Lecture 9:

Generating Behaviors: AI Agents for Virtual Worlds

Visual Computing Systems

Stanford CS348K, Spring 2025

Why is there interest in creating agents that Can successfully perform tasks in virtual worlds?

Application area 1: robotics

Train in simulation to learn behaviors that work in the real world.

Train embodied agents in simulation prior to deploying them in the real world

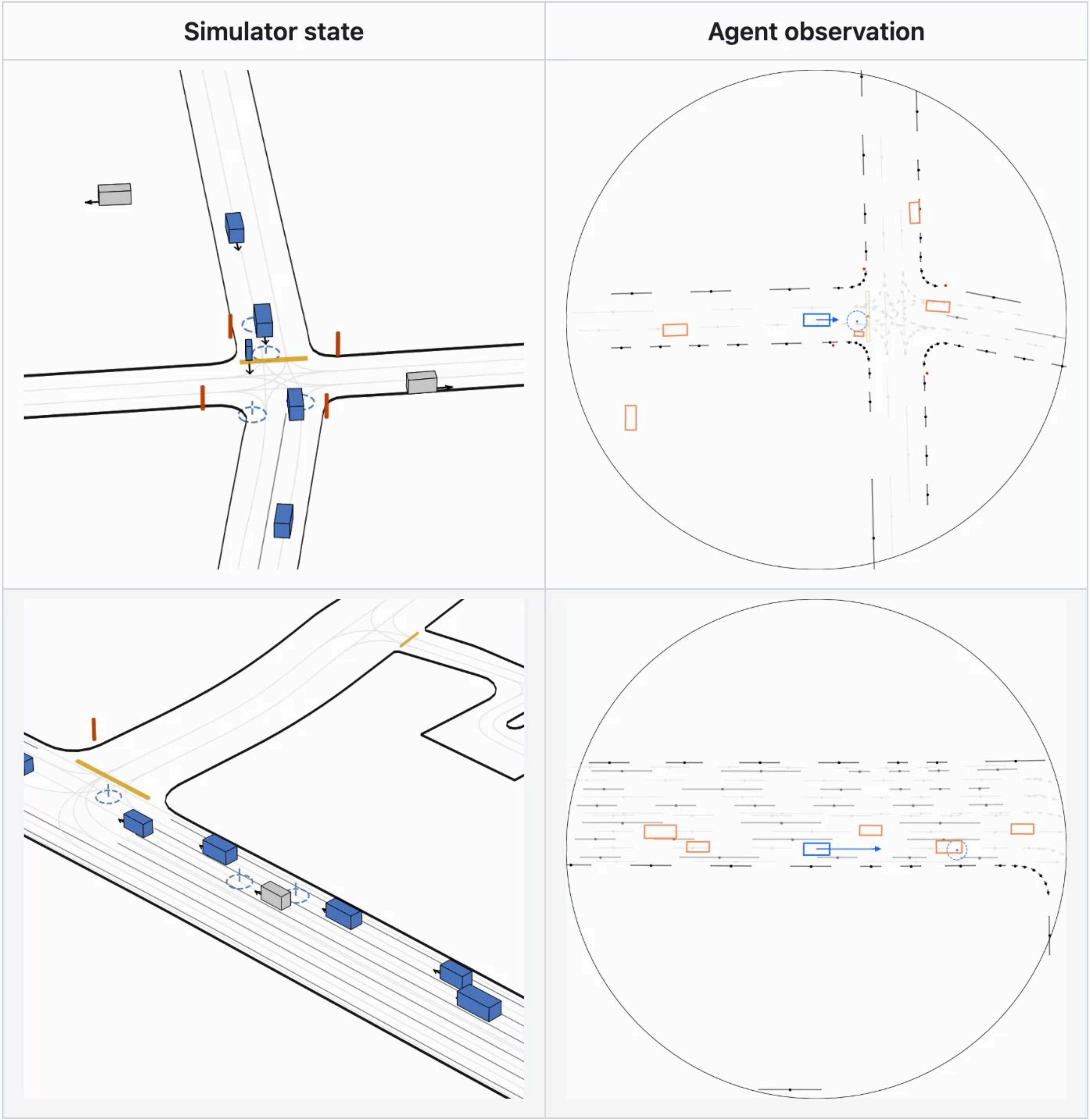
Using the virtual world as a proxy for the real world.

Example task: navigating an autonomous vehicle

NV Drive Sim

Simulation need not be "photorealistic"

- Choose to simulate at level of detailed needed for task at hand
  - e.g., separate needs of world perception from planning



Train embodied agents in simulation prior to deploying them in the real world

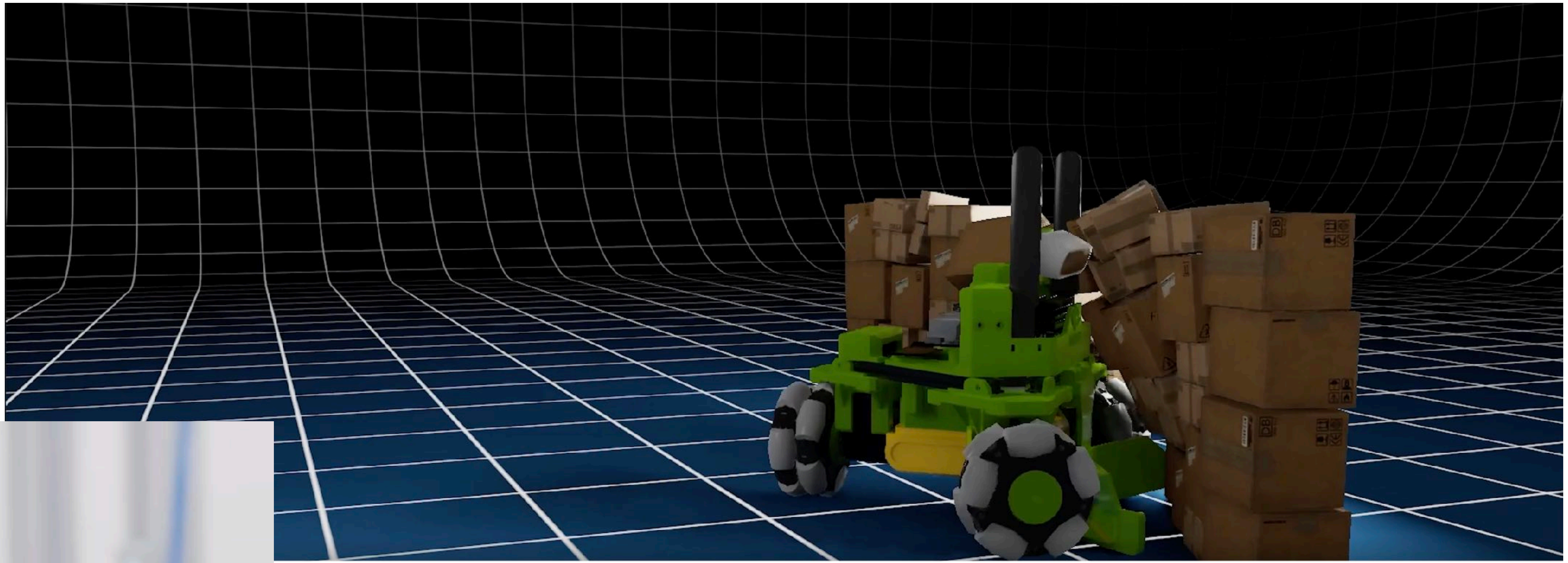
Using the virtual world as a proxy for the real world.

