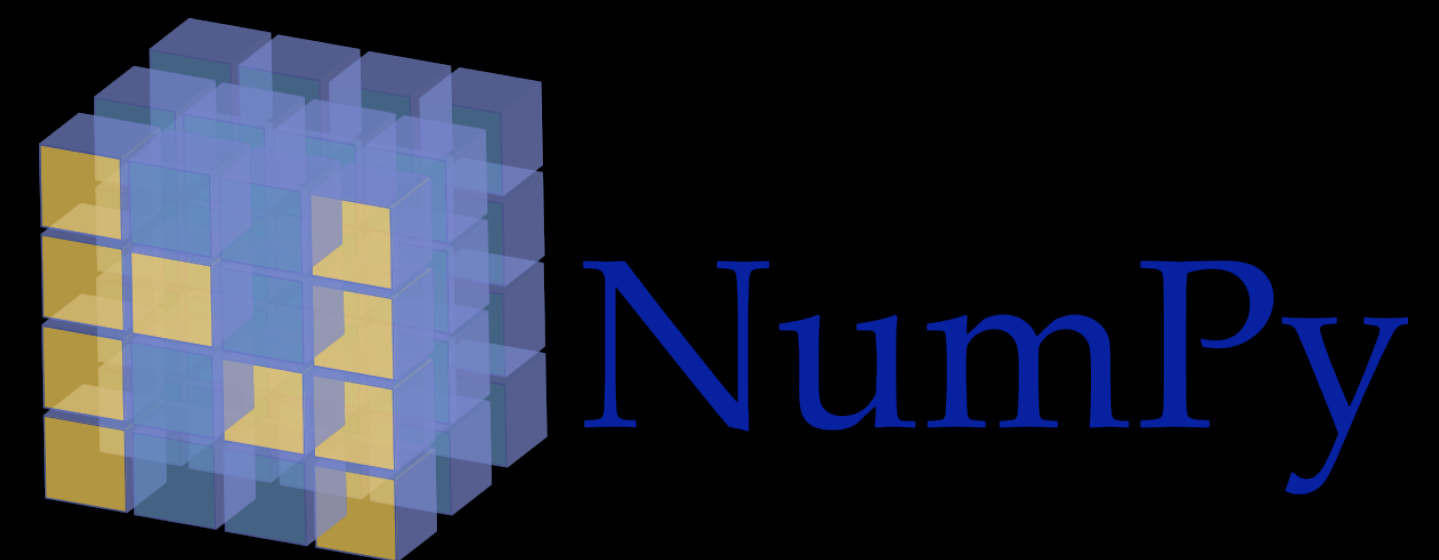
```
In [ ]: #We add v1 and v2 but the output is not as expected
v4=v2+v1
print('The sum of a column vector and numpy vector:\n{}'.format(v4))
v5 = v2.flatten() + v1
print('The expected result of summing two numpy vectors: {}'.format(v5))
```

Numpy arrays, as objects, pass by reference. This means that when you set return an array, you're returning a pointer (reference) to the real array in memory, and doing operations on the array will modify its values for every variable pointing to the array. Because of this, numpy provides a `.copy()` function, which will create a completely new array with the same values so that you can safely edit variables independently.

```
In [ ]: a = np.array([1,2,3])
b = a
c = a.copy()
print('Array a: {}'.format(a))
print('Array b (passed by reference): {}'.format(b))
print('Array c (passed by value): {}'.format(c))
a[0] = 5
print('Array a after editing a: {}'.format(a))
print('Array b after editing a: {}'.format(b))
print('Array c after editing a: {}'.format(c))
```

Warm up project (Project -1)

- NUMPY proficiency test
- Will turn into your own cheat-sheet
- Please take it seriously, this is for your own good



Vocareum and its autograder

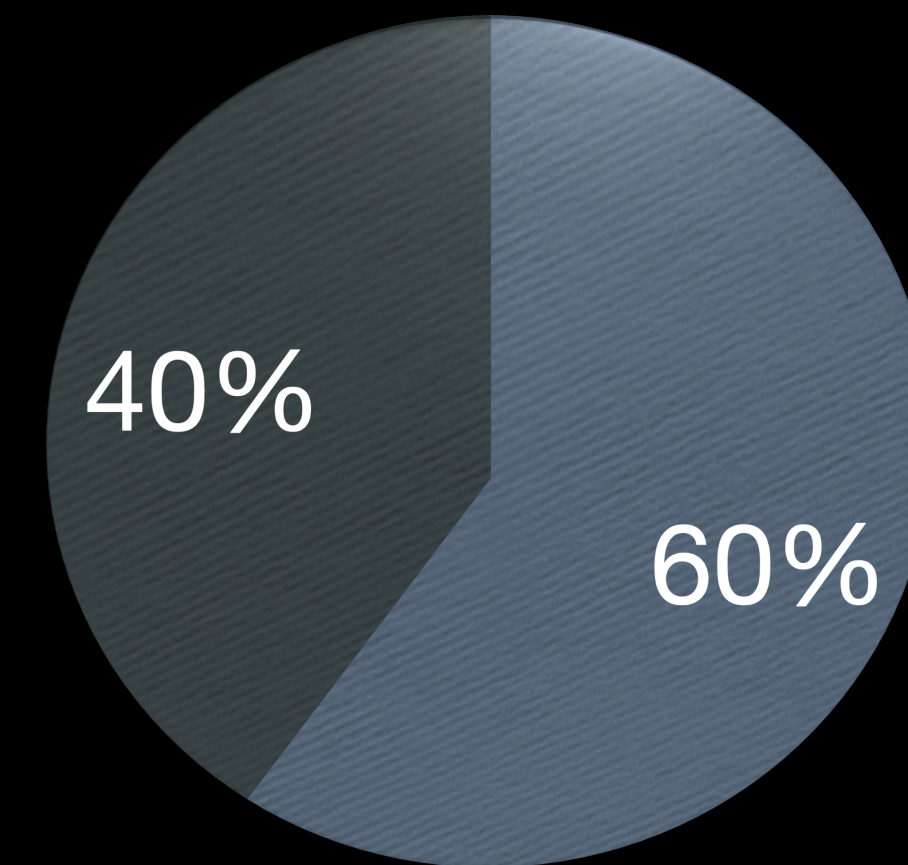
- Important notes:
 - Only text with `#<GRADED>` and `#</GRADED>` will be graded
 - **!!!!You MUST form teams before you get started!!!!!!**

For those in 5780

- Intermittent paper comprehension quizzes
- Read and answer questions on relevant ML papers
- Helps build “research comprehension” in the field
- Quizzes completed on Canvas
- Required for everyone in 5780, if you are in 4780 you can complete them if you like

Course Grade Breakdown 4780

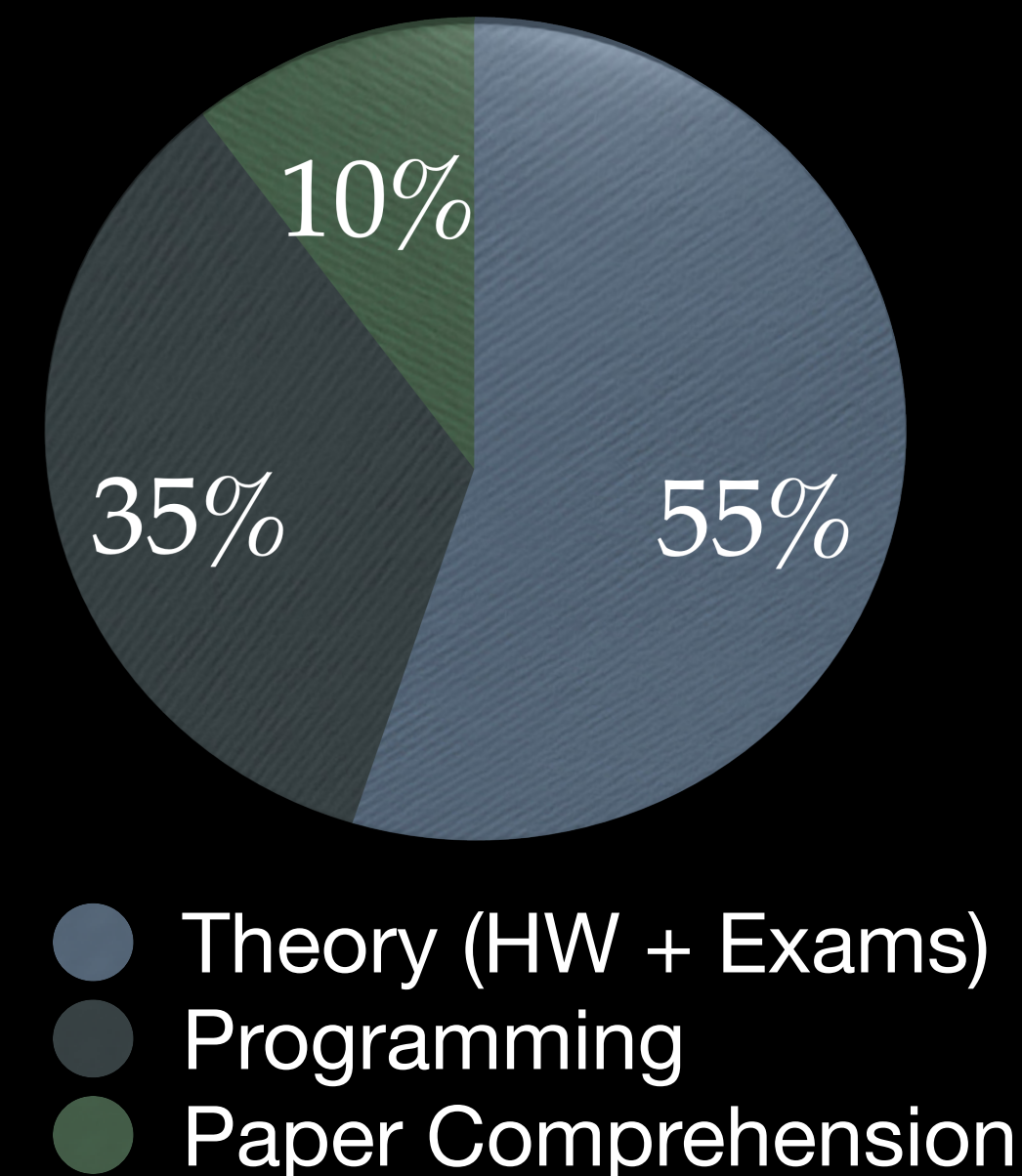
- **50% Theory: Midterm + Final**
 - Closed book
 - No cheat sheets!
 - No personal notes
- **40% Programming Assignments**
 - Up to **2** members in each team
 - **2 days extension per team per project**
 - Autograder (unlimited resubmissions)
 - *Extra credit available at times*
- **10% Homeworks**
 - Up to **3** members in each team
 - Preparation for exam



● Theory (HW + Exams) ● Programming

Course Grade Breakdown 5780

- **45% Theory: Midterm + Final**
 - Closed book
 - No cheat sheets!
 - No personal notes
- **35% Programming Assignments**
 - Up to **2** members in each team
 - **2 days extension per team per project**
 - Autograder (unlimited resubmissions)
 - *Extra credit available at times*
- **10% Paper Comprehension (mandatory)**
 - Original Research Papers in ML
 - Canvas Quizzes
- **10% Homeworks**
 - Up to **3** members in each team
 - Preparation for exam



