What's happening in Multi-Agent Reinforcement Learning?

Eugene Vinitsky, NYU Tandon, CUE

Overview

The goal of this lecture is to:

- Give you a sense of the challenges in multi-agent RL
- Show you why it’s exciting and important
- A high-level sense of some interesting subfields that you can explore further!

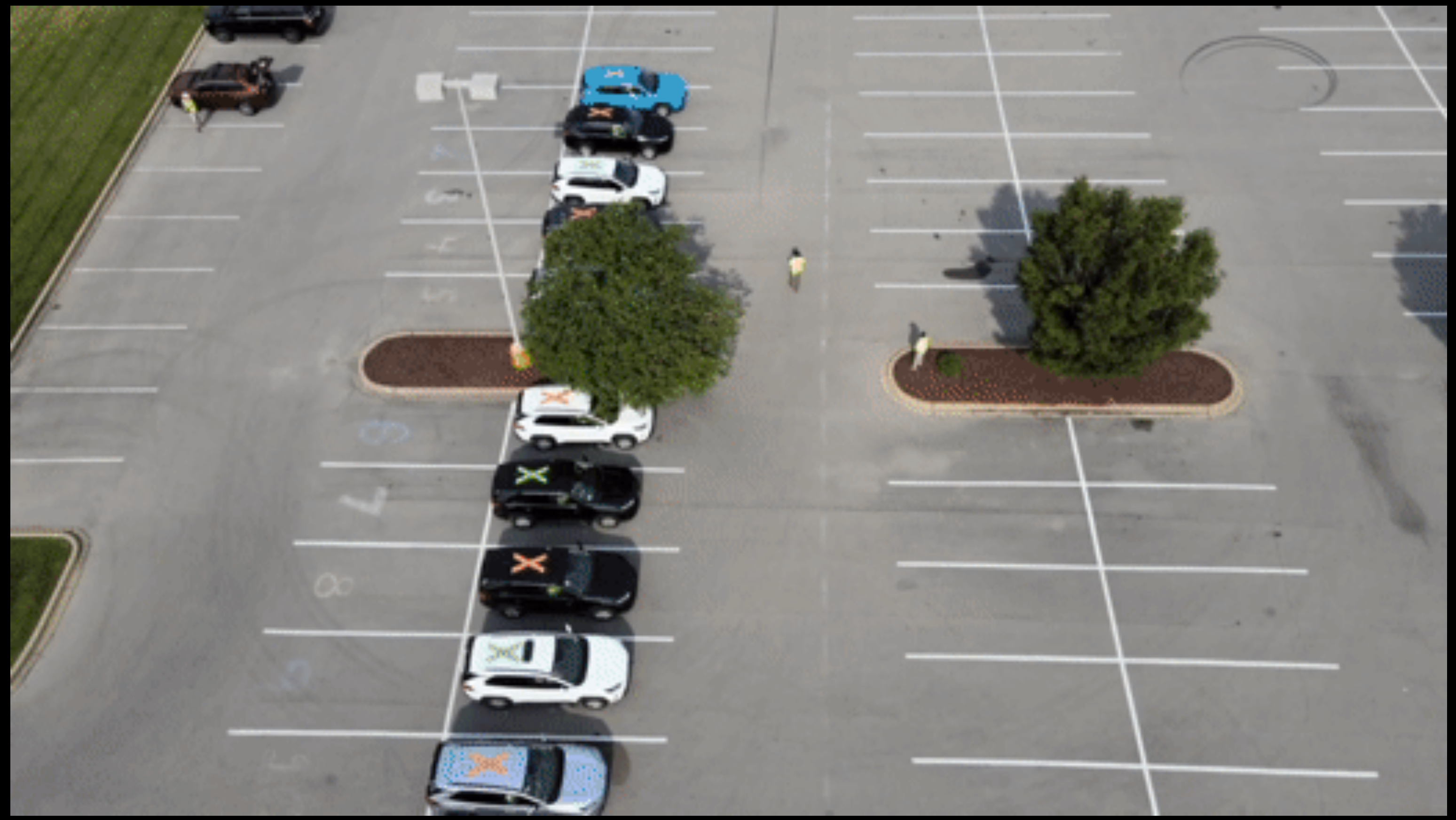
Caveat: we’re not going to talk about search-based techniques today. These are important but wouldn’t fit in time.

About me!

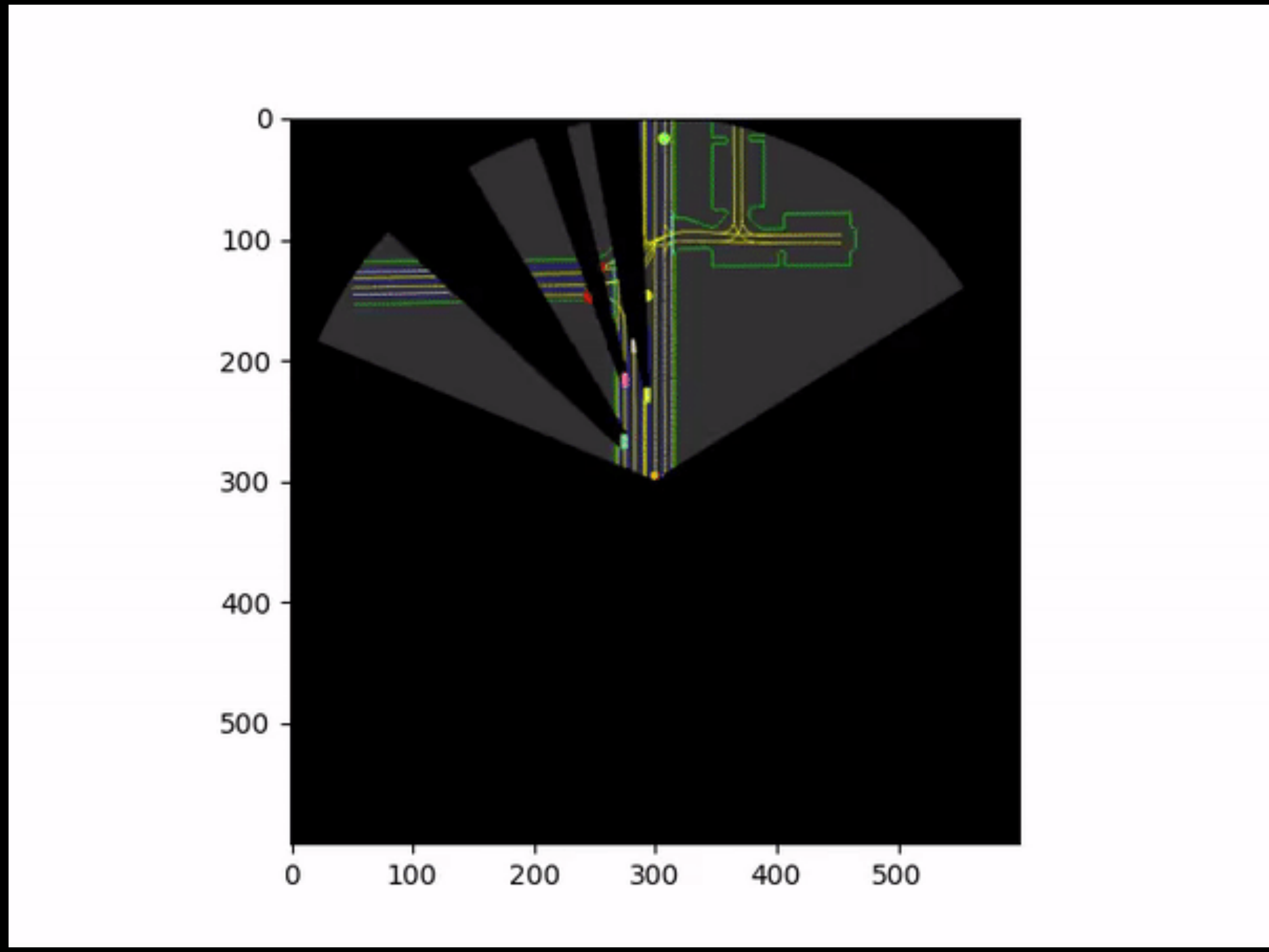
- Got my PhD at UC Berkeley in Controls (go bears)
- I’m a research scientist at Apple -> Professor at NYU Civil and Environmental Engineering
- Mostly work on multi-agent learning in the context of
  - Autonomous vehicles
  - Transportation systems
  - Mixed autonomy traffic

About me

Some work my colleagues and I have done

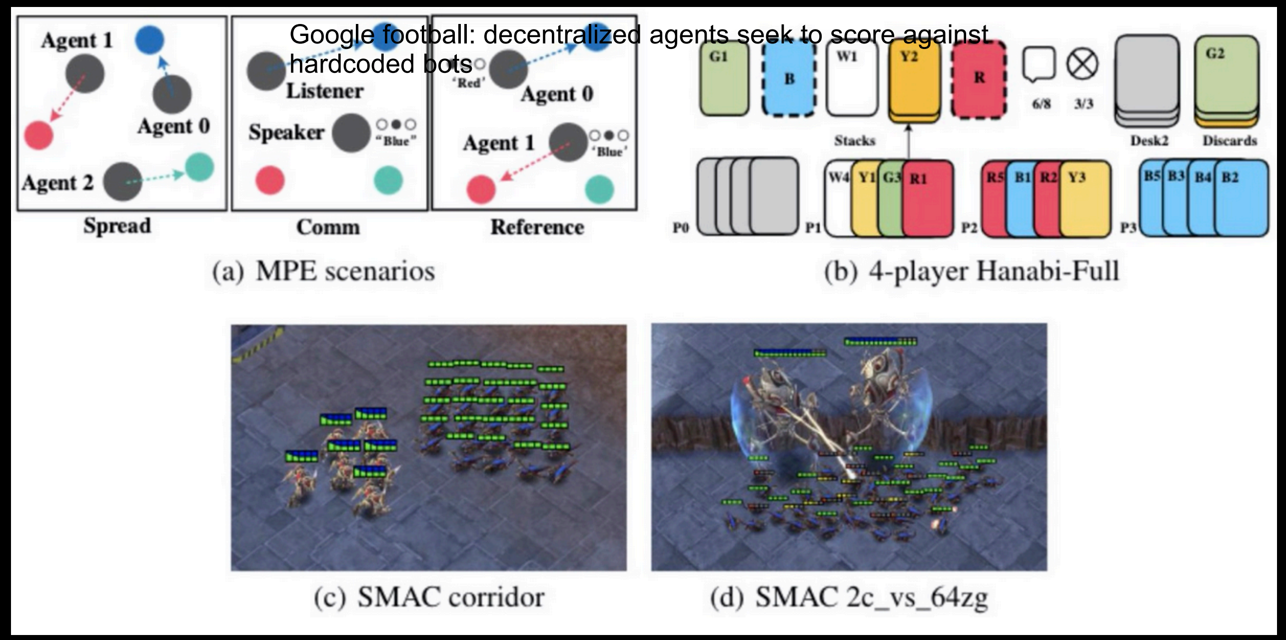


Lichtlé, N., Vinitsky, E., Nice, M., Seibold, B., Work, D., & Bayen, A. M. (2022, May). Deploying Traffic Smoothing Cruise Controllers Learned from Trajectory Data. In *2022 International Conference on Robotics and Automation (ICRA)* (pp. 2884-2890). IEEE.



Vinitsky, E., Lichtlé, N., Yang, X., Amos, B., & Foerster, J. (2022). Nocturne: a scalable driving benchmark for bringing multi-agent learning one step closer to the real world. *arXiv preprint arXiv:2206.09889*.

Google football: decentralized agents seek to score against hardcoded bots



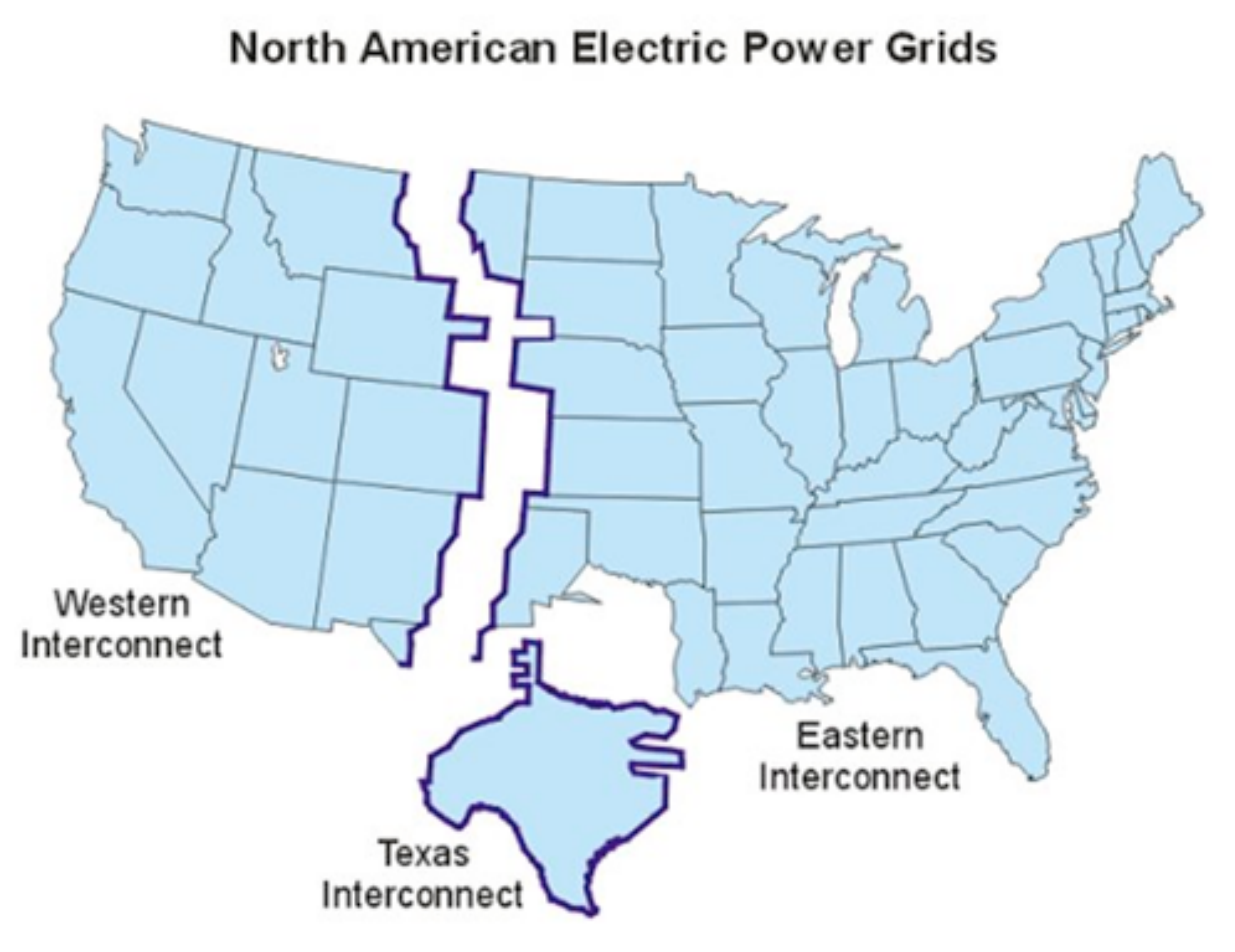
"The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games", Yu, Velu, **Vinitsky** et.al.[1]

Motivation

Why multi-agent learning?

