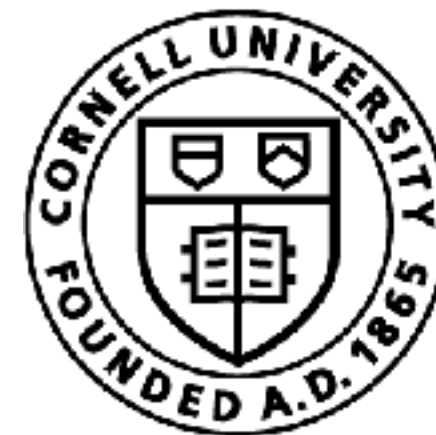


# CS 6756: Learning for Robot Decision Making

Sanjiban Choudhury



Cornell Bowers CIS  
**Computer Science**

WHAT A TIME TO BE  
T I M E  
A L L I V E !

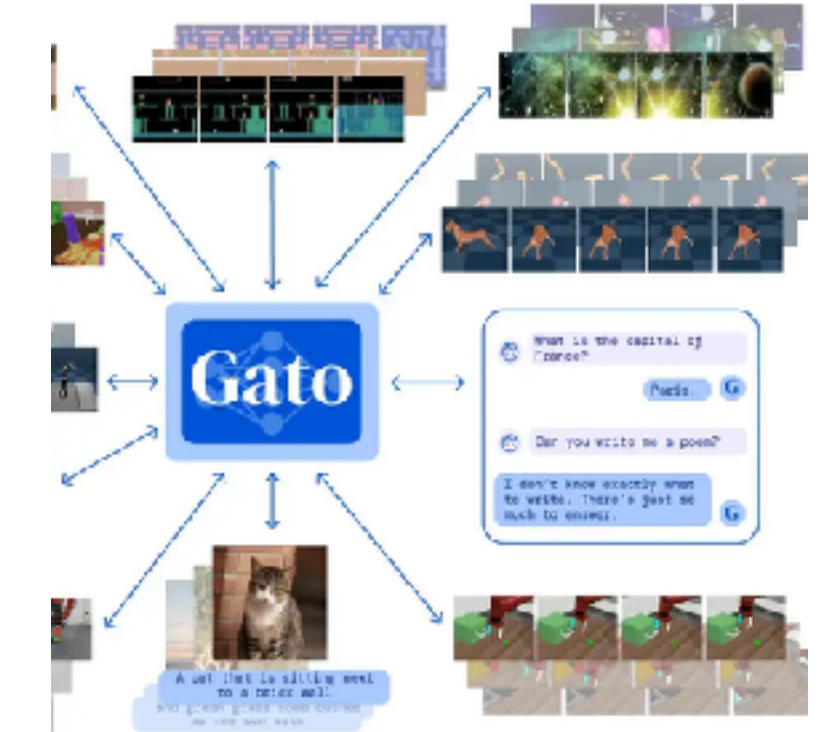
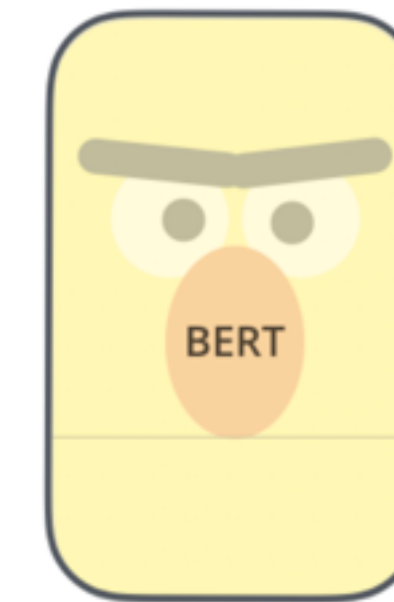
The image features a 3D, isometric-style graphic design. The text is arranged in three rows. The top row contains the words 'WHAT A TIME TO BE' in a bold, sans-serif font. The middle row contains the letters 'T I M E' in a larger, bold font. The bottom row contains the word 'A L L I V E !' in the largest, bold font. Each letter is rendered in a vibrant red color with a thick black outline. The 3D effect is achieved through blue shading on the top and side surfaces of the letters, and black shading on the bottom and back surfaces. The entire composition is set against a solid black background.

# Exciting time for Artificial Intelligence

## AlphaGo



## Transformers



## Deep Q Networks

### Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih Koray Kavukcuoglu David Silver Alex Graves Ioannis Antonoglou  
Daan Wierstra Martin Riedmiller  
DeepMind Technologies

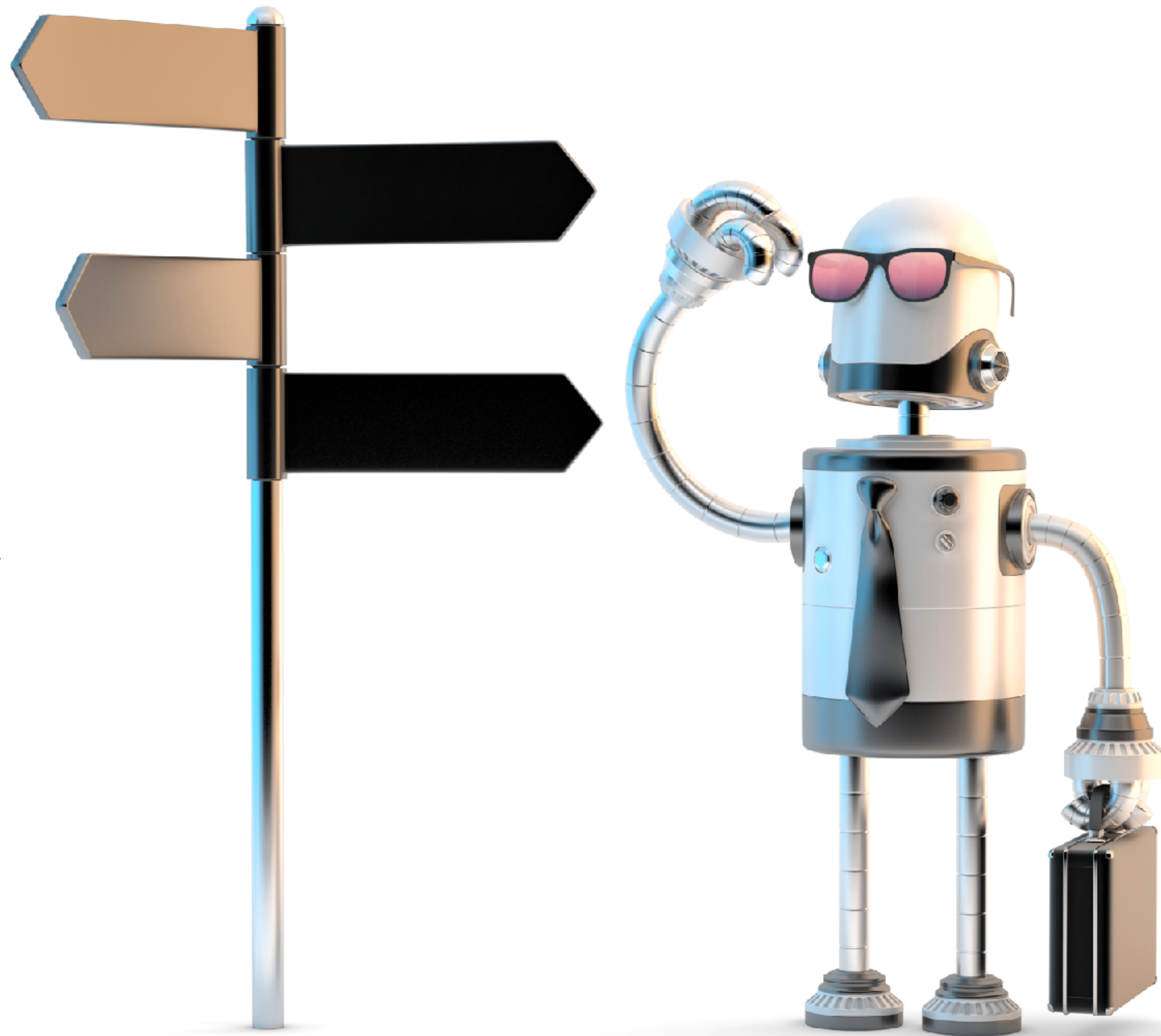


2013

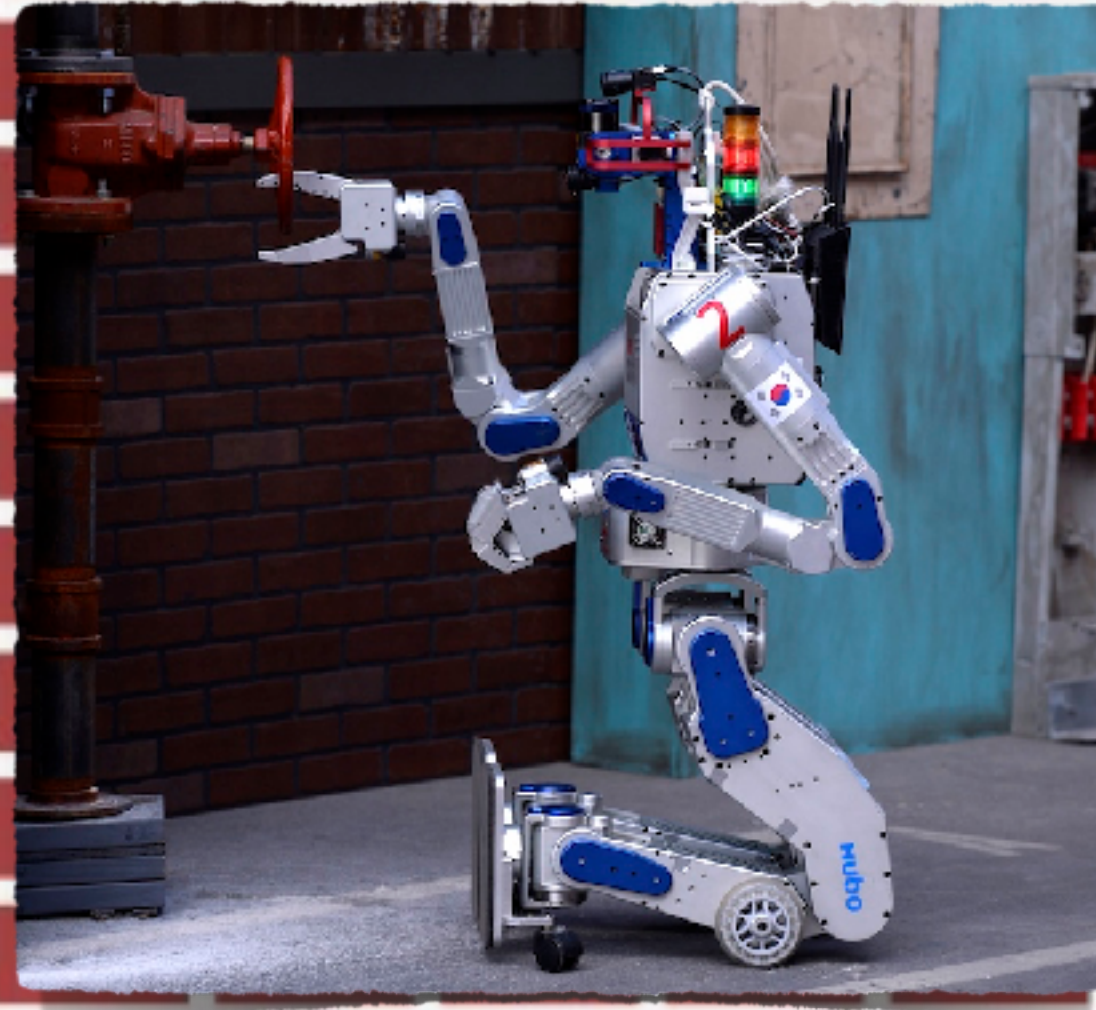
2016

Today

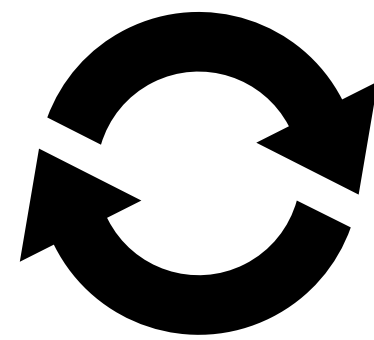
What about  
robotics?



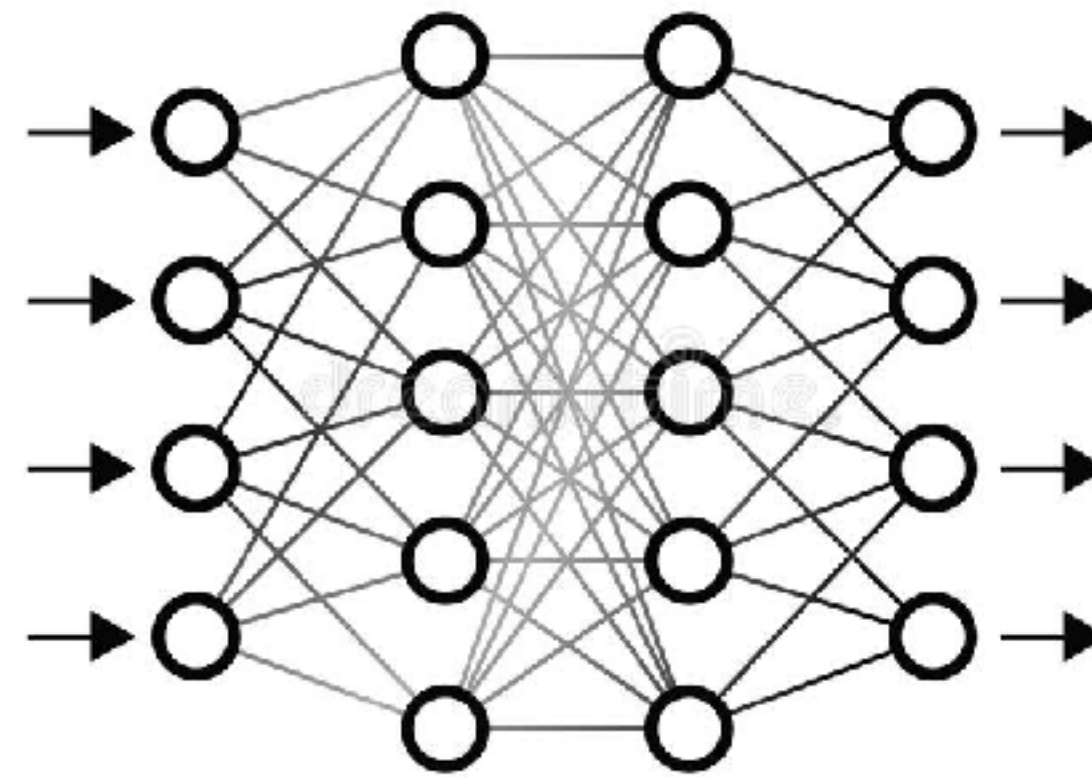
# Robotics 1.0: Building things brick by brick



