# Trustworthy Deep Learning

CS294-131

Instructors: Darrell, Song, Steinhardt

# What Is This Course?

Seminar on **technical problems** intersecting **social aspects** of ML

Intended for students with ML background

Series of guest lectures: intro to topic + technical details

**Come with questions!**

# Prerequisites

**Assume basic knowledge in deep learning:**

- Part I & II in Deep Learning Book (Goodfellow et al.)
- CS294-129 or CS231n (Stanford class)

# Prerequisites

**Assume basic knowledge in deep learning:**

- Part I & II in Deep Learning Book (Goodfellow et al.)
- CS294-129 or CS231n (Stanford class)

**Undergrads:**

- Have taken AI/ML courses e.g. CS188/9, CS294-129, CS294-112
- Or have done some deep learning research/projects

# Prerequisites

**Assume basic knowledge in deep learning:**

- Part I & II in Deep Learning Book (Goodfellow et al.)
- CS294-129 or CS231n (Stanford class)

**Undergrads:**

- Have taken AI/ML courses e.g. CS188/9, CS294-129, CS294-112
- Or have done some deep learning research/projects

Contact course staff if you have questions!

# Course Staff

## Instructors:

Trevor Darrell

Dawn Song

Jacob Steinhardt

## TA:

Samaneh Azadi

# Speaker List

- Nicolas Papernot (Google Brain)
- Justin Gilmer (Google Brain)
- Stefan Wager (Stanford)
- Guillaume Bouchard (Facebook)
- Zachary Lipton (CMU)
- Been Kim (Google Brain)
- Balaji Lakshminarayanan (DeepMind)
- And more! (Send suggestions)

# Today

Logistics

Overview of Topics

Some Basic Frameworks

# Today

**Logistics**

Overview of Topics

Some Basic Frameworks

# Format

Weekly lecture

- Main lecture (60 minutes)
- In-depth discussion (20 minutes)

Weekly reading, discussion questions

Projects

# Weekly Readings

Research papers on week's topic

- main reading vs. background reading

# Discussion Questions

Questions due **Sunday at noon** (through Piazza)

Think about what you would like to learn from the speaker

Use these questions to spark discussion in class

# Projects

3 options:

- Distill-type Literature Review (cf. https://distill.pub)
- Reimplementation + open source (cf. ICLR repro challenge)
- **Conference quality** research project

Groups of 2-3 people

- Talk to instructors if need to form group of 1 or 4 people

# Project Schedule

- **2/25:** Project proposal due
- **4/1:** Project milestone report due
- **4/29:** Poster Presentation
- **5/6:** (tentative) Final project report due

# Grading and Variable Units

- 20% Class Participation
- 25% Weekly Reading Assignment
- 55% Project
- 1 unit for readings, 2 units for readings+project

# Grading and Variable Units

- 20% Class Participation
- 25% Weekly Reading Assignment
- 55% Project
- 1 unit for readings, 2 units for readings+project

**If you've taken the class for $> 2$ units, please consider changing to 1 or 2 units!**

# Next steps

- Join Piazza: https://piazza.com/class/joy4z1cunad9h
- Reading for next lecture: on course website/piazza soon
- Plan for course project
- Form for submitting reading questions: Will be posted on Piazza

Website: https://berkeley-deep-learning.github.io/cs294-131-s19/

Webcasting: https://www.youtube.com/playlist?list=PLkFD6_40KJIxG6I7MWd4LXAKI-kQO54_8

# Enrollment

If you need to get enrolled: find Samaneh after class!

# Today

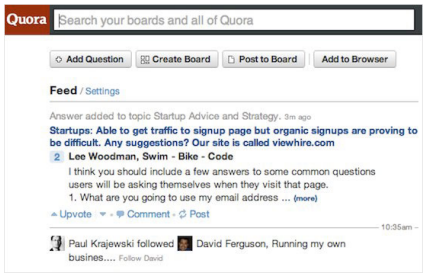
Logistics

**Overview of Topics**

Some Basic Frameworks

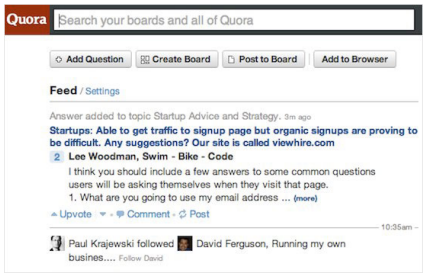
# Why This Class?

Machine learning systems deployed to billions of users

# Why This Class?

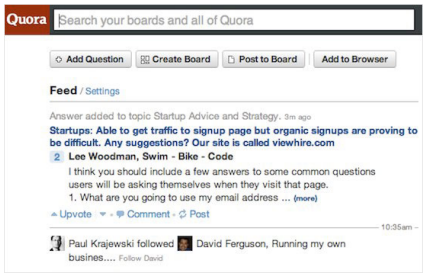
Machine learning systems deployed to billions of users


Existing paradigm: supervised learning

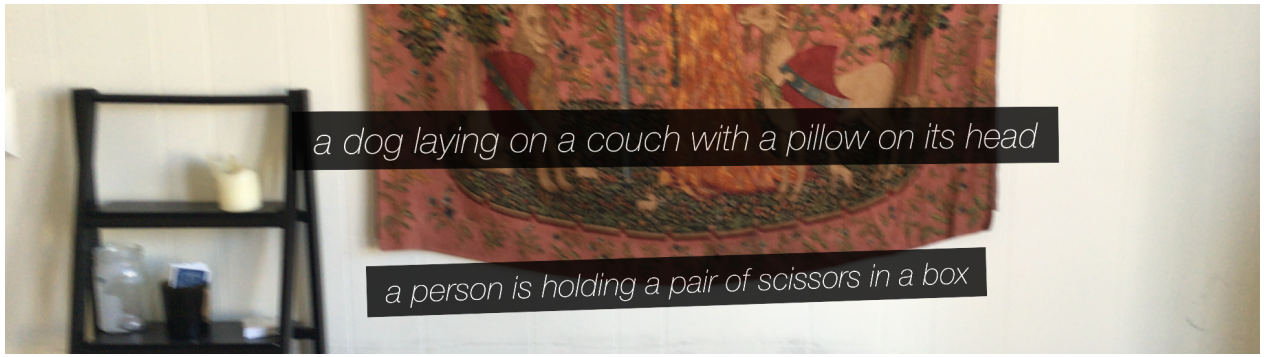
# Why This Class?