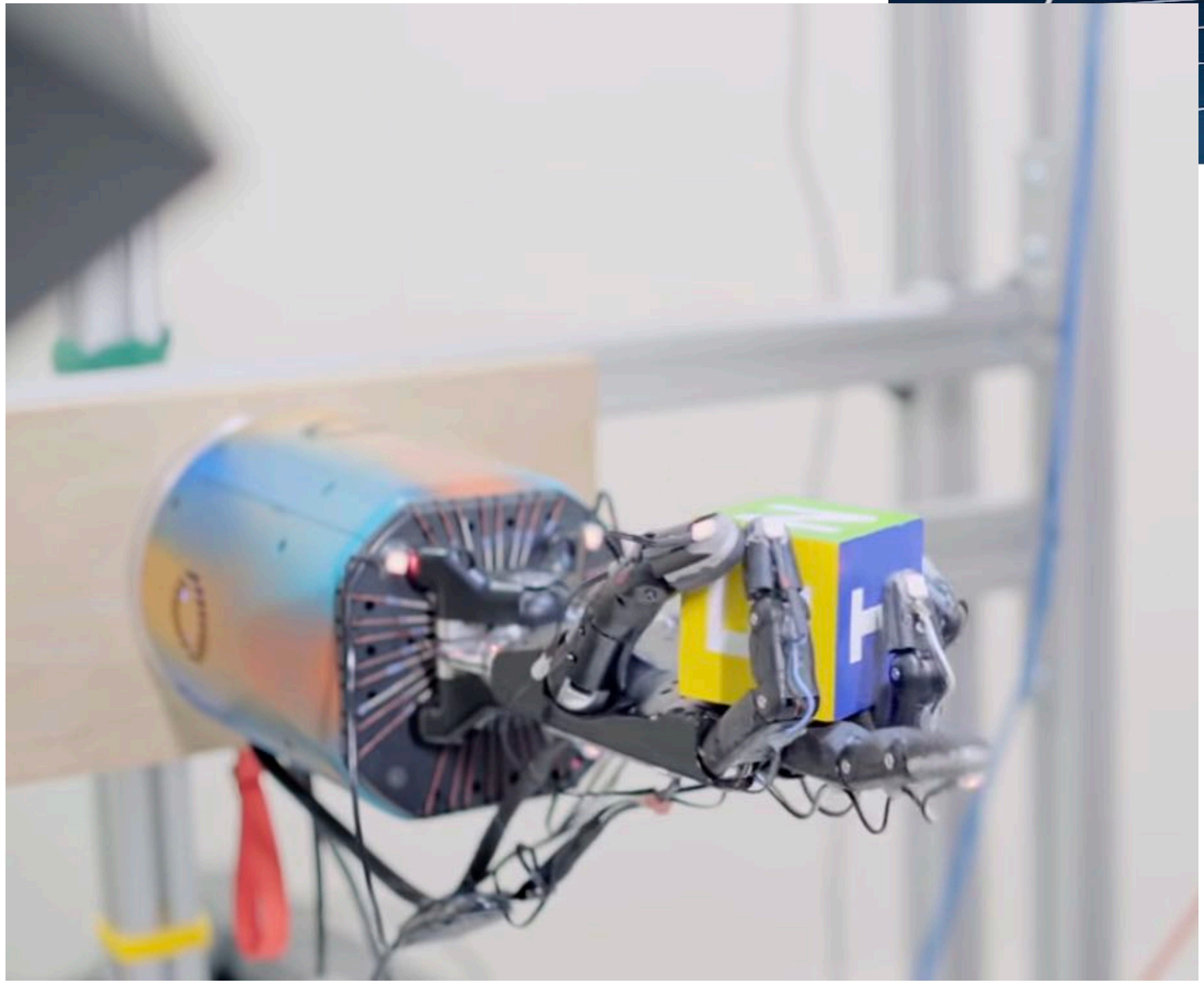
Example task: navigating the home and manipulating items in the home.