# Robotics 2.0: Scale and improve with data



PyTorch

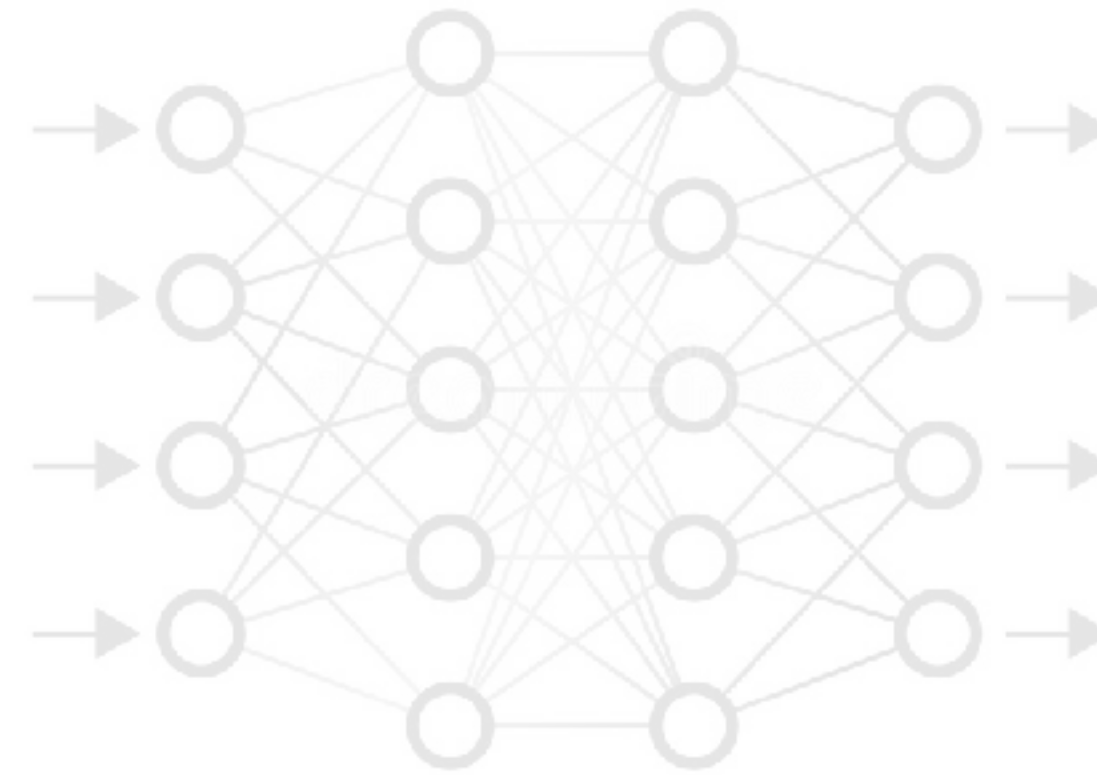


aws

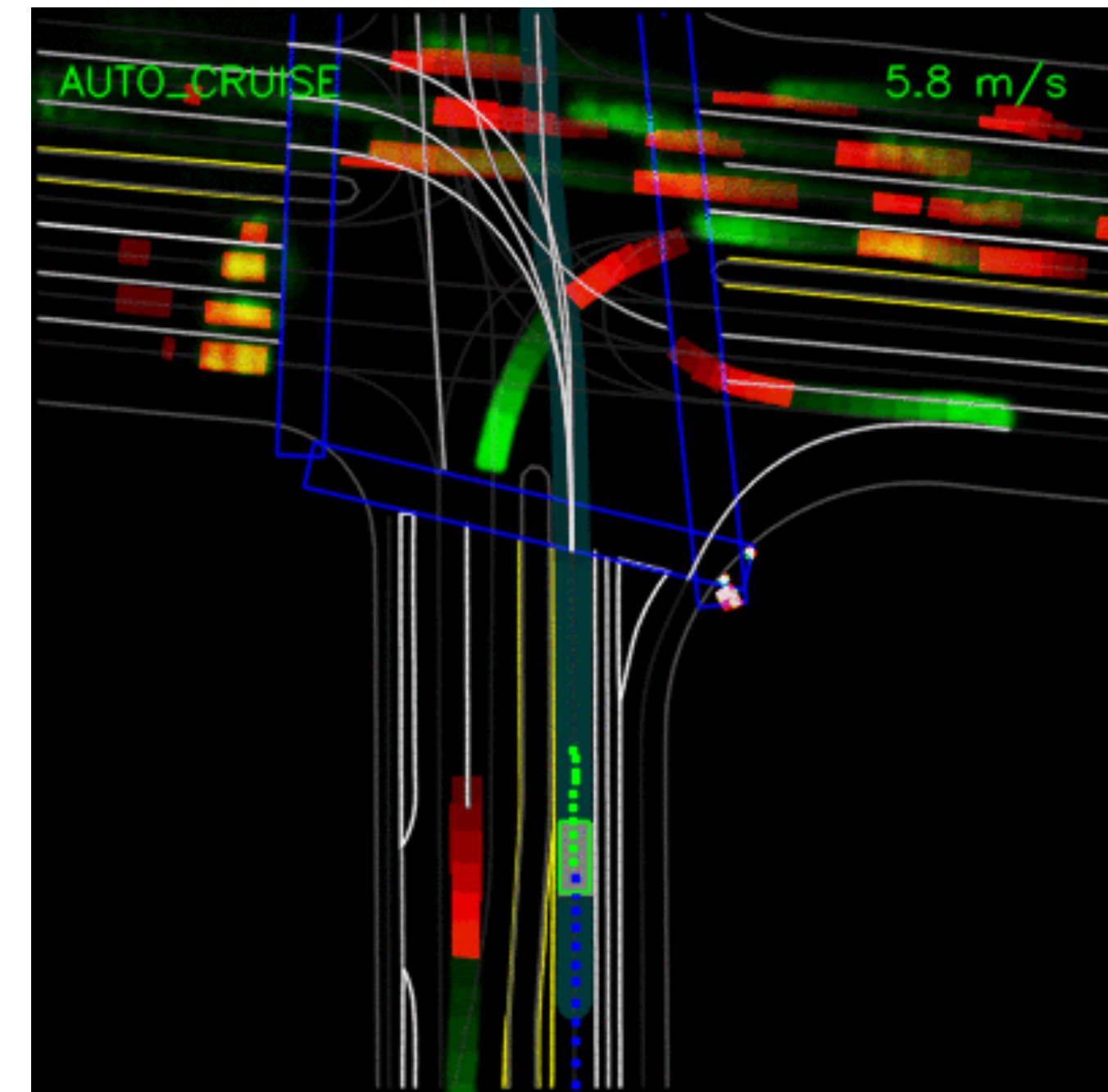
Invest in good  
ML pipelines

Formulate as a **learning** problem

# Robotics 2.0: Scale and improve with data



Invest in good ML pipelines



Self-driving led the way!

Formulate as a learning problem



Carnegie Mellon  
THE ROBOTICS INSTITUTE



PhD

W PAUL G. ALLEN SCHOOL  
OF COMPUTER SCIENCE & ENGINEERING



Postdoc

Aurora



Startup  
(that went public!)

Sanjiban  
Choudhury  
He / Him

*How should robots learn to make good decisions?*

Love the  
ocean!



And  
traveling  
with my  
partner



How should robots **learn** to make **good** decisions?



# WHY this course?



**Formulate** as a Markov Decision Problem (MDP)



**Solve** MDPs using an all-purpose toolkit  
(Imitation/Reinforcement learning, Model based/free)



**Deploy** learners in real-world  
(Safety, distribution shift, value alignment)

Take *any* robot application

# Belonging



How should robots **learn** to make **good** decisions?



# Self-driving

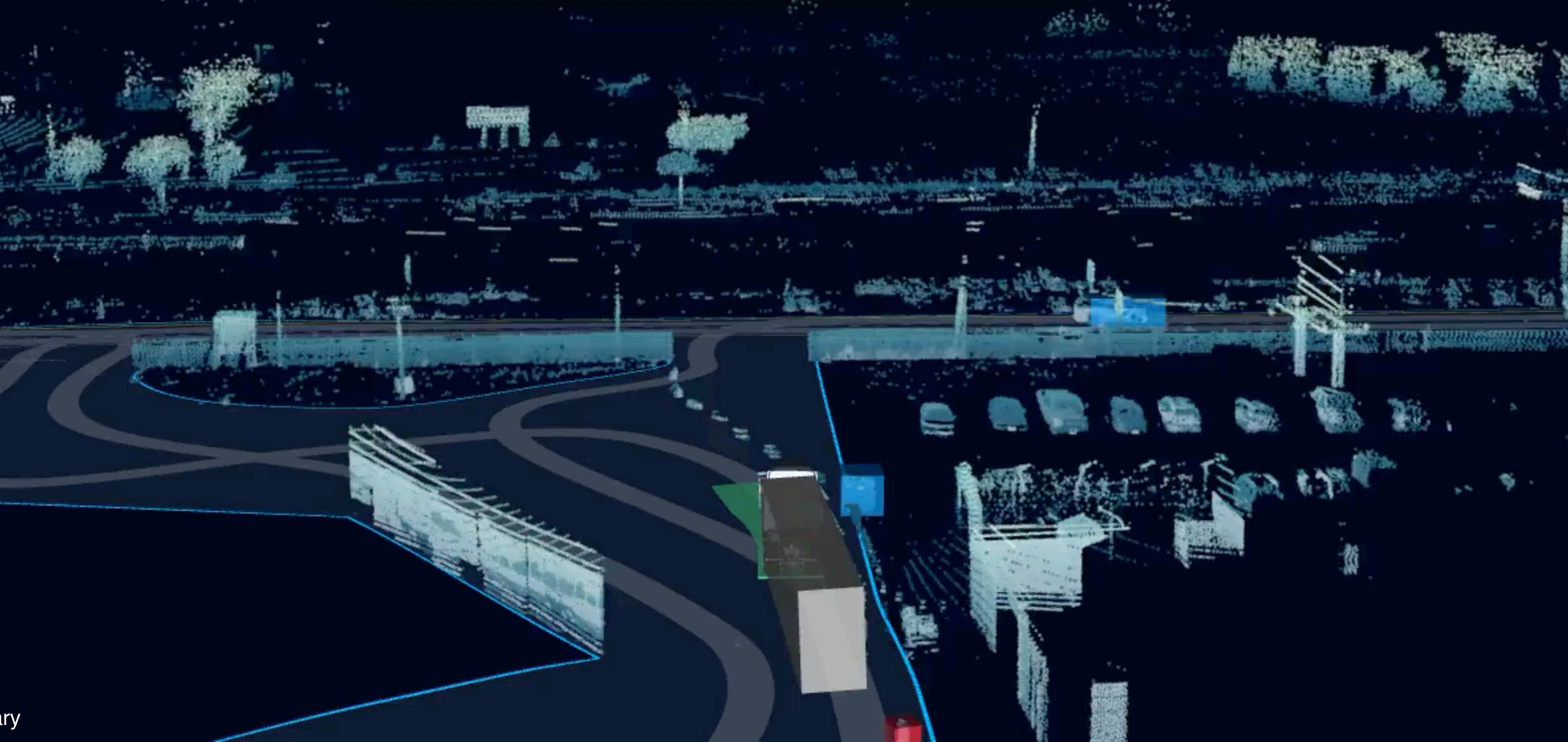
RIGHT TURN  
-10 FT

READY



0  
MPH

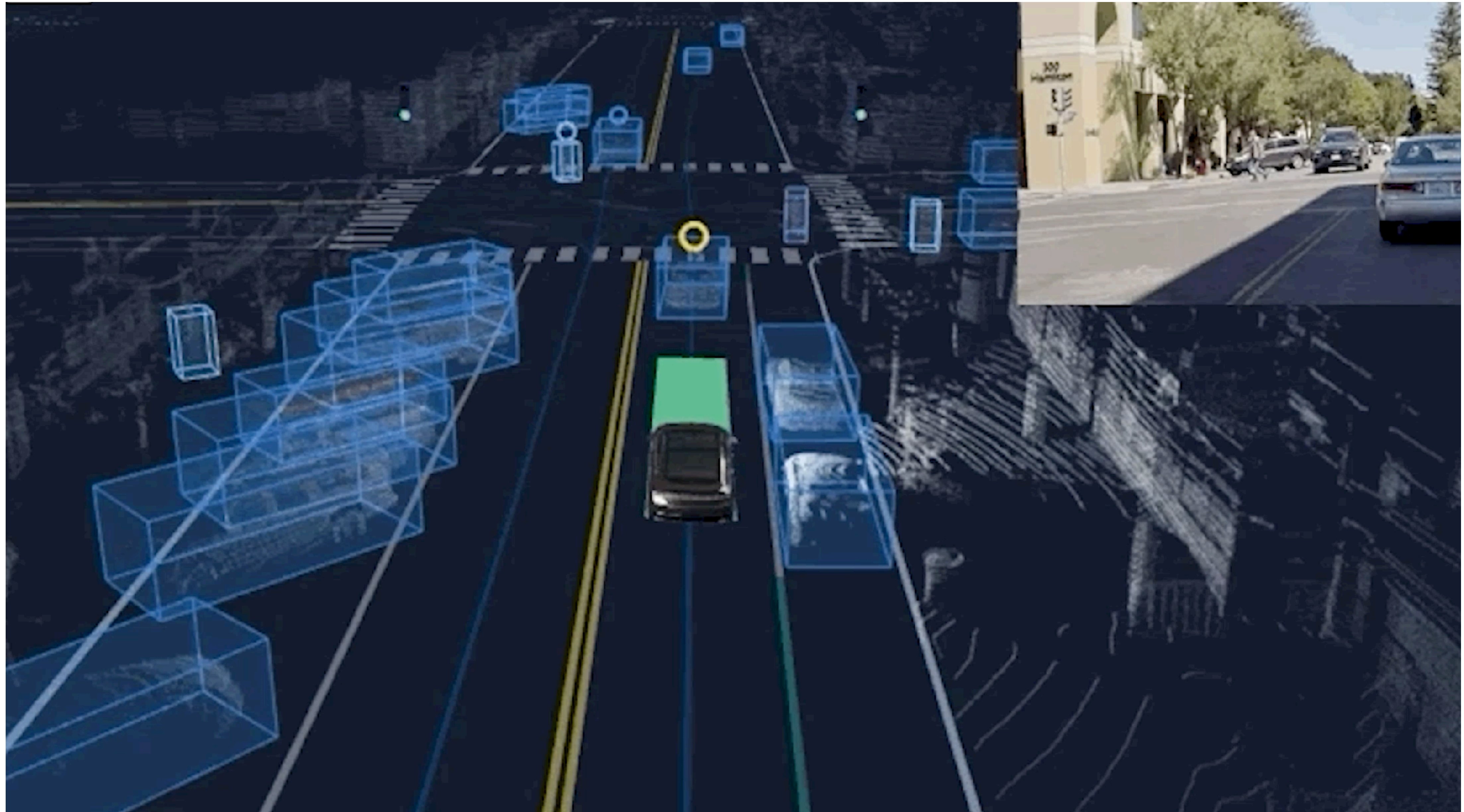
SPEED  
LIMIT  
5



Activity!



# Activity: What is “good” behavior in a left turn?





# Activity: What is “good” behavior in a left turn?



How should robots **learn** to make **good** decisions?