Some problems are just unavoidably multi-agent

- Power grid pricing
  - Generator owners / load operators submit day-ahead bids
  - Grid operators solve an optimization problem to assign loads
  - Game repeats daily
  - Opportunities for collusion



Why multi-agent learning?

Some problems are just unavoidably multi-agent

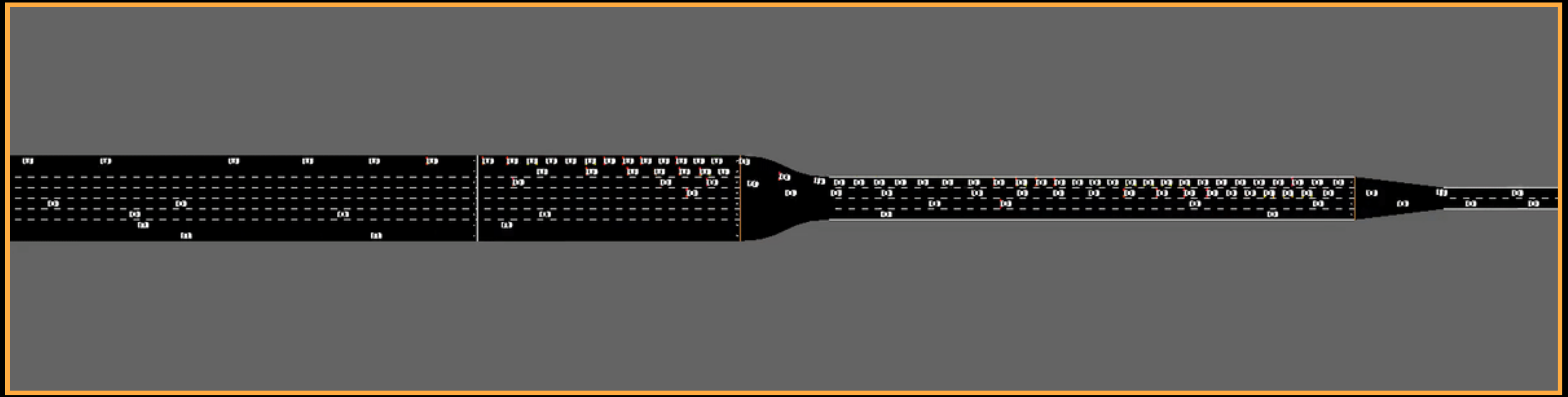
- Security games
  - You want to patrol with limited resources
  - The “poacher” can observe your strategy
  - How to allocate resources knowing that the “poacher” will respond?

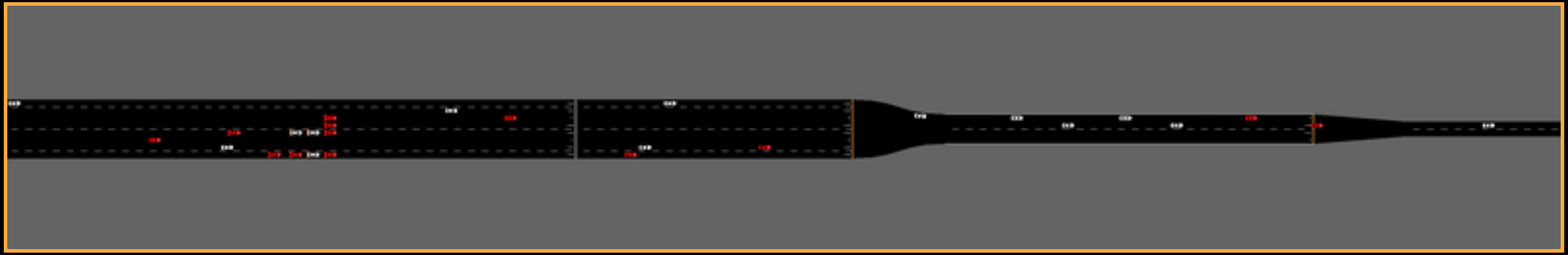
Sinha, Arunesh, et al. "Stackelberg security games: Looking beyond a decade of success." IJCAI, 2018.

Why multi-agent learning?

Some problems are just unavoidably multi-agent

- Cooperative Autonomous Vehicles
  - You have some limited number of vehicles
  - You want them to collectively accomplish some desirable goal





Why multi-agent learning?

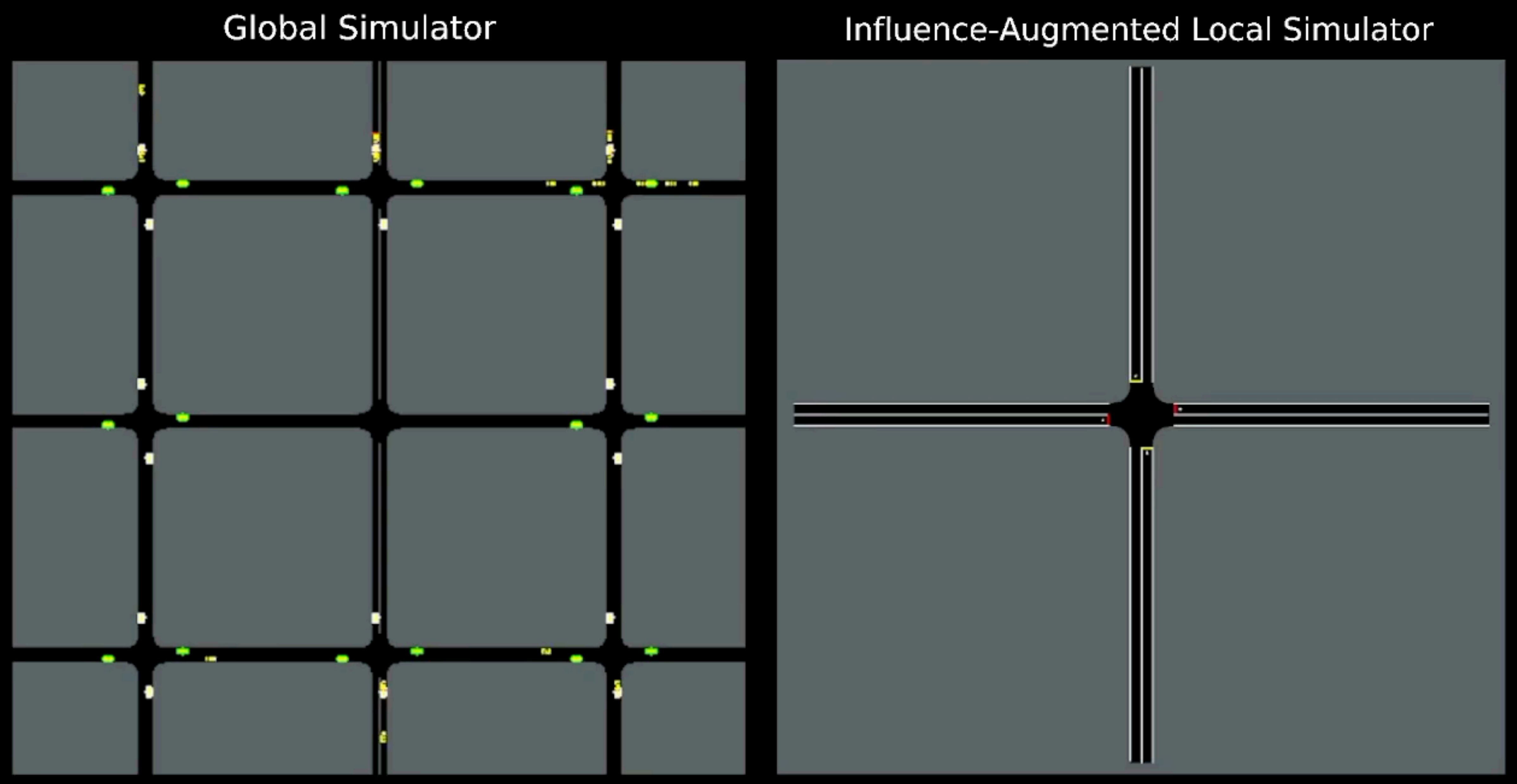
Some problems are just unavoidably multi-agent

- Self-driving
- Human-robot collaboration
- Decentralized traffic light grids
- Mixed-autonomy traffic
- Social science
- Economics
- Games

Why multi-agent learning?

Multi-agent can be *efficient*

- Suppose your simulator scales super-linearly with the number of agents
- Decomposing your system into coupled agents can be faster!



Suau, Miguel, et al. "Distributed Influence-Augmented Local Simulators for Parallel MARL in Large Networked Systems." *arXiv preprint arXiv:2207.00288* (2022).

Why multi-agent learning?

The multi-agent perspective on intelligence

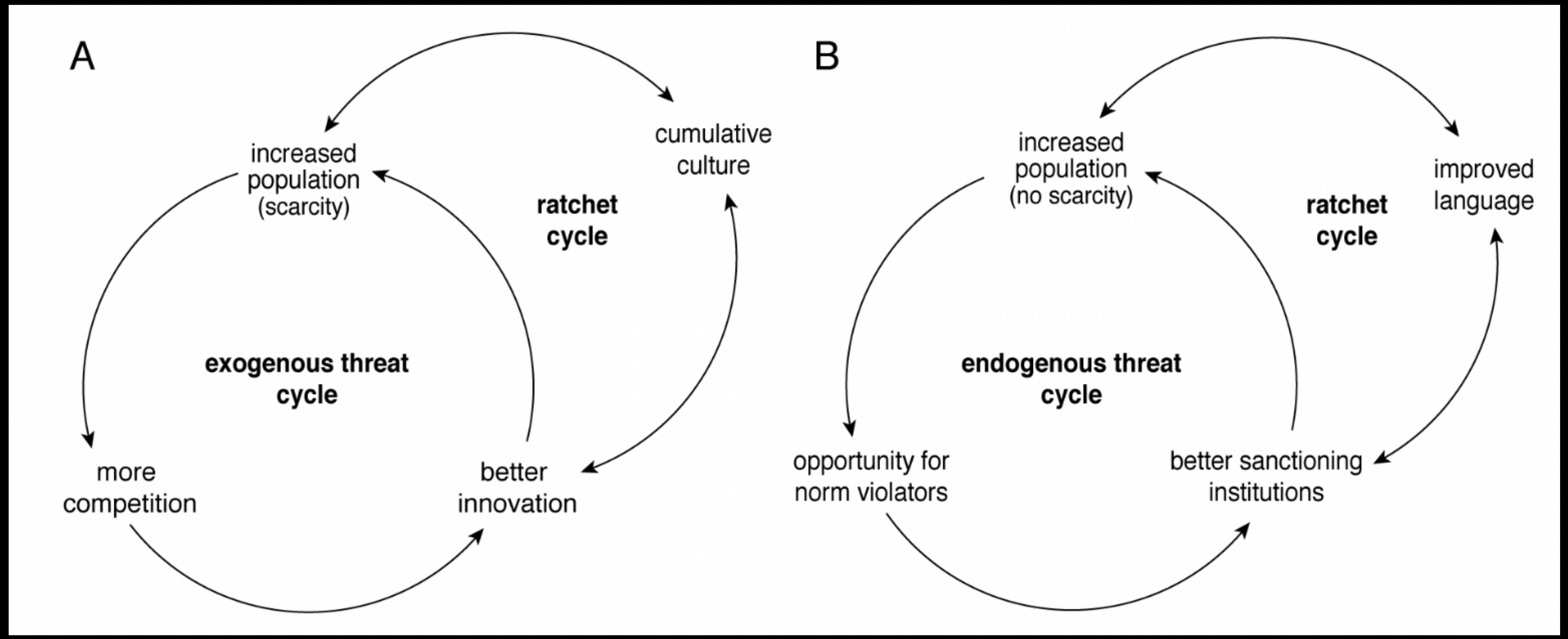
- Where do tasks come from? Human designers

Leibo, Joel Z., et al. "Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research." *arXiv preprint arXiv:1903.00742* (2019).

Why multi-agent learning?