Machine learning systems deployed to billions of users


Existing paradigm: supervised learning

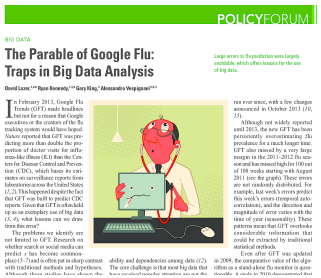
Many ways this can go wrong...



# Real World Deployment

Inputs can change unexpectedly, or subtly over time

Cook et al., 2011



POLICYFORUM

BIG DATA

**The Parable of Google Flu: Traps in Big Data Analysis**

Large errors in flu prediction were largely avoidable, which offers lessons for the use of big data.

David Lazer,[1,2]* Ryan Kennedy,[1,3]* Gary King,[4] Alessandro Vespignani[5,6,7]

In February 2013, Google Flu Trends (GFT) made headlines but not for a reason that Google executives or the creators of the flu tracking system would have hoped. *Nature* reported that GFT was predicting more than double the proportion of doctor visits for influenza-like illness (ILI) than the Centers for Disease Control and Prevention (CDC), which bases its estimates on surveillance reports from laboratories across the United States (*1*, *2*). This happened despite the fact that GFT was built to predict CDC reports. Given that GFT is often held up as an exemplary use of big data (*3*, *4*), what lessons can we draw from this error?

The problems we identify are not limited to GFT. Research on whether search or social media can predict *x* has become commonplace (*5*–*7*) and is often put in sharp contrast with traditional methods and hypotheses. Although these studies have shown the ... ability and dependencies among data (*12*). The core challenge is that most big data that ...

run ever since, with a few changes announced in October 2013 (*10*, *11*).

Although not widely reported until 2013, the new GFT has been persistently overestimating flu prevalence for a much longer time. GFT also missed by a very large margin in the 2011–2012 flu season and has missed high for 100 out of 108 weeks starting with August 2011 (see the graph). These errors are not randomly distributed. For example, last week's errors predict this week's errors (temporal autocorrelation), and the direction and magnitude of error varies with the time of year (seasonality). These patterns mean that GFT overlooks considerable information that could be extracted by traditional statistical methods.

Even after GFT was updated in 2009, the comparative value of the algorithm as a stand-alone flu monitor is questionable. A study in 2010 demonstrated that ...

Sensors can fail

Test cases may involve novel classes unseen during training

Adversaries might try to hack the system

Papernot et al., 2016

# Price of Autonomy

Deployment scale: too large for humans to effectively monitor

Sculley et al., 2015

# Price of Autonomy

Deployment scale: too large for humans to effectively monitor

Sculley et al., 2015

Time scale: too short to wait for human feedback

autonomous vehicles: Temizer et al., 2010; Geiger et al., 2012

# Price of Autonomy

Deployment scale: too large for humans to effectively monitor

Sculley et al., 2015

Time scale: too short to wait for human feedback

autonomous vehicles: Temizer et al., 2010; Geiger et al., 2012


Stakes: too high to tolerate errors

surgery: Taylor et al., 2008

# Why This Class? (II)

- Make sure ML systems do what people expect, avoid silent/unexpected/extreme failures

# Why This Class? (II)

- Make sure ML systems do what people expect, avoid silent/unexpected/extreme failures

- Particularly important in ML: correlated failures

# Why This Class? (II)

- Make sure ML systems do what people expect, avoid silent/unexpected/extreme failures
- Particularly important in ML: correlated failures
- Increasingly important as systems become more complex and autonomous

# Why This Class? (II)

- Make sure ML systems do what people expect, avoid silent/unexpected/extreme failures
- Particularly important in ML: correlated failures
- Increasingly important as systems become more complex and autonomous
- A new discipline: Reliability Engineering of ML Systems

# Why This Class? (II)

- Make sure ML systems do what people expect, avoid silent/unexpected/extreme failures
- Particularly important in ML: correlated failures
- Increasingly important as systems become more complex and autonomous
- A new discipline: Reliability Engineering of ML Systems
- Be forward-looking, care about impacts of ML on society

# Failures and Challenges

**Failures:**

- Can easily stop performing accurately when out-of-distribution

**Challenges:**

- Care about more than just accuracy when intersecting society

# Failure: Distribution Shift

POLICYFORUM

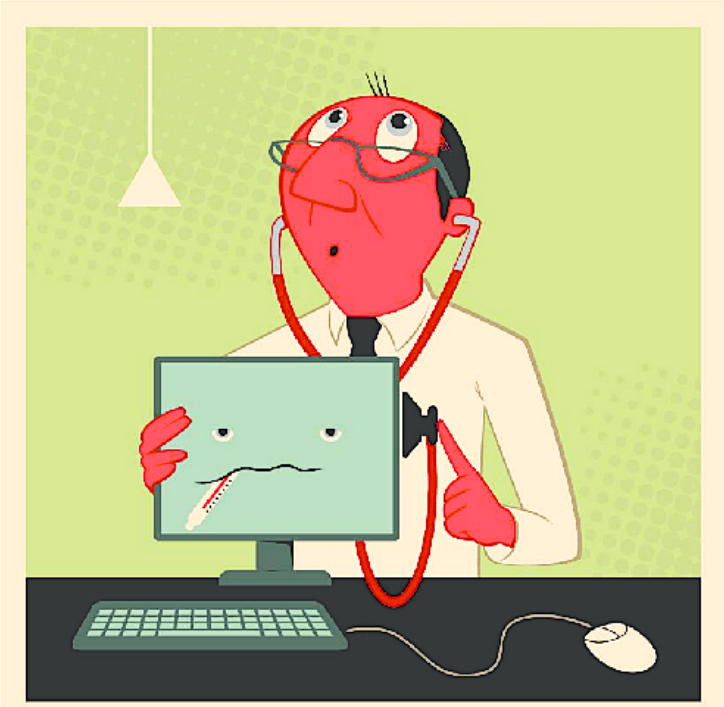
BIG DATA

## The Parable of Google Flu: Traps in Big Data Analysis

Large errors in flu prediction were largely avoidable, which offers lessons for the use of big data.