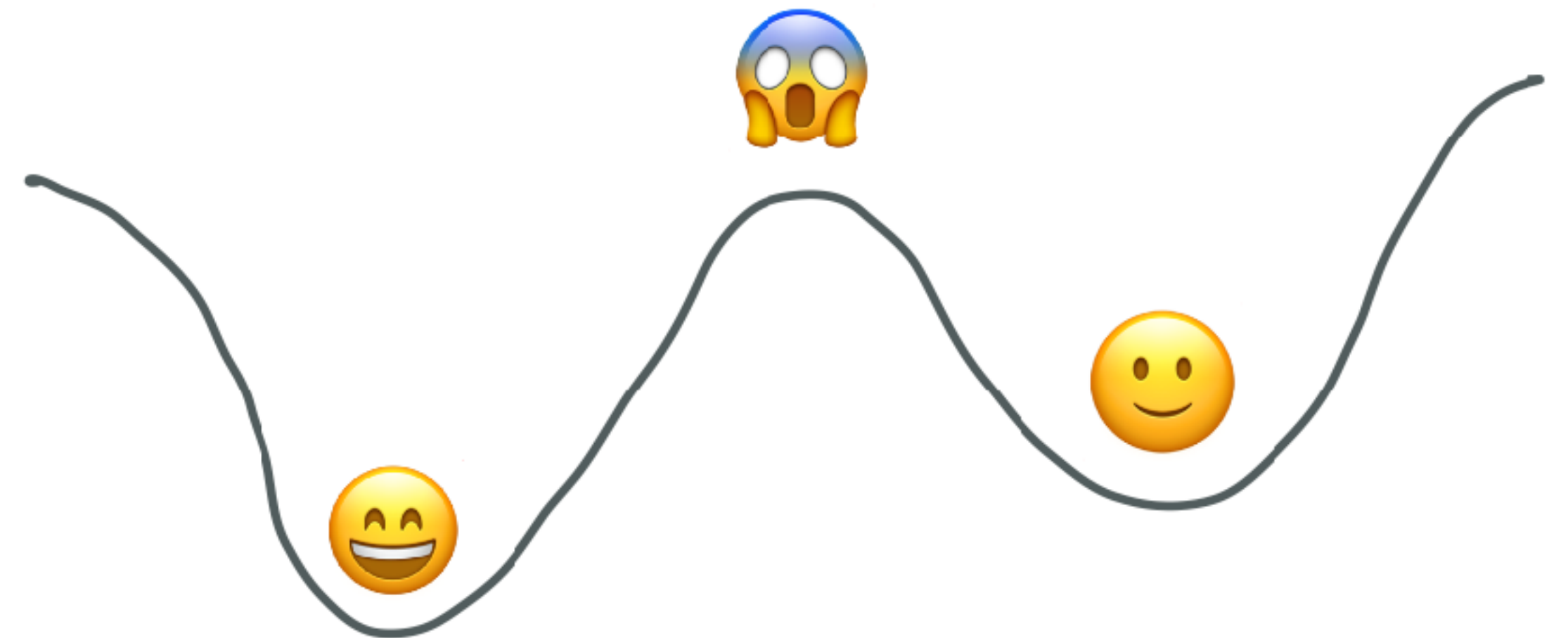




# Three fundamental questions

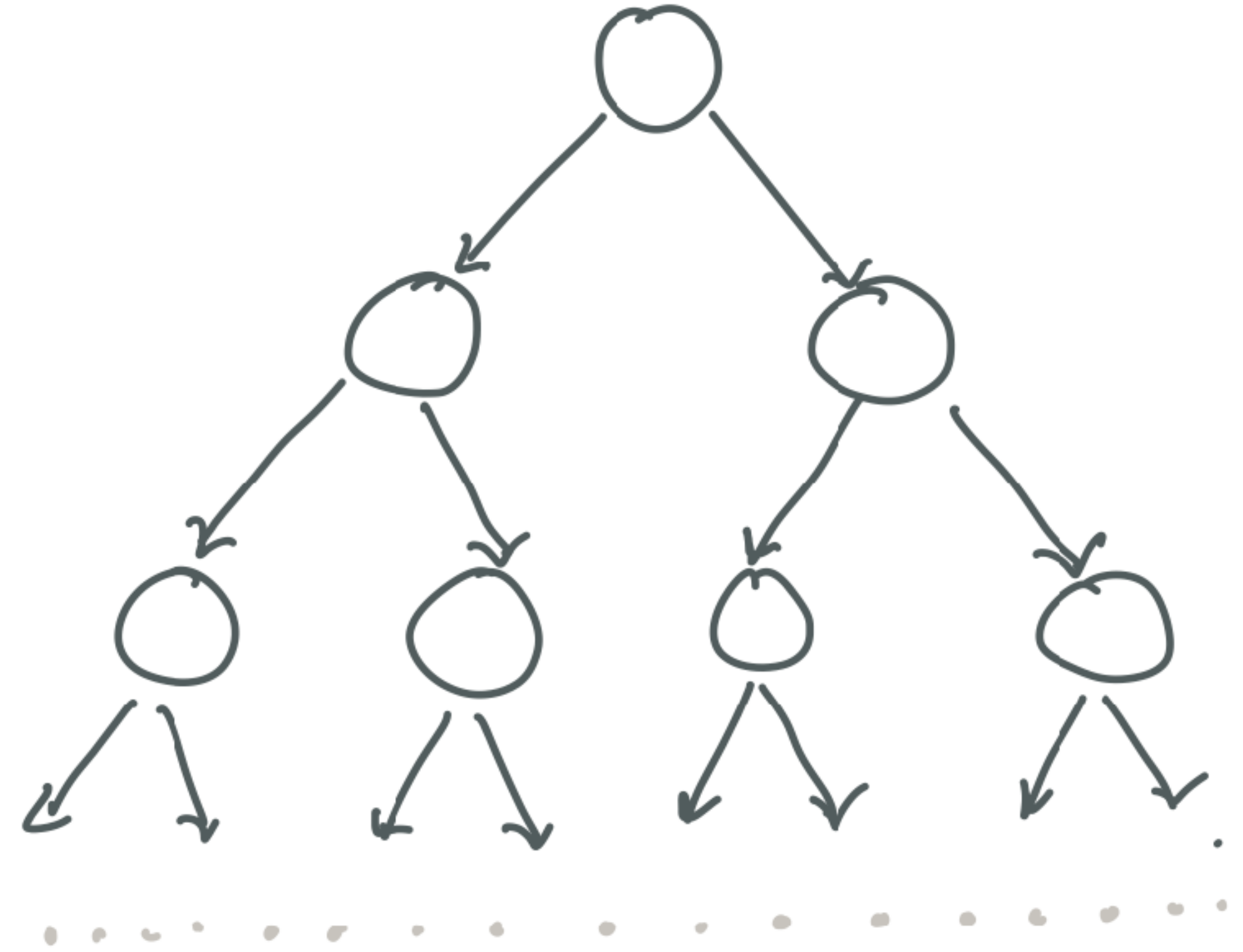
# Values

What are good / bad states?



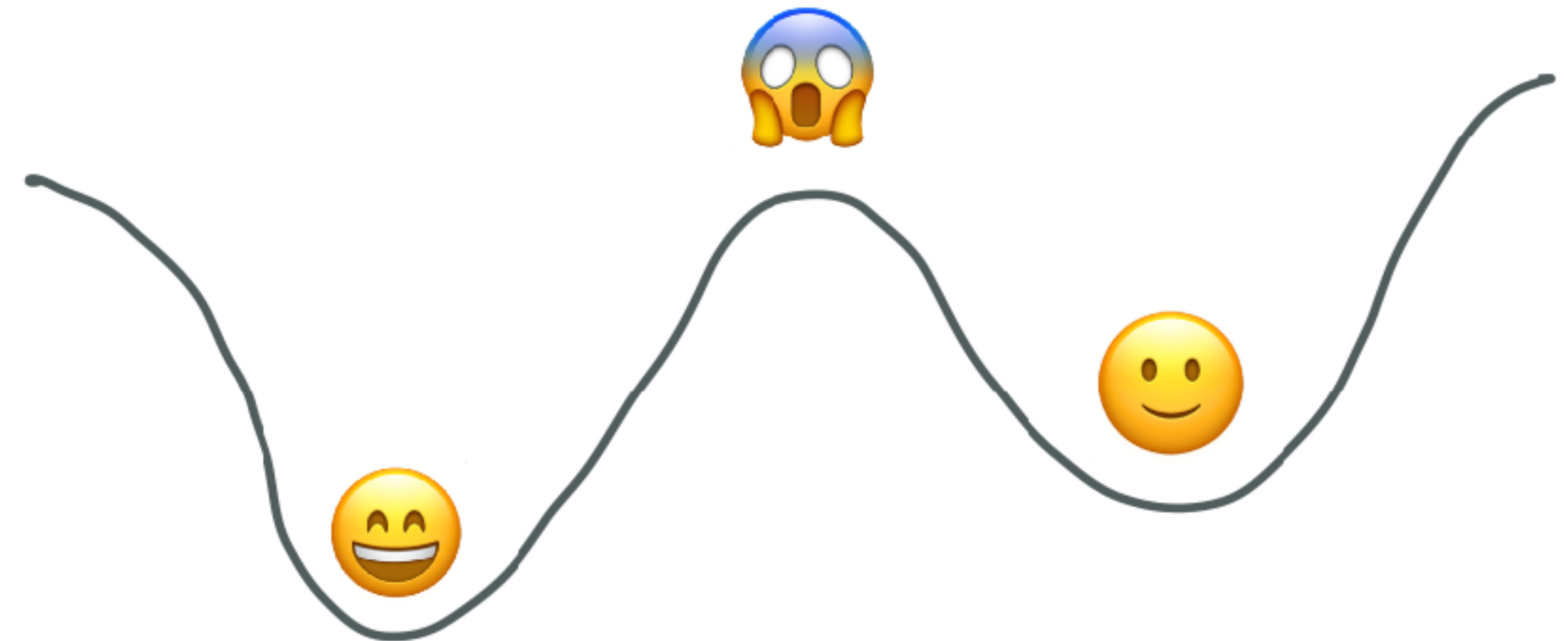
# Models

How do decisions affect states?



# Values

What are good / bad states?



# Optimization

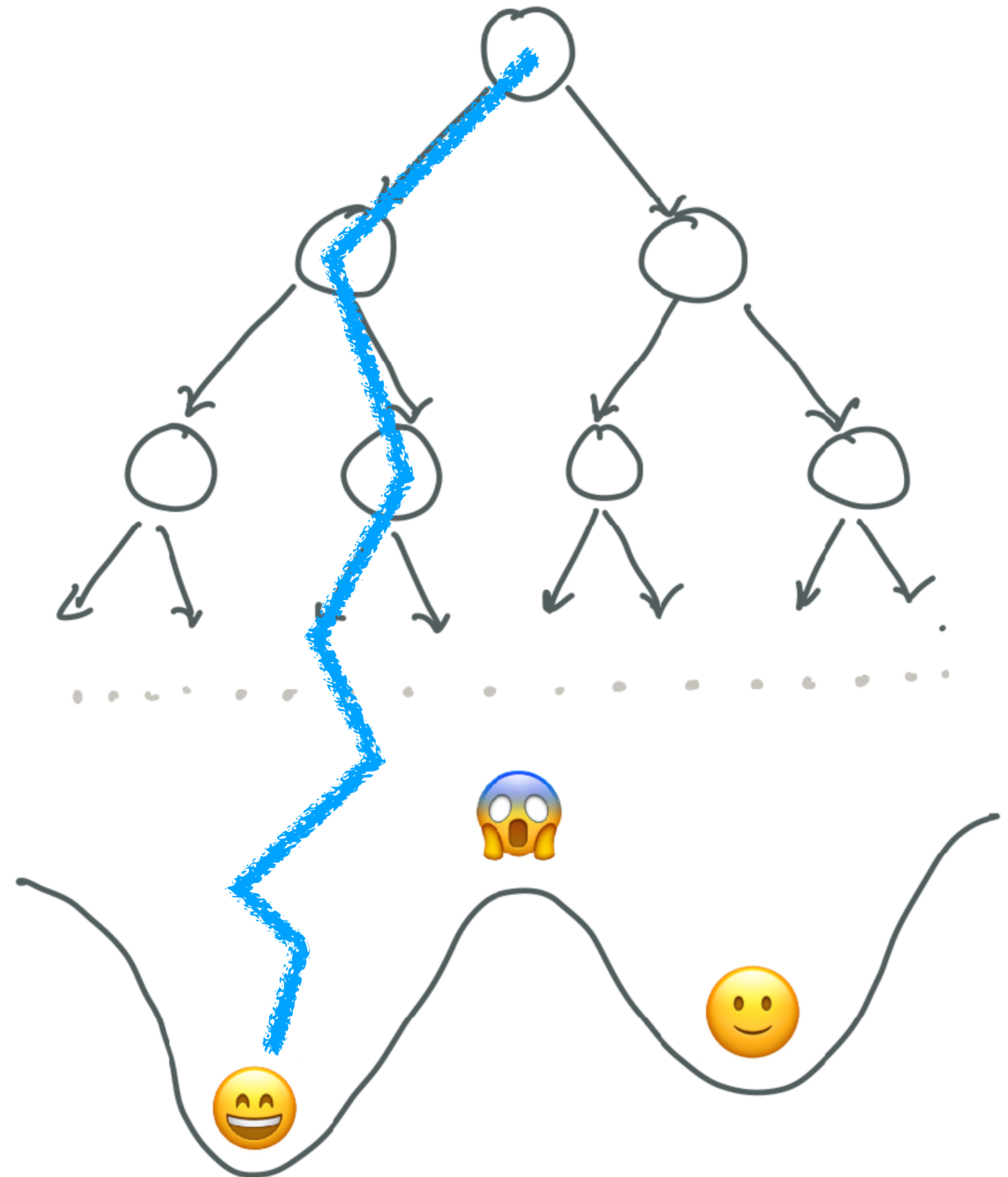
How do we efficiently find the optimal sequence of decisions?

## Models

How do decisions affect states?

## Values

What are good / bad states?