The problem-problem*

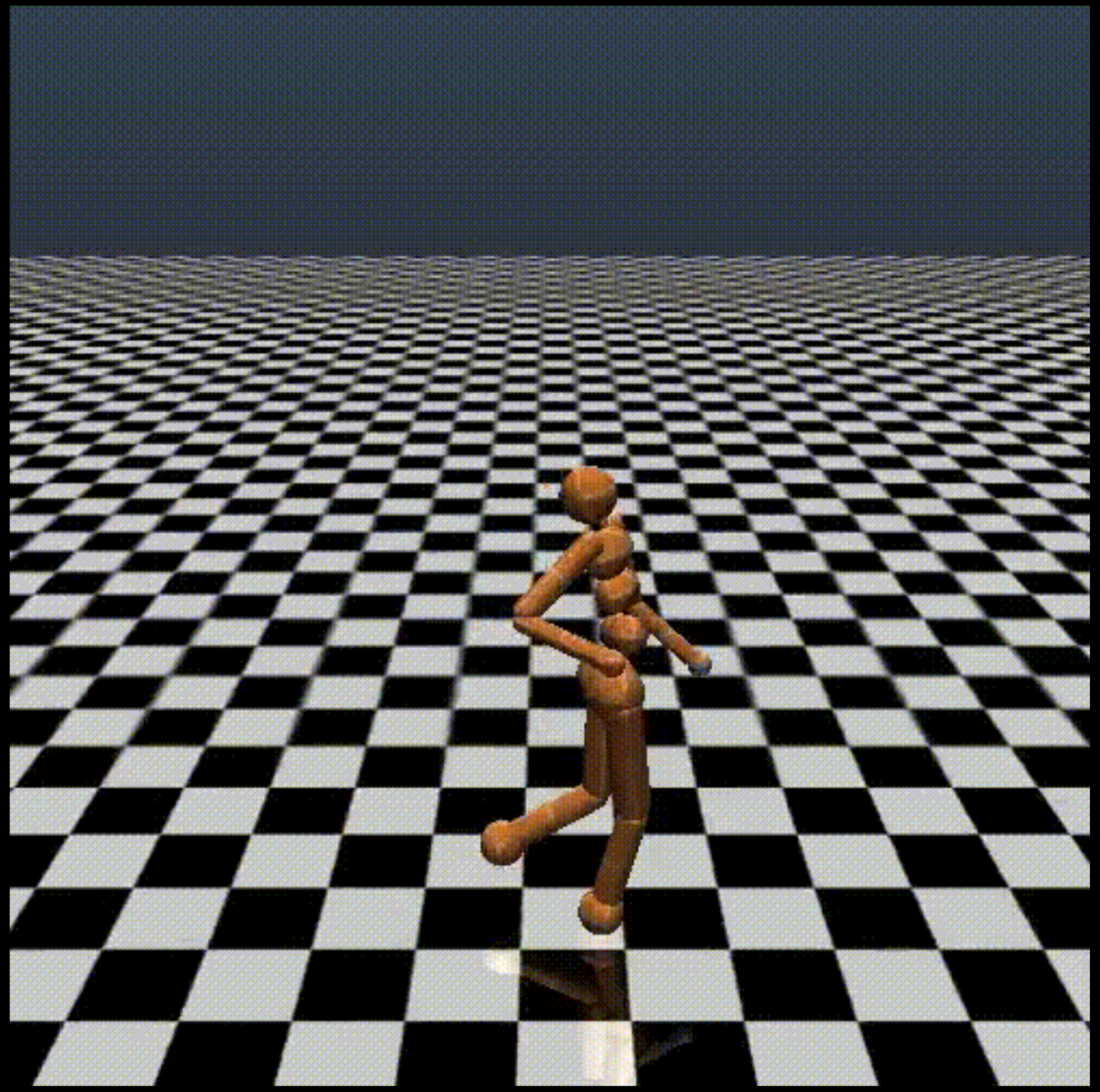
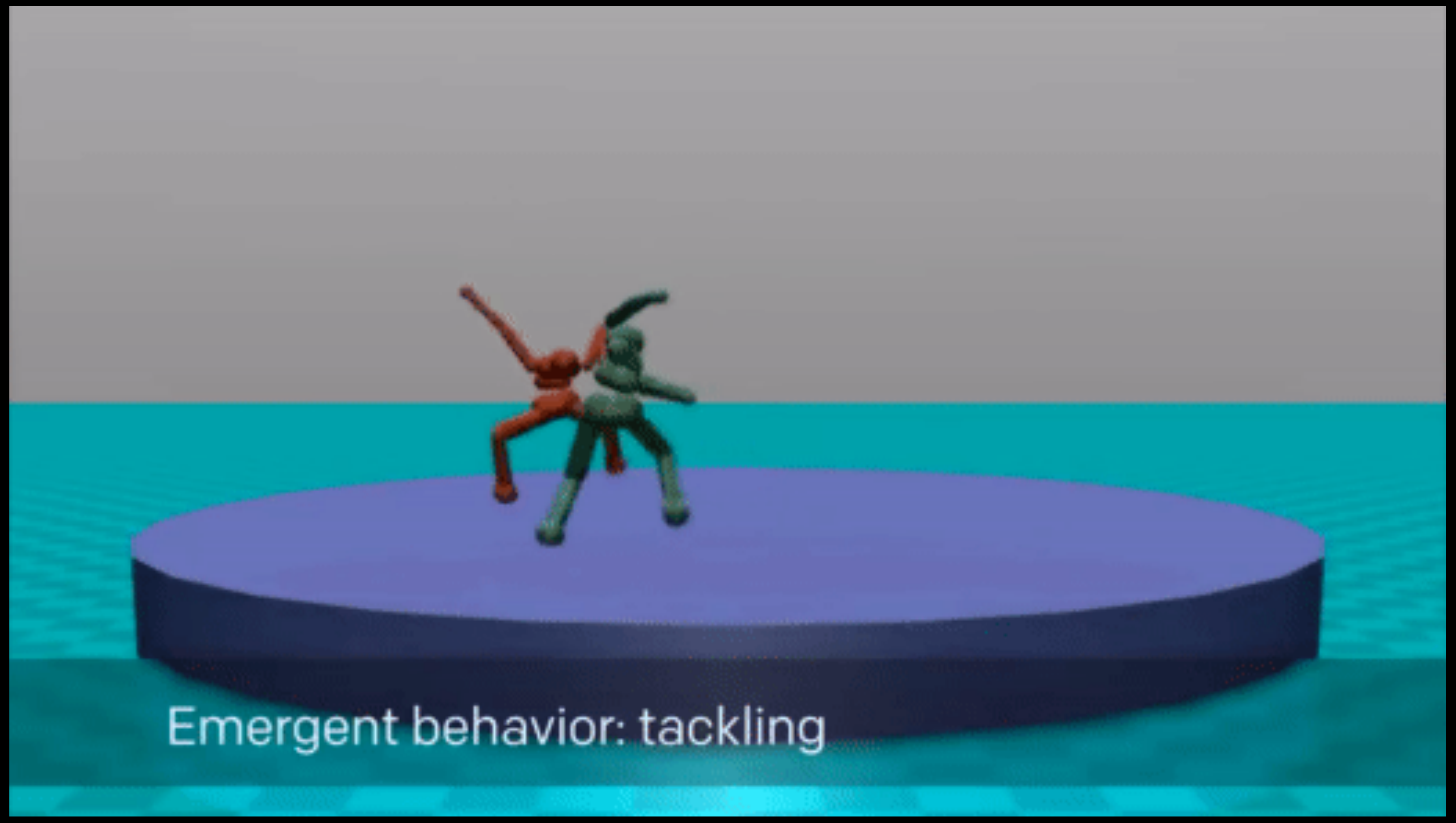
- Where do tasks come from?
- What if instead tasks were emergent from agent interaction?
- Pressure between and within societies creates challenges and spurs innovation
- Some evidence that human “intelligence” is mostly cultural intelligence



Leibo, Joel Z., et al. "Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research." *arXiv preprint arXiv:1903.00742* (2019).

Why multi-agent learning
The problem problem*

With a single agent, the range of behaviors is bounded

If multi-agent is the answer what is the question?*

- Multi-agent learning mixes many distinct agendas together
  - **Computational**: how do we algorithmically find equilibria?
  - Descriptive: how do we build models that match how people learn?
  - **Prescriptive**: how *should* agents learn?
    - What objectives should we adopt?
    - What equilibria should agents strive for?
    - Etc.

*Shoham, Yoav, Rob Powers, and Trond Grenager. "If multi-agent learning is the answer, what is the question?." *Artificial intelligence (2007)*

Preliminaries

Refresher: MDPs

- **MDP** defined by a tuple: $< \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma >$
    - $\mathcal{S}$ is a state space, what the agent will take as input
    - $\mathcal{A}$ is an action space, the set of actions an agent can take
    - $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is a reward function, scoring the value of a state action pair
    - $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ is a transition function, indicating how likely a next state is
    - γ is a discount factor, indicating how much we care about future reward

Refresher: RL

- Given an MDP, we want to find a policy π such that

$$\underset{\pi}{\text{argmax}}\, J^{\pi} = \mathbb{E}_{\pi,\mathscr{P}}\left[\sum_i \gamma^i r(s_i, a_i)\right]$$

- Lets take a second to go through that expectation
  - π: our policy is a probability distribution over actions $a_i \sim \pi(s_i)$
  - $\mathscr{P}$: the environment dynamics are stochastic $s_{i+1} \sim P(s_i, a_i)$
- **Note:** what I'm talking about is discounted-reward-maximizing RL. There are other quantities you might want to maximize.

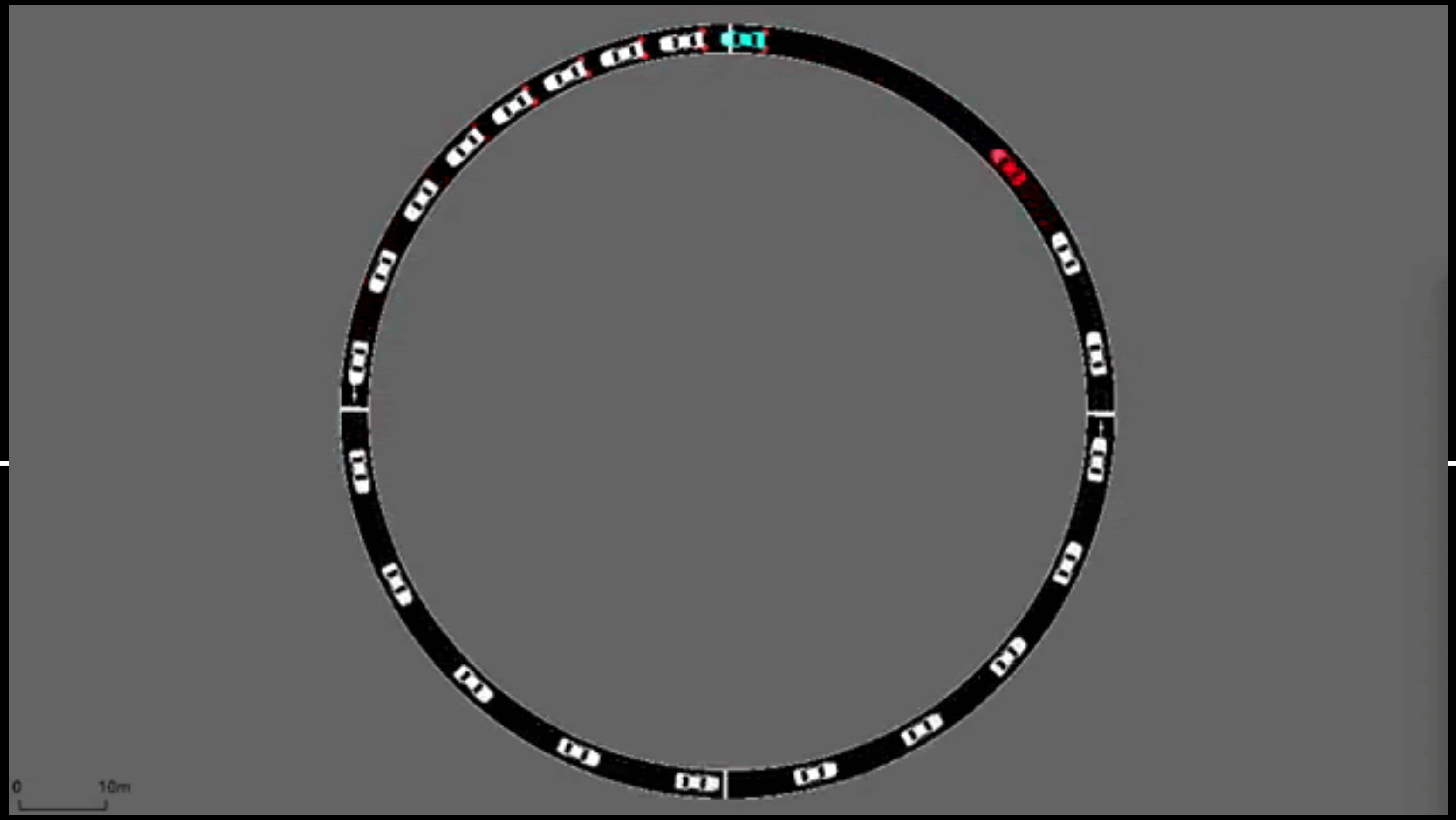
Example

State
Position / speed of all vehicles

Rewards:
Vehicle Speed

Actions
Accelerate
Decelerate

The Environment



Notation: what changes?