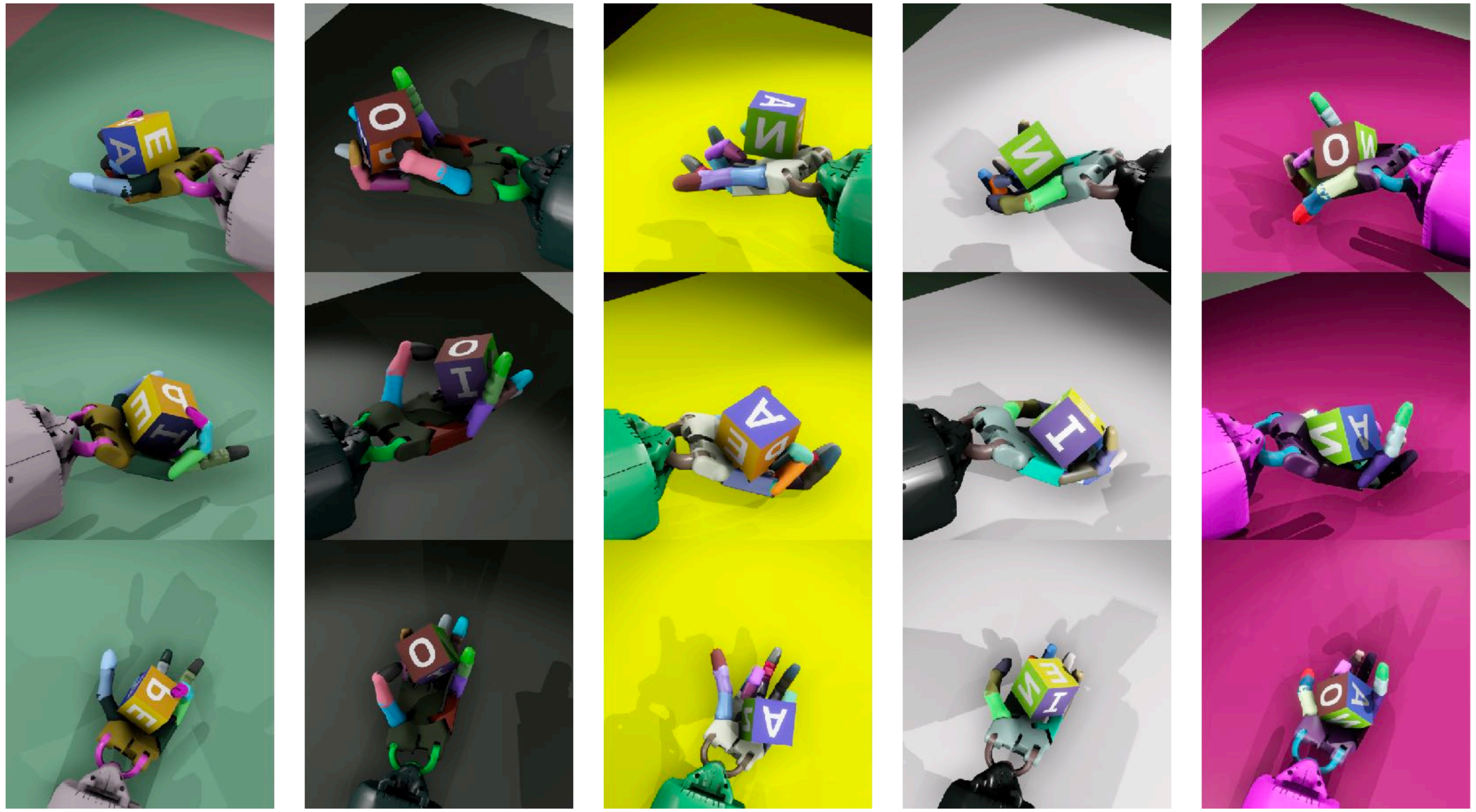
Szot et al, NeurIPS 2021

Dexterous manipulation

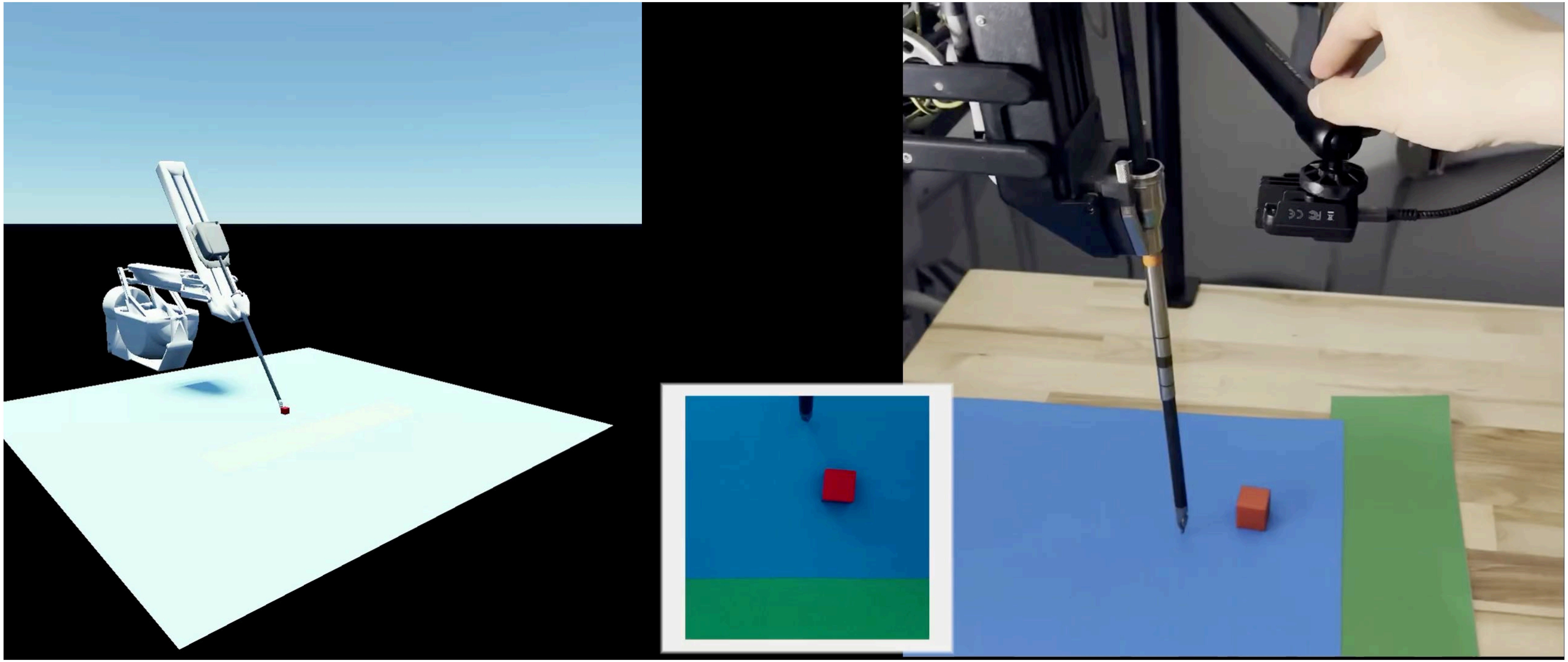
Example task: precise manipulation of objects







Simple sim-to-real transfer with domain randomization



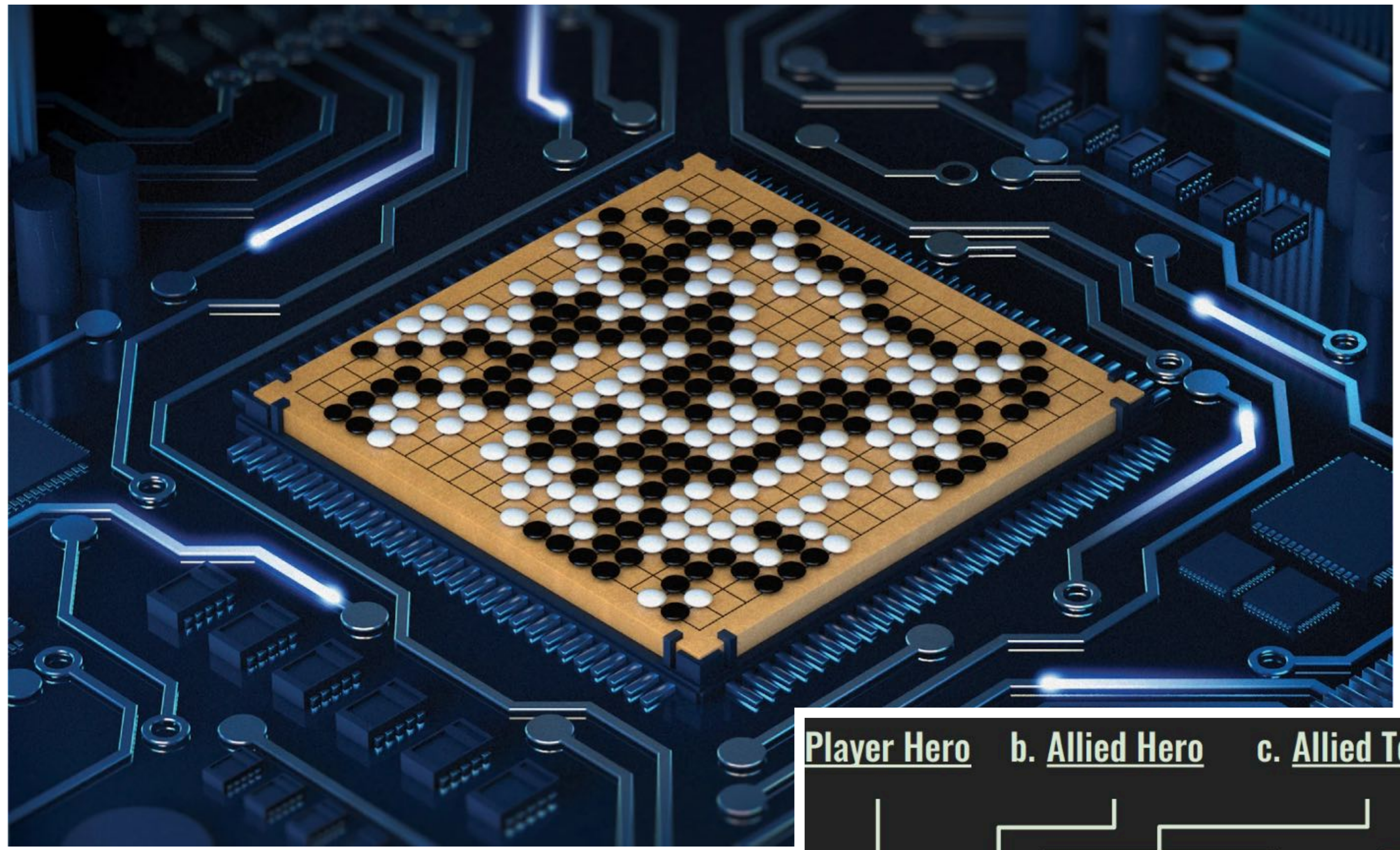
Application area 2: video game agents

Train agents to function as autonomous game players

(Both in pursuit of better bots, and as a pure science exercise)