# Optimization

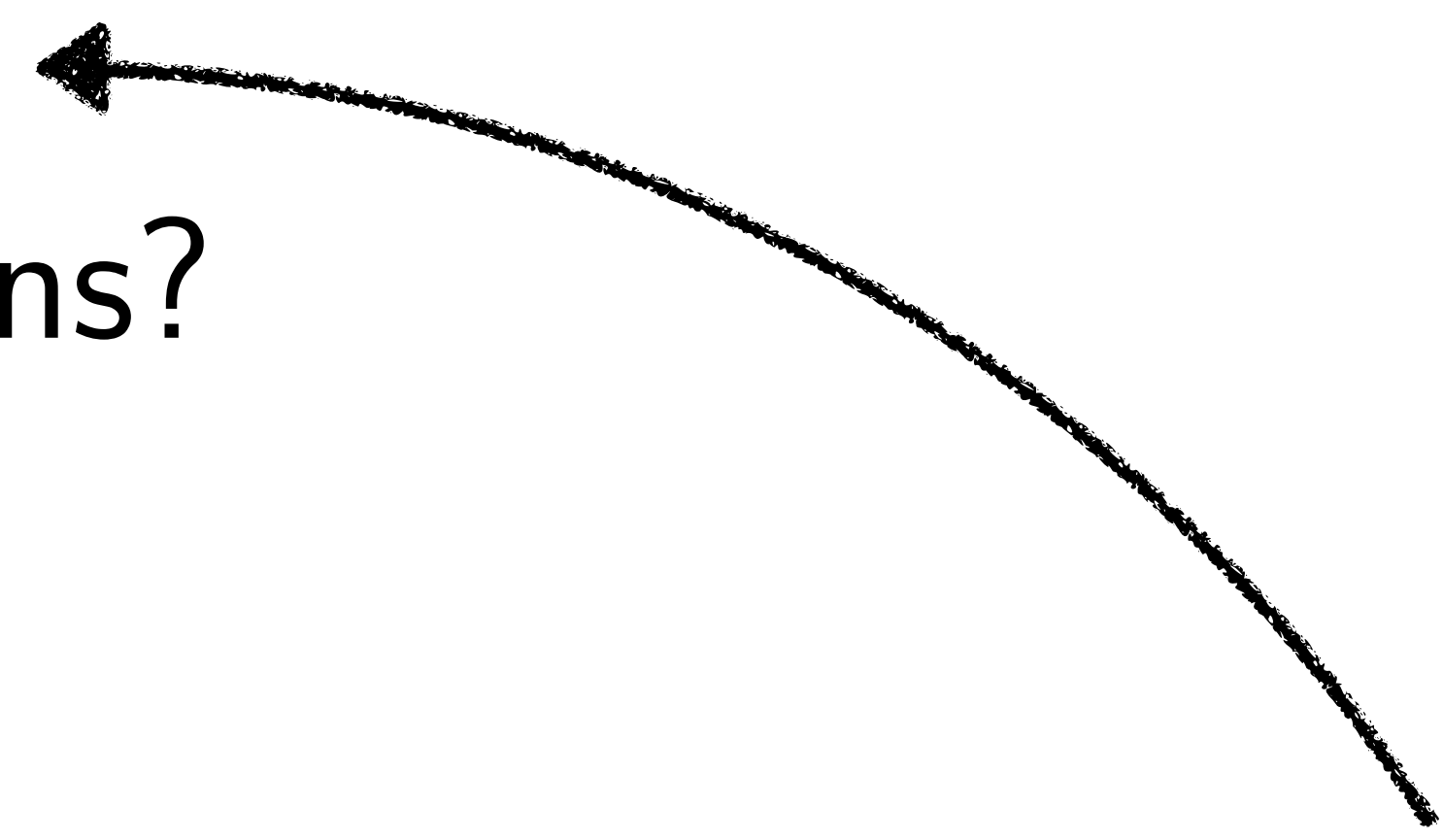
How do we efficiently find the optimal sequence of decisions?

# Models

How do decisions affect states?

# Values

What are good / bad states?



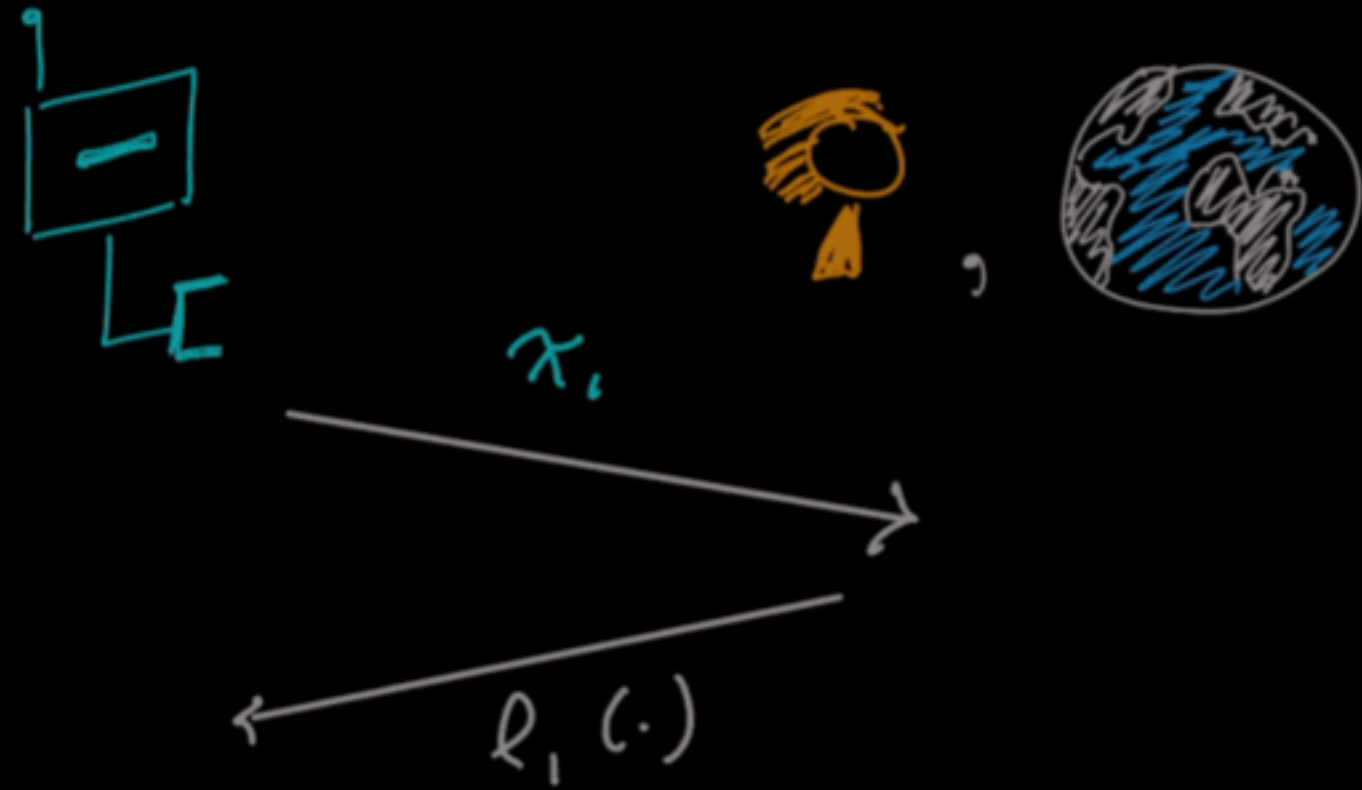
# Learning



# 5 Levels

of

# Robot Learning



$$\min_{\pi} \sum_{i=1}^{\infty} Q^*(s, \pi(s)) - Q^*(s, \pi^*(s))$$

min  
 $\pi$   
ROBOT

max  
 $Q^*$   
ACTION  
VALUE



# Values

*What are good / bad states?*

# What are good / bad states?



## Bad

- Collision
- Cutting off pedestrians
- Cutting off oncoming car
- Getting stuck in intersection when light turns red
- Excessive braking / braking speed limit

## Good

- Completing the turn quickly



Question:

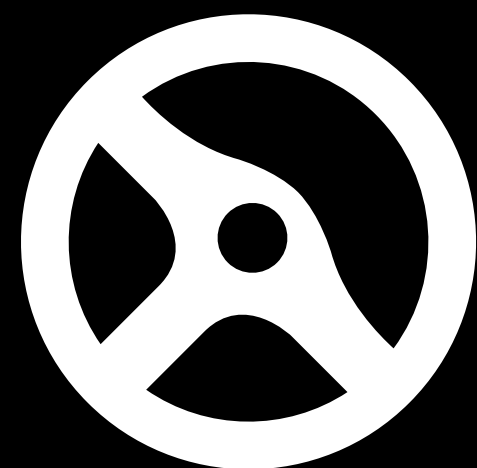
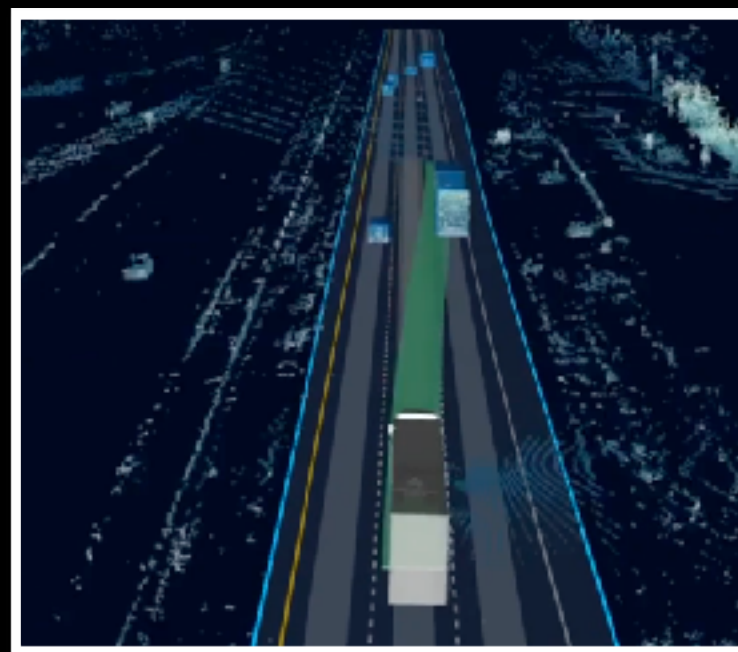
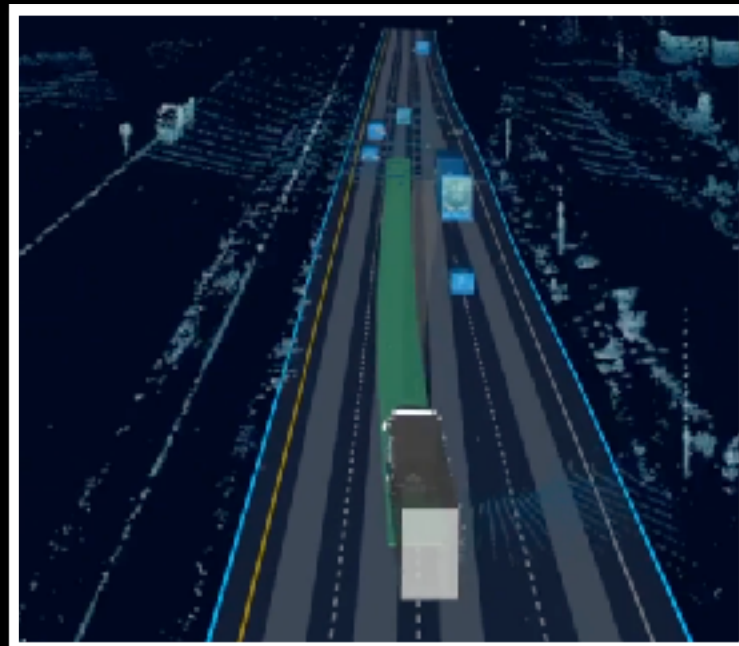
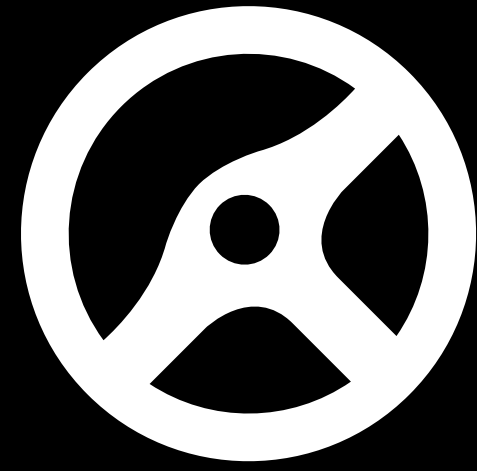
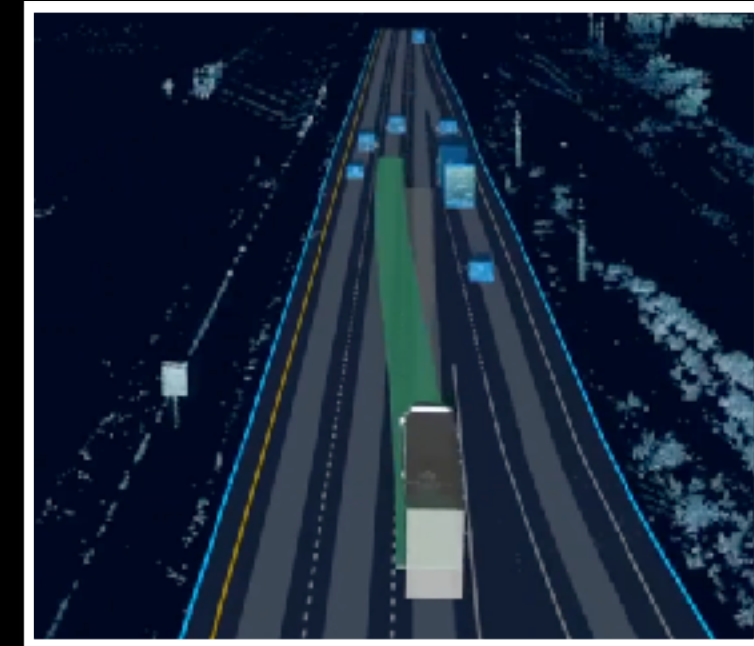
How do we program in  
these values?



Why don't we simply  
*imitate* **good** human  
driving?

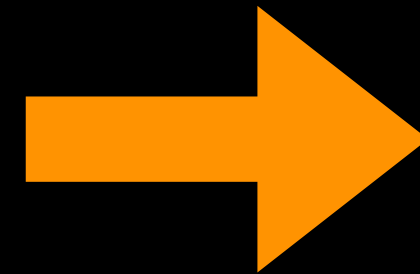
# SUPERVISED LEARNING

#1 Get Expert Data



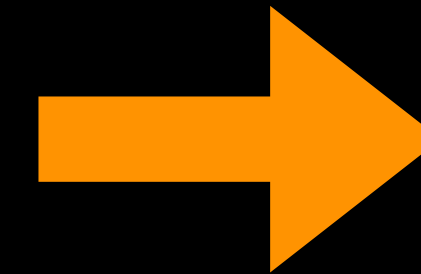
Input (s)

Output (a)



#2  
Train  
Policy

$$\pi : s \rightarrow a$$



#3 Deploy!



An aerial, high-angle photograph of a dense urban street intersection. The street is filled with a variety of vehicles, including cars, buses, and vans, moving in different directions. The scene is captured in a slightly desaturated, dark color palette. Overlaid on the center of the image is the text "Train ≠ Test" in a large, bold, orange-red font. The text is centered horizontally and vertically, with the "≠" symbol being a prominent feature between the two words.