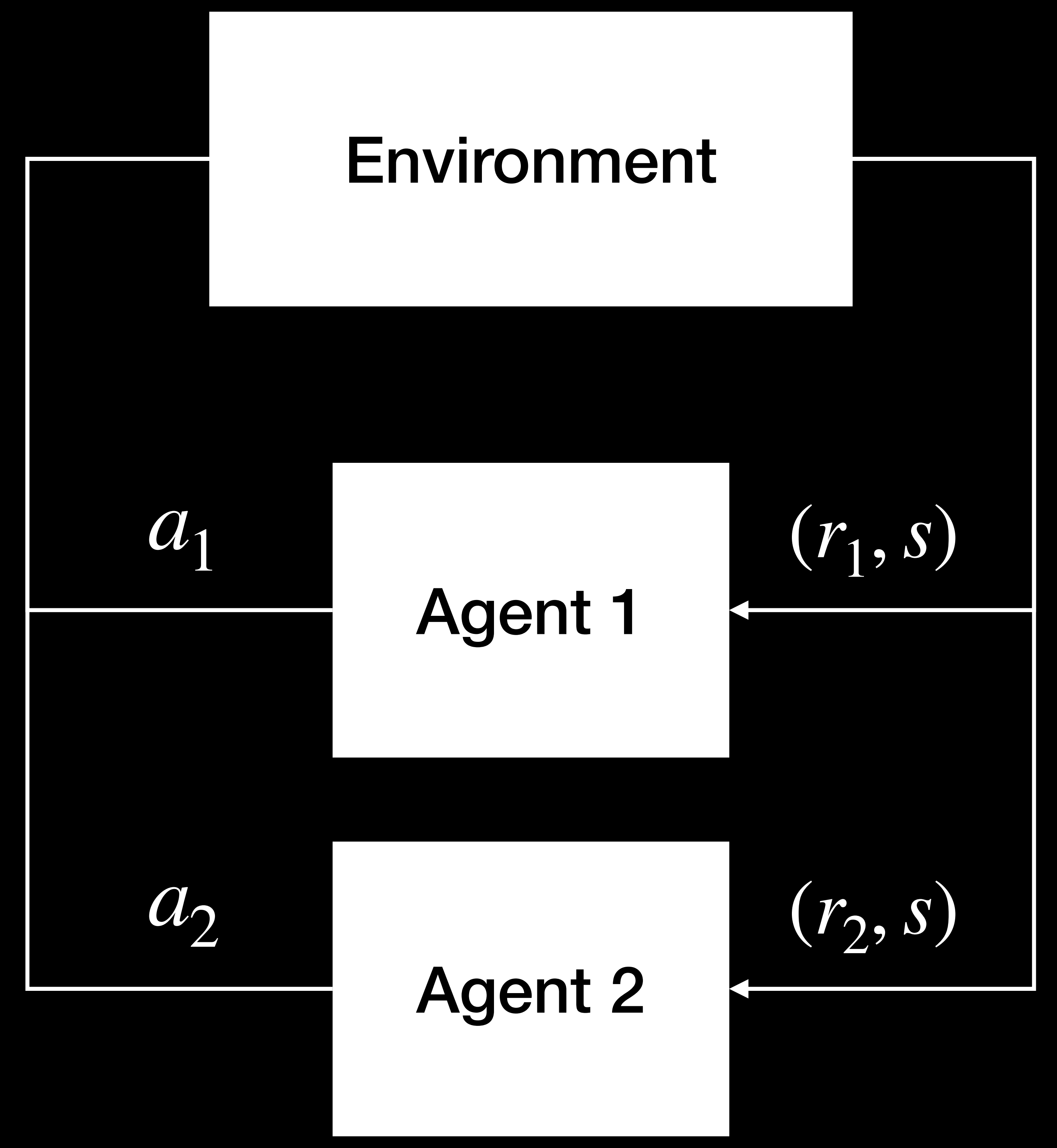
- We're mostly going to be talking about **Markov Games**
- Each player has a tuple: $< \mathcal{S}, \mathcal{A}_i, \mathcal{R}_i, \mathcal{P}, \gamma_i >$
- i indexes a player
- The action of all agents: $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \ldots \times \mathcal{A}_n$
- The transition $\mathcal{P}$ is now $\mathcal{P} : \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2 \ldots \times \mathcal{A}_n \times \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$
- The reward is now: $\mathcal{R}_i : \mathcal{S} \times \mathcal{A}_1 \times \mathcal{A}_2 \ldots \times \mathcal{A}_n \rightarrow \mathbb{R}$
- Multiple agents can act in the system, sometimes simultaneously

Some more notation

- Writing out 1, 2, …, N is really annoying
- We will often use i to index a particular agent and $-i$ to index all the other agents
- For example:
  - a_1 is the action of agent 1, a_{-1} is the joint action of all other agents
  - π_1 is the policy of agent 1, π_{-1} is the joint policy of all other agents
  - We'll use $J^{\pi_i, \pi_{-i}}$ as the expected reward of the policy pair

Markov Games

- Note, **in this lecture**, we are neglecting the possibility of hidden information
- The challenges today come from
  - Simultaneous actions
  - Rewards dependent on all agents
  - Transitions dependent on all agents

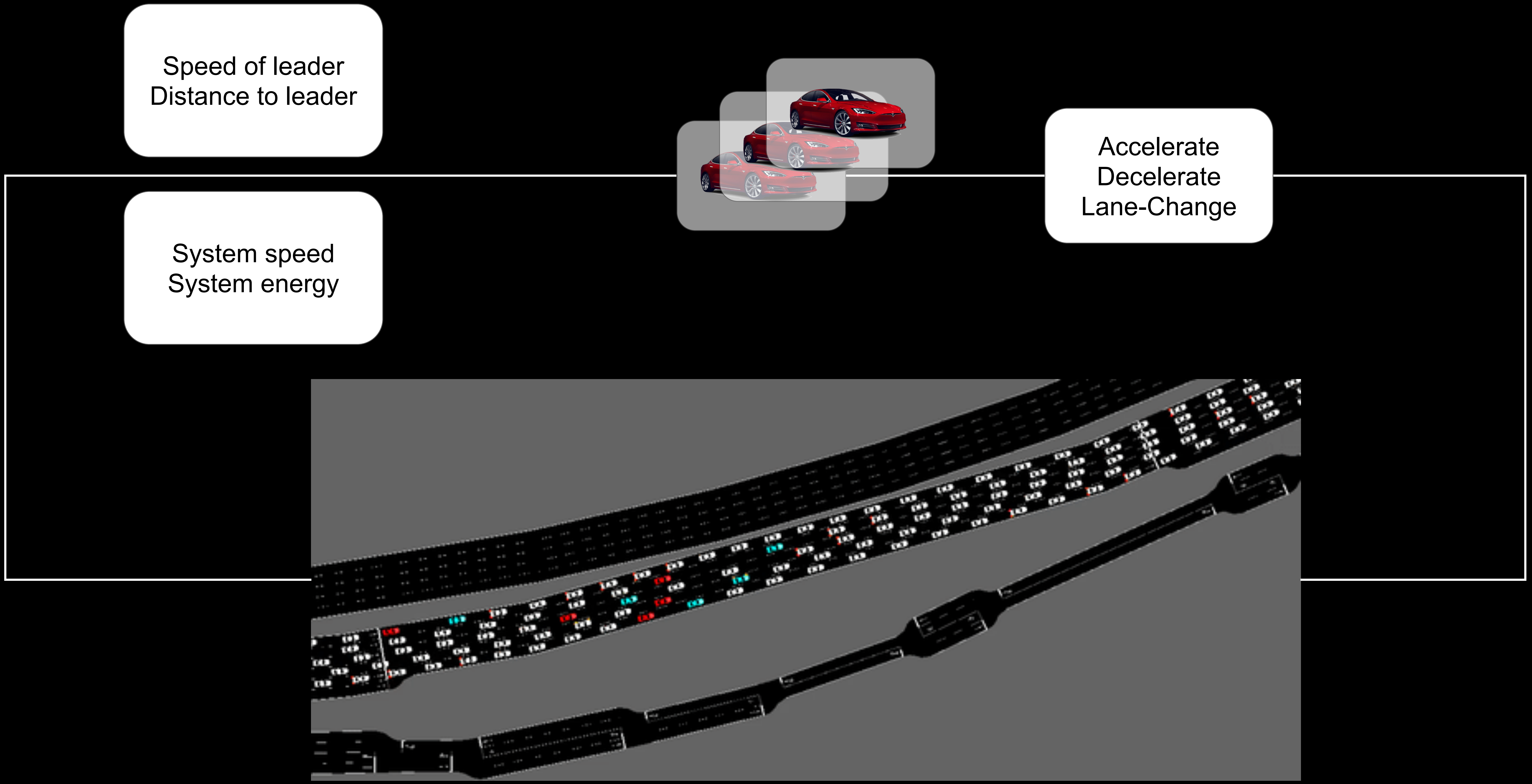


How is MARL different than RL?

- Every environment looks non-stationary to each agent
- (Almost) Every environment is partially observed due to other agents
- Single-agent RL algorithms often lose their convergence guarantees
- Multiple notions of convergence / equilibria appear

Example

- The rewards can be dependent on the actions of every agent
- Agents **might** act simultaneously



What's an "optimal" policy now?

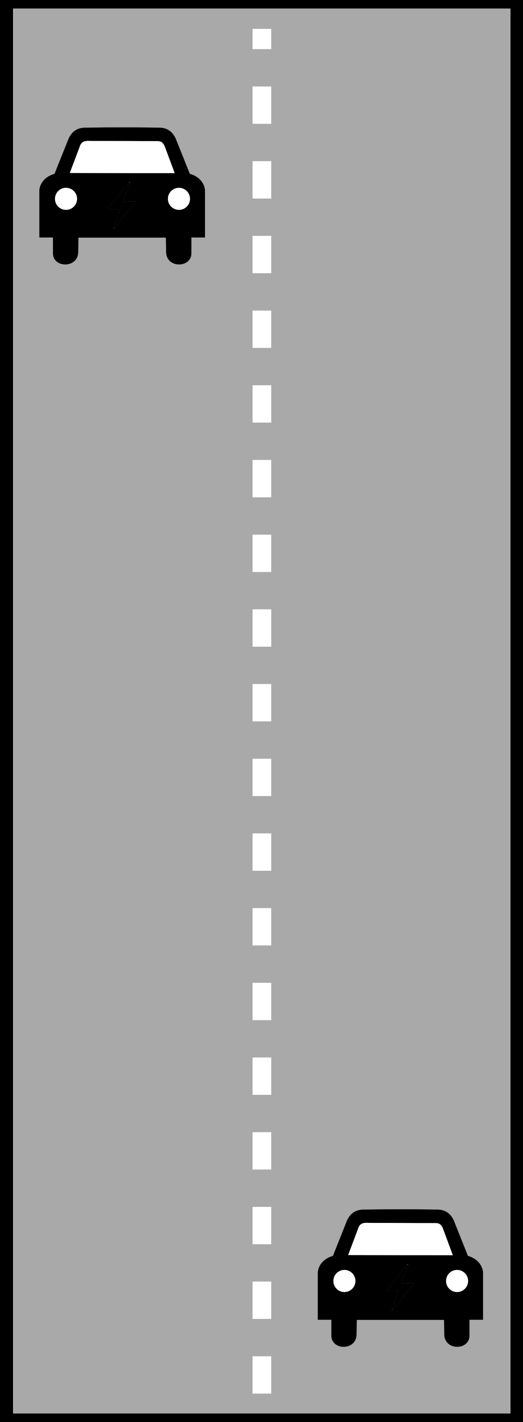
- In single-agent RL, we're happy once we've found π^* where

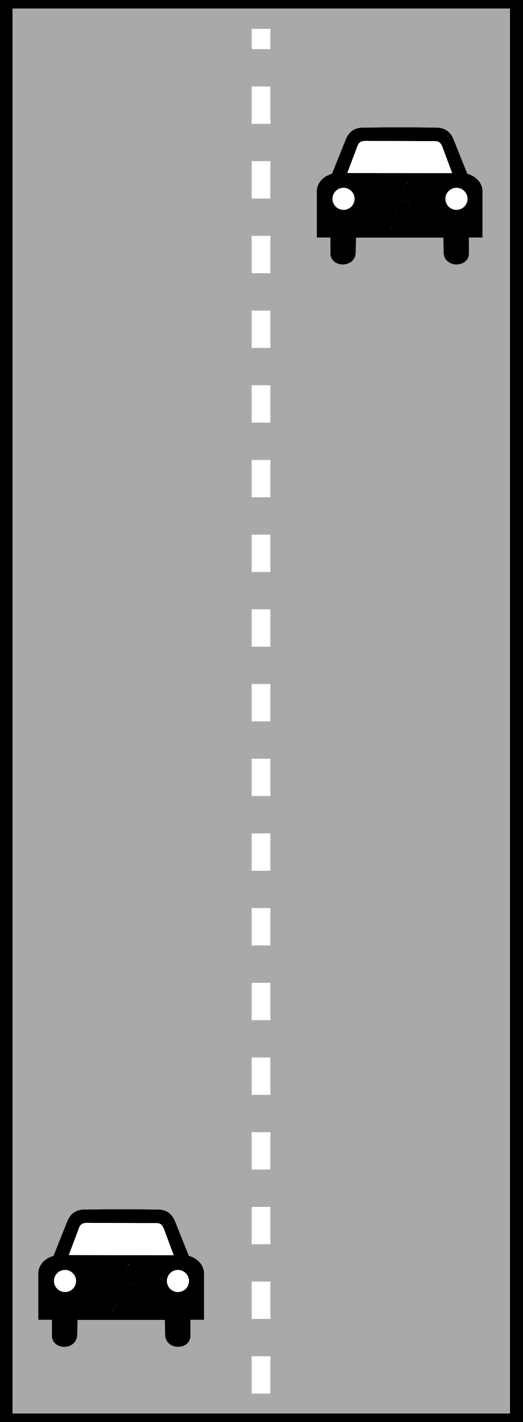
$$\pi^* = \underset{\pi}{\text{argmax}}\, J^\pi = \mathbb{E}_{\pi,\mathscr{P}}\left[\sum_i \gamma^i r(s_i, a_i)\right]$$

- But now, we have N agents, each with their own reward functions, looking for their own π_i^*
- For a given problem, is there one optimal policy?

What's an "optimal" policy now?