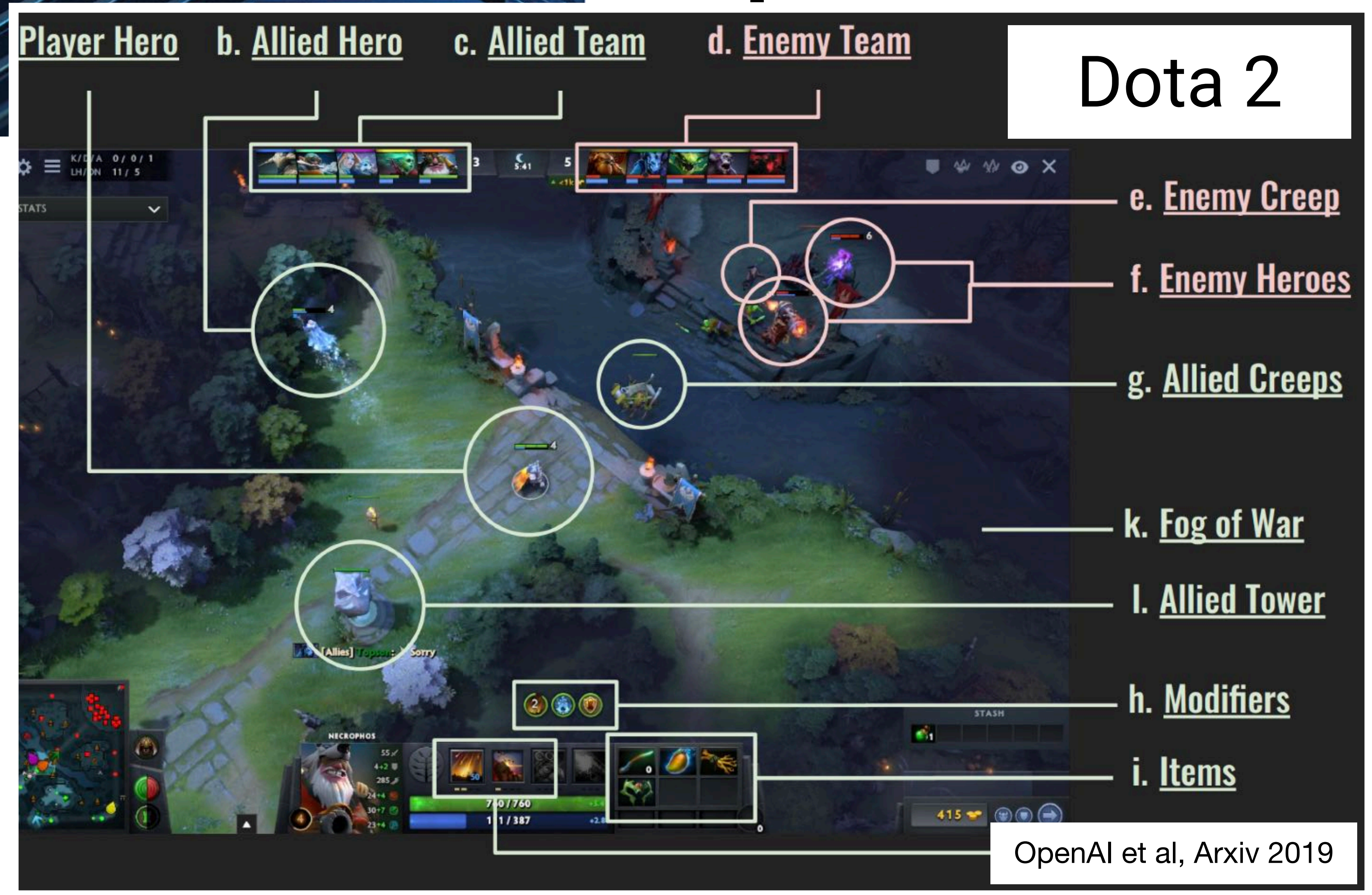
Mid-to-late 2010's: pursuit of superhuman performance via large-scale deep RL

DeepMind AlphaGo



Demonstrations of large-scale use of reinforcement learning to learn "creative" play and expert levels skills

OpenAI 5 (Dota)

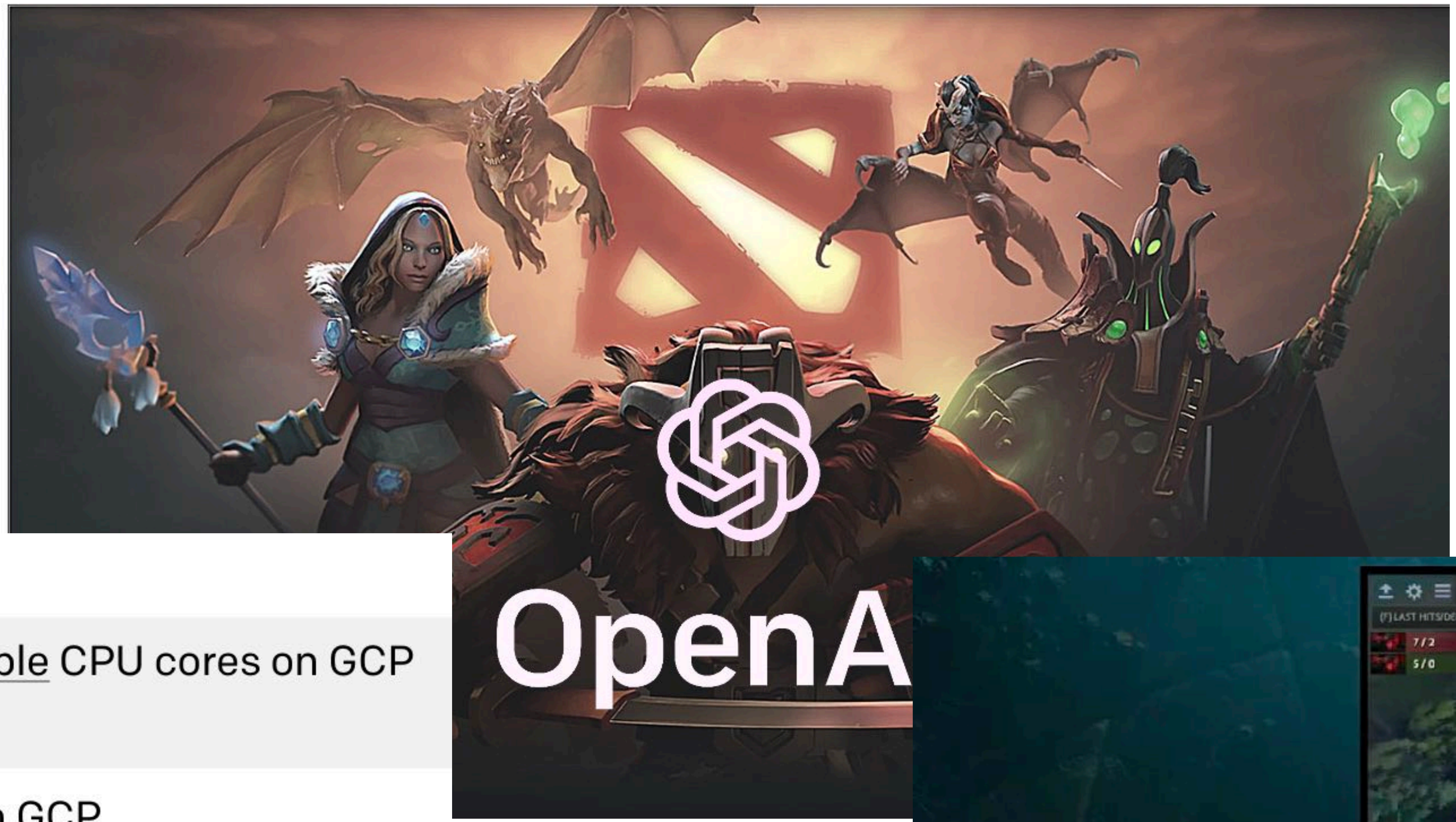


DeepMind AlphaStar


Video Credit: OpenAI


Random incoming ball + Random target
Practice in simulation…

Scale of experience collected: OpenAI's "OpenAI 5" Dota 2 bot



	OPENAI FIVE
CPUs	128,000 preemptible CPU cores on GCP
GPUs	256 P100 GPUs on GCP
Experience collected	~180 years per day (~900 years per day counting each hero separately)
Size of observation	~36.8 kB
Observations per second of gameplay	7.5
Batch size	1,048,576 observations
Batches per minute	~60

Cam play uses of virtual characters in interactive experiences

- Virtual teammates in team-based games
- Narrative elements
- Coaches, etc.