**Train  $\neq$  Test**



# Lesson #1

Feedback drives

covariate shift

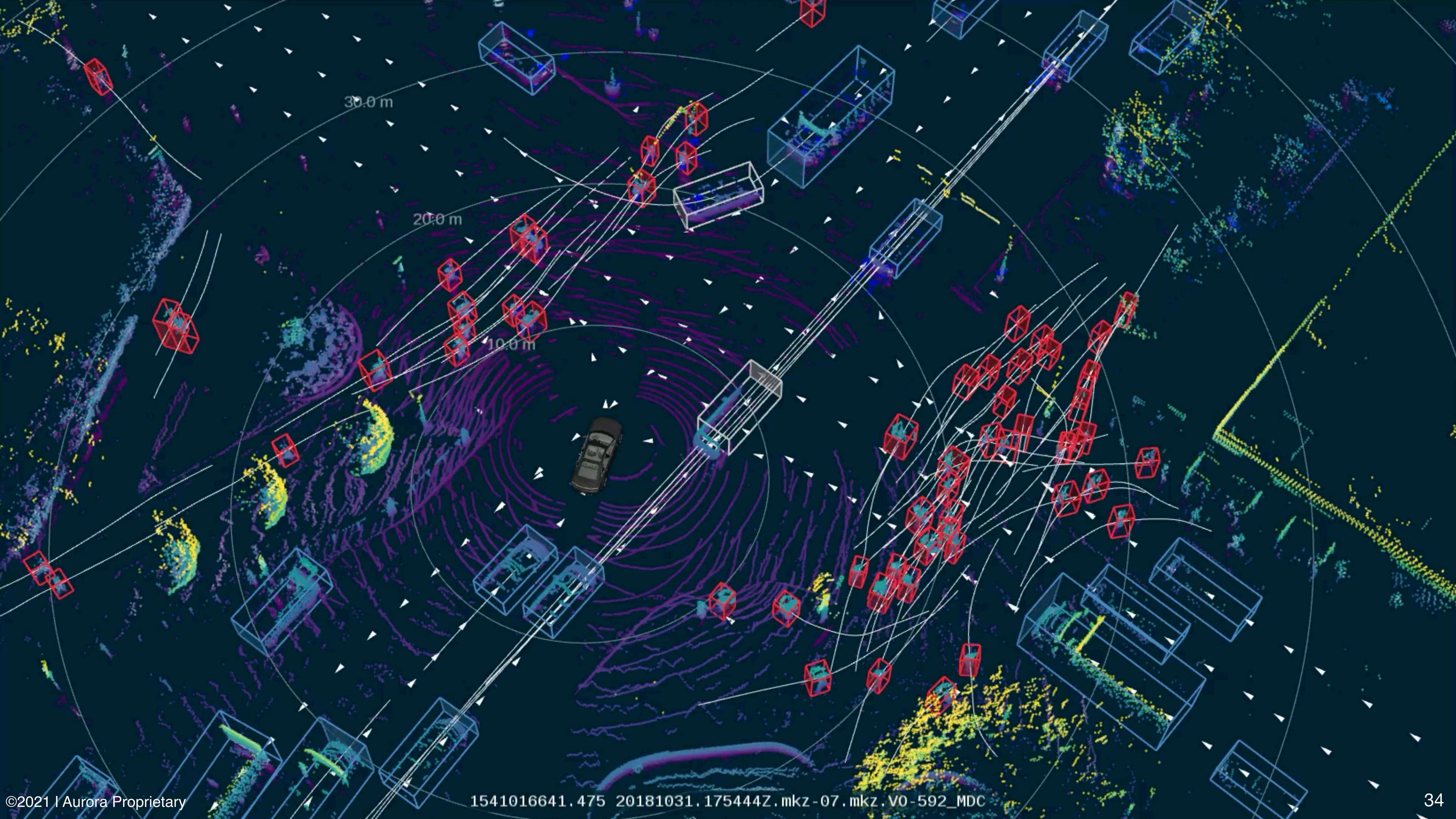
# Models

*How do decisions affect states?*



Activity!



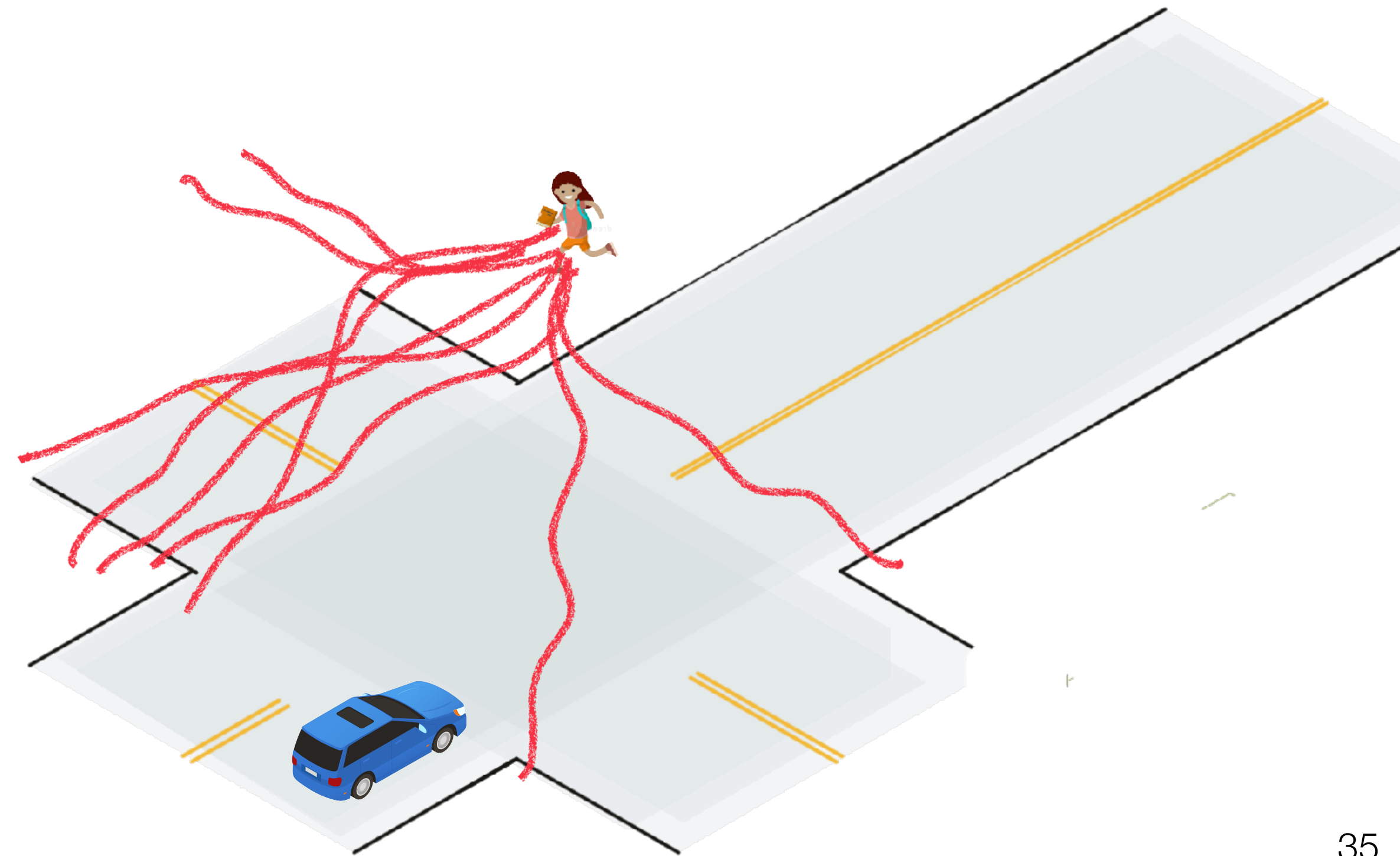


# Think-Pair-Share

Think (30 sec): How do we train a model of how pedestrians move?  
What are some of the challenges?  
What makes for a good model?

Pair: Find a partner

Share (45 sec): Partners exchange ideas



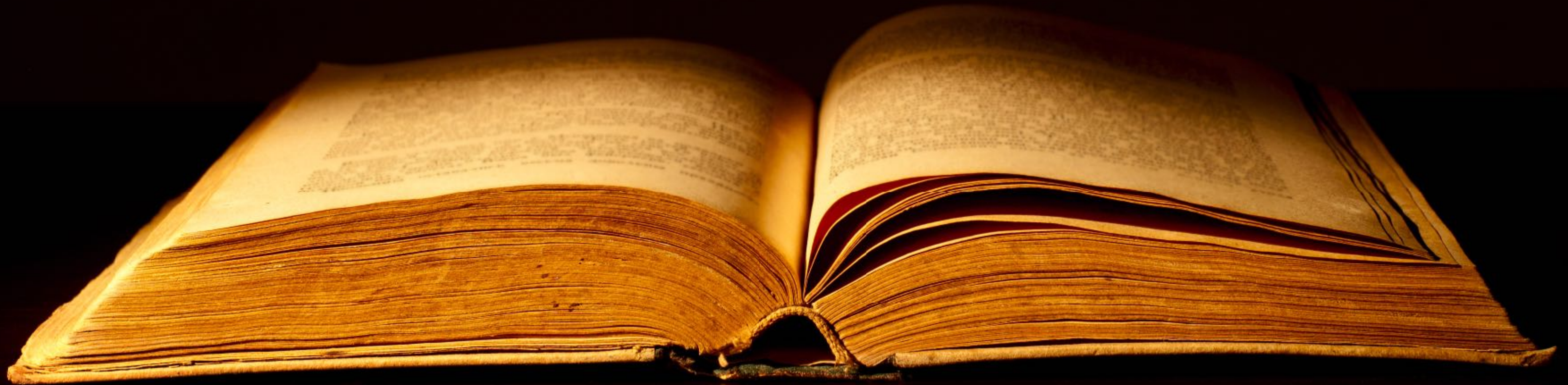
*“When solving a problem of interest,  
do not solve a more general problem  
as an intermediate step.”*

Vladmir Vapnik

The Nature of Statistical Learning

# Lesson #2

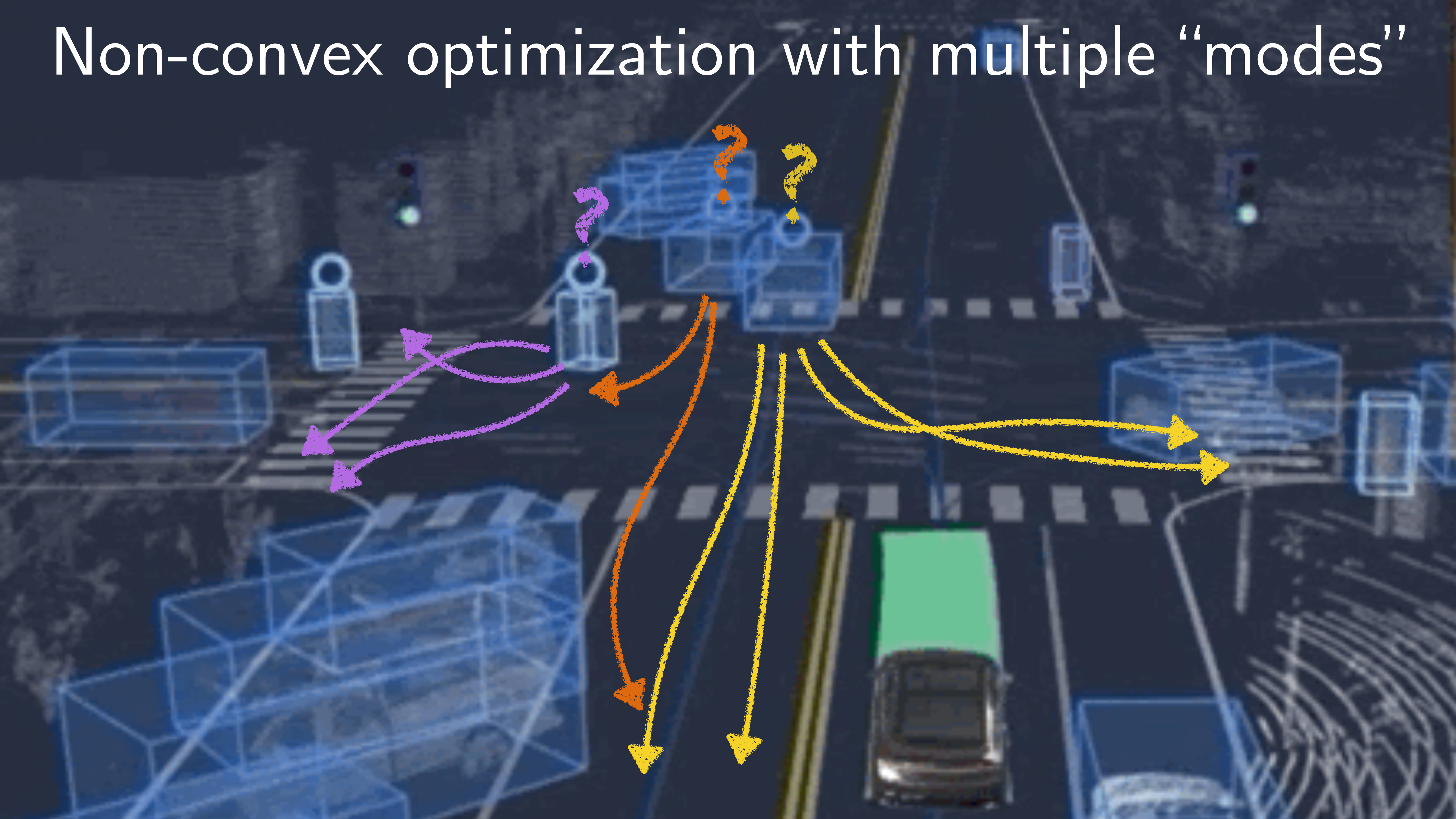
Models are useful fictions



# Optimization

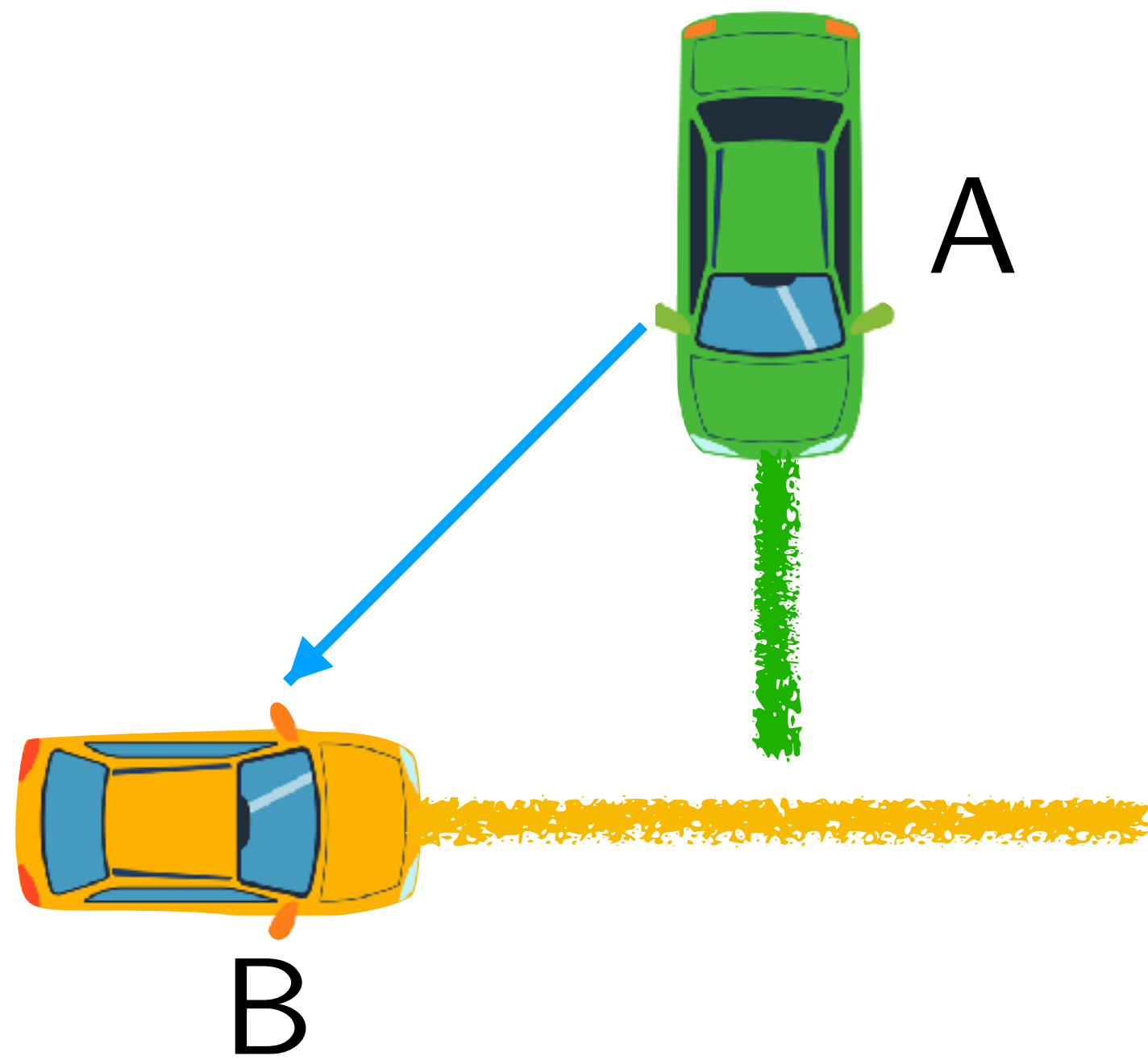
*How do we efficiently find  
the optimal sequence of decisions?*