- In general, there isn't one optimal policy, it depends on the other agents





Equilibria Notions

Nash equilibrium

- For a policy to be desirable, it must be “compatible” with the policies of the agents it plays with
- A common way of describing this compatibility is called a **Nash Equilibrium**
- A set of policies $\{\pi_1, \pi_2, \ldots, \pi_N\}$ is a Nash equilibrium if

$$J^{\pi_i^*, \pi_{-i}^*} \geq J^{\pi_i, \pi_{-i}^*}, \forall \pi_i, \forall i$$

Equilibria Notions

Nash equilibrium

- A set of policies $\{\pi_1^*, \pi_2^*, \ldots, \pi_N^*\}$ is a Nash equilibrium if

$$J^{\pi_i^*, \pi_{-i}^*} \geq J^{\pi_i, \pi_{-i}^*}, \forall \pi_i, \forall i$$

- In intuitive terms, any agent is worse off if it changes its policy while holding the other policies fixed
- No individual agent has an incentive to change

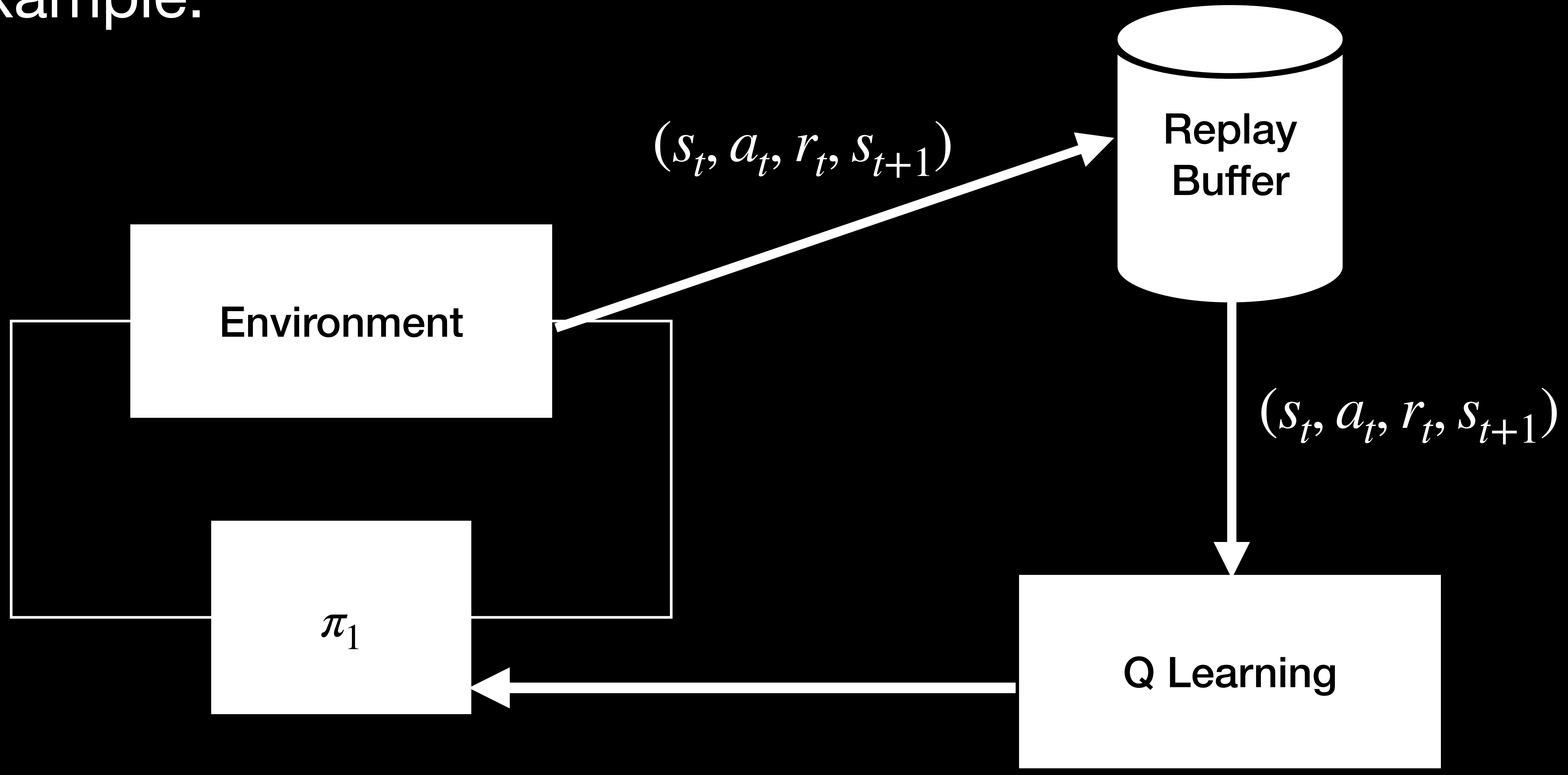
Equilibria Notions

Other options

- Nash is a central equilibrium notion, there are many others
    - Correlated equilibria
    - Coarse correlated Equilibria
    - Trembling Hand Equilibria
    - Cyclic equilibria
- No time to discuss these, but want to name them so **you** can look them up!

What's challenging about MARL?
Non-stationarity due to changing policies

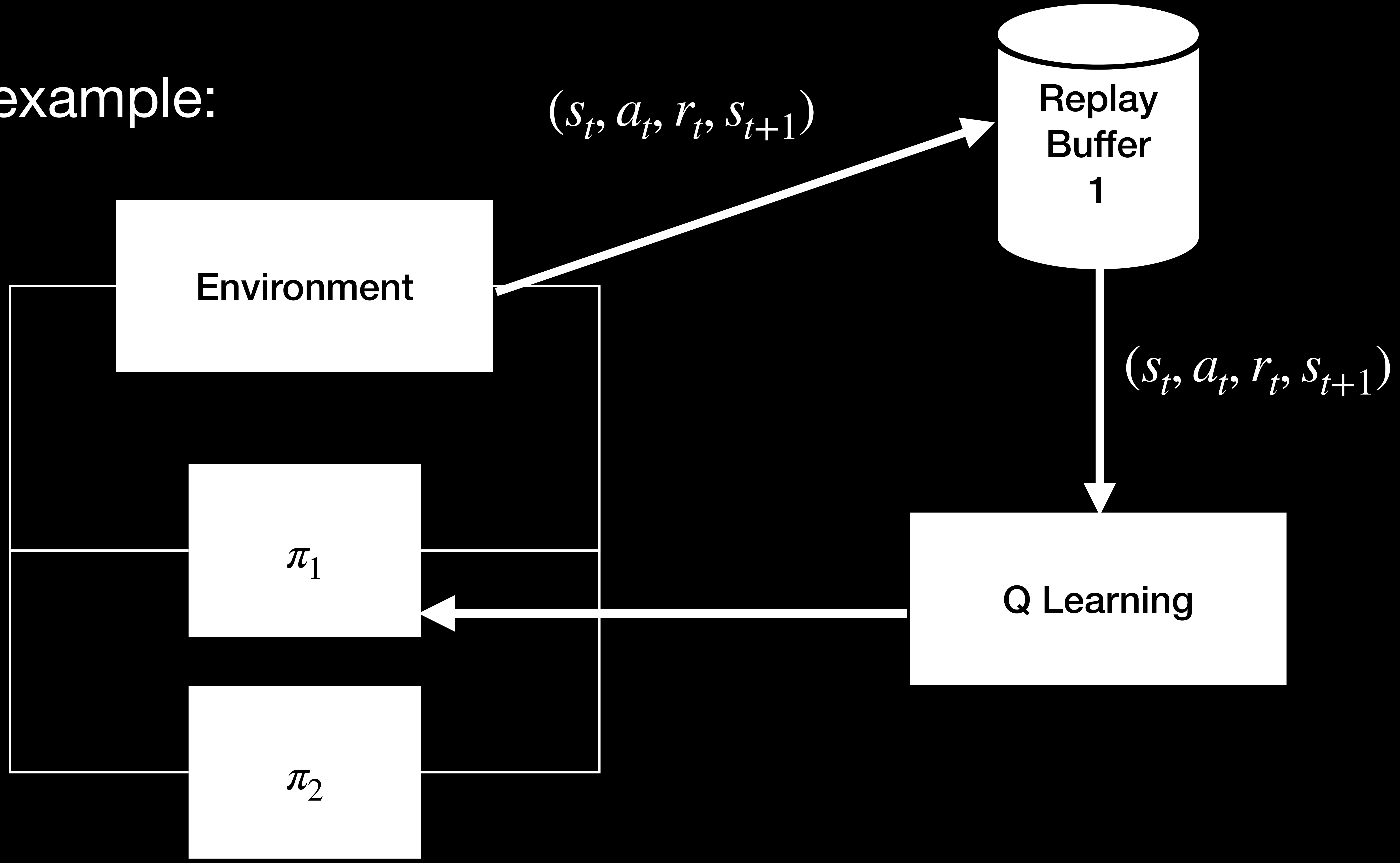
- Stale data. Almost every RL method implicitly or explicitly has a replay buffer
- DQN example:



What's challenging about Markov Games?

Non-stationarity due to changing policies

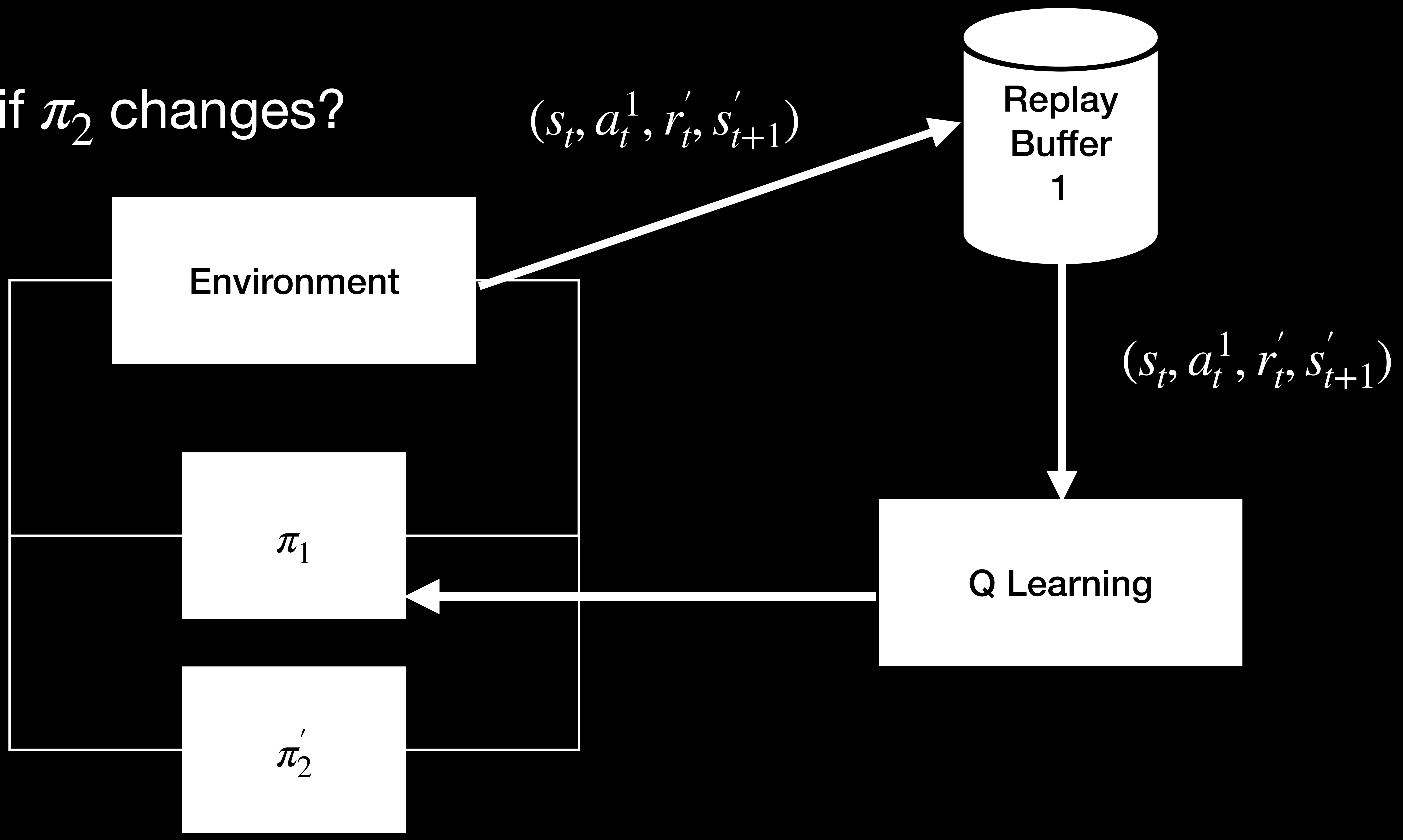
- DQN example:



What's challenging about Markov Games?

Non-stationarity due to changing policies

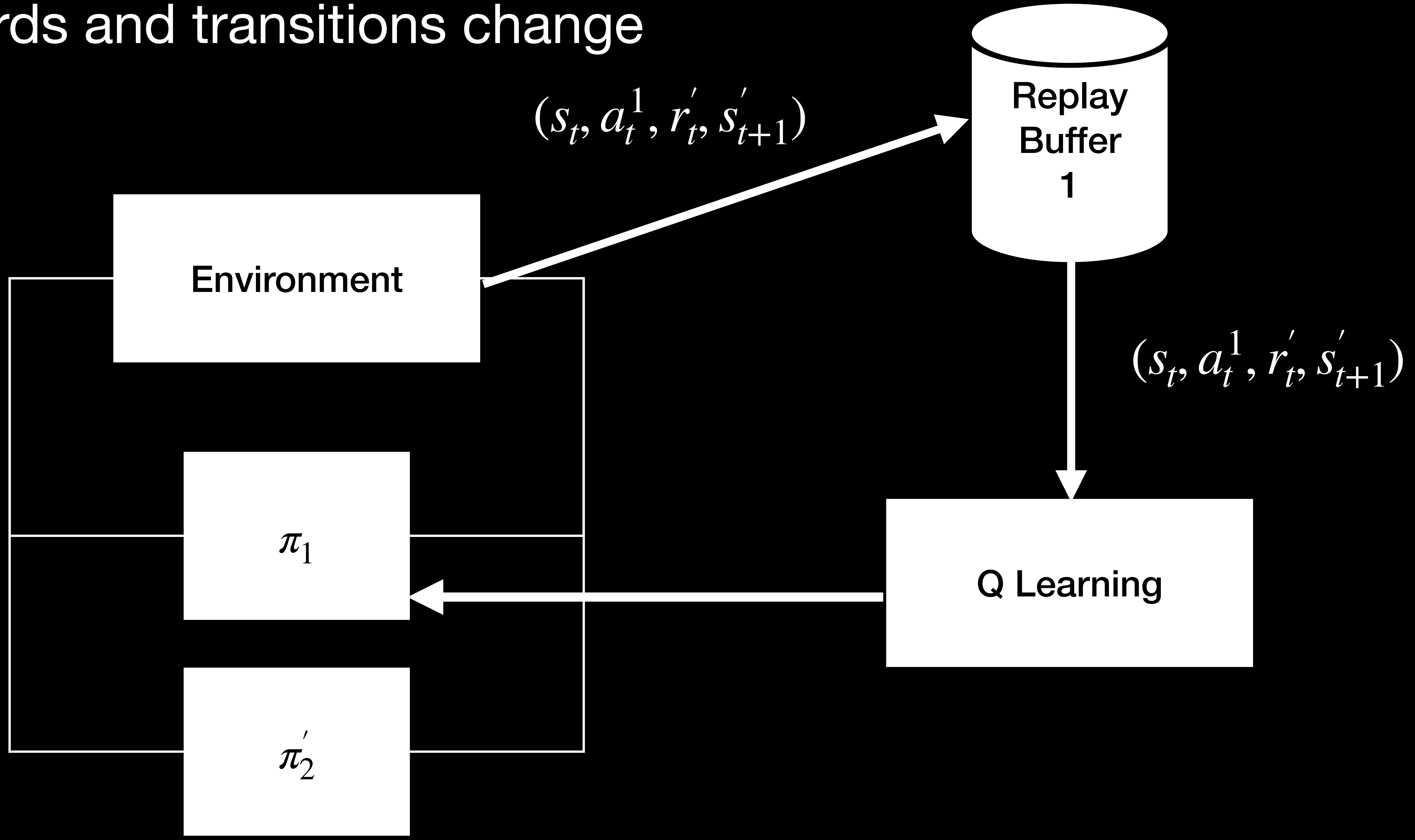
- What if π_2 changes?