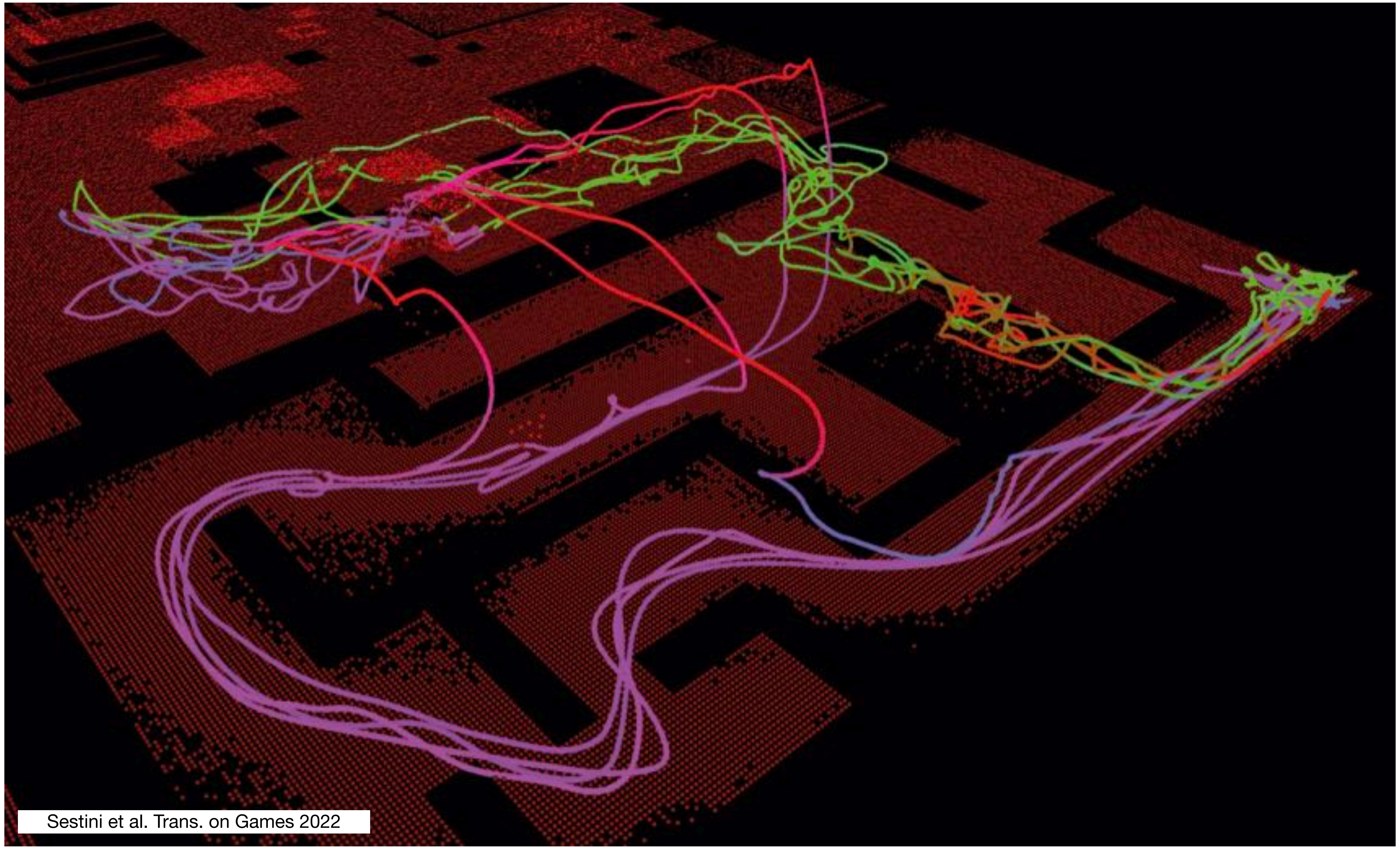
Learning to race in Tran Truism Sport
[Wurman et al. 2022]

Application area 3: video game "verifiers"

Using agents as proxy play testers to aid game design

As proxies for predicting future behavior of real humans

- Feedback on game design
- Game testing



Where can a player get to on a map?

What sections are not reachable?

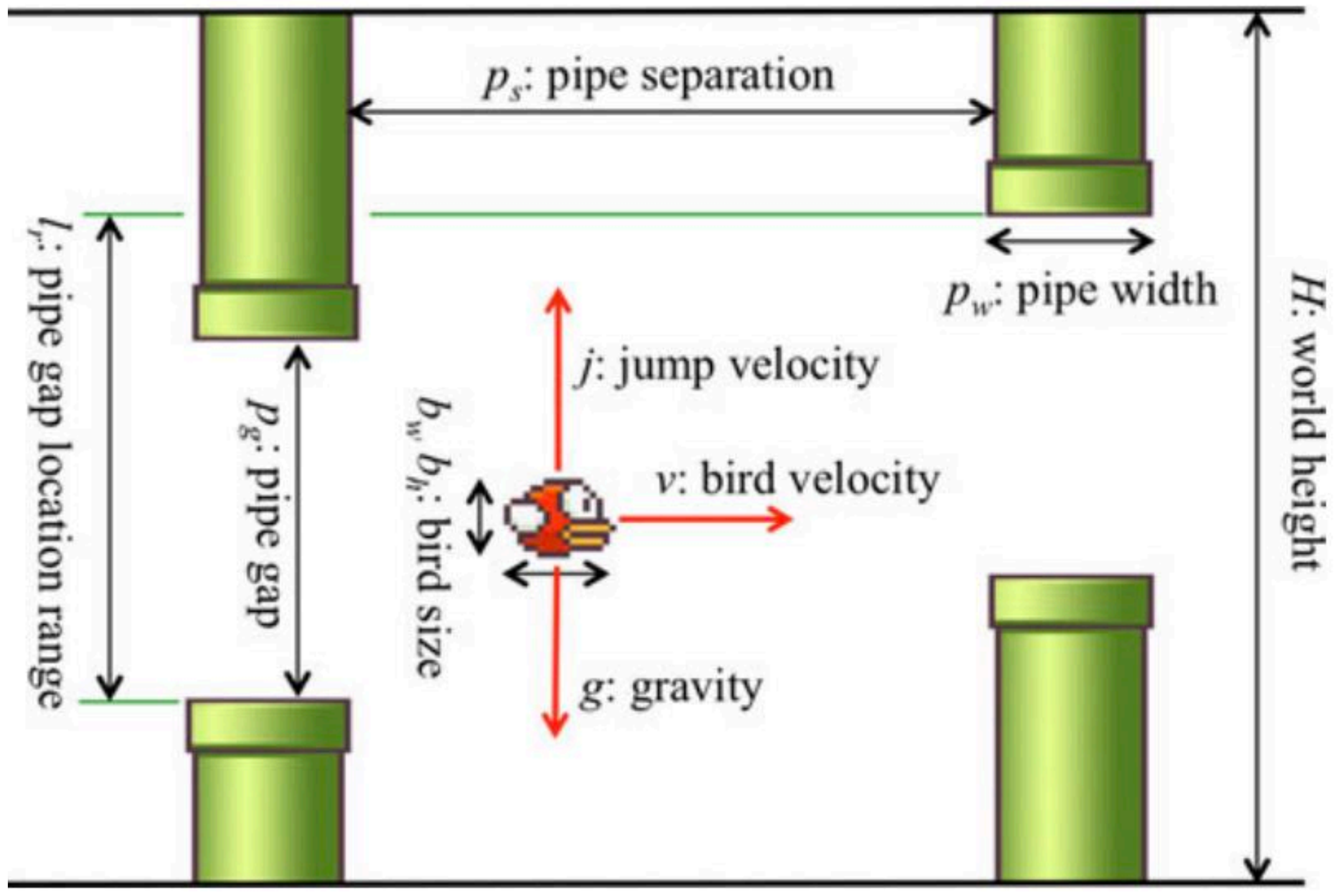
What sections are reachable but should not be?

How long does it take to get there?