David Lazer,[1,2]* Ryan Kennedy,[1,3,4] Gary King,[3] Alessandro Vespignani[5,6,3]

In February 2013, Google Flu Trends (GFT) made headlines but not for a reason that Google executives or the creators of the flu tracking system would have hoped. *Nature* reported that GFT was predicting more than double the proportion of doctor visits for influenza-like illness (ILI) than the Centers for Disease Control and Prevention (CDC), which bases its estimates on surveillance reports from laboratories across the United States (*1*, *2*). This happened despite the fact that GFT was built to predict CDC reports. Given that GFT is often held up as an exemplary use of big data (*3*, *4*), what lessons can we draw from this error?



The problems we identify are not limited to GFT. Research on whether search or social media can predict *x* has become commonplace (*5*–*7*) and is often put in sharp contrast with traditional methods and hypotheses. Although these studies have shown the ability and dependencies among data (*12*). The core challenge is that most big data that have received popular attention are not the

run ever since, with a few changes announced in October 2013 (*10*, *15*).

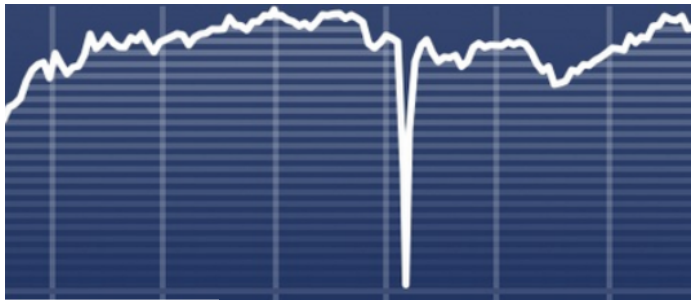
Although not widely reported until 2013, the new GFT has been persistently overestimating flu prevalence for a much longer time. GFT also missed by a very large margin in the 2011–2012 flu season and has missed high for 100 out of 108 weeks starting with August 2011 (see the graph). These errors are not randomly distributed. For example, last week's errors predict this week's errors (temporal autocorrelation), and the direction and magnitude of error varies with the time of year (seasonality). These patterns mean that GFT overlooks considerable information that could be extracted by traditional statistical methods.

Even after GFT was updated in 2009, the comparative value of the algorithm as a stand-alone flu monitor is ques-

# Failure: Exploitability

Syrian hackers compromise @AP:


$\Longrightarrow$

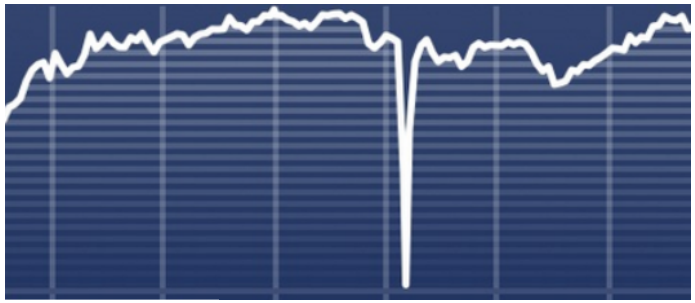


$136 billion drop

# Failure: Exploitability

Syrian hackers compromise @AP:


$\Longrightarrow$



$136 billion drop

Bots influenced U.S., other elections [Marwick & Lewis '17]

- presidential debates, #MacronLeaks
- affect trending topics

# Failure: Calibration

"My object detector is 99% accurate"

# Failure: Calibration

"My object detector is 99% accurate"

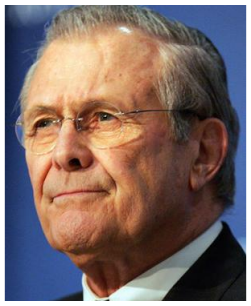
**Issue:** 99% with respect to what distribution?

# Failure: Calibration

"My object detector is 99% accurate"

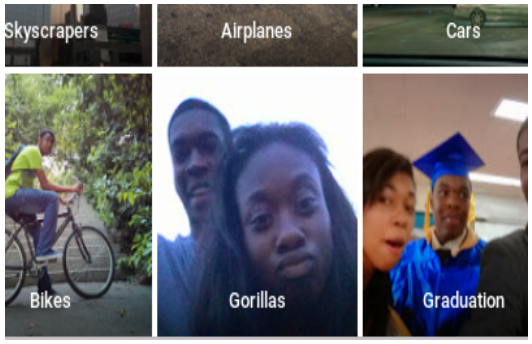
**Issue:** 99% with respect to what distribution?