What's challenging about Markov Games?

Non-stationarity due to changing policies

- Rewards and transitions change



What’s challenging about Markov games?
Stochastic policies

- In MDPs we have a wonderful fact: there is an optimal **deterministic** policy
- In a Markov game, some **Nash Equilibria** can be stochastic
- Example: Bach Stravinsky

	Player 2	
Player 1	2 , 1	0 , 0
	0 , 0	1 , 2

What's challenging about Markov Games?

The Curse of Dimensionality

- The sample complexity of RL algorithms often depend on the action dimension
- Consider the case where each agent has |A| actions
- If we have N agents, we have $|A|^N$ actions
- Exponential in the number of agents

Research Directions, Key Ideas, and Major Results

Some Key Ideas

Self-play

The secret to go

- Suppose we have a game that is
  - **2-player zero sum**
    - If the reward for player 1 is r
    - The reward for player 2 is -r
  - Markovian

Self-play

The secret to go

- Self-play: start with a random policy π_0
- For iterations I=0:N
  - Use some algorithm (MCTS) to find policy π_{i+1} that beats policy π_i
- “Simple” as that
- In reality, the policy improvement technique (MCTS) and how we order self-play is complicated
- Read “Mastering the game of Go without Human Knowledge” for technical details

From self-play to league play

- Is self-play guaranteed to keep improving?
- **No.** Consider rock paper scissors.
- Suppose policy 0 is 30% rock, 40% scissors, 30% paper.
- Policy 1 will be 100% rock.
- Policy 2 will be 100% paper.
- Policy 3 will be 100% scissors.
- And so on. This is sometimes called **cyclic dynamics**

From self-play to league play

- How do we fix this? One attempt: **Policy Space Response Oracles**[1]
- **Basic idea:**
    - Keep a pool of policies for each player
    - Compute a distribution over policies for each player (a “meta strategy”)
    - Have one player learn a best response to the other player’s meta-strategy
    - Add the best response to the pool

1: Lanctot, Marc, et al. "A unified game-theoretic approach to multiagent reinforcement learning." *Advances in neural information processing systems* 30 (2017).

From self-play to league play

- How do we fix this? One attempt: **Policy Space Response Oracles**[1]
- **What does this look like:**
- π_i^j, policy j for agent I
- p_i^j, probability of playing π_i^j

		p_1^0 π_1^0	p_1^1 π_1^1
p_0^0	π_0^0	r_1, r_2	r_3, r_4
p_0^1	π_0^1	r_5, r_6	r_7, r_8

1: Lanctot, Marc, et al. "A unified game-theoretic approach to multiagent reinforcement learning." *Advances in neural information processing systems* 30 (2017).

From self-play to league play

- How do we fix this? One attempt: **Policy Space Response Oracles**[1]
- **What does this look like:**
- π_i^j, policy j for agent I
- p_i^j, probability of playing π_i^j

		p_1^0	p_1^1
		π_1^0	π_1^1
p_0^0	π_0^0	r_1, r_2	r_3, r_4
p_0^1	π_0^1	r_5, r_6	r_7, r_8
p_0^2	π_0^2	r_9, r_{10}	r_{11}, r_{12}

1: Lanctot, Marc, et al. "A unified game-theoretic approach to multiagent reinforcement learning." *Advances in neural information processing systems* 30 (2017).

From self-play to league play

- How do we fix this? One attempt: **Policy Space Response Oracles**[1]
- Other variants of this idea (for you to look into):
    - Fictitious play (play against a uniform distribution over prior opponents)
    - Double oracle
    - XDO

1: Lanctot, Marc, et al. "A unified game-theoretic approach to multiagent reinforcement learning." *Advances in neural information processing systems* 30 (2017).

League Play / Population Play

Starcraft*

- Building pools of agents is quite common in games with cycles
- Starcraft II has cyclic dynamics like RPS: Zerg, Terran, Protoss
- They use three types of agent pools:
  - **Main agents:** 35% self-play, 50% play against all old agents, 15% exploiters
  - **League exploiters:** 100% play against old agents
  - **Main exploiters:** mostly play against the main agents

*Vinyals, Oriol, et al. "Grandmaster level in StarCraft II using multi-agent reinforcement learning." *Nature* 575.7782 (2019): 350-354.

Why does population play help?
A provocative theory

- Games can be decoupled into cyclic and transitive components

- Suppose $f(\pi_i, \pi_j) > 0$ implies π_i beats π_j

  - Transitive:
    $f(\pi_i, \pi_j) > 0, f(\pi_j, \pi_k) > 0 \rightarrow f(\pi_i, \pi_k) > 0$

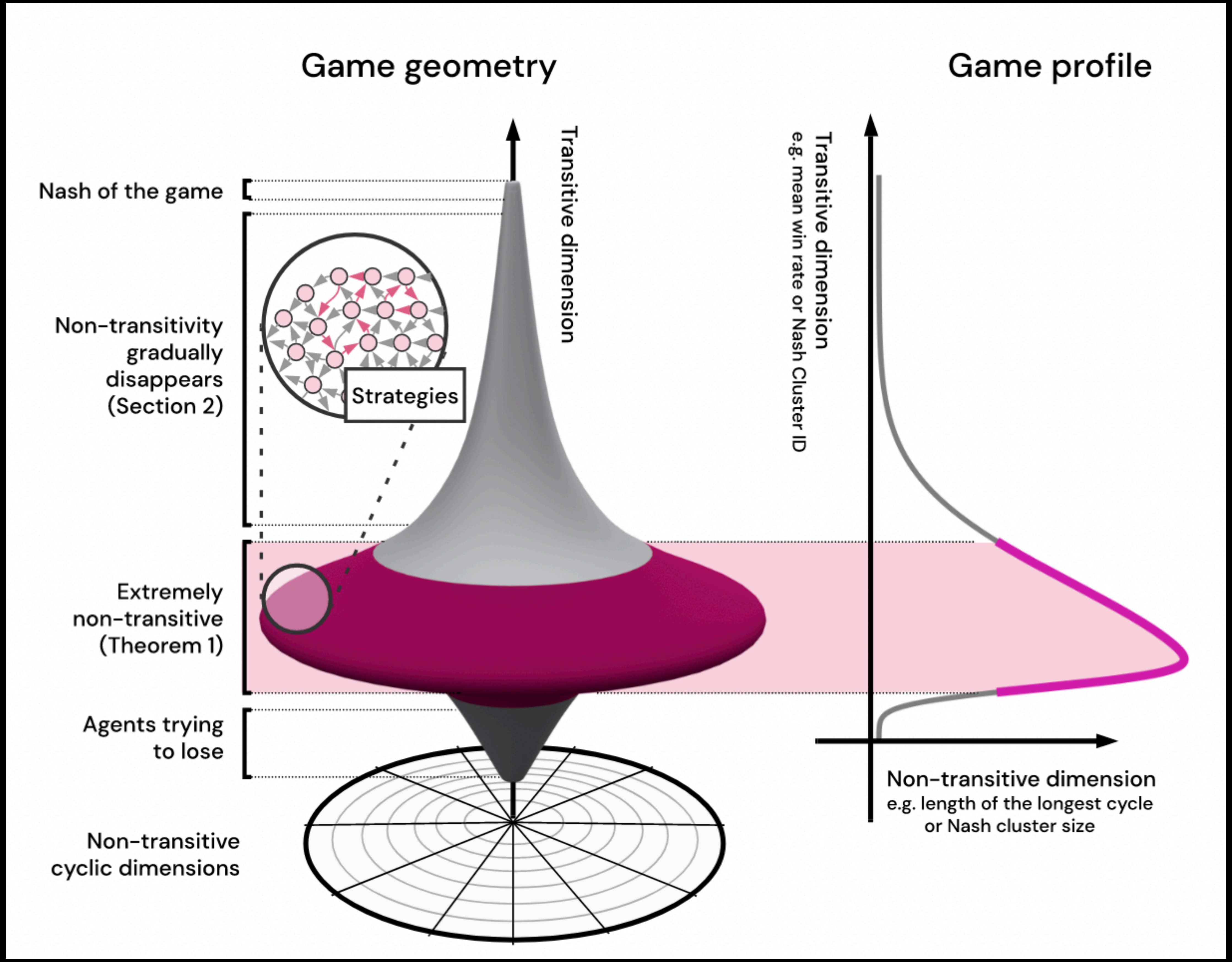
  - Cyclic of length l
    $f(\pi_{i+1}, \pi_i) > 0 \forall i > 0, f(\pi_0, \pi_l) > 0$

Czarnecki, Wojciech M., et al. "Real world games look like spinning tops." *Advances in Neural Information Processing Systems* 33 (2020): 17443-17454.

Why does population play help?

A provocative theory

- Think about RPS which is cyclic:
  - Rock beats paper beats scissors beats rock
- Transitive: one tic-tac-toe strategy is strictly better than all others
- The point of this paper: most games are a mix and can contain long cycles

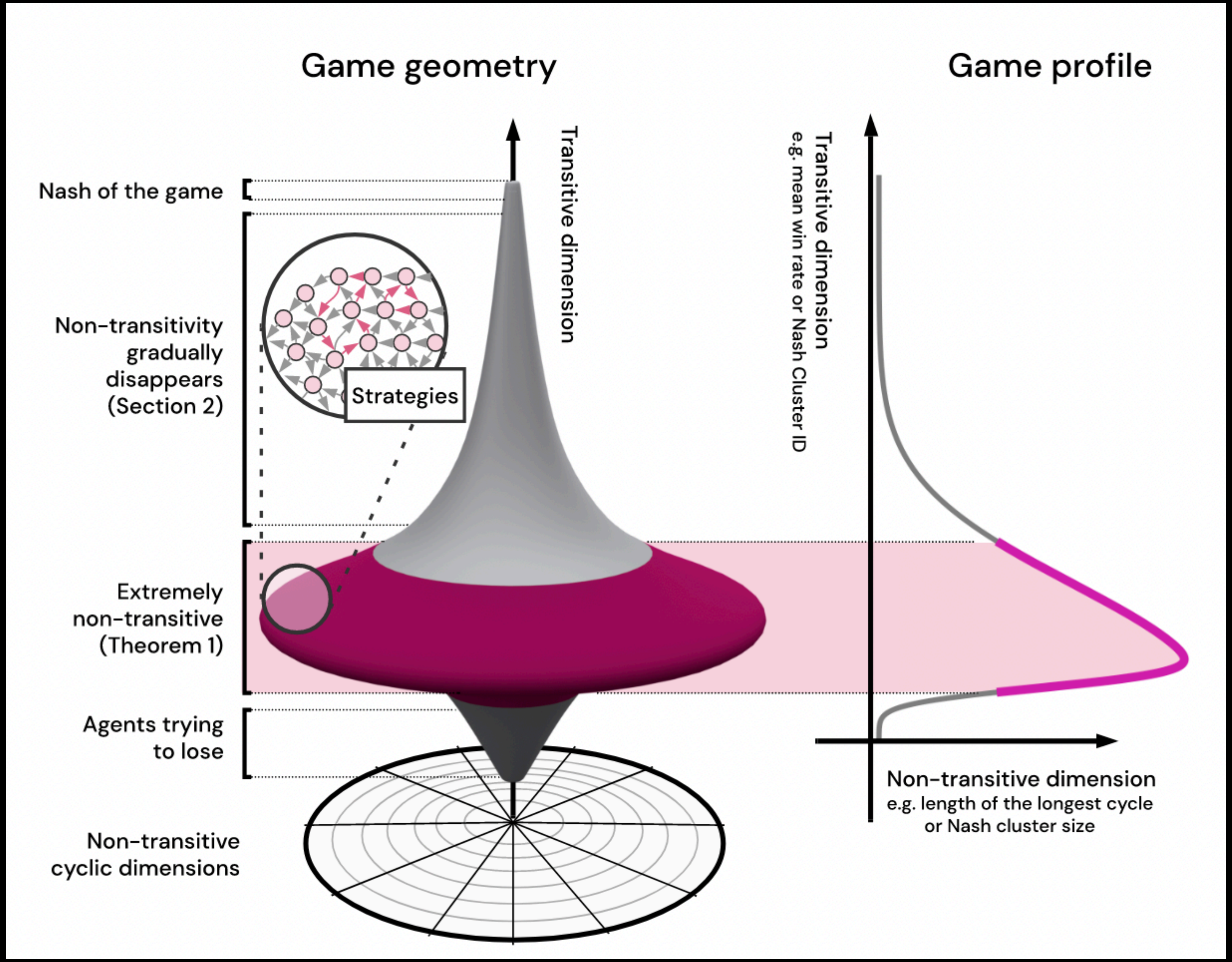


Czarnecki, Wojciech M., et al. "Real world games look like spinning tops." *Advances in Neural Information Processing Systems* 33 (2020): 17443-17454.

Why does population play help?