About this course

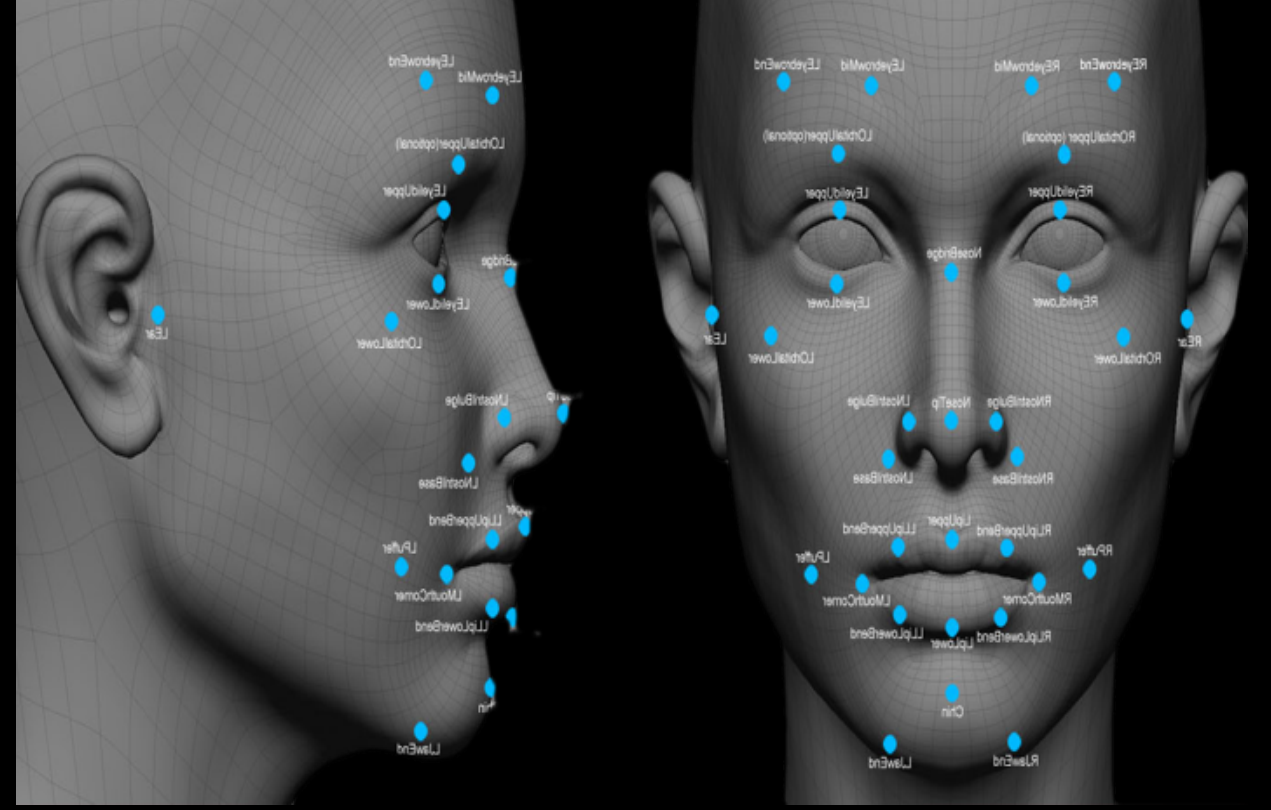
- Take this course if ...
 - you are interested in Machine Learning
 - you are comfortable with a decent amount of mathematics
 - you are not scared of programming
- Don't take this course if ...
 - matrices scare you
 - you don't remember how to take derivatives
- We discourage taking the course if you find the warm up quiz very unfamiliar and challenging
 - In that case, take appropriate prerequisites and we would love to see you in a future offering

Student comments (truth in advertising)

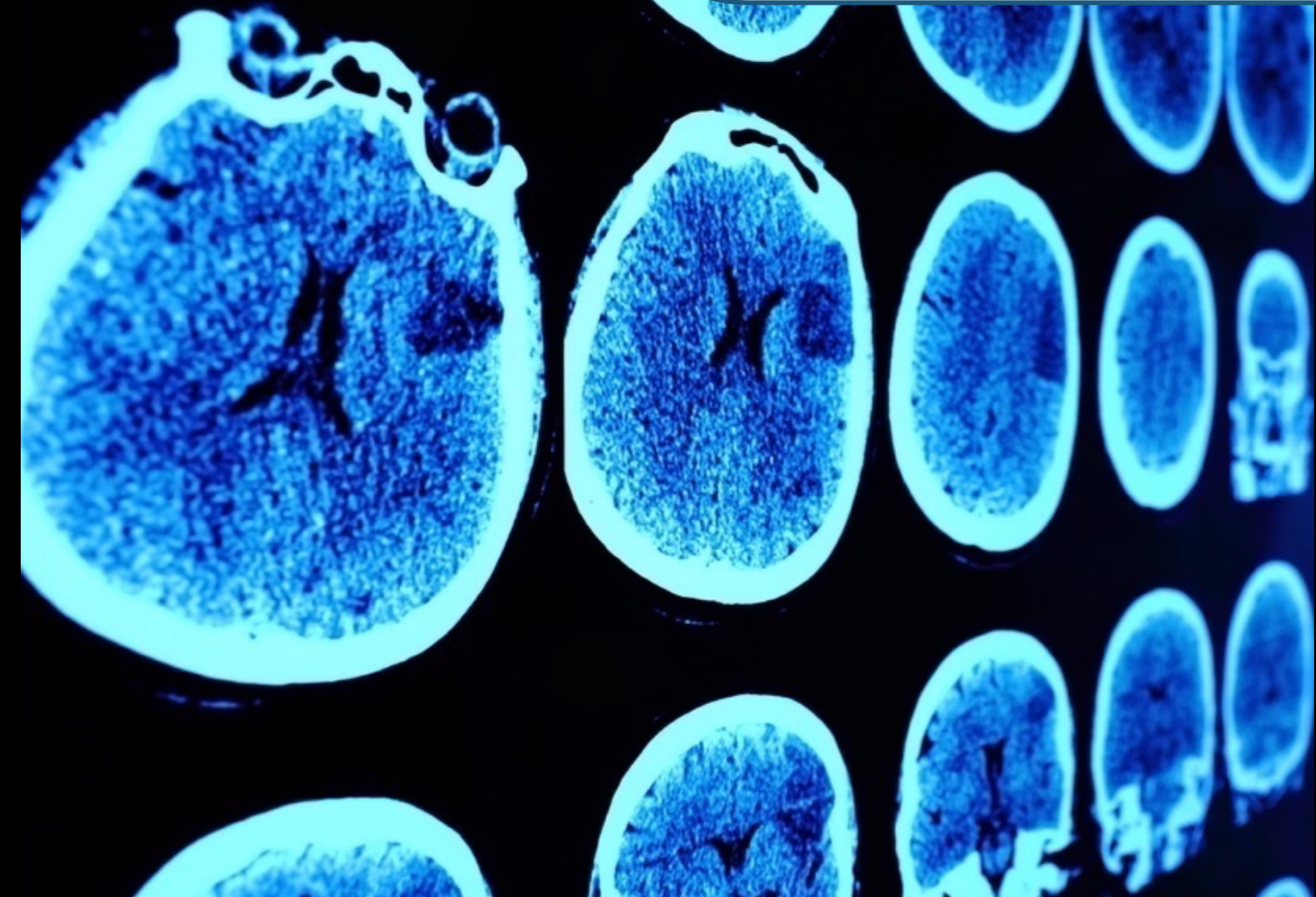
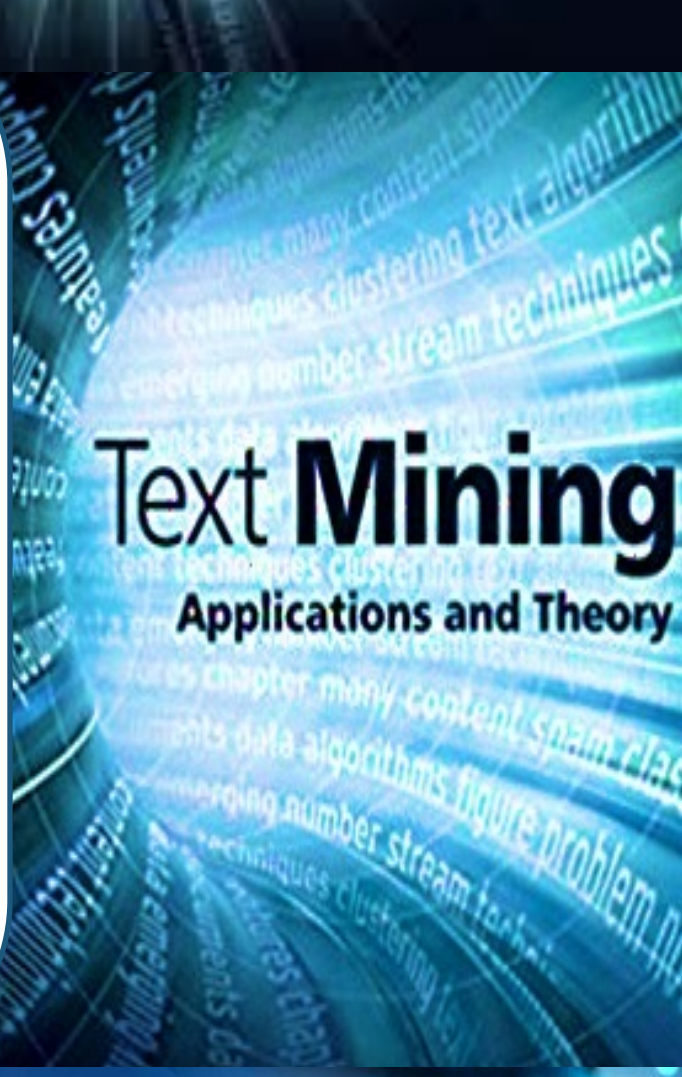
- “[...] Requires a good knowledge in math and derivatives.”
- “A TON of work, but mostly worth it for a very valuable skill.”
- “great course, but prepare to work your butt off.”
- “The topics were pretty complicated and difficult to understand quickly. I would have preferred a slightly slower pace.”
- “It's mostly a math class”

Academic Integrity

- Zero tolerance policy: all occurrences will be reported
- We **actively** look for academic conduct violations
- The autograder checks for plagiarism
- AI tool policy: feel free to use it, but need to provide references and include the details in the submissions (e.g., what prompt you used, how did you use the solutions from the tool, what's the difference between you wrote and the solution provided by the tool;)



What is Machine Learning?