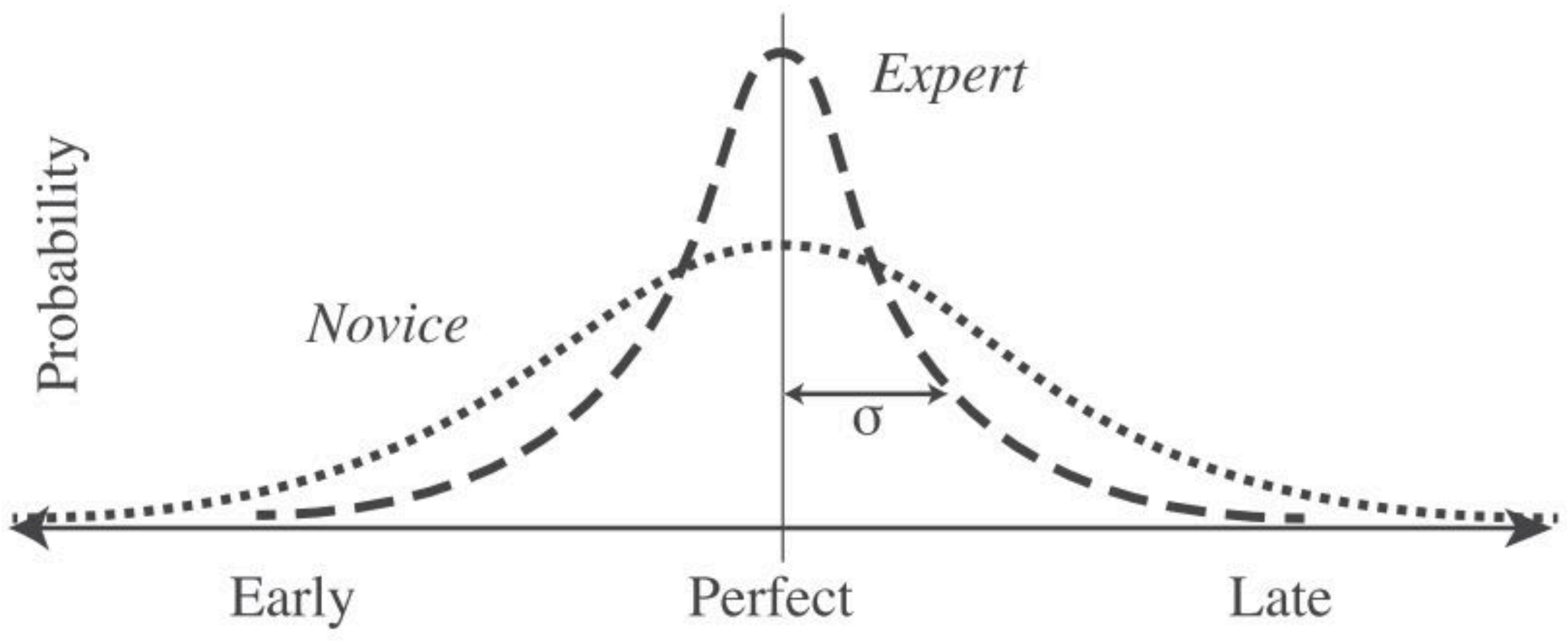
As proxies for predicting future behavior of real humans

How difficult will future players find a game? And can we adjust difficulty to meet certain experience goals?

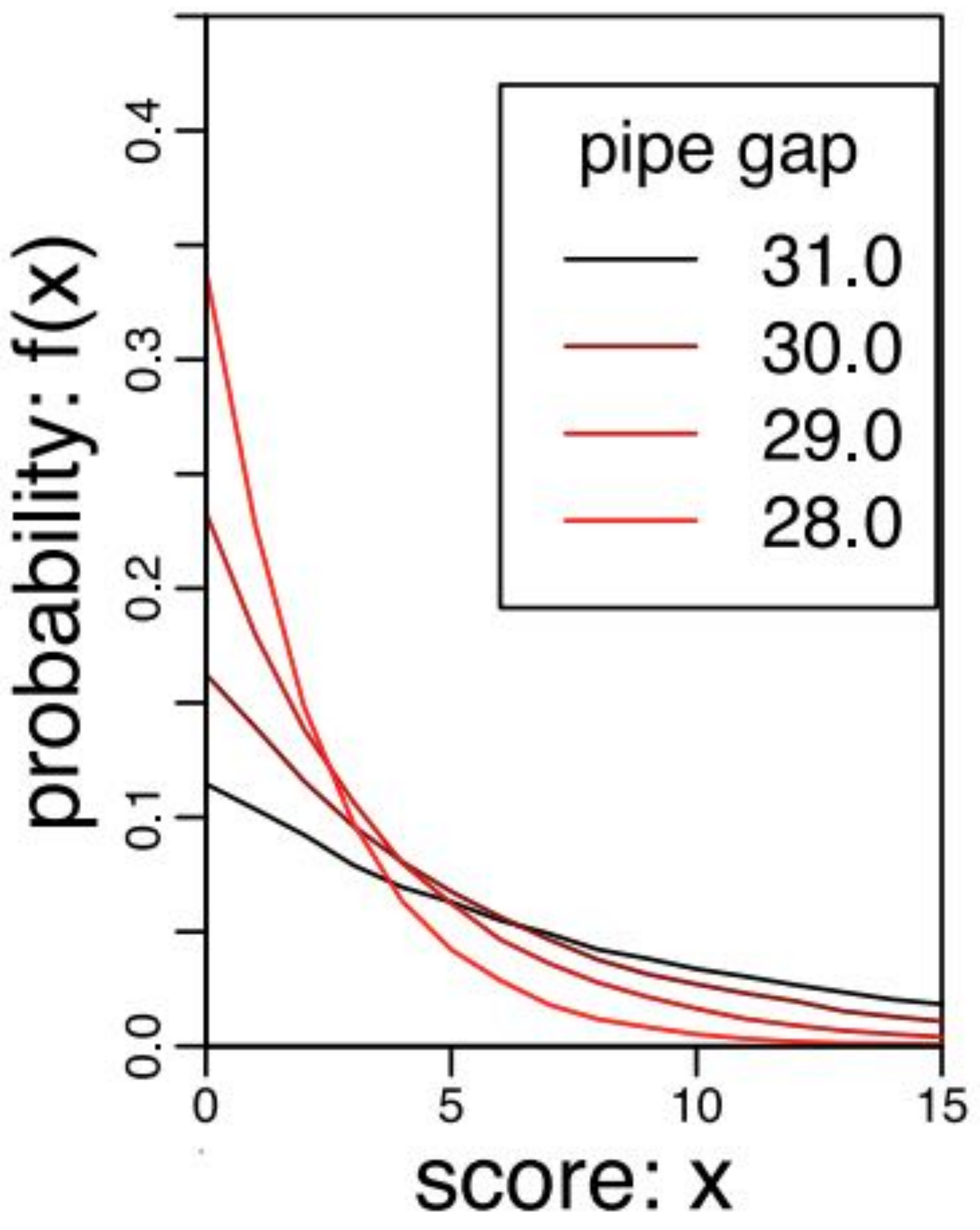
Example from tonight's reading: [Isaksen et al. 2018]



A parameterized game



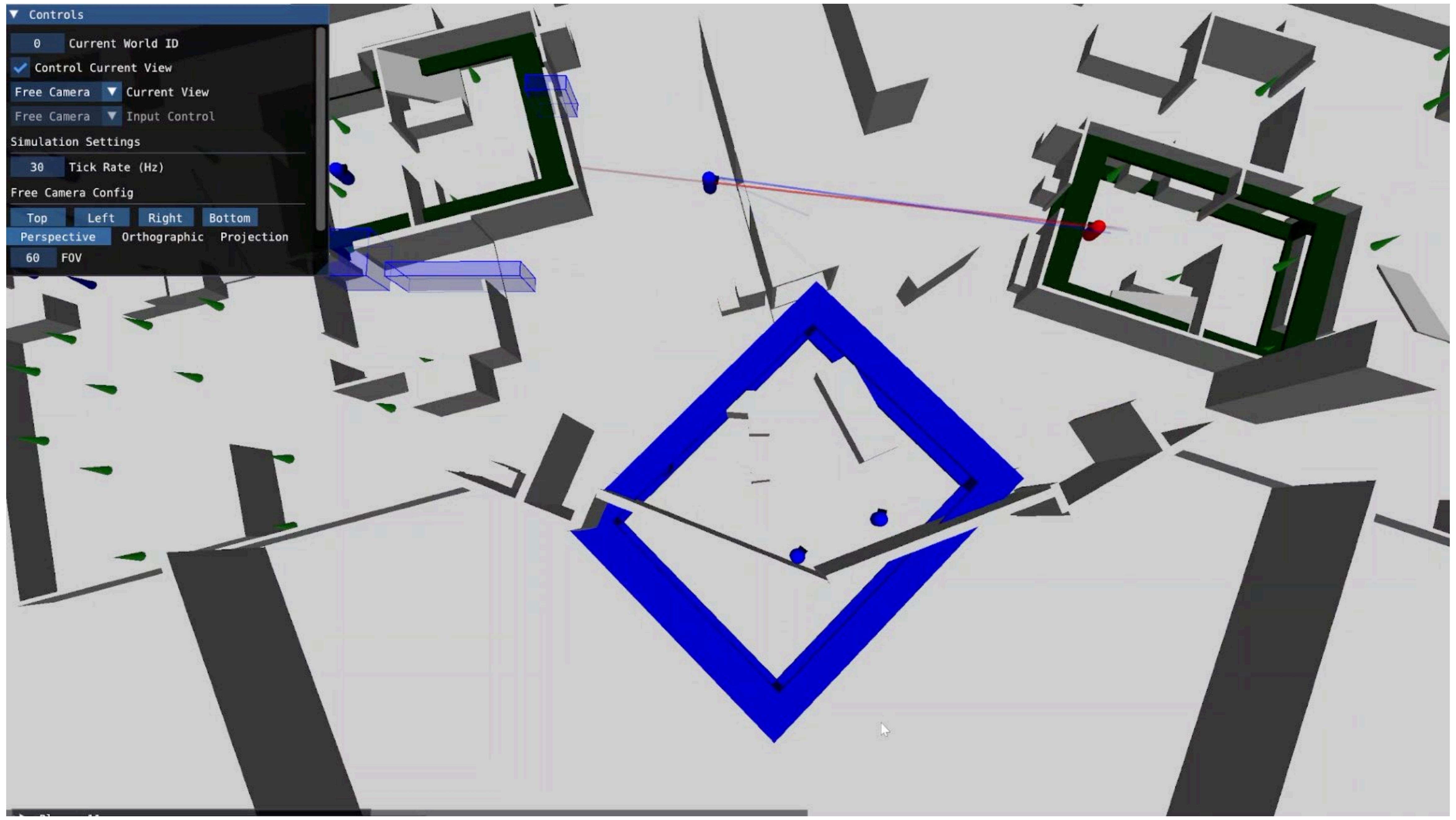
A model of player skill
(Deviation from optimal play)