A provocative theory

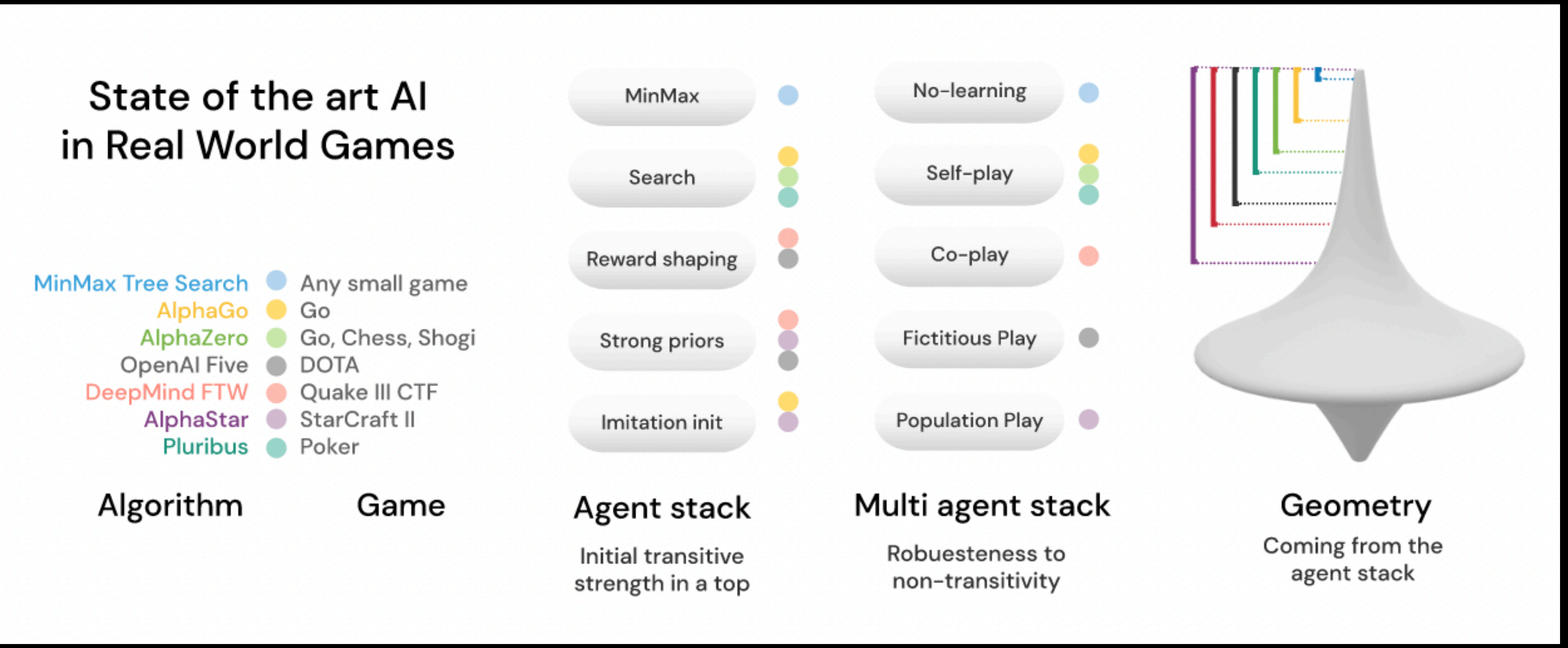
- A theorem asserted by the paper (loosely): once you have a large enough population of skills, you can jump to the next transitive dimension
- I.e. to improve, your population should cover all game styles



Czarnecki, Wojciech M., et al. "Real world games look like spinning tops." *Advances in Neural Information Processing Systems* 33 (2020): 17443-17454.

Why does population play help?
A provocative theory

- A theorem asserted by the paper (loosely): once you have a large enough population of skills, you can jump to the next transitive dimension
- I.e. to improve, your population should cover all game styles

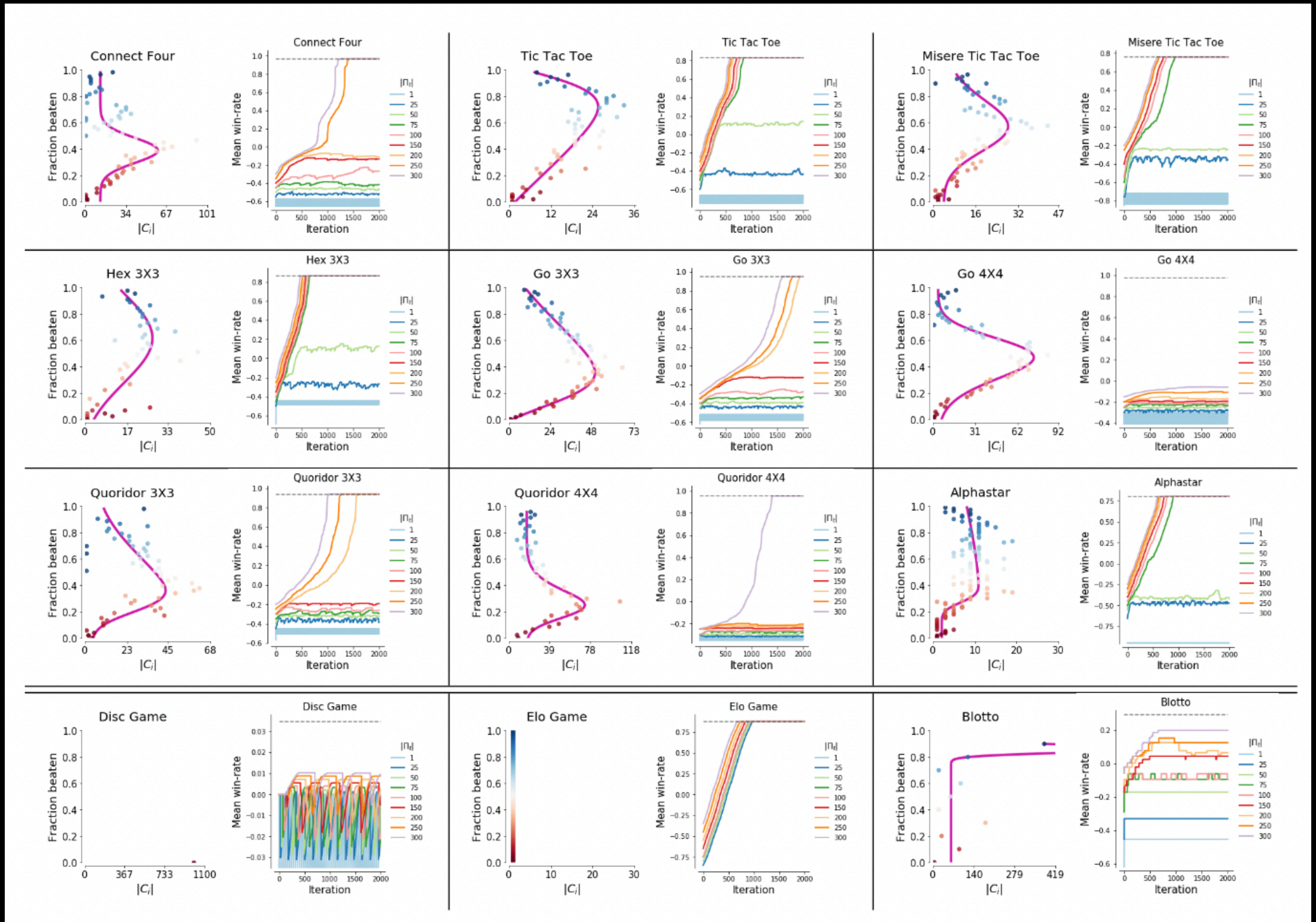


Czarnecki, Wojciech M., et al. "Real world games look like spinning tops." *Advances in Neural Information Processing Systems* 33 (2020): 17443-17454.

Why does population play help?

A provocative theory

- They empirically measure this in a bunch of games



Czarnecki, Wojciech M., et al. "Real world games look like spinning tops." *Advances in Neural Information Processing Systems* 33 (2020): 17443-17454.

Centralized Training - Decentralized Execution

- We talked about how multi-agent renders replay buffers non-stationary?
- What if we could make the replay buffer stationary?
- **Idea:**
    - **Use** all the actions in a global Q function
    - Hide the global actions from the actor
- **Algorithms:** MADDPG**,** MAPPO, COMA, QMIX, 1000+ other algorithms

Centralized Training - Decentralized Execution

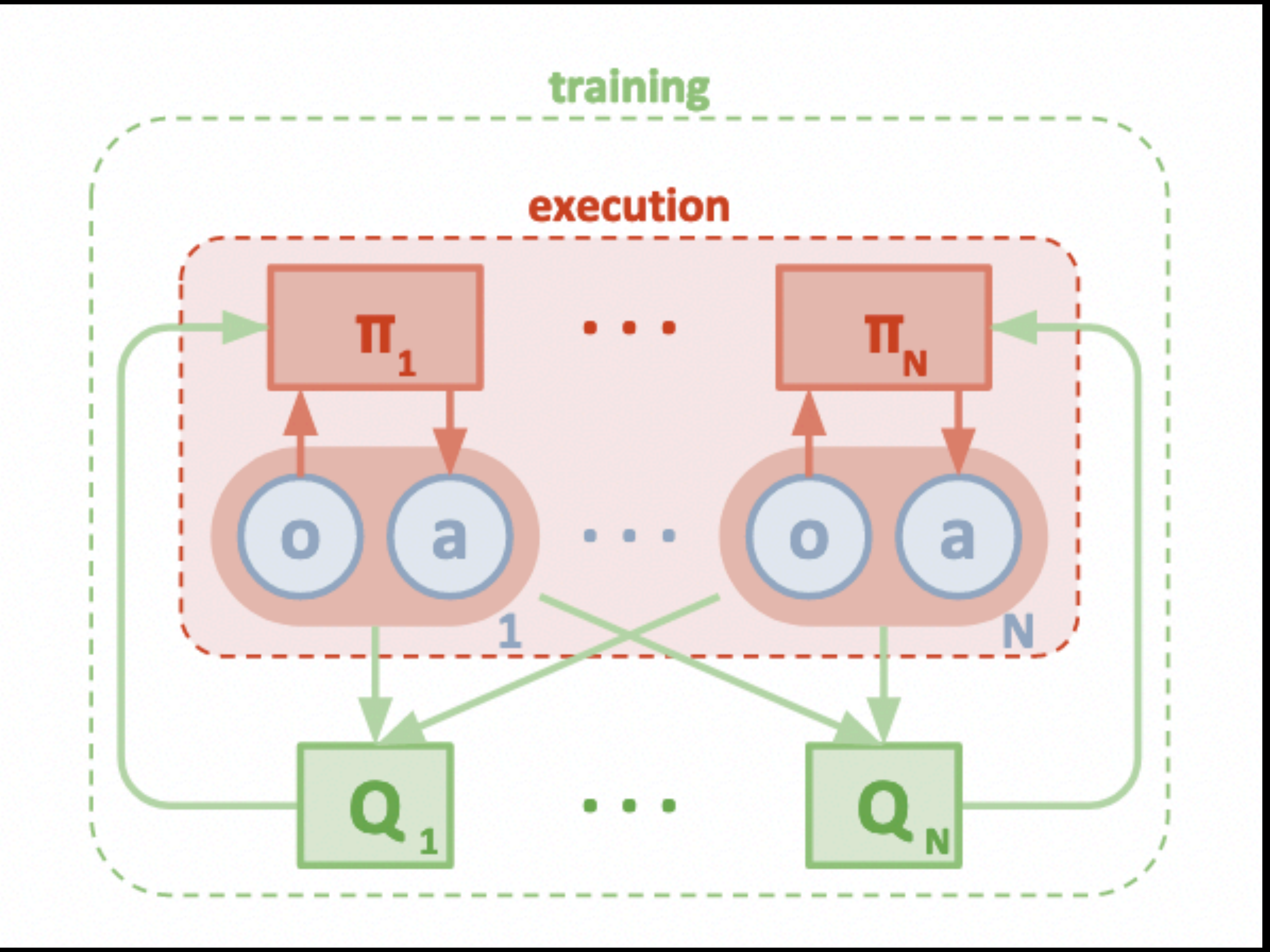
Quick refresher: DDPG

- We have a Q-function $Q(s, a)$ that we learn via Q-learning

- We learn a separate, continuous actor μ via

$$\underset{\mu}{\operatorname{argmax}}\, Q(s, \mu(s))$$

- **Quick warning:** figure on the right has notation **o,** lets pretend it says **s**