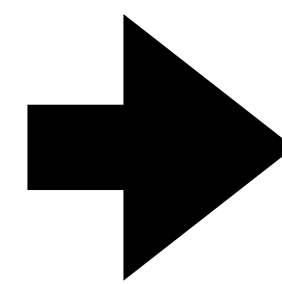
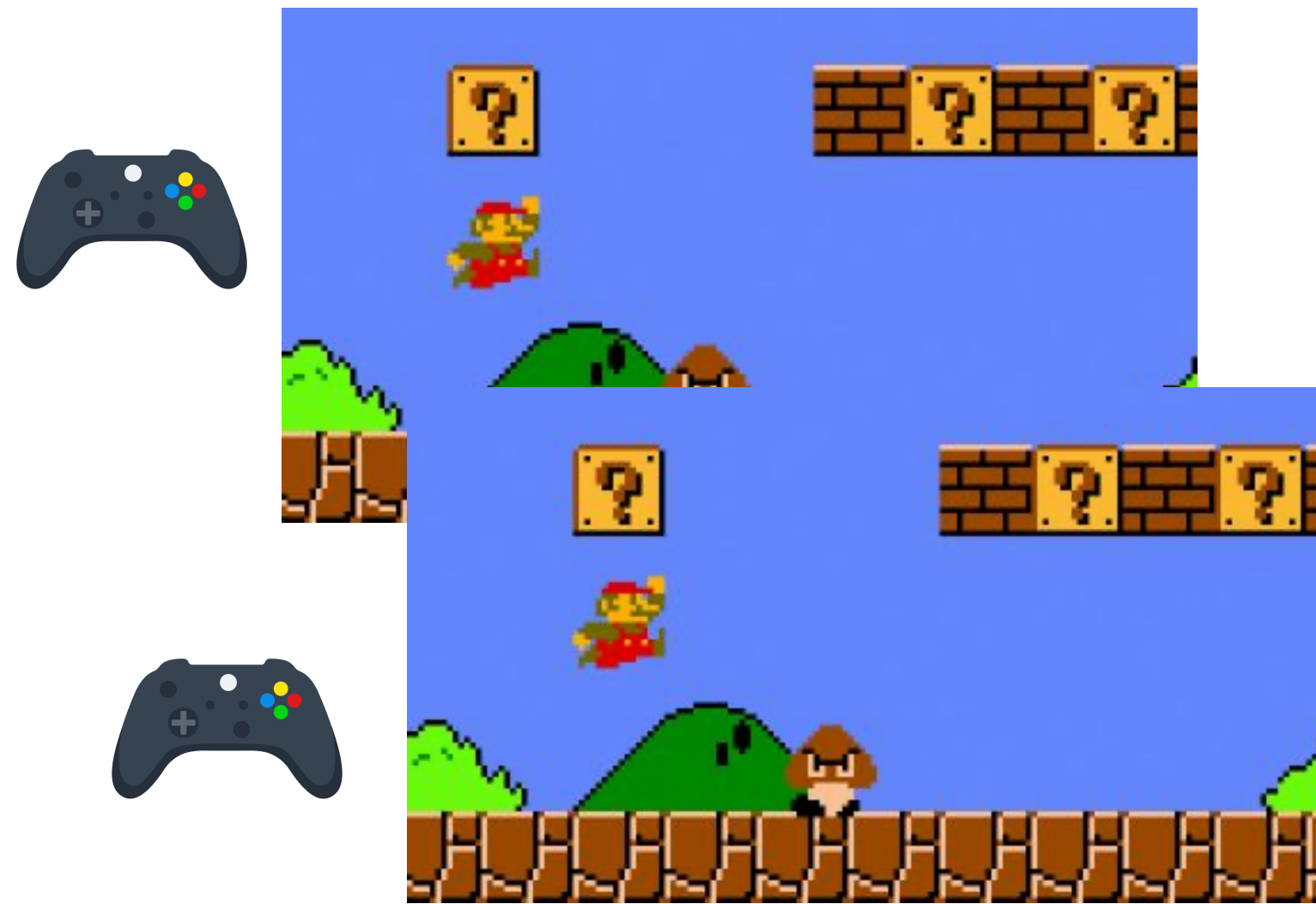
Machine Learning vs Traditional Computer Science

SUPER MARIO BROS.

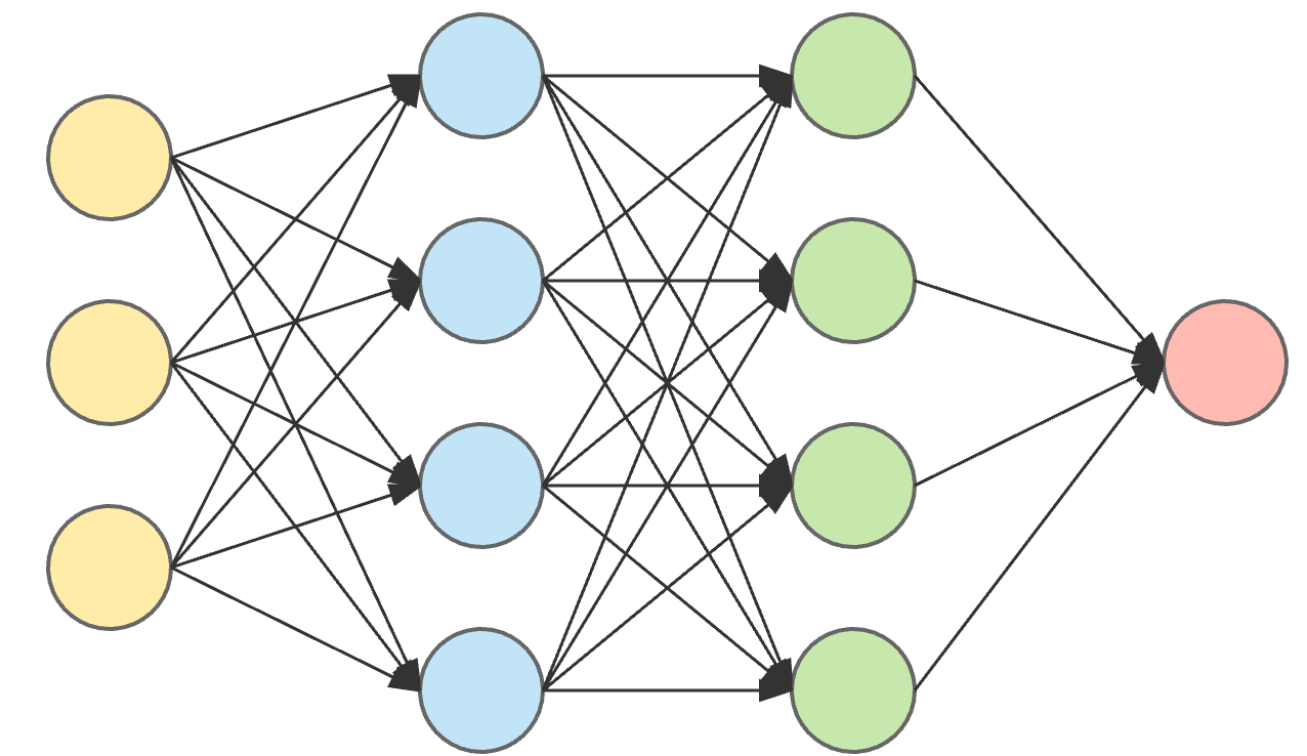
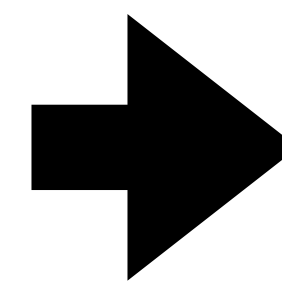


Machine Learning can make it easy

Human demonstrations



**ML
algorithm**



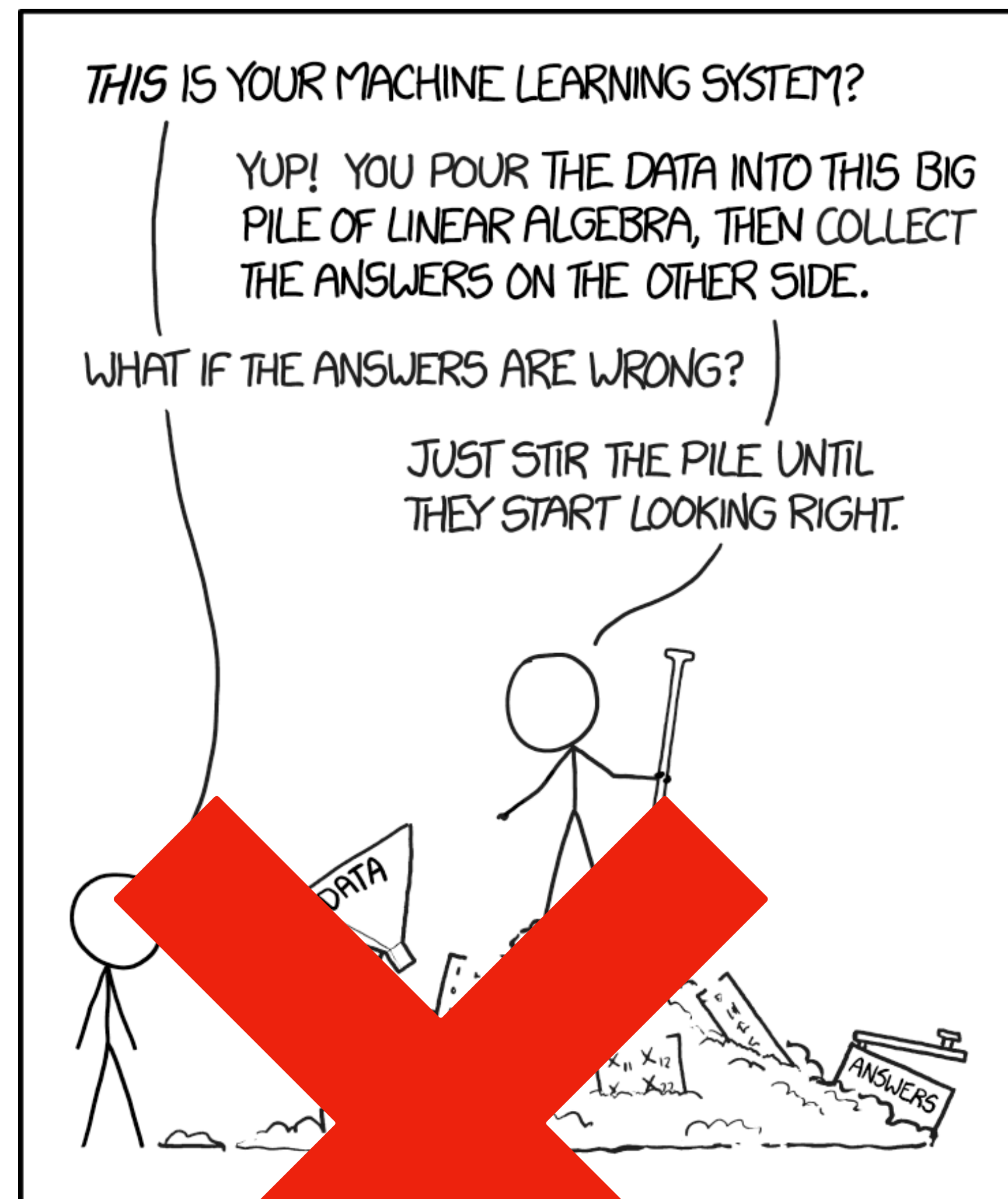
"a program" (e.g.,
a neural network)

selected Actions:

RIGHT

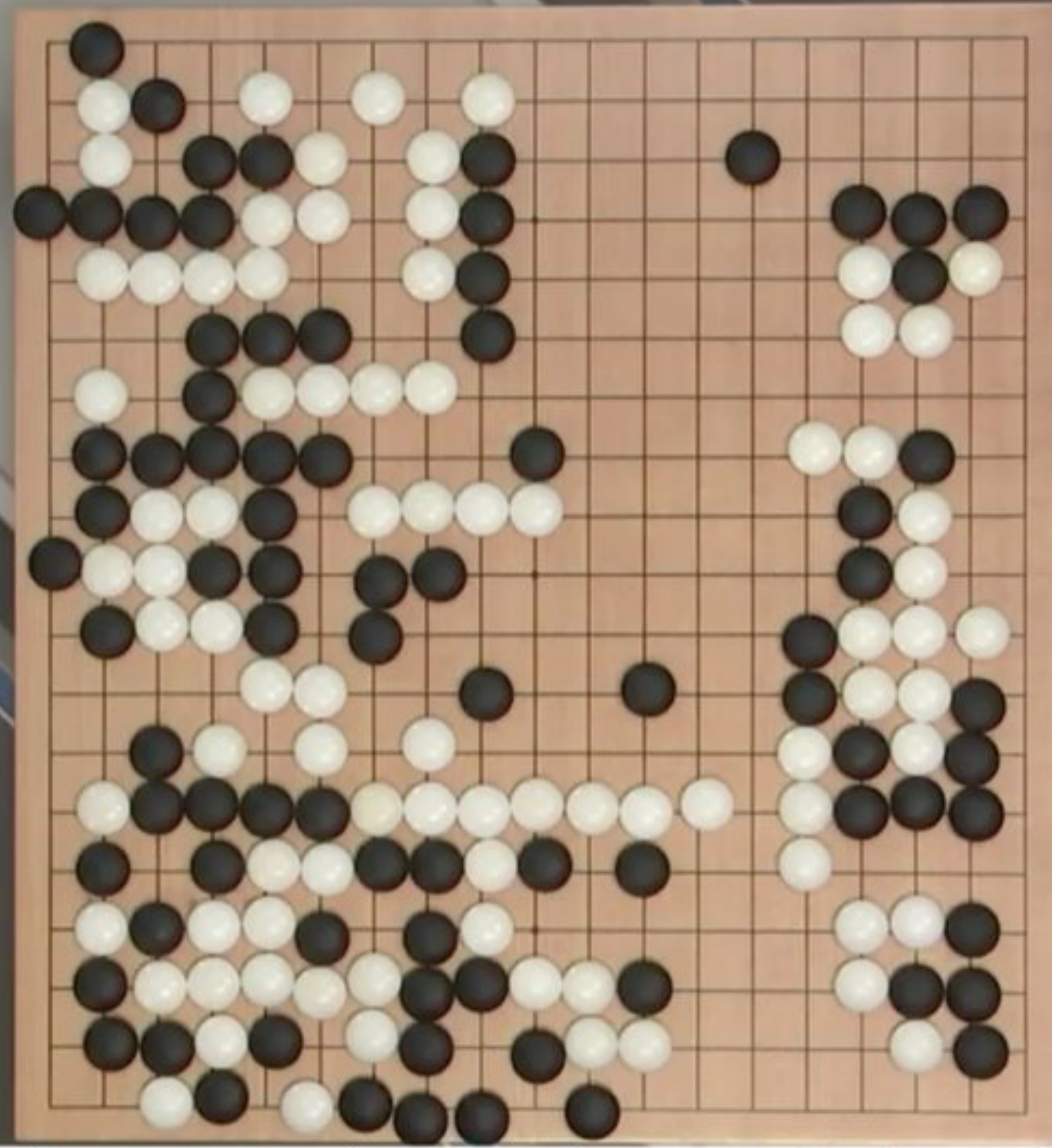
SPEED

*Machine Learning gives the computer ability to **learn from data without explicitly being programmed***



A (very brief) history of ML, started from Turing test around 1950...

● ALPHAGO
00:08:32



● LEE SEDOL
00:00:27



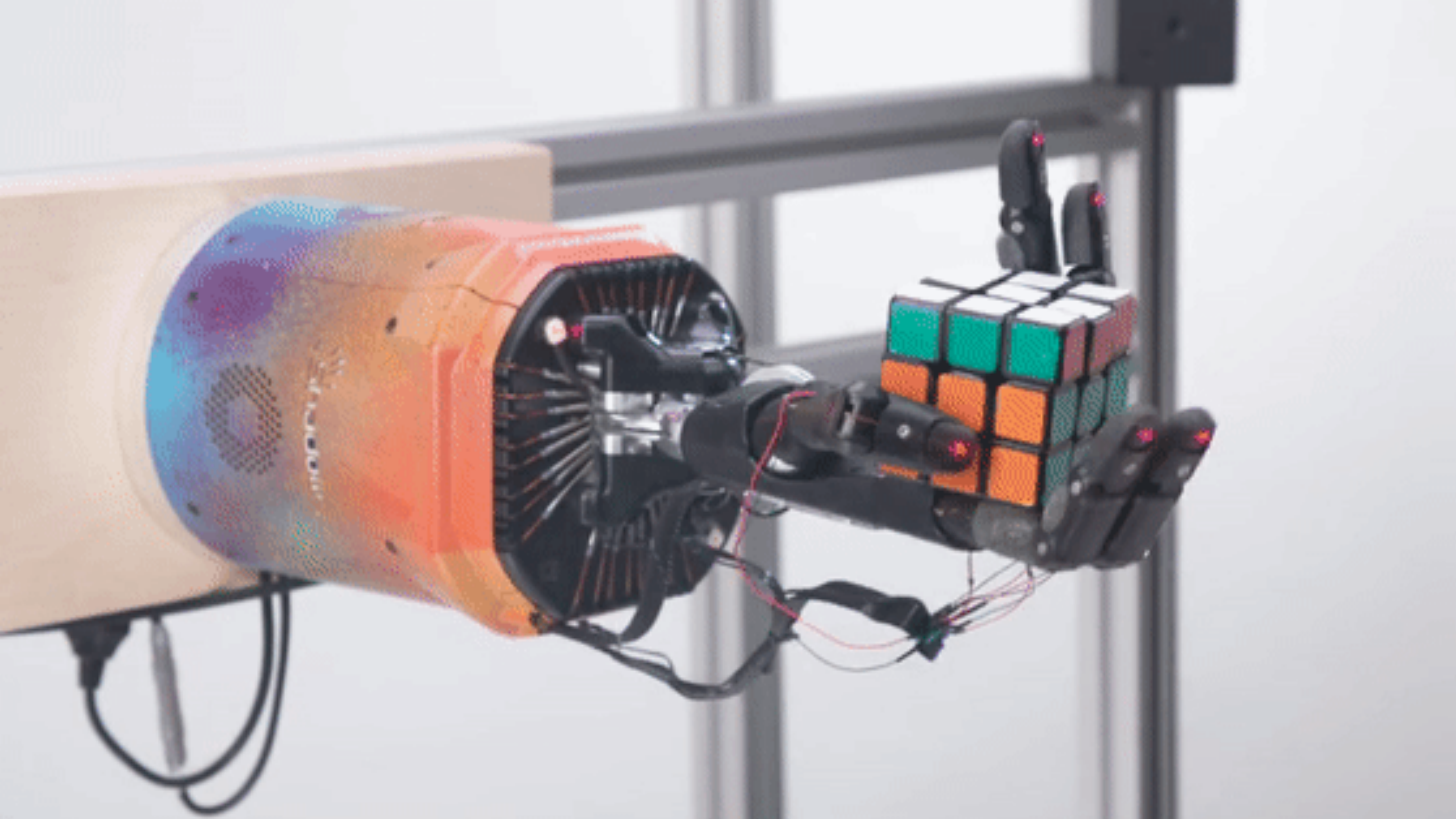


Image generation

Text prompt:
teddy bears working on new
AI research on the moon in
the 1980s



Image generation

Text prompt:
teddy bears mixing sparkling
chemicals as mad scientists in
a steampunk style



The key principle behind the Diffusion Model: Maximum likelihood Estimate + variational lower bound

$$\begin{aligned}
 L_{\text{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \\
 &= \mathbb{E}_q \left[\log \frac{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right] \\
 &= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right] \\
 &= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
 &= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \left(\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \cdot \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} \right) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
 &= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
 &= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
 &= \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \\
 &= \mathbb{E}_q \left[\underbrace{D_{\text{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]
 \end{aligned}$$

Should be easy if
you do well in this
class!

USER **this code is not working like i expect — how do i fix it?**

```
resultWorkerErr := make(chan error)
defer close(resultWorkerErr)
go func() {
    defer cancel()
    resultWorkerErr <- b.resultWorker(ctx)
}()

err := b.worker(ctx)
cancel()
if err == nil {
    return <-resultWorkerErr
}
return multierror.Append(err, <-resultWorkerErr)
```

CHATGPT It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?

USER **just part of it — the error never surfaces. I think there is something wrong with the channel**