Running agent in game generates statistics of play (expected score)

In an area of generative AI, many interesting possibilities for using AI agents to "verify" created content

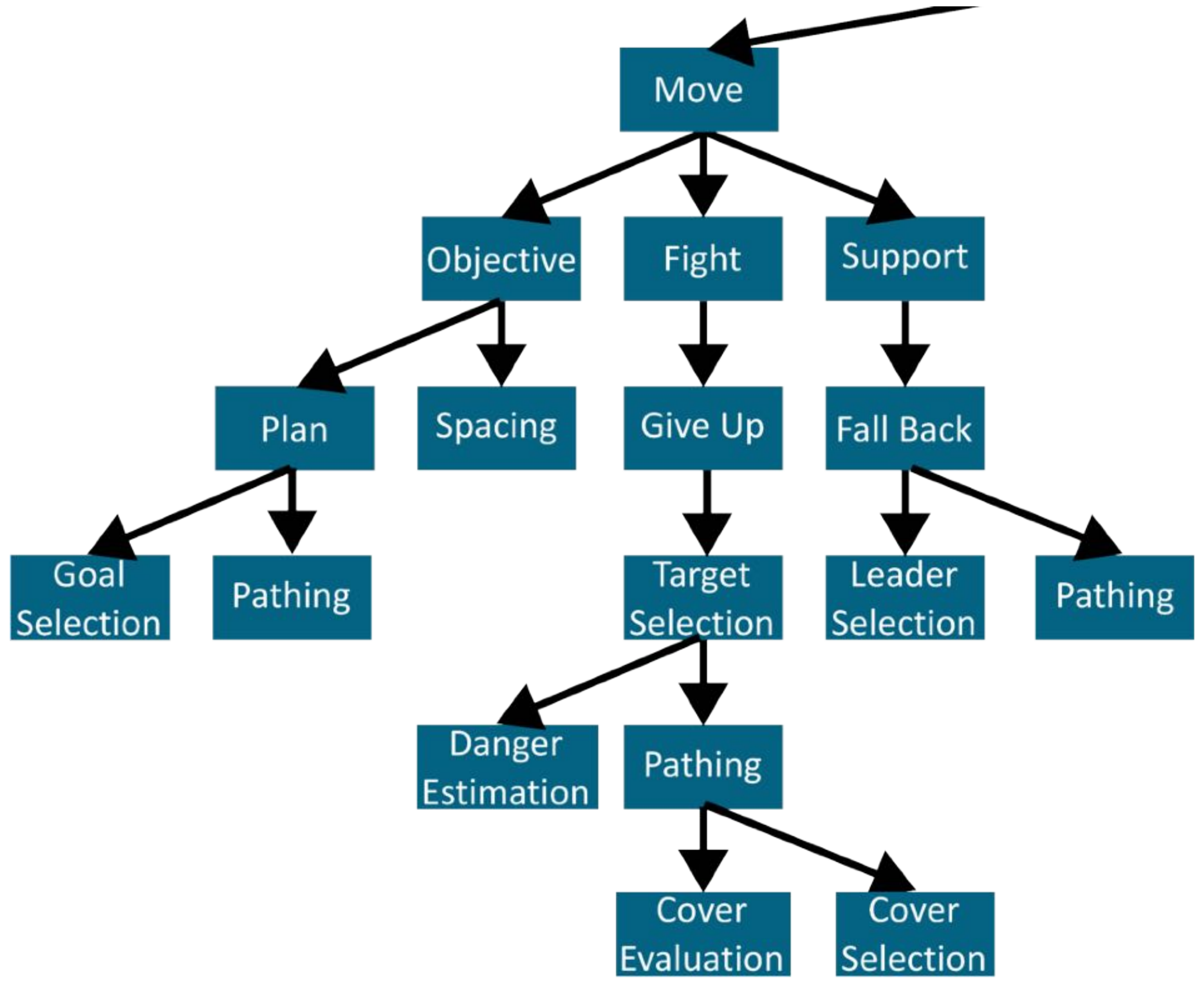
- Is a map for a FPS game fair?
- Does a puzzle have a solution?
- How does changing the damage done by a weapon change game strategy?



Strategies for designing agents

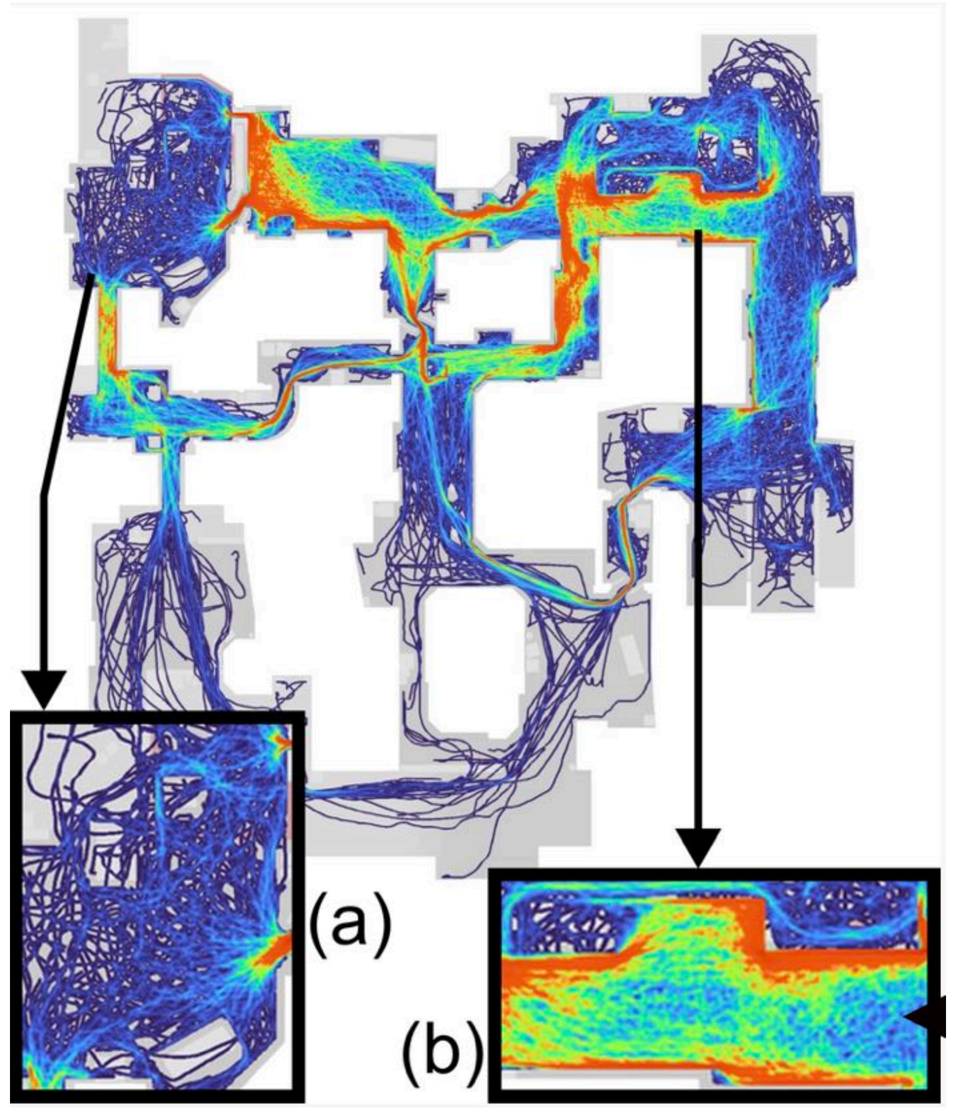
- Major theme of upcoming readings/lectures
- Pros/cons of each approach