Would like models to **know what they know** and abstain on bad inputs



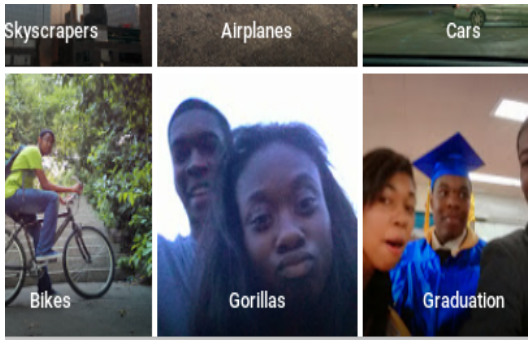
# Challenge: Fairness

Minority subpopulations

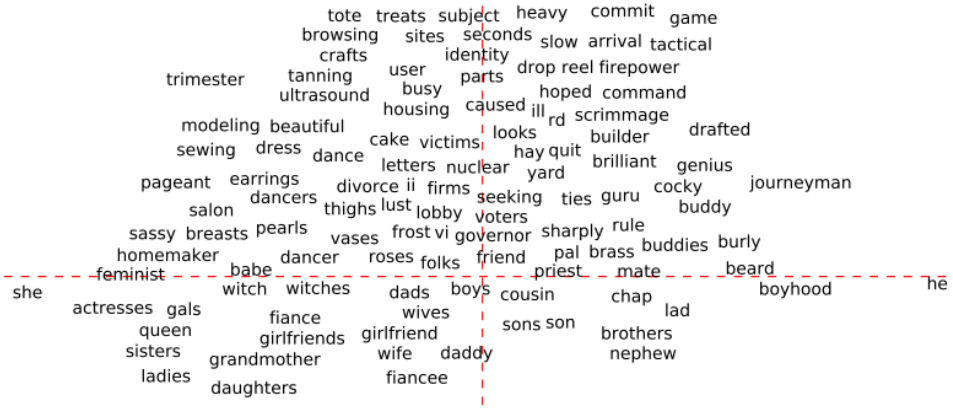


# Challenge: Fairness

## Minority subpopulations

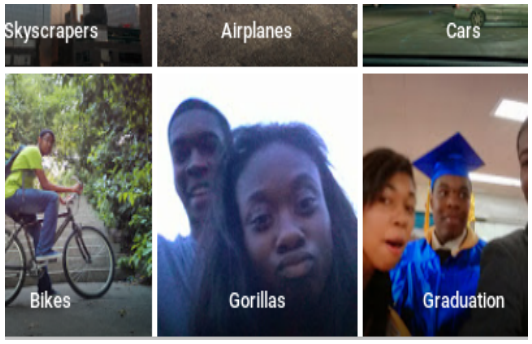



## Bias in data

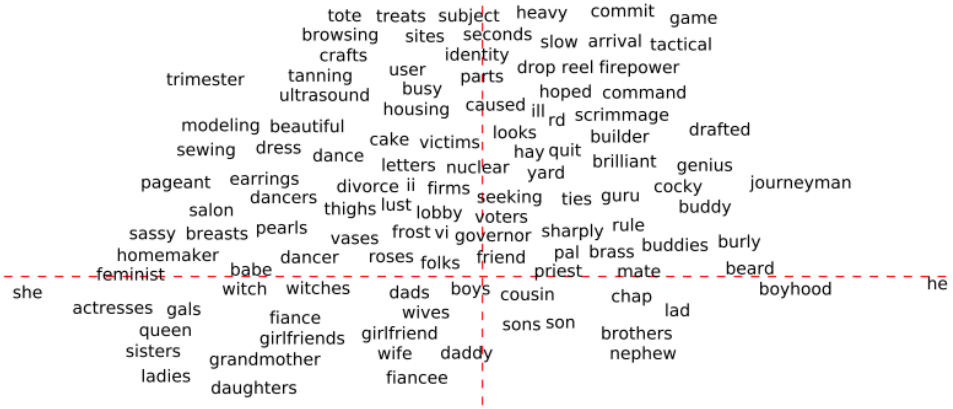


# Challenge: Fairness

Minority subpopulations




Bias in data




Crucial decisions: recidivism, loans

# Challenge: Causality

ML for:

- healthcare
- economics

**Key question:** What is the effect of intervention/policy X?

**Out-of-distribution**

# Challenge: Privacy

# Challenge: Systems

## Hidden Technical Debt in Machine Learning Systems

**D. Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips**
{dsculley,gholt,dgg,edavydov,toddphillips}@google.com
Google, Inc.

**Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, Dan Dennison**
{ebner,vchaudhary,mwyoung,jfcrespo,dennison}@google.com
Google, Inc.

### Abstract

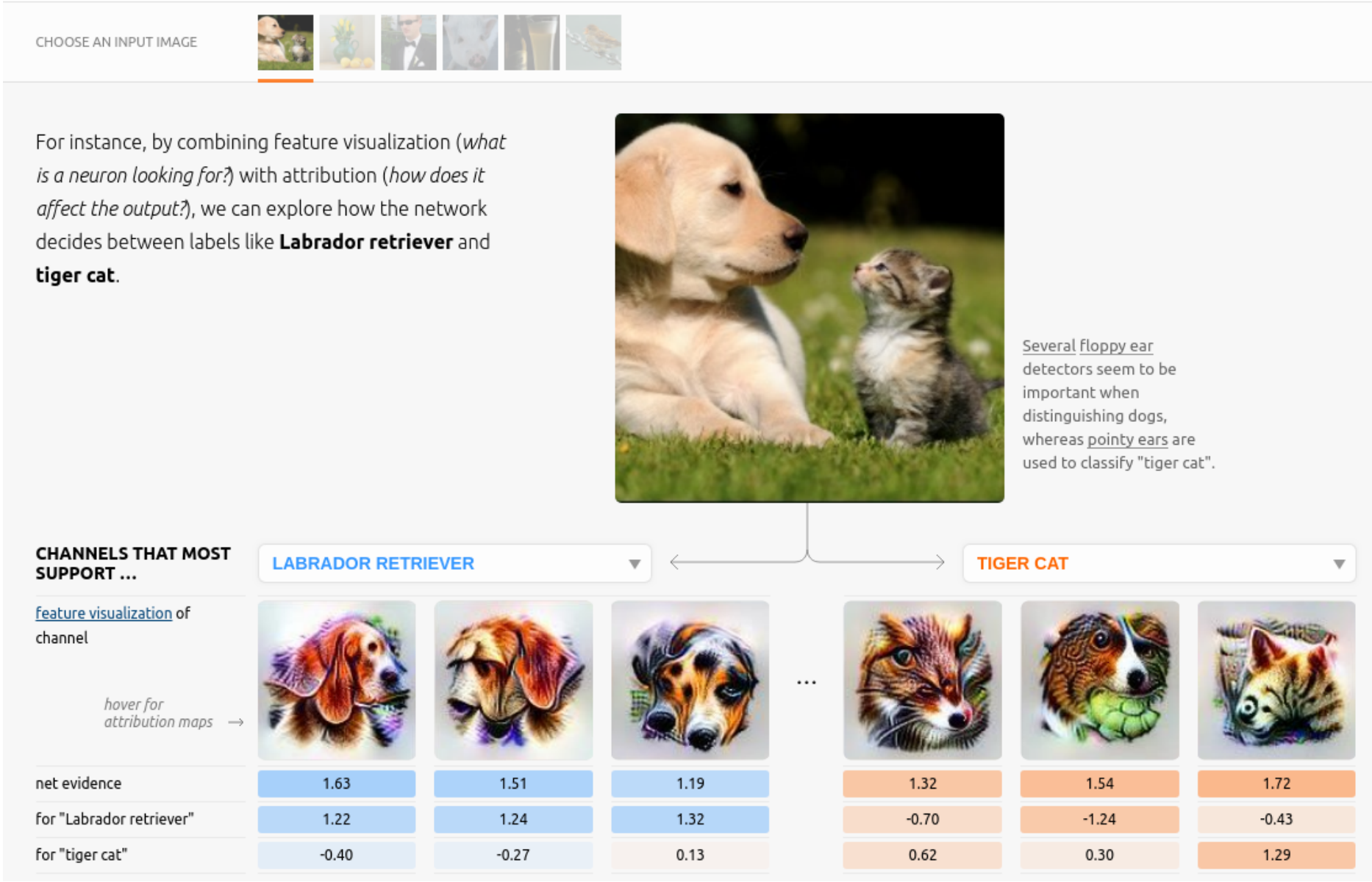
Machine learning offers a fantastically powerful toolkit for building useful complex prediction systems quickly. This paper argues it is dangerous to think of these quick wins as coming for free. Using the software engineering framework of *technical debt*, we find it is common to incur massive ongoing maintenance costs in real-world ML systems. We explore several ML-specific risk factors to account for in system design. These include boundary erosion, entanglement, hidden feedback loops, undeclared consumers, data dependencies, configuration issues, changes in the external world, and a variety of system-level anti-patterns.