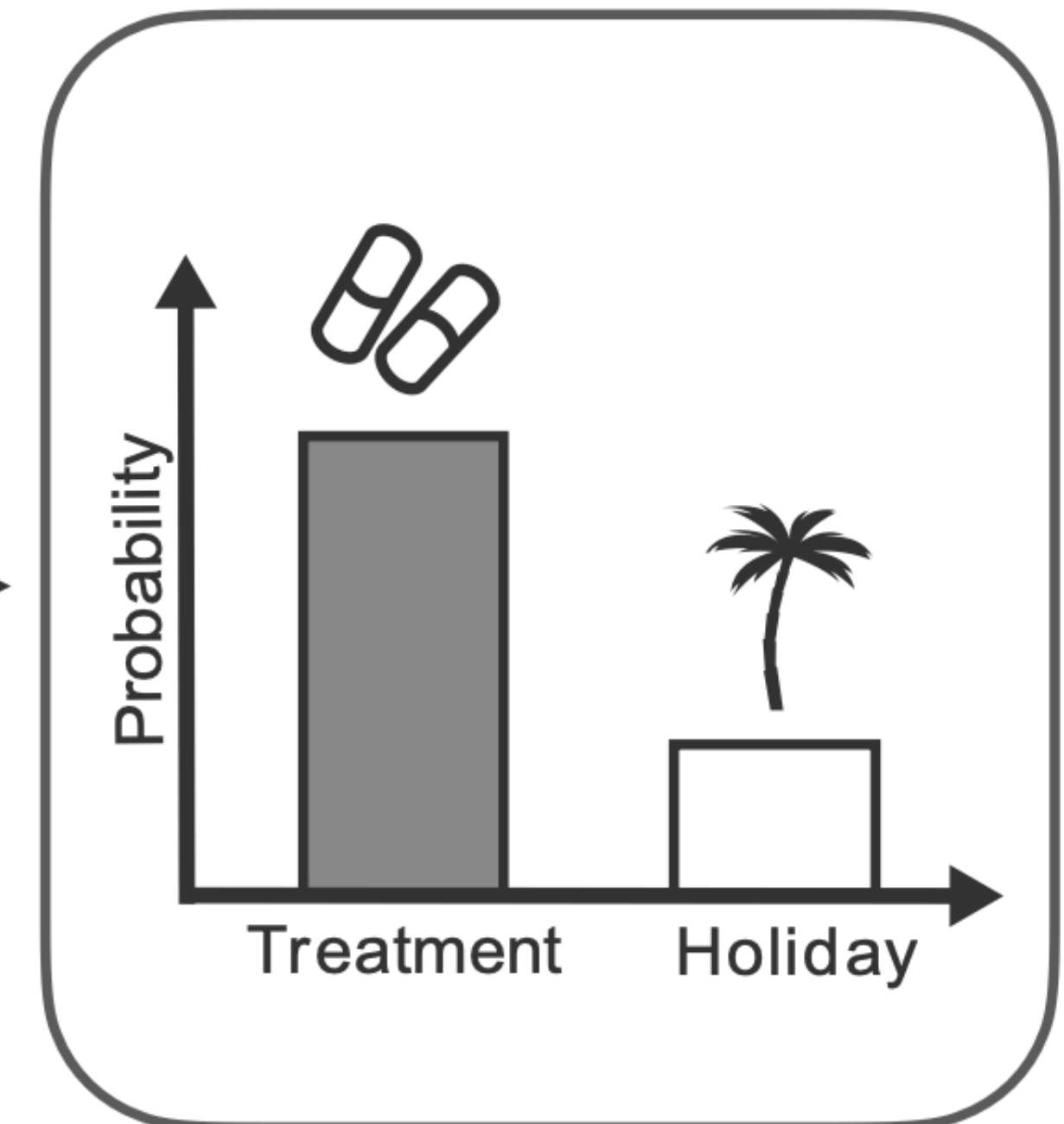
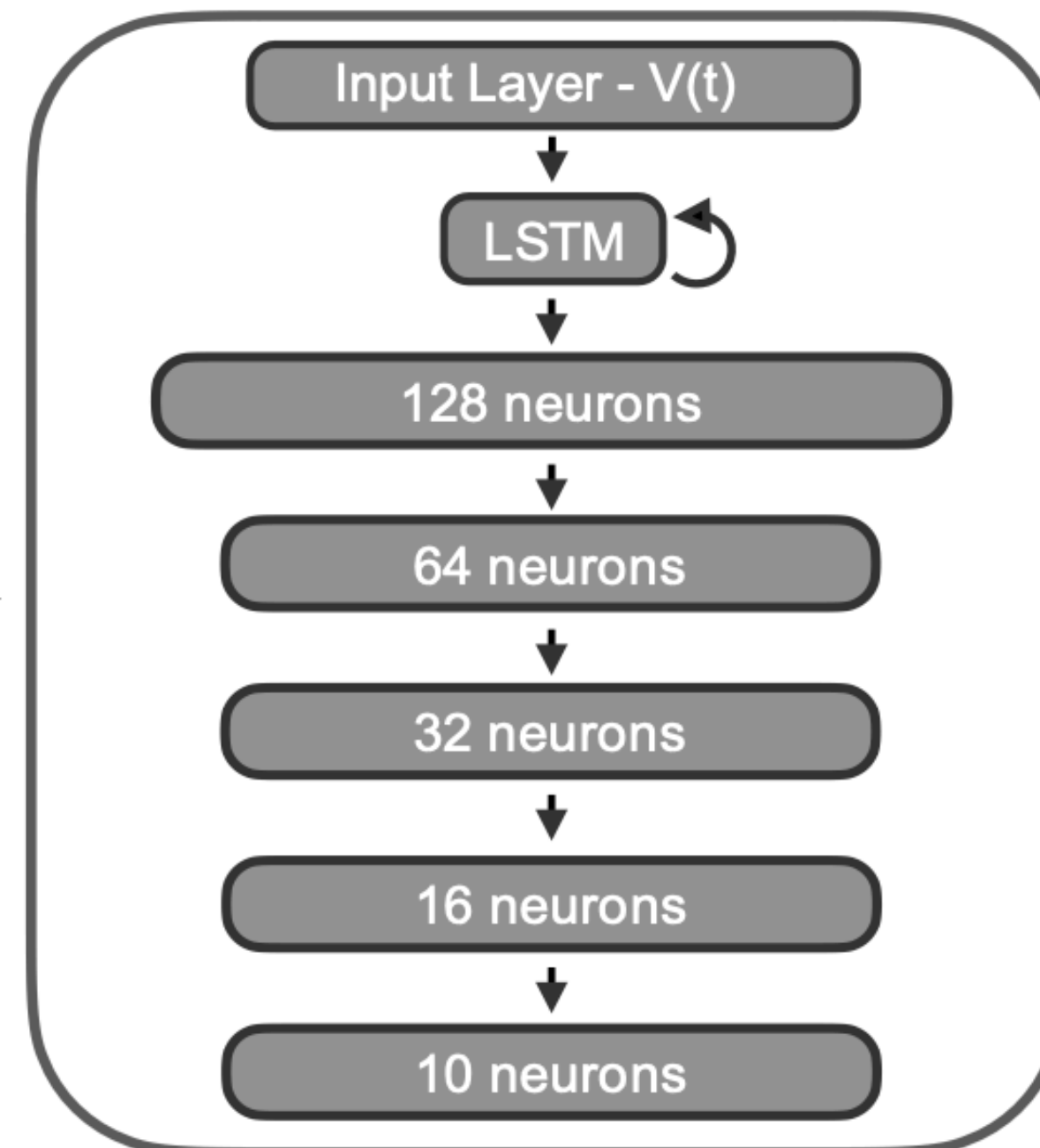
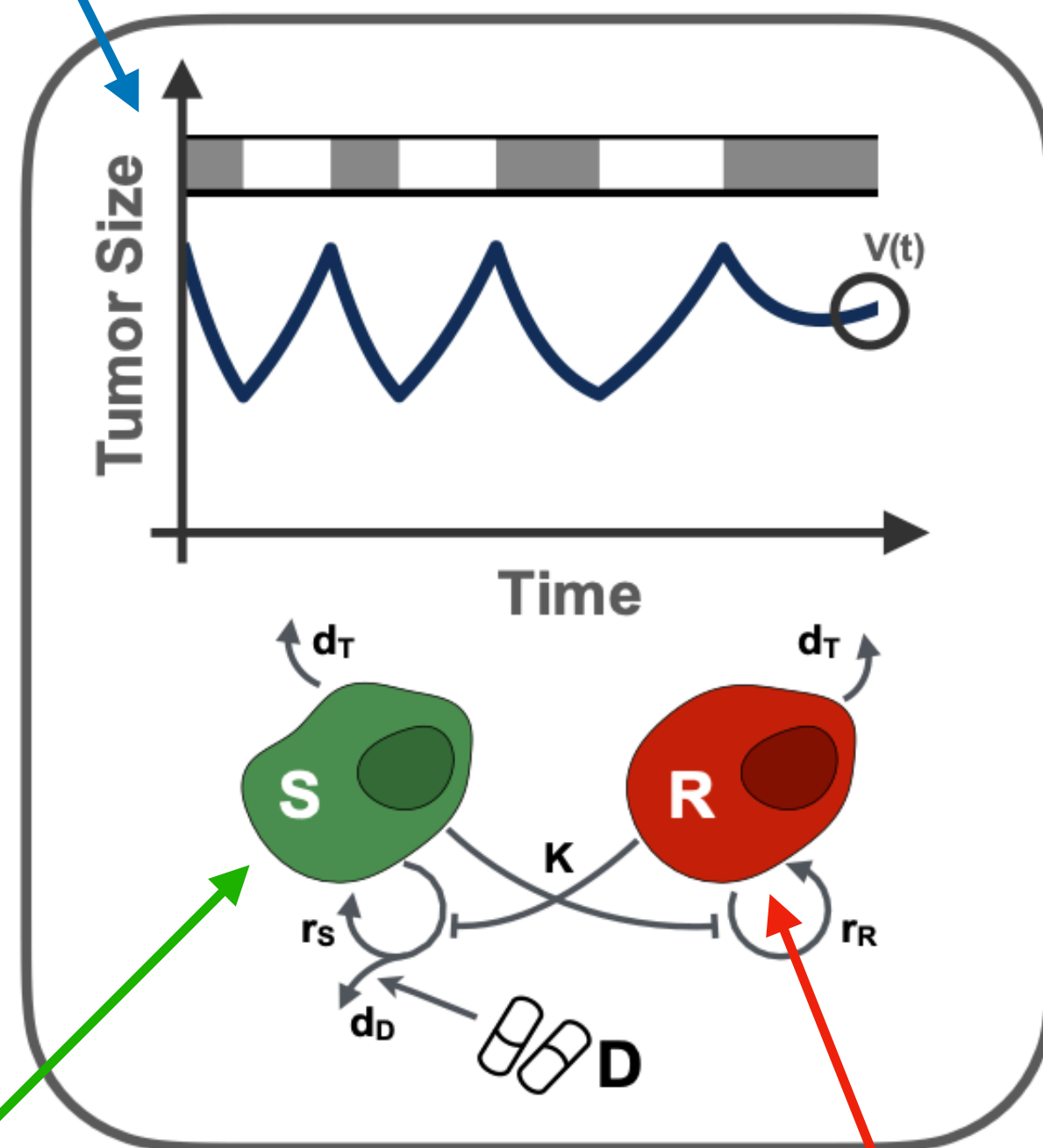
CHATGPT It's difficult to say without more information about what the code is supposed to do and what's happening when it's executed. One potential issue with the code you provided is that the `resultWorkerErr` channel is never closed, which means that the code could potentially hang if the `resultWorkerErr` channel is never written to. This

**Large Language Models,
e.g., ChatGPT**

Machine Learning for adaptive cancer therapy

Using ML model to recommend when to give treatment based on the patient's status

History of the total tumor size



Prediction of whether to treat given the patient's status

Sensitive cells

Resistant cells

DANGER



Will AI / ML take over the world?

This is an ongoing debate between many ML scientists these days!