Behavior defined by human-crafted rules



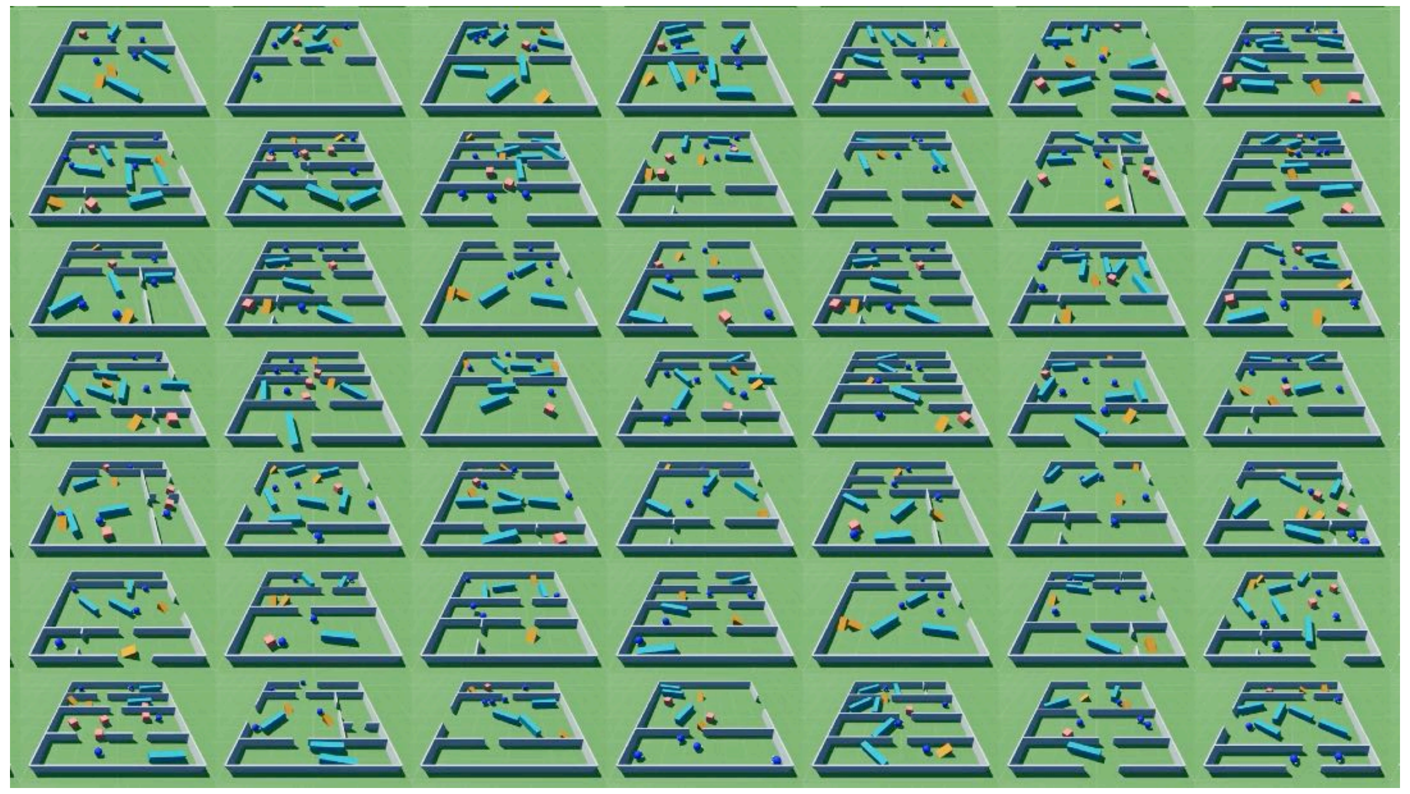
Good: controllable
Bad: formulaic behavior, brittle, laborious to create

Learn from data (traces of human play)



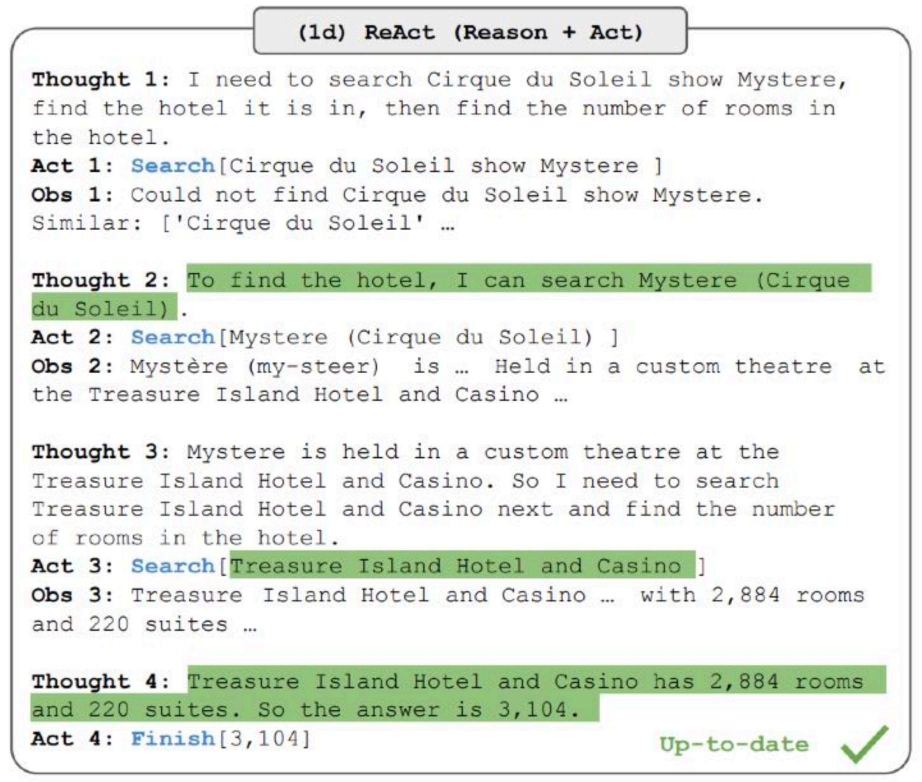
Good: "humanlike" by construction
Bad: data collection challenges, Hard to generalize to unseen situations (need massive datasets)

Learn from brute force trial-and-error (RL)



Good: superhuman skill potential,
Bad: high cost training, difficult to successfully train, hard to control behavior

Leverage common sense embodied by LLM



Good: low effort to create, surprising adept at many tasks
Bad: lack of grounding in actual environment, high runtime cost, unpredictable failures

Full-stack challenges

- Agent design challenges

- Systems challenges
  - High-performance execution of training for millions/billions of trials
  - Large-scale logging of human traces, new data selection challenges

- Evaluation challenges
  - How do we design automated metrics of success beyond just "success on task"?
  - What is an agent that is "human like"? "Fun to play with?", "Lacks this skill"
  - You'll see an increasing emphasis on quality of evaluation in our in-class discussions, because evaluating this work is a challenge on its own!

Example thought experiment

- You are trying to train an agent to be "fun" to play with in your favorite FPS game
- What does "fun" mean, and how would you evaluate success?