### 1 Introduction

As the machine learning (ML) community continues to accumulate years of experience with live systems, a wide-spread and uncomfortable trend has emerged: developing and deploying ML systems is relatively fast and cheap, but maintaining them over time is difficult and expensive.

This dichotomy can be understood through the lens of *technical debt*, a metaphor introduced by Ward Cunningham in 1992 to help reason about the long term costs incurred by moving quickly in software engineering. As with fiscal debt, there are often sound strategic reasons to take on technical

# Challenge: Interpretability



# Other topics (not this class)

Reward/goal specification

Human-in-the-loop

Incentives

Safe exploration

Robust control (robotics)

# How to Get the Most Out of This Class

**Ask lots of questions:** both motivation and technical details

Think about the decisions made by the speaker

Form your own opinions!

# Today

Logistics

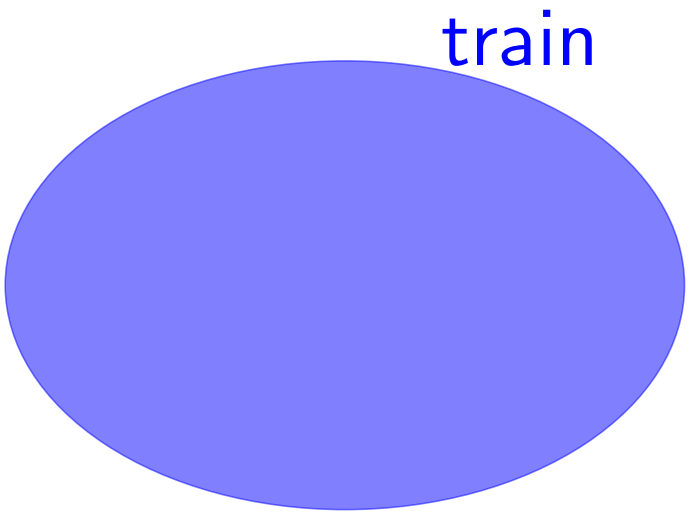
Overview of Topics

**Some Basic Frameworks**

# Train vs. Deployment

Most ML systems assume:

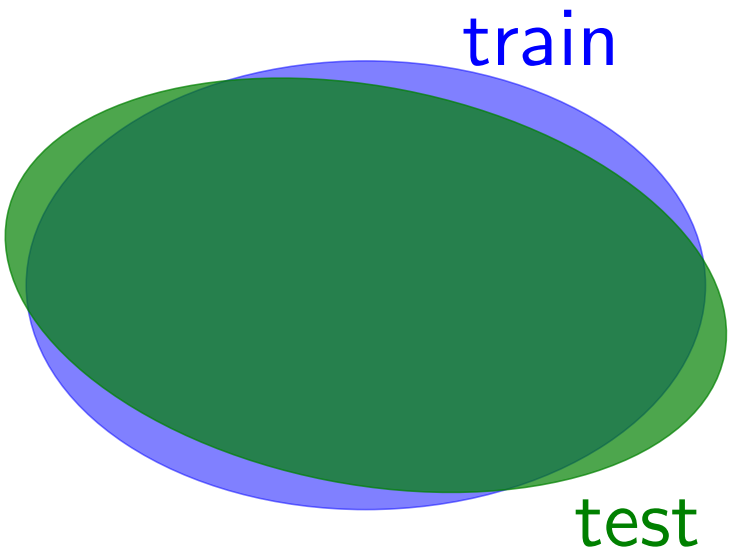
**train (data collection)** $\approx$ **test (deployment)**



# Train vs. Deployment

Most ML systems assume:

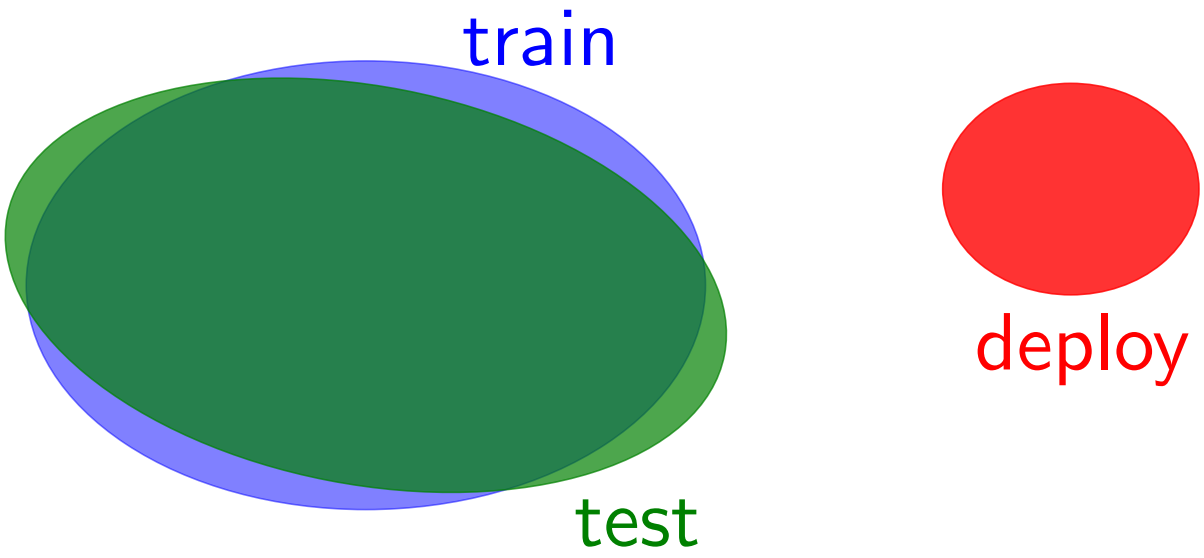
**train (data collection) $\approx$ test (deployment)**