IDEAS MADE TO MATTER | ARTIFICIAL INTELLIGENCE

Why neural net pioneer Geoffrey Hinton is sounding the alarm on AI

by Sara Brown | May 23, 2023

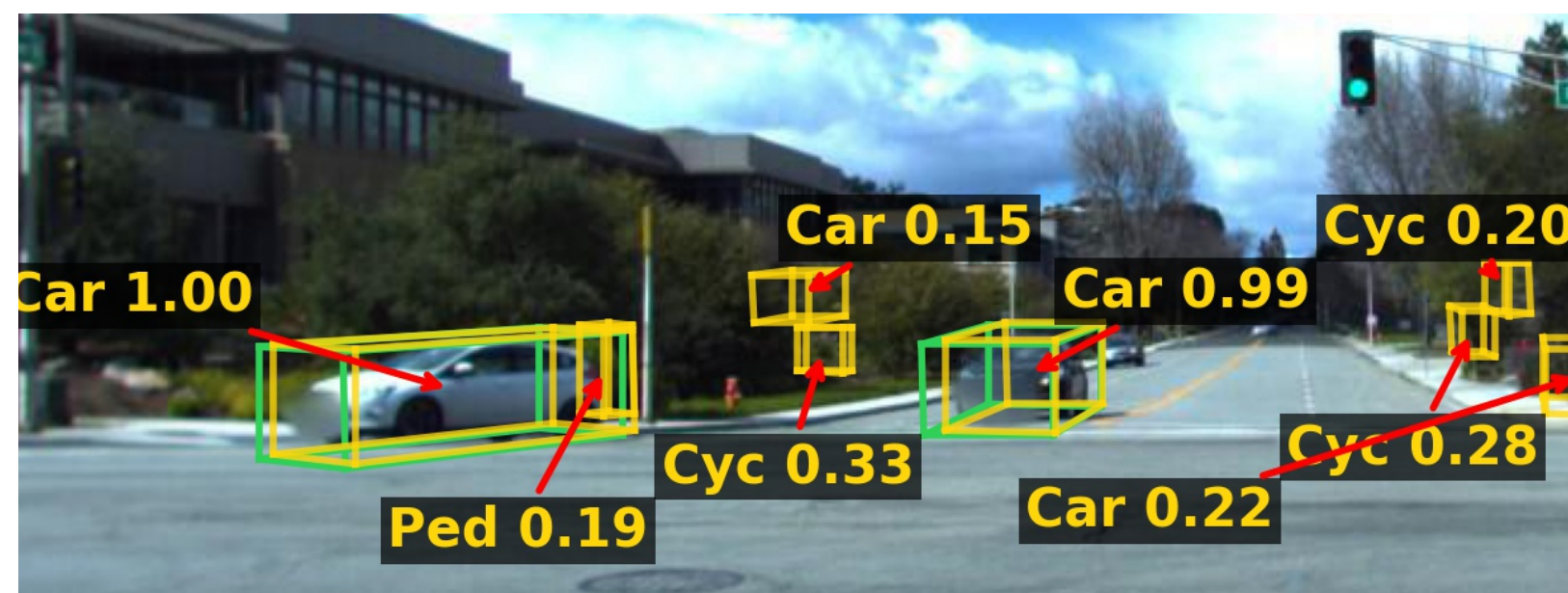
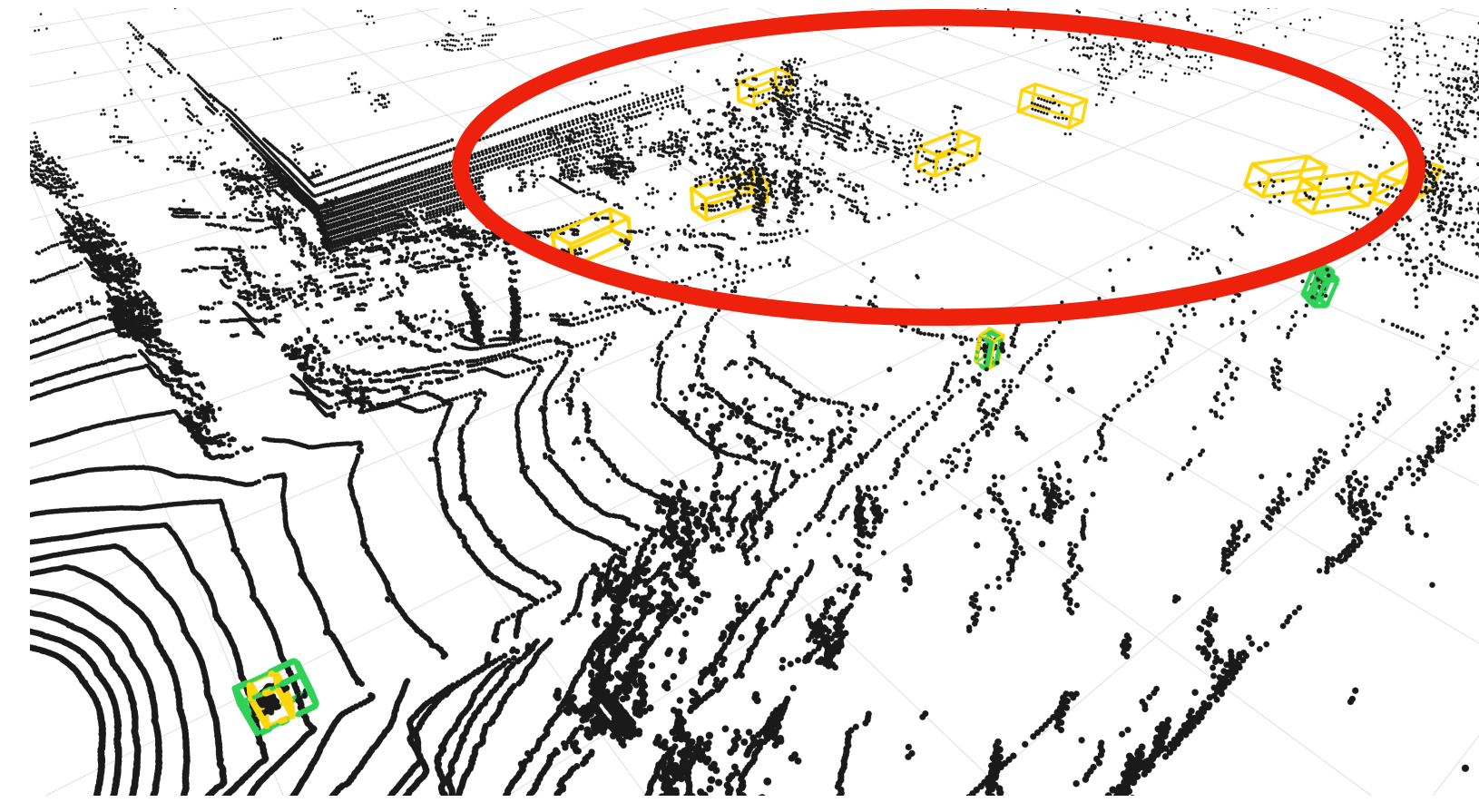
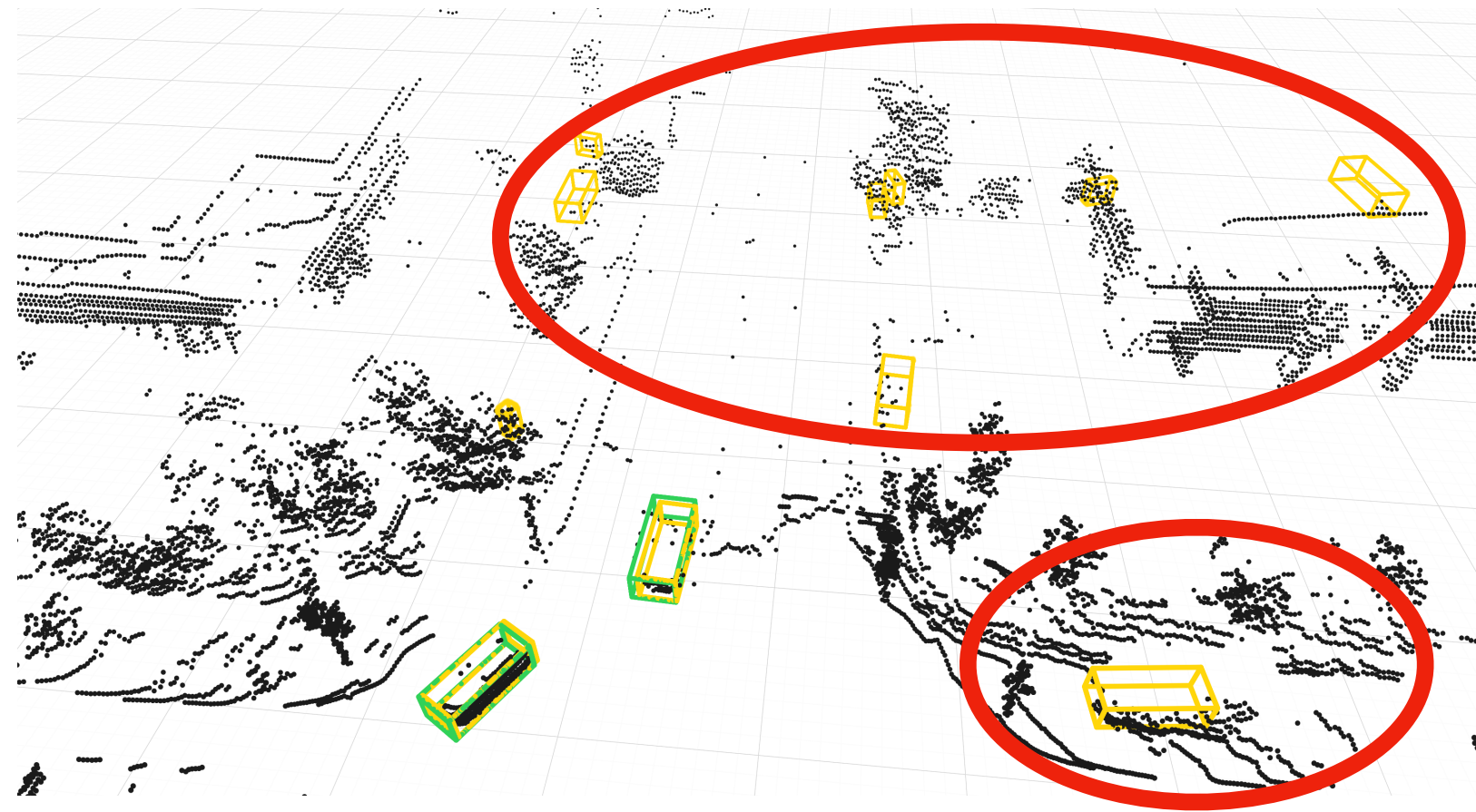


Andrew Ng ✓
@AndrewYNg

Do Large Language Models really "understand" the world, or just give the appearance of understanding? Evidence (e.g., Othello-GPT) shows LLMs build models of how the world works, which makes me comfortable saying they do understand. More in The Batch:

It's time to confront the existenti

ML powered AI systems sometimes make stupid mistakes



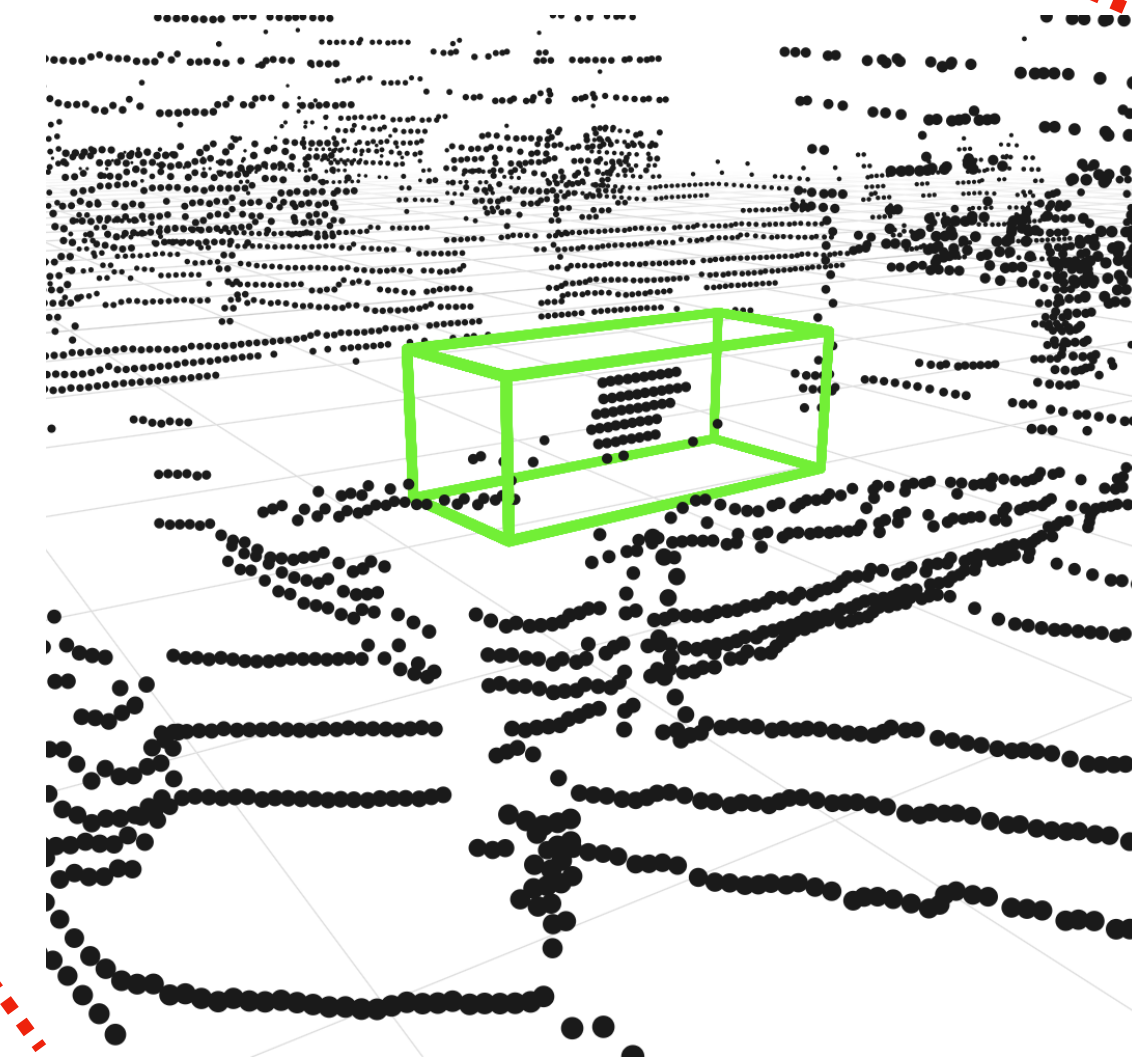
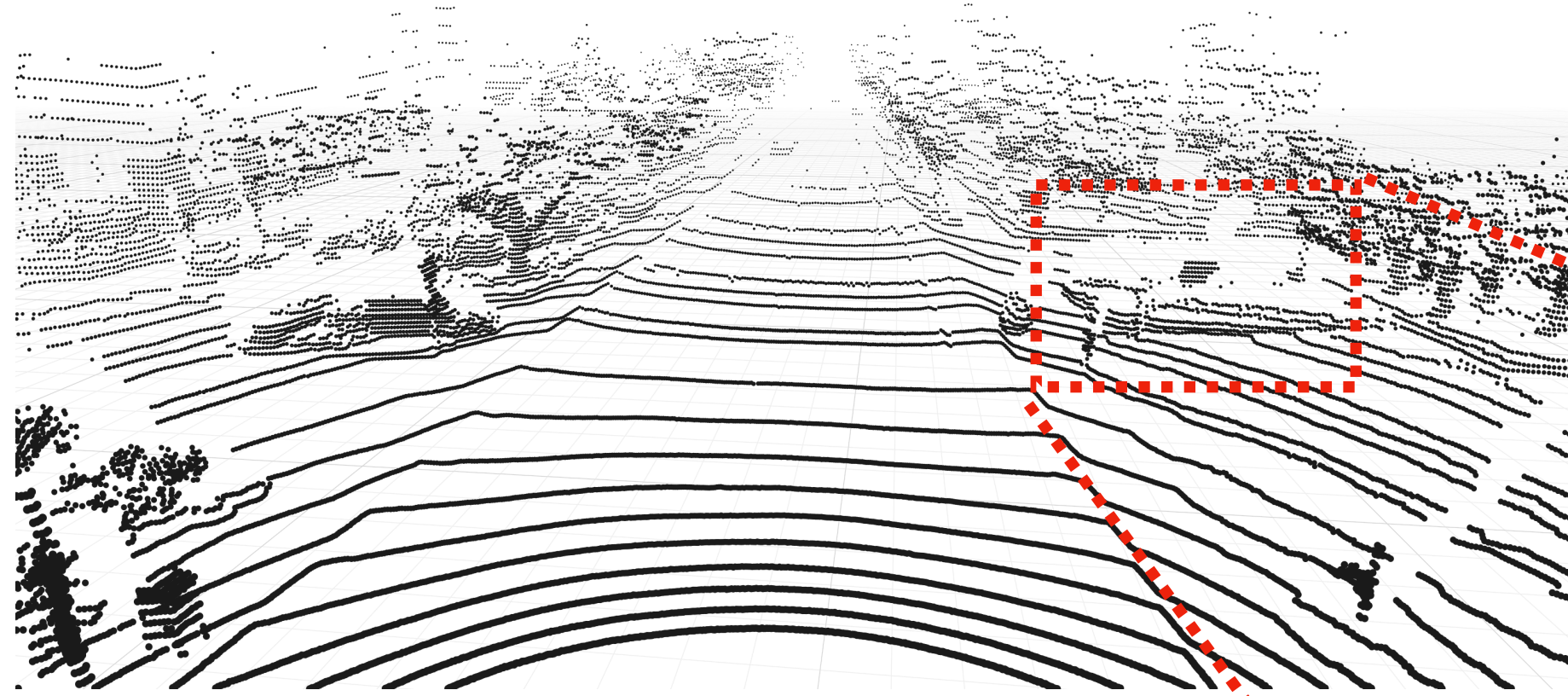
Example Scene 1