# Non-convex optimization with multiple “modes”

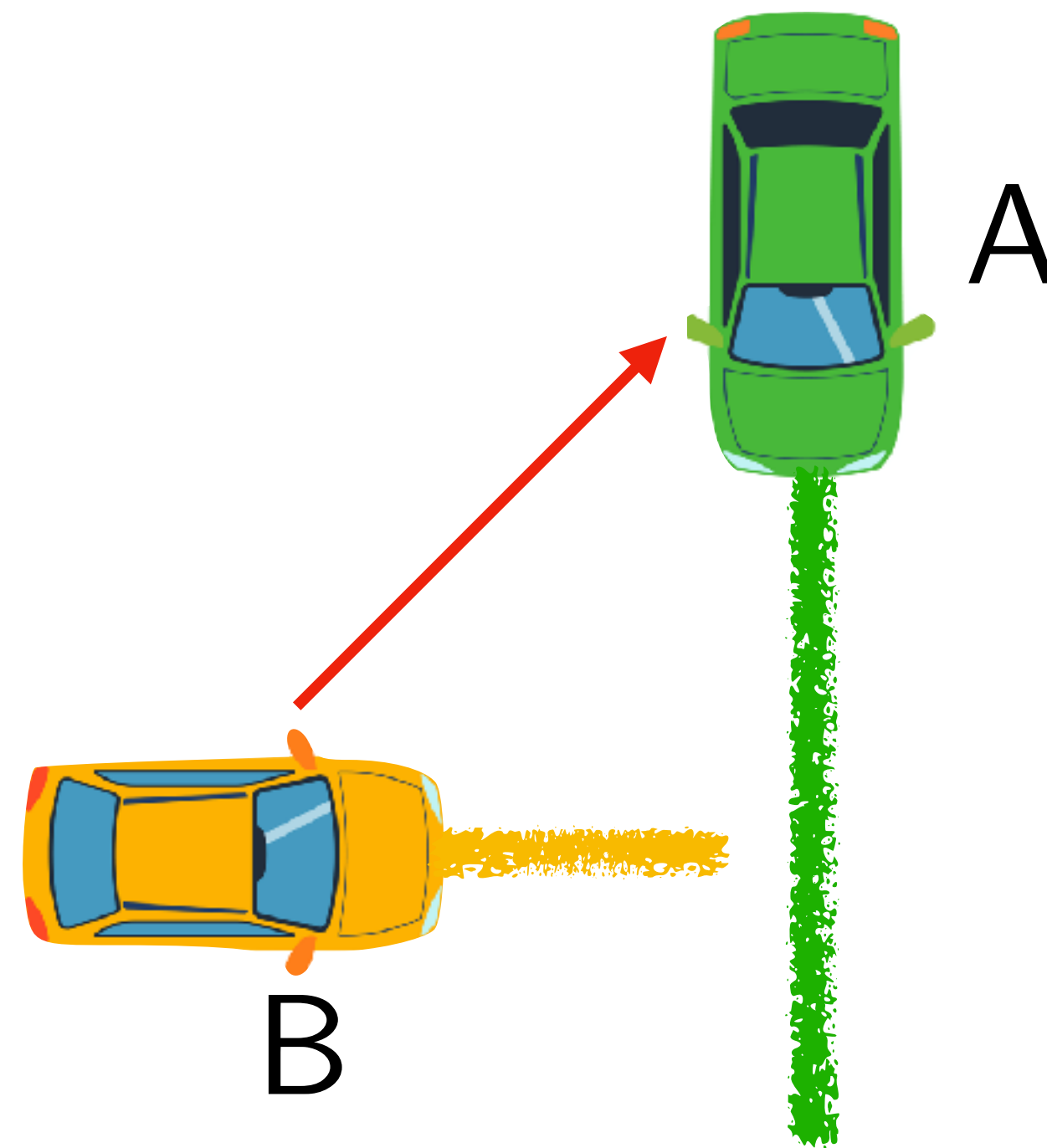


# 3 discrete **modes** of space-time paths

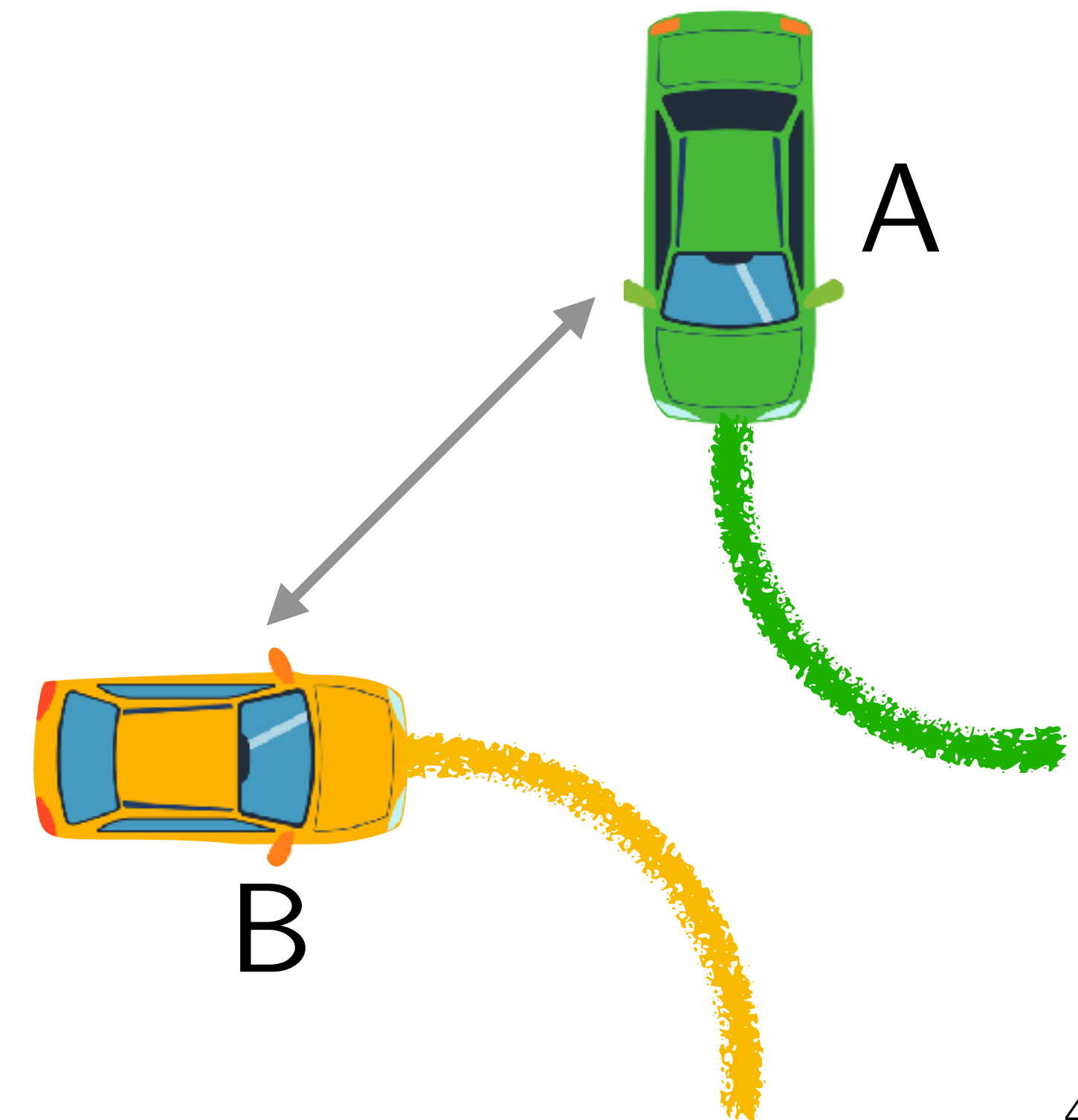
A Yields to B



B Yields to A

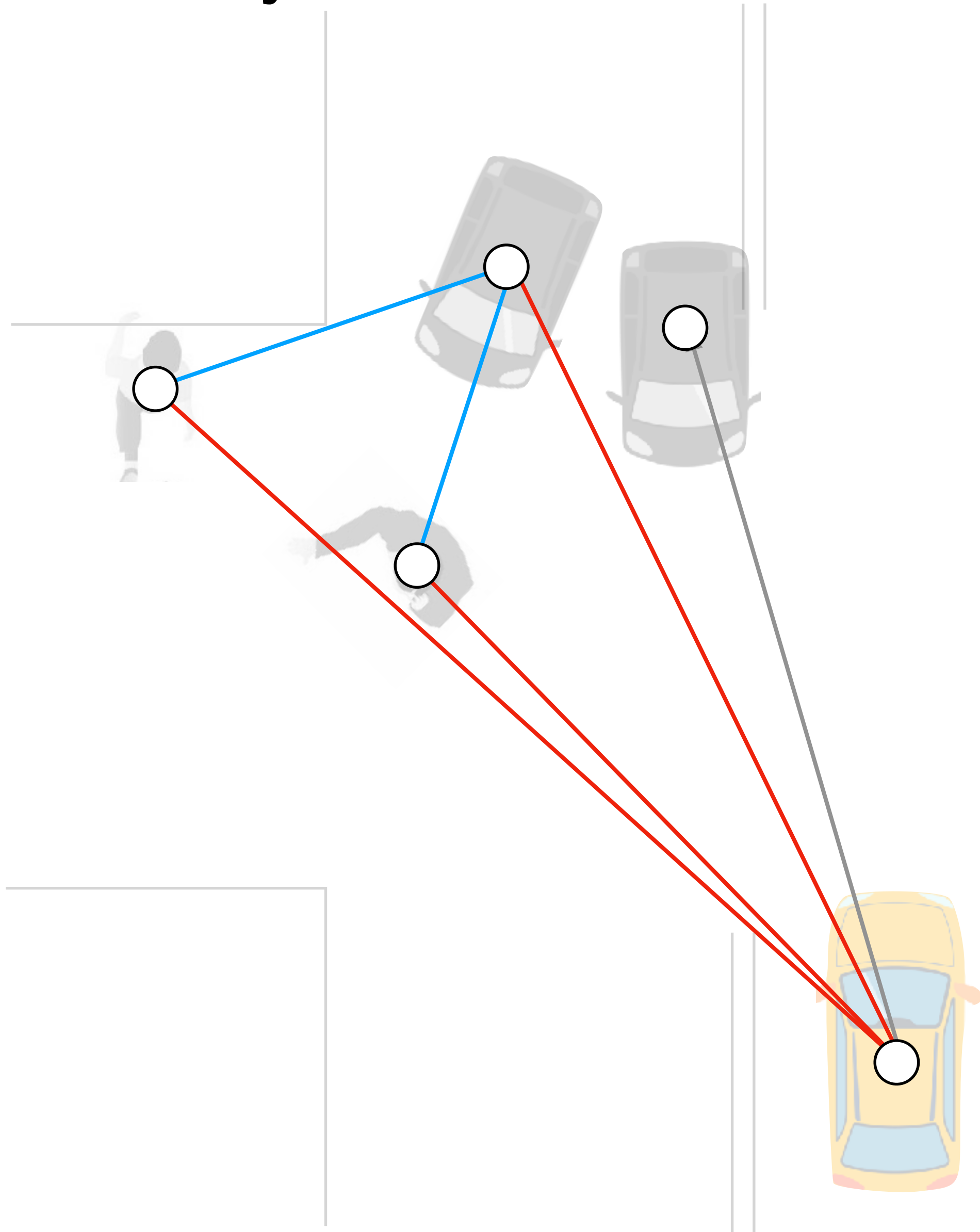


Not Yield

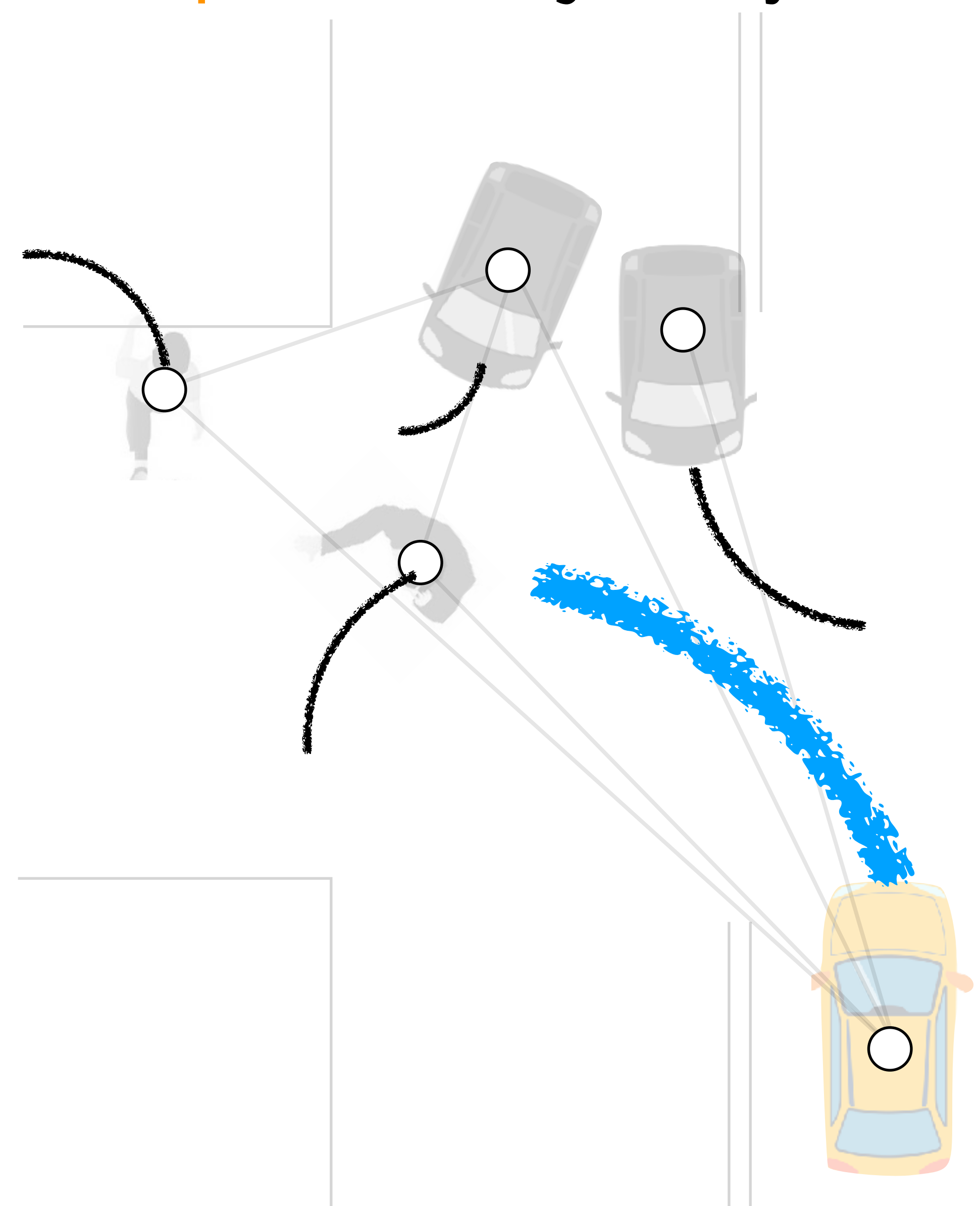




Use **learning** to recall likely discrete modes

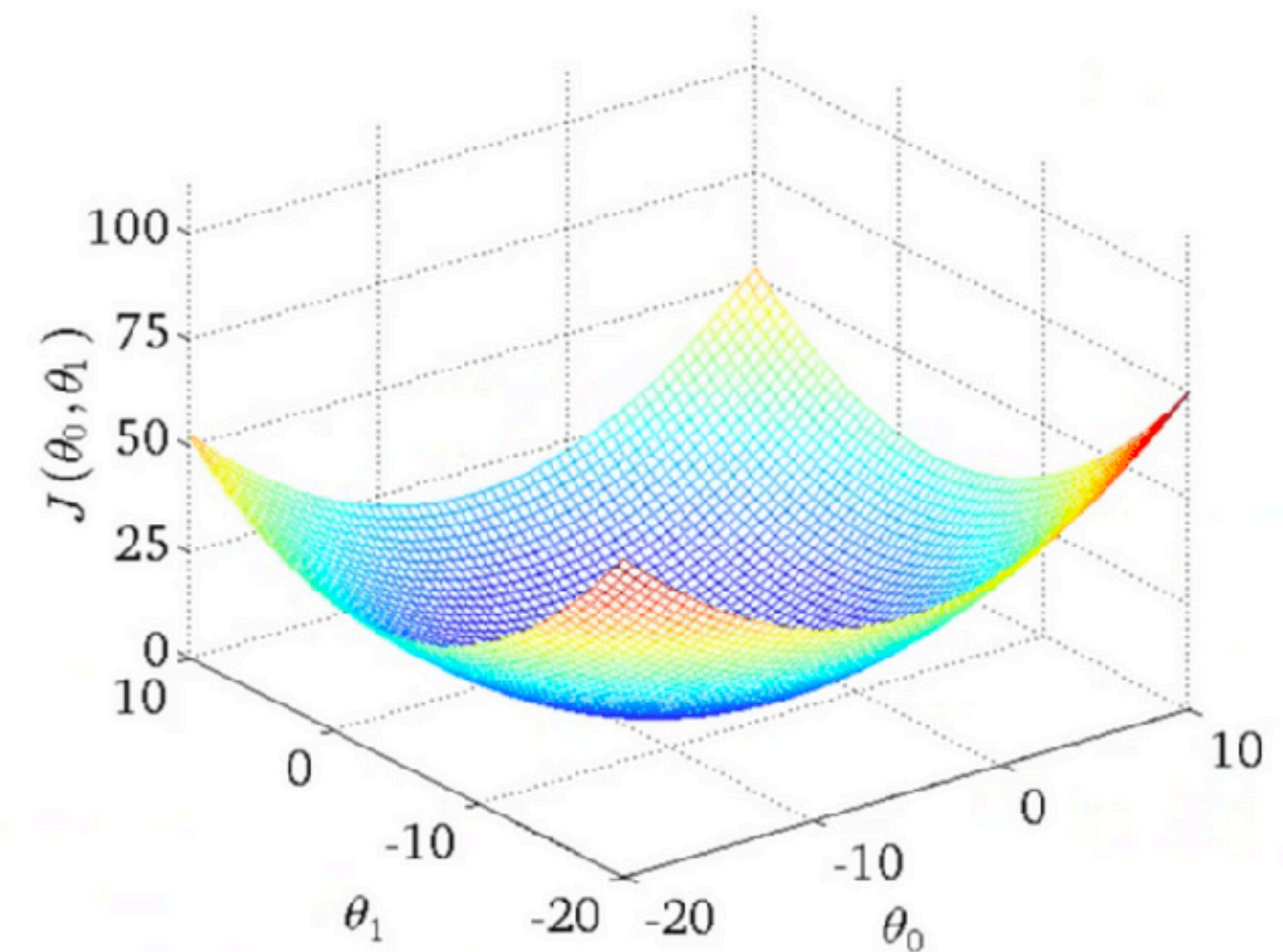