# Train vs. Deployment

Most ML systems assume:

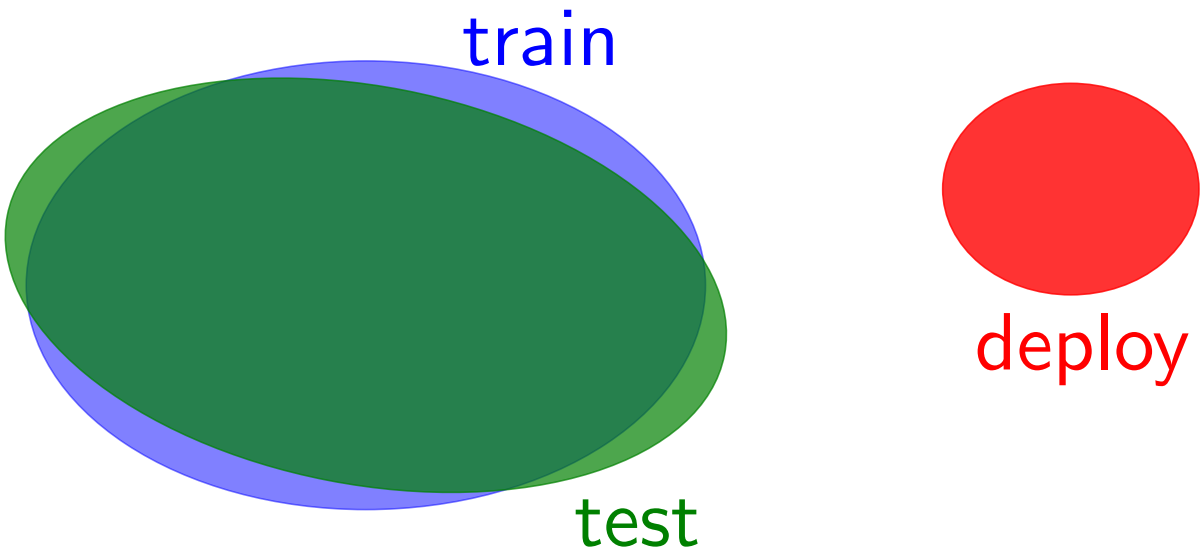
**train (data collection) $\approx$ test (deployment)**



# Train vs. Deployment

Most ML systems assume:

**train (data collection)** $\approx$ **test (deployment)**




**Real systems** can easily violate assumption!

# Alternative metrics

Typical metric: accuracy on held-out dev set

Many other metrics:

- accuracy on **variety** of dev sets (distribution shift)
- accuracy on **sub-populations** (fairness)
- **worst-case** over nearby points (robustness)
- accuracy after applying **intervention** (causality)

# Concept: Model Mis-specification

Model family said to be **well-specified** if true distribution is in family

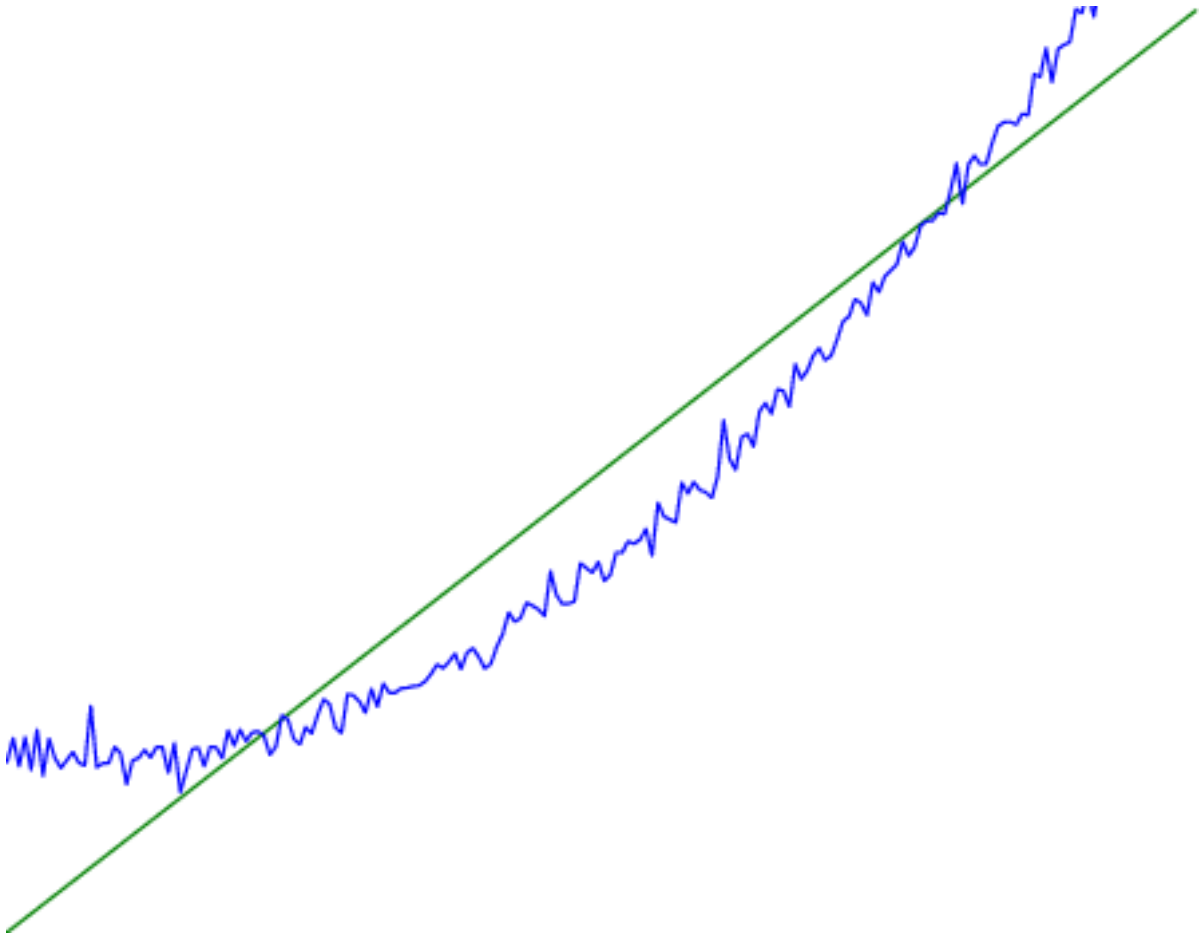
> All models are **wrong**, but some models are **useful**.
>
> George Box