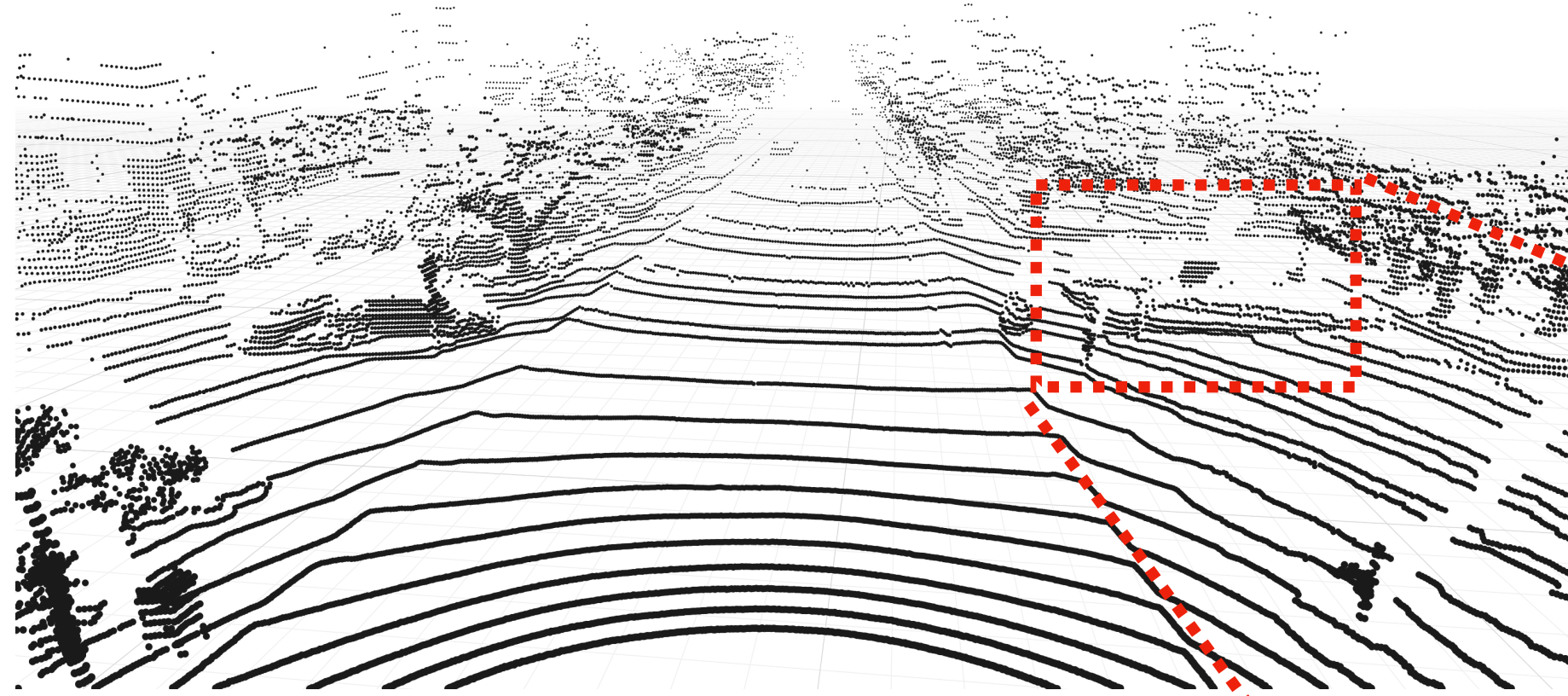
Example Scene 2

Detection results from PointRCNN [Shi, et al., CVPR19]
LiDAR only detection (image shown here for visualization only)

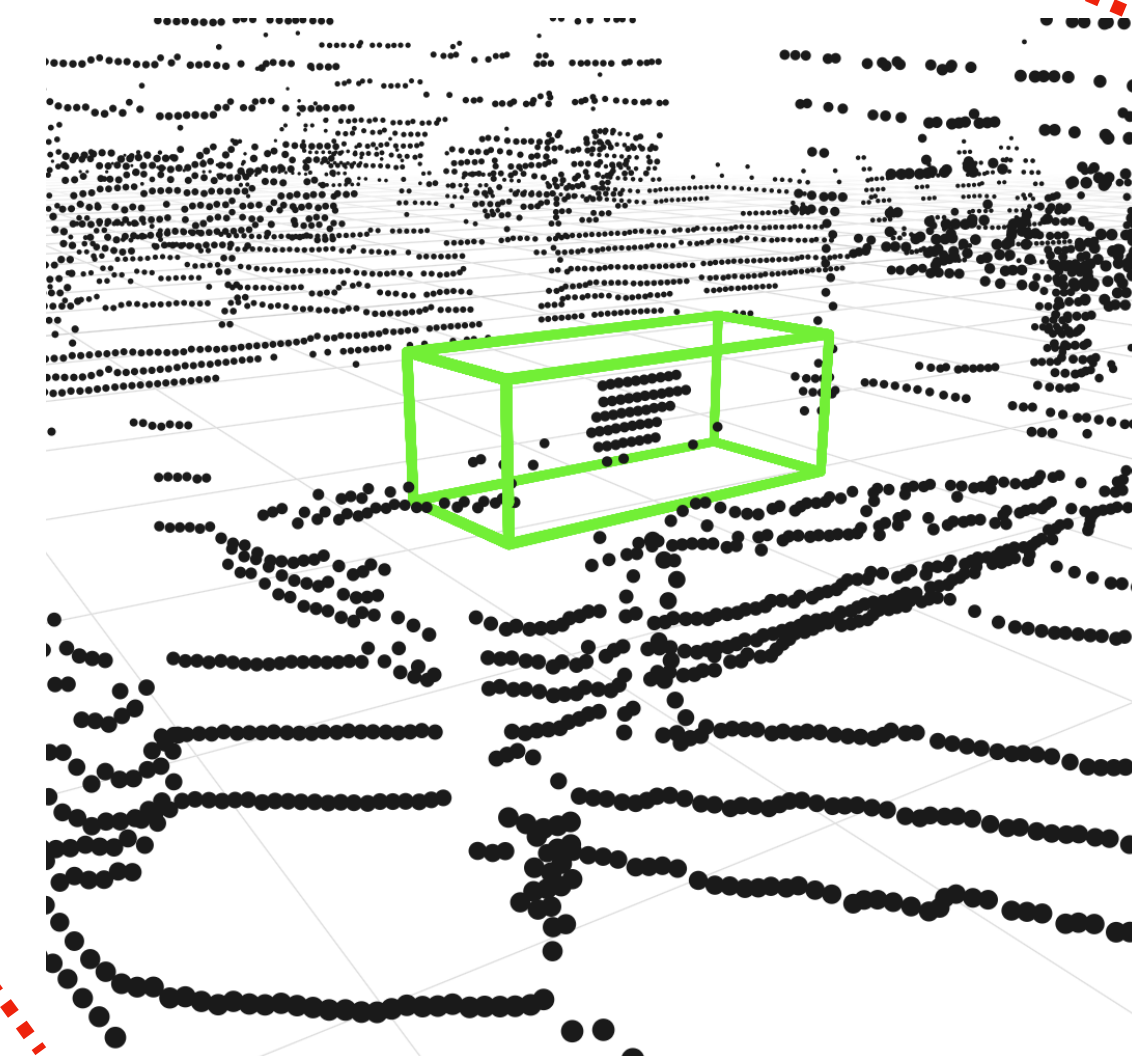
Yellow: detection, Green: ground-truth



Our self-driving car vision system does not have the ability to reason and understand scenes!



Actually, it is just a static sign.



ML algorithms can fail to learn true causation

Example: using ML to learn when to brake from human driving data



ML algorithm thinks: *brake light on* => *brake*

ML algorithms is good at learning
spurious correlations;

Correlations \neq Causation

Even the powerful GPT4 cannot do reasonings!

Let's (1) play a tic-tac-toe game w/ GPT4, (2) look at the convex question again

Will AI / ML take over the world?

Good news:

AI seems still nowhere near to general Intelligence

Bad news:

AI doesn't have to be smarter than us to be harmful

Biases and fairness issues in ML

The Best Algorithms Struggle to Recognize Black Faces Equally

Google's algorithm shows prestigious job ads to men, but not to women. Here's why that should worry you.