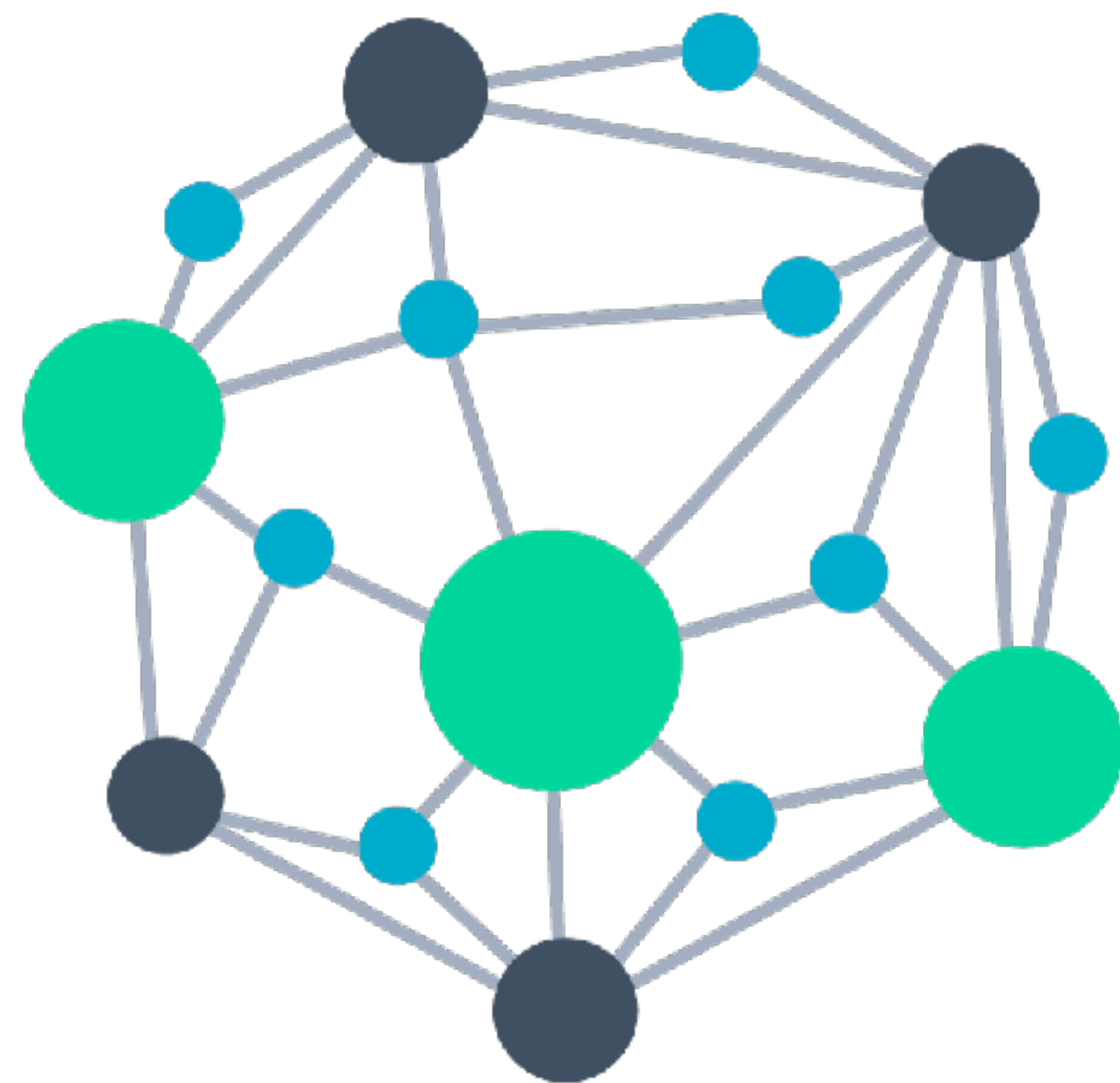


Use **optimization** to find the **precise trajectory**



# Lesson #3

ML for recall + Optimization for precision



# Optimization

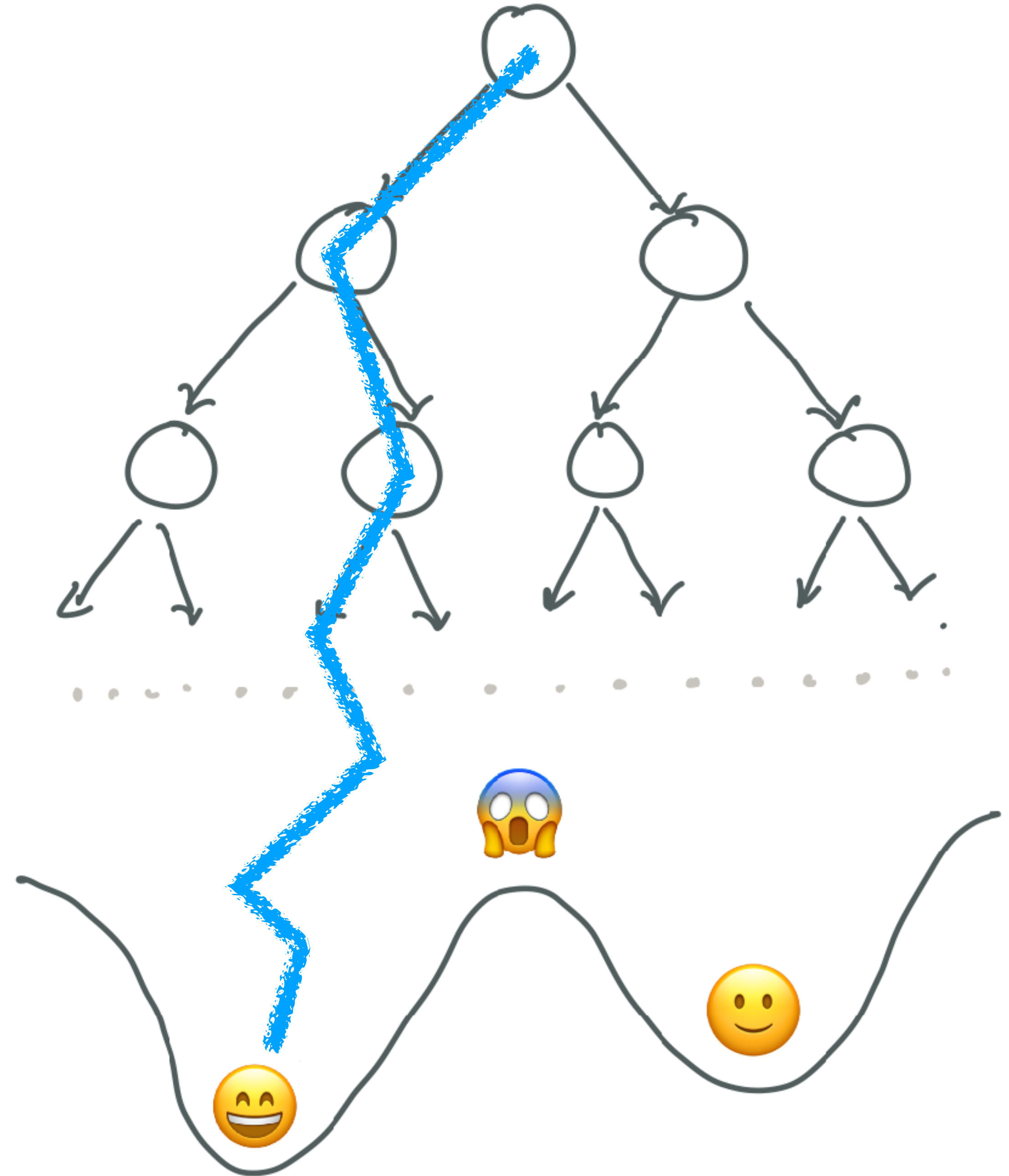
How do we efficiently find the optimal sequence of decisions?

## Models

How do decisions affect states?

## Values

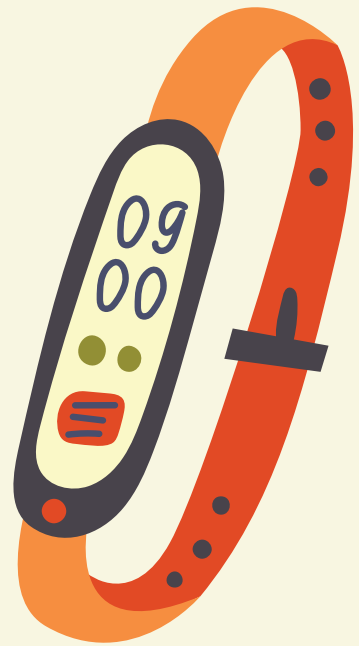
What are good / bad states?





The  
journey  
ahead!