**Supervised learning** (with train=test):

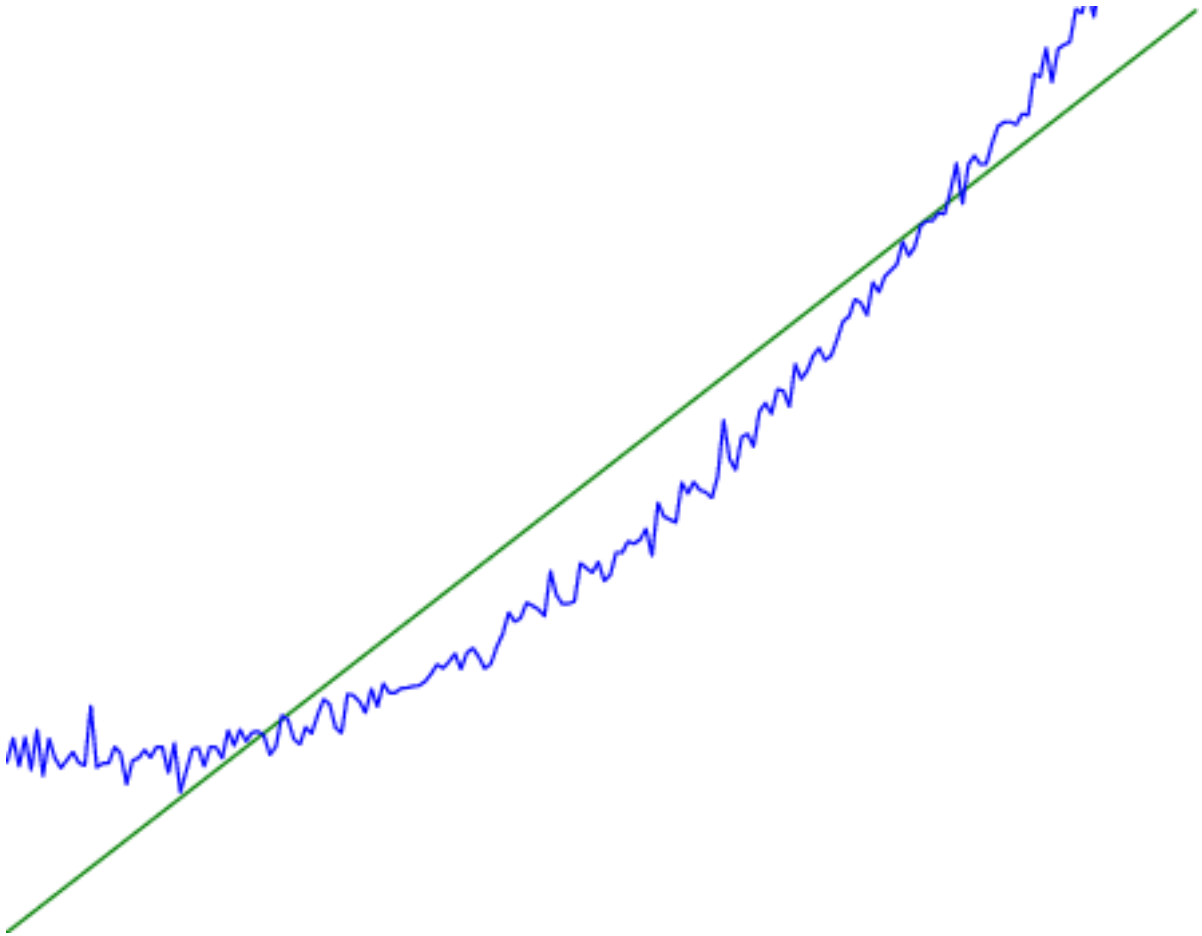
- mis-specification doesn't matter

**Everything else:** mis-specification matters!

# Mis-specification: Examples

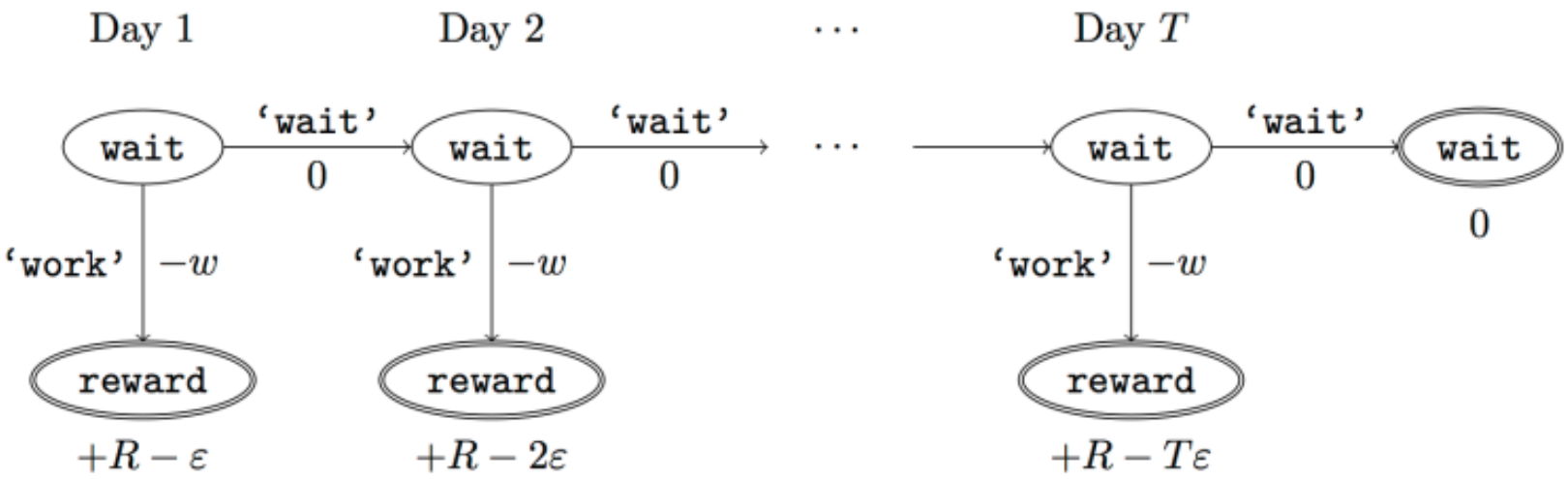


# Mis-specification: Examples

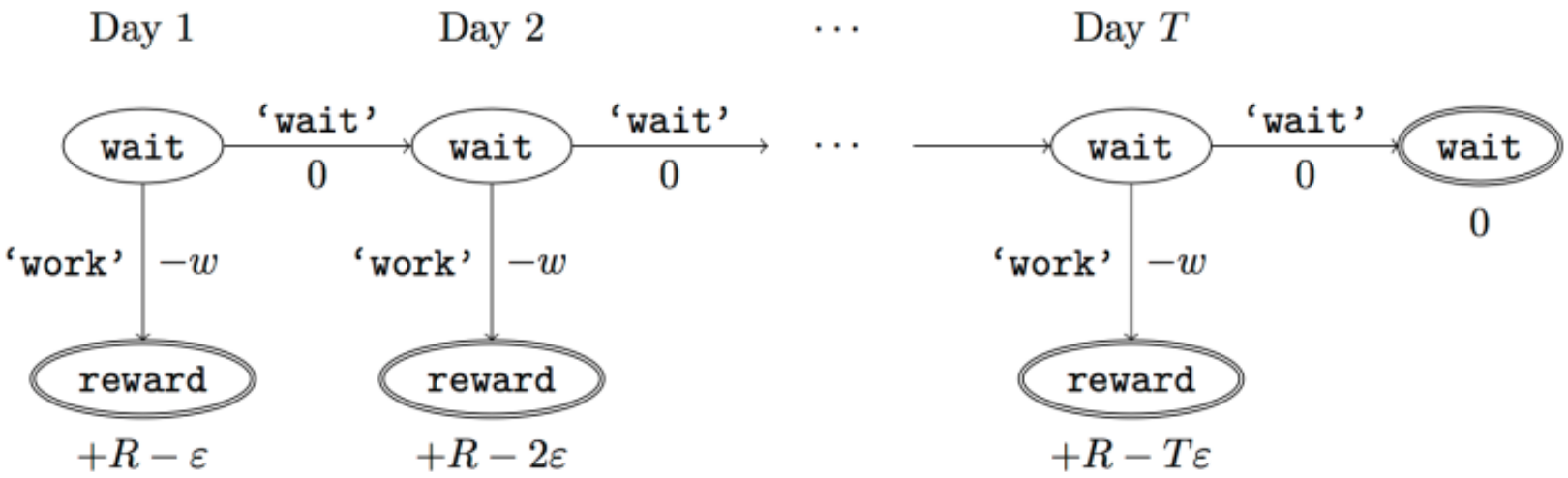


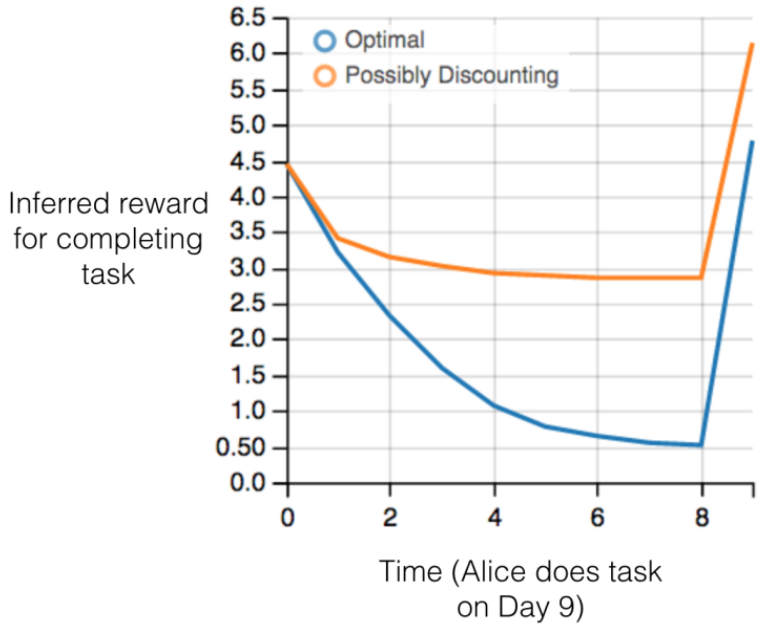
Different input distributions **conflict**

# Mis-specification: Examples



# Mis-specification: Examples



# Aspects of ML

**Engineering:** build systems that work

**Science:** understand why they work

**Concepts:** mental frameworks for designing/understanding systems

**Math:** formal underpinnings of the above

# Recommended Reading

*Two High-Stakes Challenges in ML* (Bottou)

*Concrete Problems in AI Safety* (Amodei et al.)

*Reflections on Random Kitchen Sinks* (Rahimi and Recht)