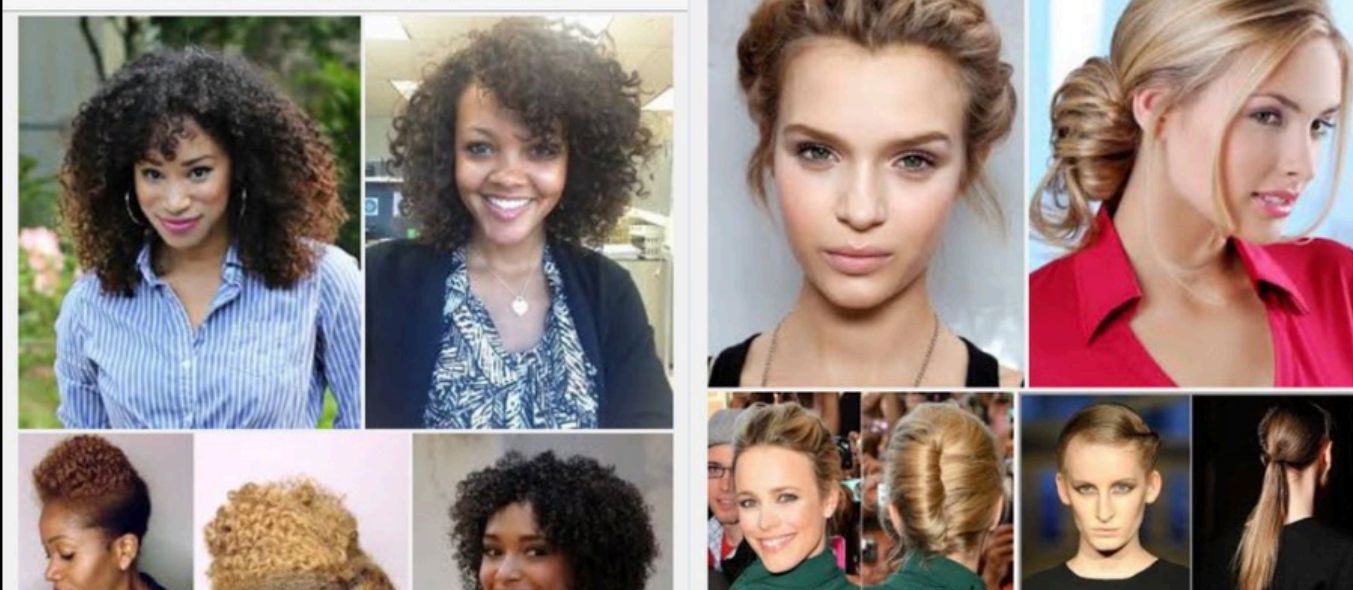
Gender and racial bias found in Amazon's facial recognition technology (again)

How Amazon Accidentally Invented a Sexist Hiring Algorithm

A company experiment to use artificial intelligence in hiring inadvertently favored male candidates.

Do Google's 'unprofessional hair' results show it is racist?

unprofessional hairstyles for work



When an Algorithm Helps Send You to Prison

By Ellora Thadaney Israni



ML-based AI systems can generate fake information

TECH · DEEPFAKES

These deepfake videos of [redacted]
viral

TECHNOLOGY

How real is the threat of AI deepfakes in the 2024 election?

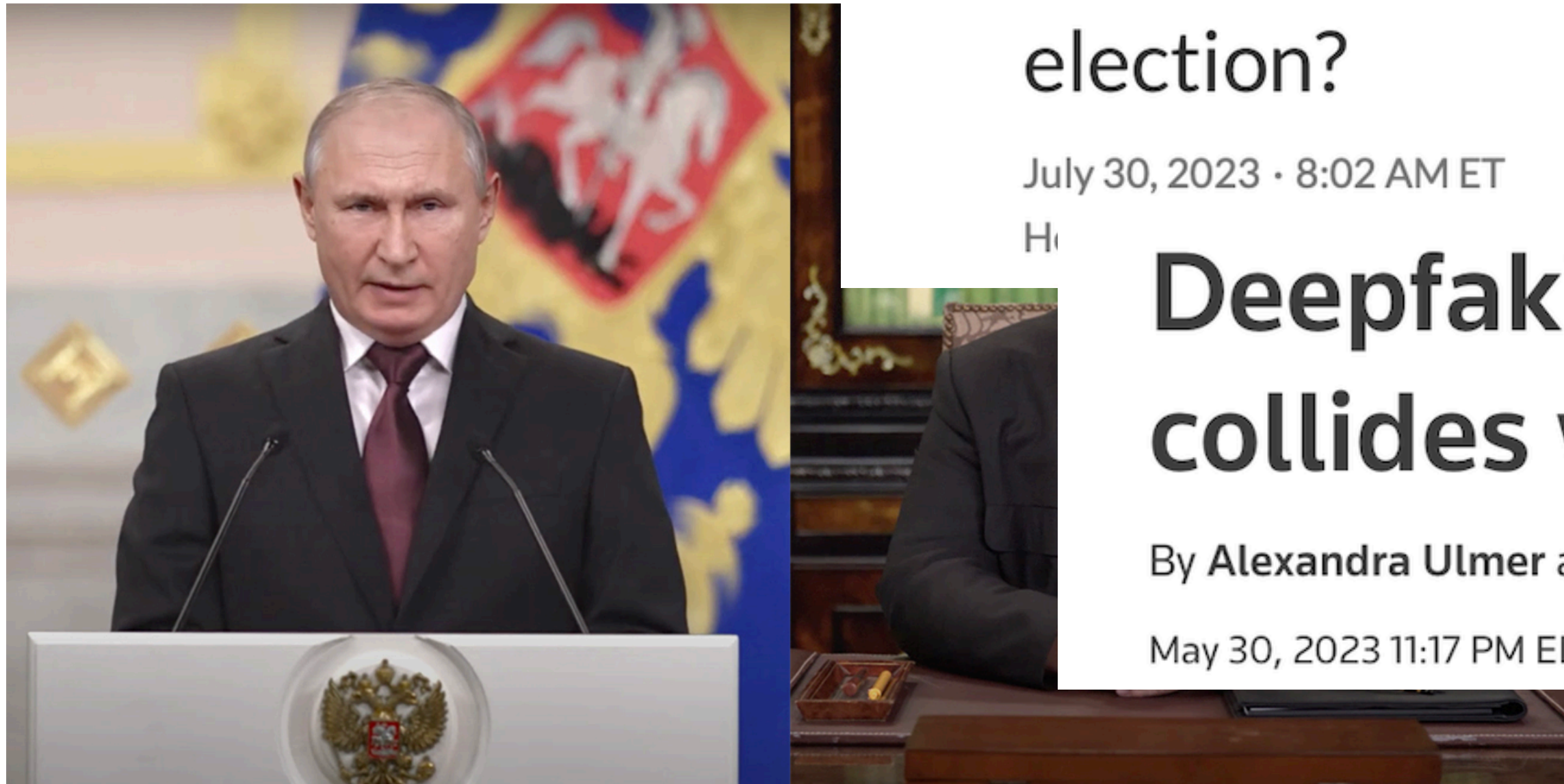
July 30, 2023 · 8:02 AM ET

H

Deepfaking it: America's 2024 election collides with AI boom

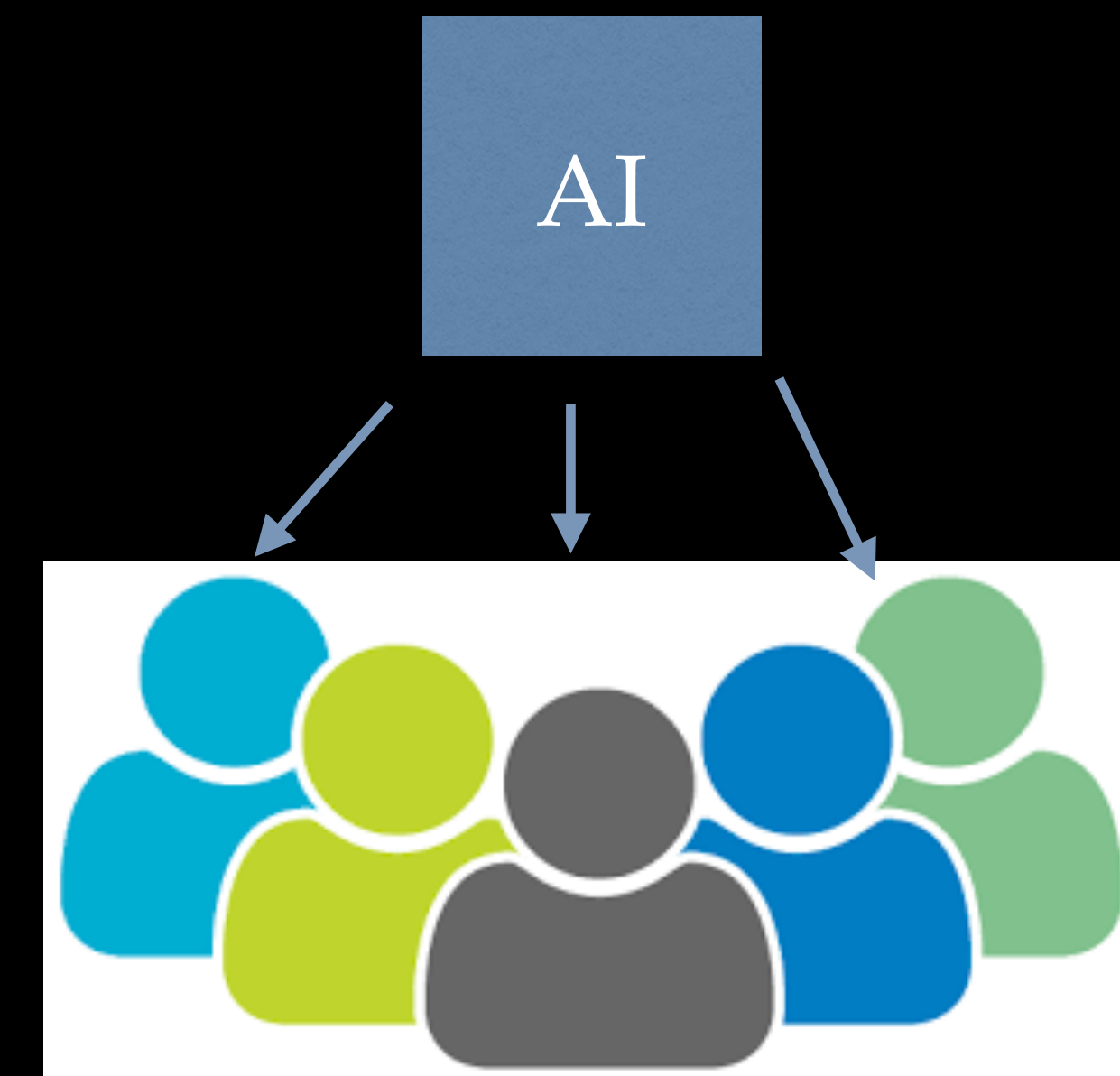
By Alexandra Ulmer and Anna Tong

May 30, 2023 11:17 PM EDT · Updated 2 months ago



ML-powered online advertising

- Advertisers pay AI companies to induce change in people's behaviors
 - AI “learns” your preferences and display things that you are interested in
 - Leads to fast clicks/ prolonged engagement (more advertising time)
 - Causes (social) anxiety, fear, undesired behavior, elevates misinformation



Summary

*Machine Learning gives the computer ability to **learn from data without explicitly being programmed***

ML is powerful, but it can be harmful. Be mindful about where and how to use this powerful technology!