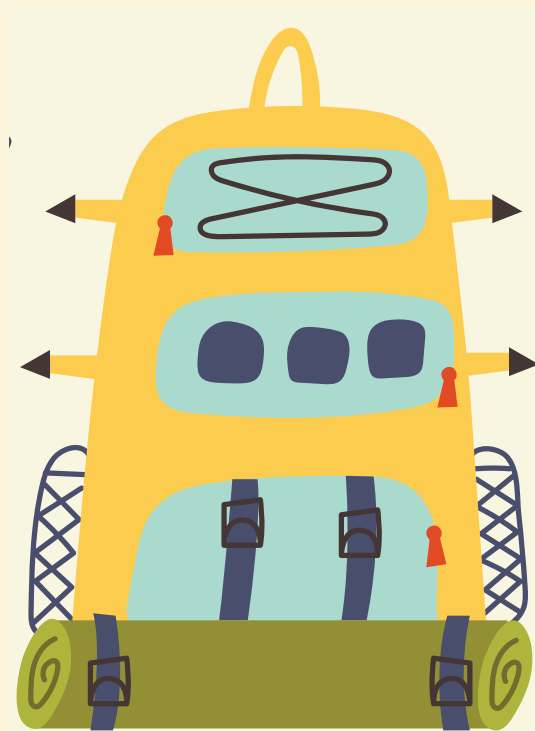
# Open Challenges



Reinforcement Learning  
(Approximate DP, TD learning,  
Policy optimization, Trust region)



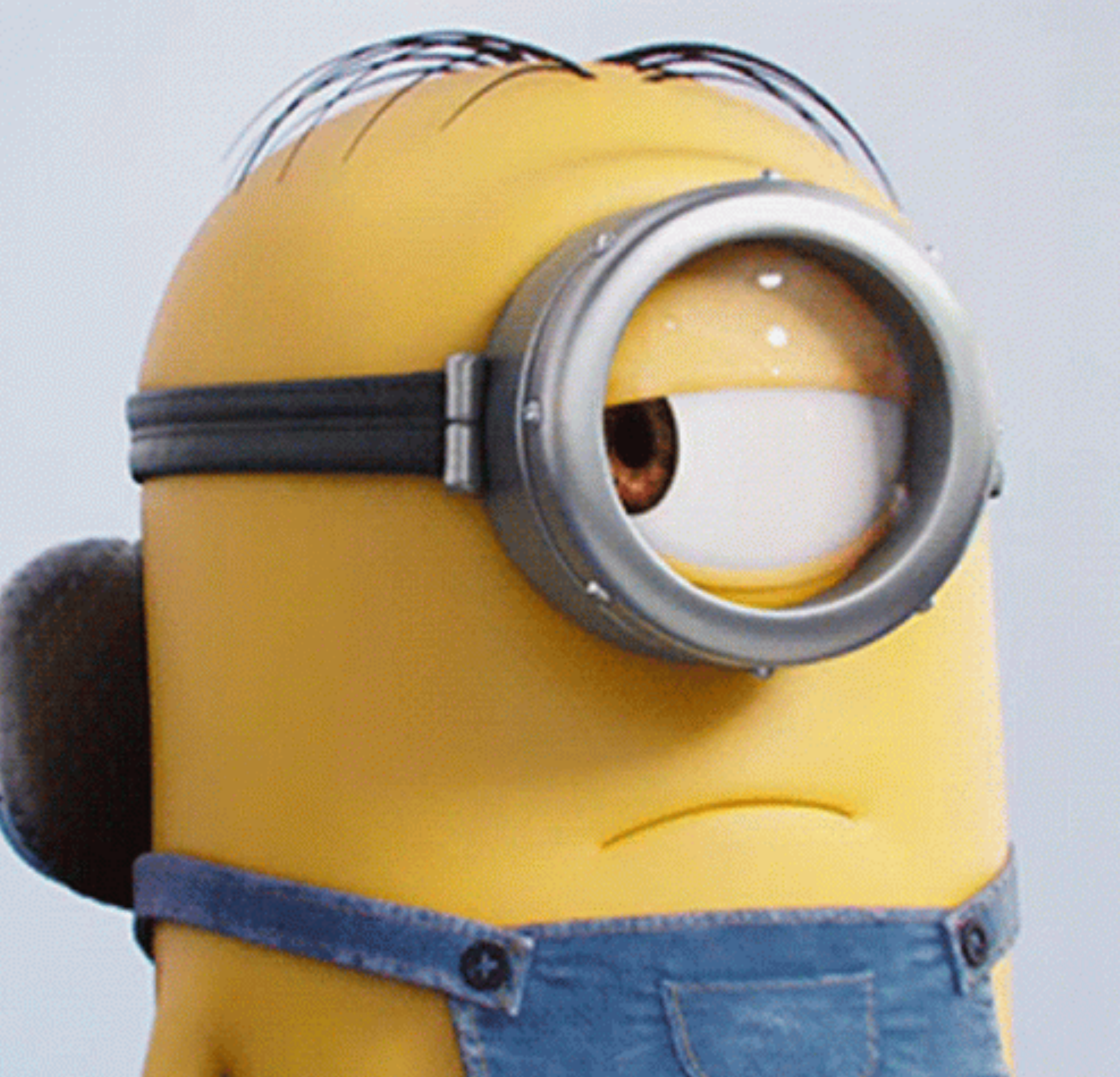
Imitation Learning  
(Behavior cloning, DAGGER,  
MaxEnt, Distribution matching)



Model Predictive Control  
(LQR, iLQR, Augmented Lagrangian)

## Fundamentals

(MDP, Online learning)



# Logistics

# TA Intro!



Dhruv  
Sreenivas

Second-year CS MS student, advised by Wen Sun.

Generally interested in reinforcement learning, with a focus in

- Imitation learning and offline reinforcement learning, specifically in continuous control
- Deep generative models and connections to model-based RL

Would love to discuss research directions with anyone interested in said topics!

# Logistics

**Website:** <https://www.cs.cornell.edu/courses/cs6756/2022fa/>

## Lectures

Assigned pre-reading (focused), lectures for interaction

**Assignments** [3 assignments \* 15% grade = 45%]

Python. HW2, HW3 involve PyTorch. Maybe a little theory. Done individually!

**Project** [45%]

Final project. Pick a research problem, apply techniques from class. Be creative!

Groups of 2. Extended abstract, final presentation, final paper. Best paper award!

**Participation** [10%]

Interaction during lectures and/or Online discussions. Help everyone engage!





# Course tools

Course Website: The ONE true hub for all information. Please check this frequently and surface any errors or sources of confusion.

Ed: The discussion forum where all announcements are sent, where all student-TA and student-student communications occur.

Gradescope: Where all assignments and projects are submitted.

Canvas: Limited to no use.

# Books and other resources

Work-in-progress book

Modern Adaptive Control and Reinforcement Learning,

James A. Bagnell, Byron Boots, and Sanjiban Choudhury

(Please feel free to send me feedback)

For other resources, keep checking website



# Assignment 0

Simple survey

- What are you hoping to get out of this class?
- Familiarity with concepts
- Etc

Released soon (today..), due end of the week

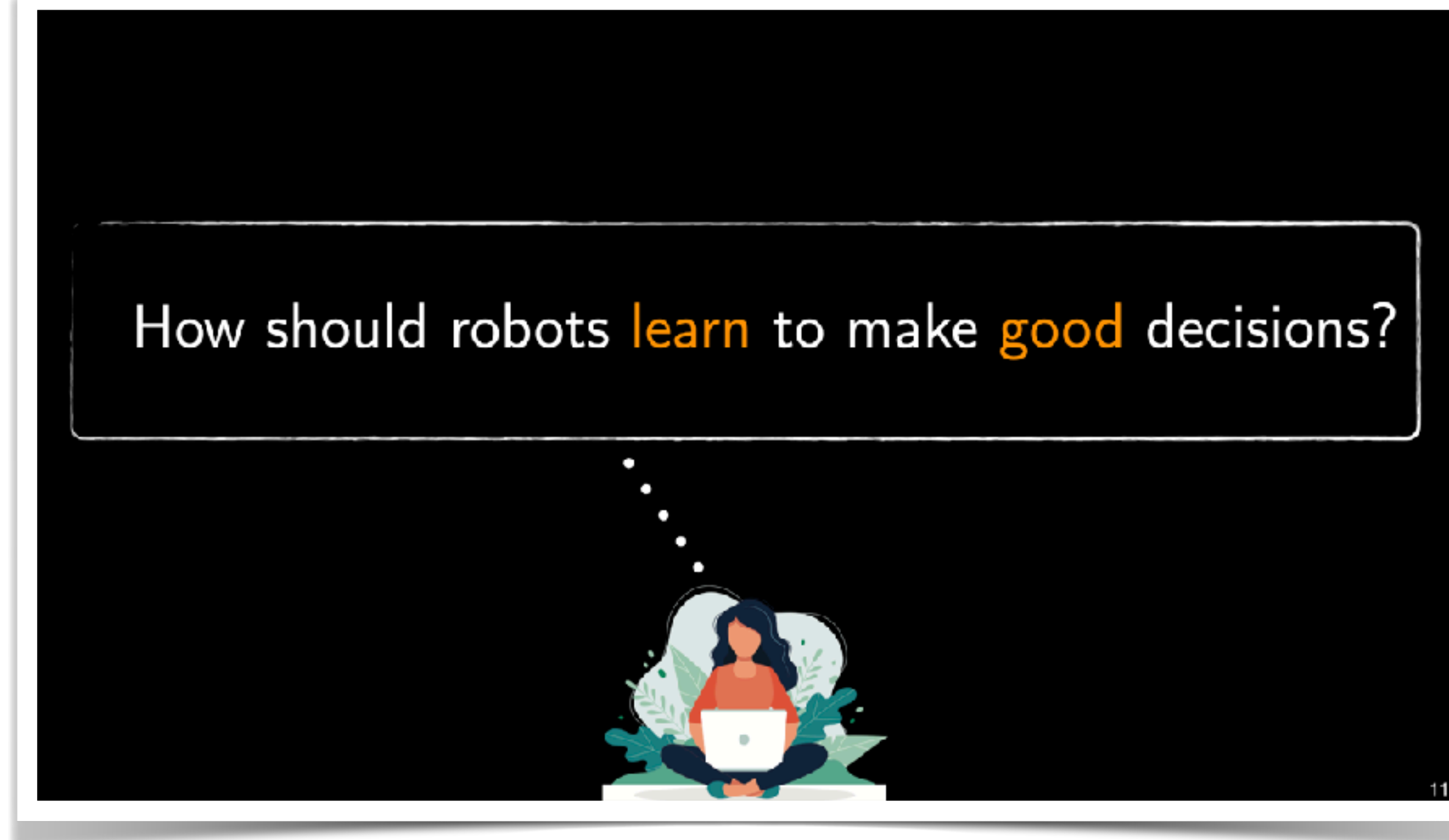
Mandatory!

# Questions?

Things that will happen soon:

- Waitlist moved over to enrolled
- Office hours finalized

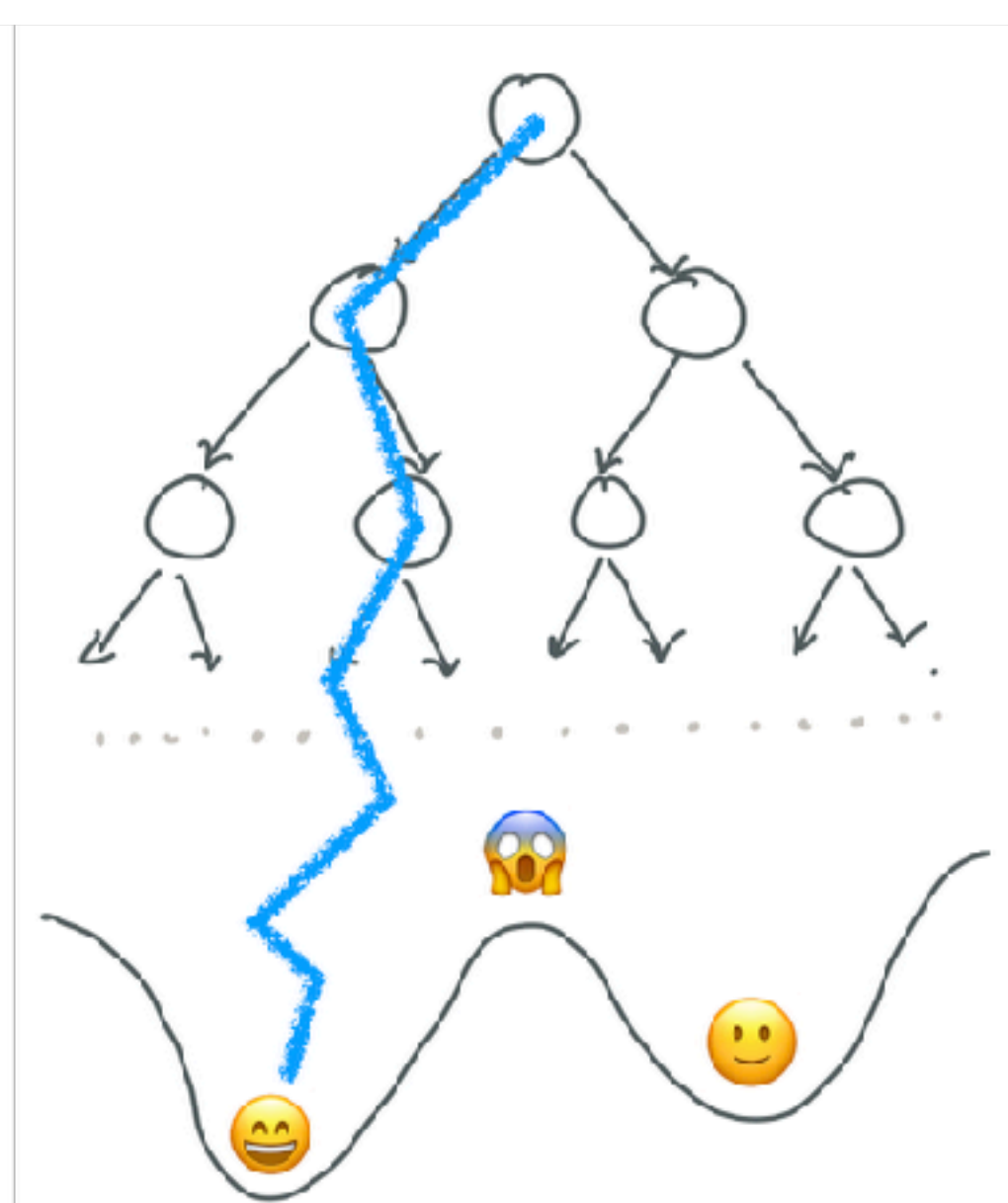
# tl;dr



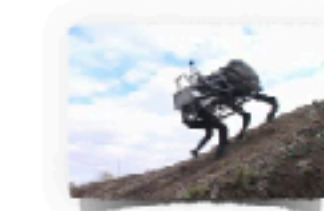
**Optimization**  
How do we efficiently find the optimal sequence of decisions?

**Models**  
How do decisions affect states?

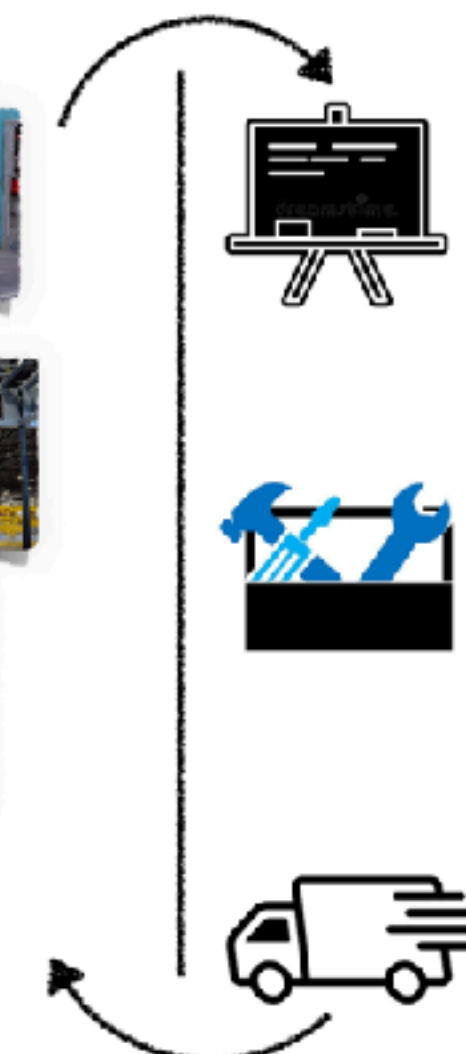
**Values**  
What are good / bad states?



## WHY this course?



Take *any* robot application



**Formulate** as a Markov Decision Problem (MDP)

**Solve** MDPs using all-purpose toolkit  
(Imitation/Reinforcement learning, Model based/free)

**Deploy** learners in real-world  
(Safety, distribution shift, value alignment)