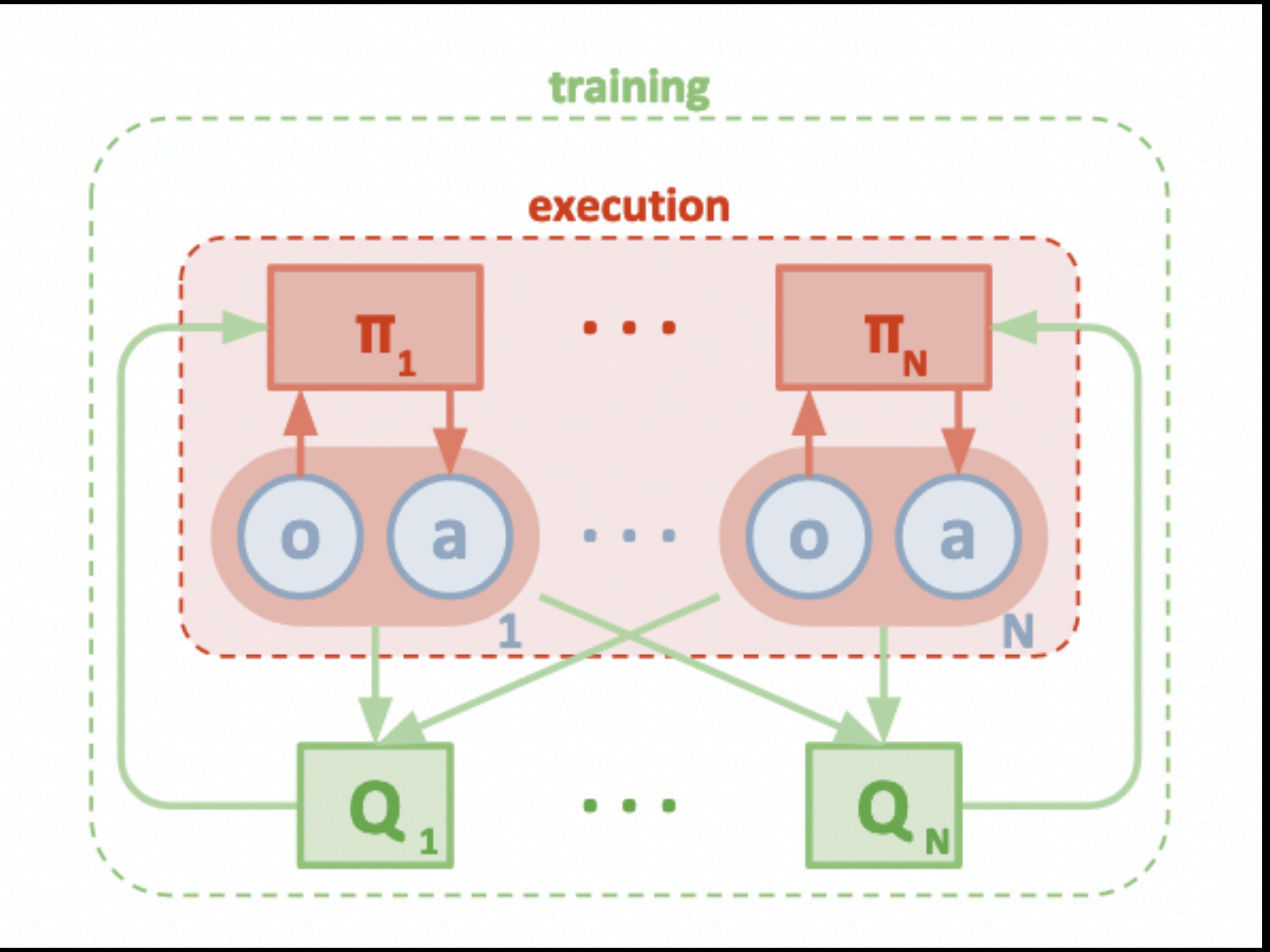


Source: Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." *Advances in neural information processing systems* 30 (2017).

Centralized Training - Decentralized Execution

Why does this “work”? MADDPG example

- We stick all the agent actions into the Q-function
- Then $P(s' \mid s, a_1, a_2, \ldots, a_n)$ is Markovian again and we return to happy-MDP-land
- But wait! We still need to be decentralized at test time.

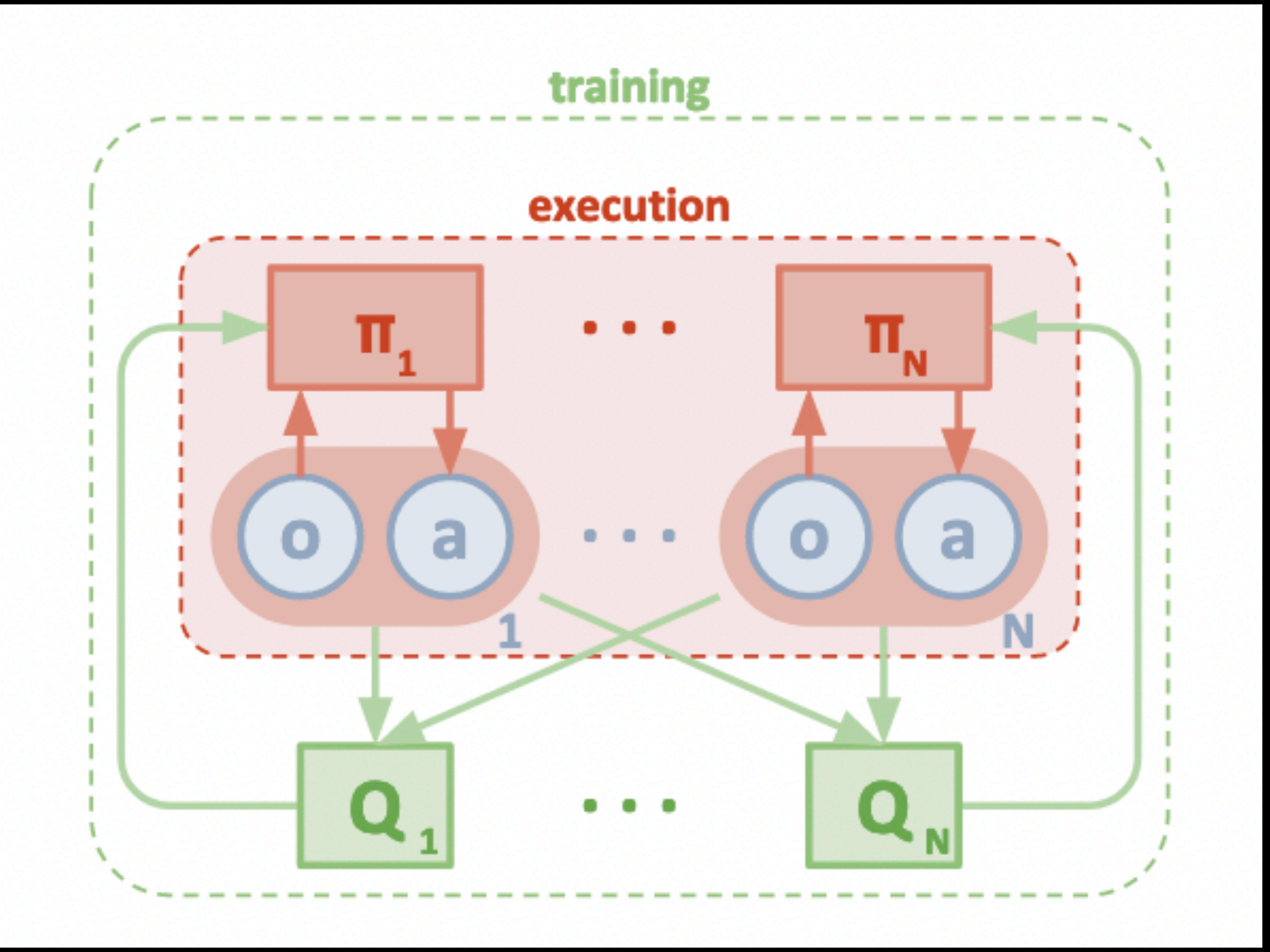


Source: Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." *Advances in neural information processing systems* 30 (2017).

Centralized Training - Decentralized Execution

Why does this “work”?

- But wait! How do we decentralize the actor?
- MADDPG example:
  - The Q-function takes in centralized obs
  - The actor takes in local obs and locally maximizes the global Q-function



Source: Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." *Advances in neural information processing systems* 30 (2017).

Centralized Training - Decentralized Execution

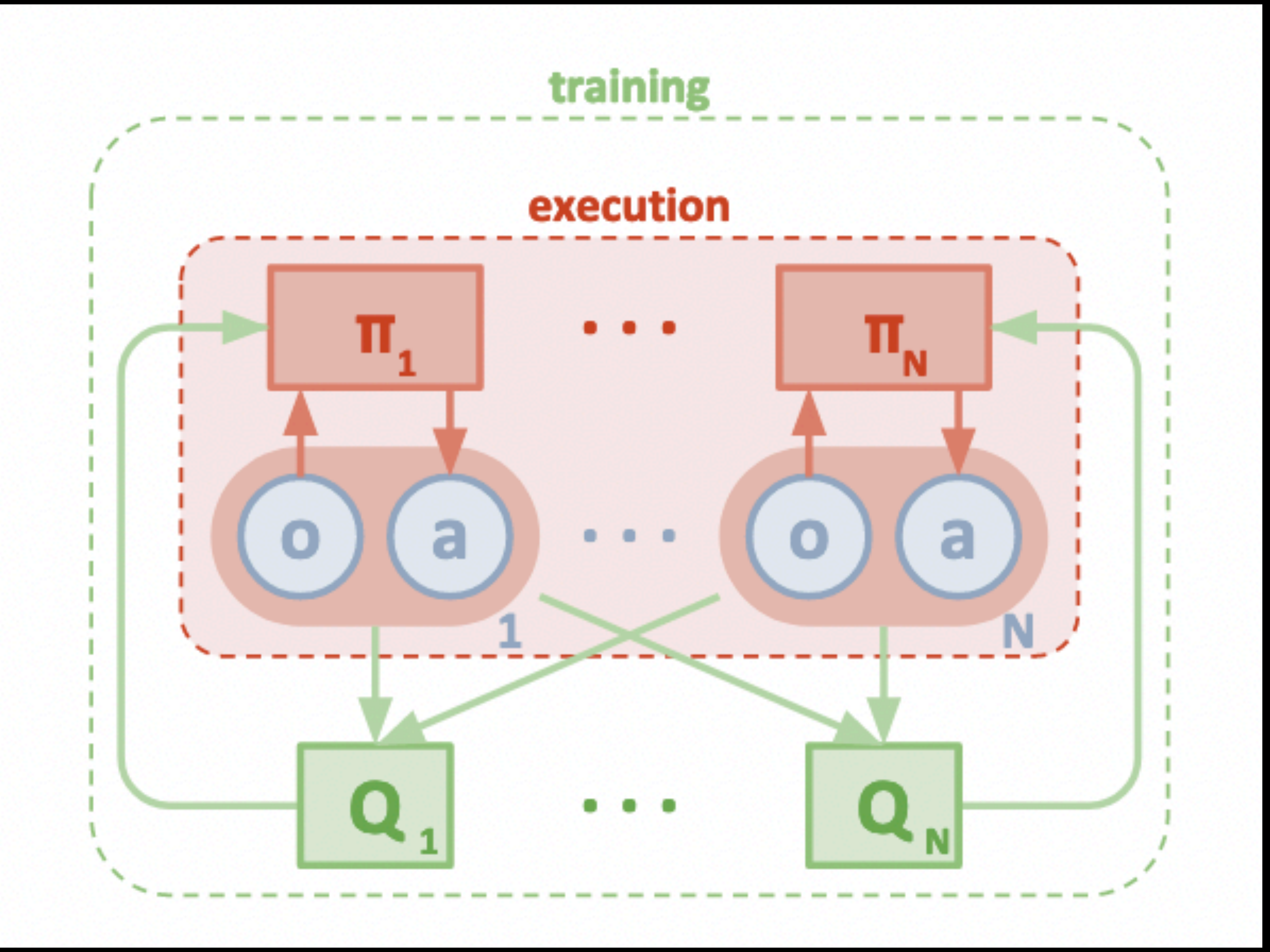
Why does this "work"?

- We now have a Q-function

$$Q(s, a_1, \dots a_i, \dots a_n)$$

- We can still learn an actor the old-fashioned way

$$\underset{\mu}{\text{argmax}}\, Q(\mu(s), a_1, \dots a_i, \dots a_n)$$

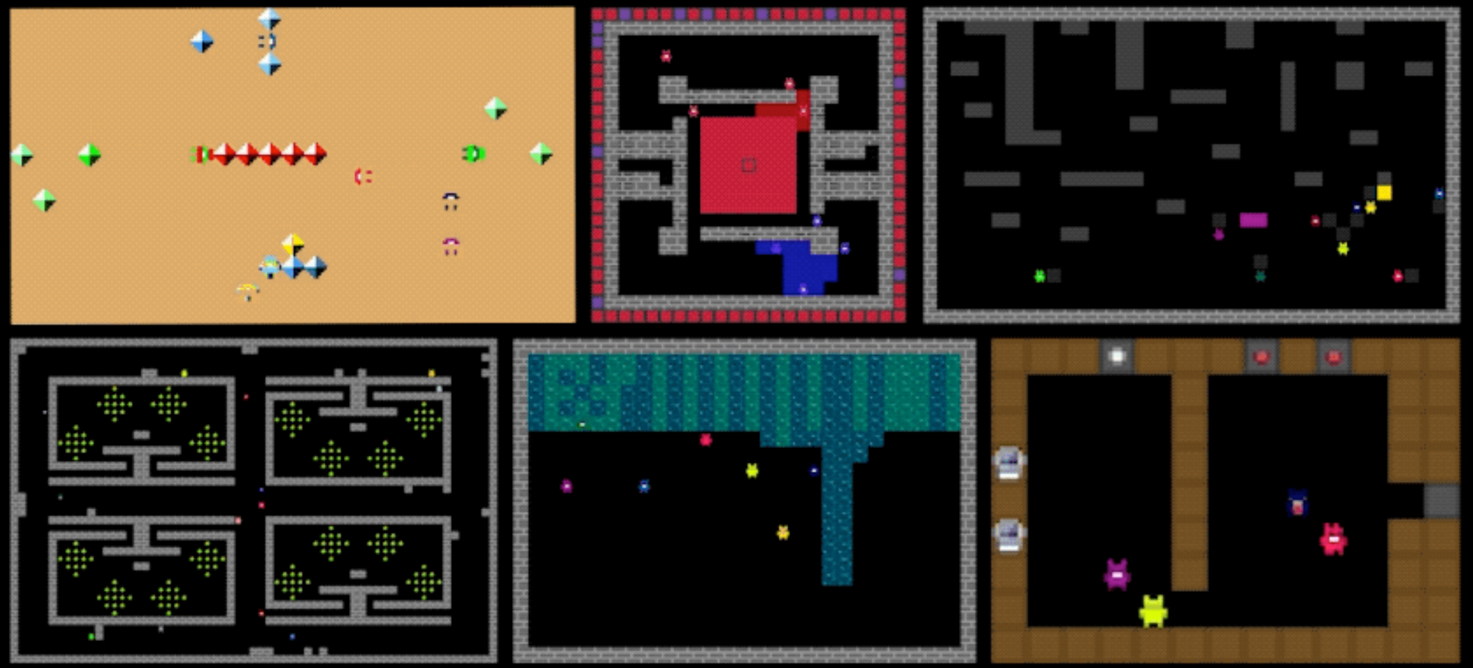


Source: Lowe, Ryan, et al. "Multi-agent actor-critic for mixed cooperative-competitive environments." *Advances in neural information processing systems* 30 (2017).

Research Directions

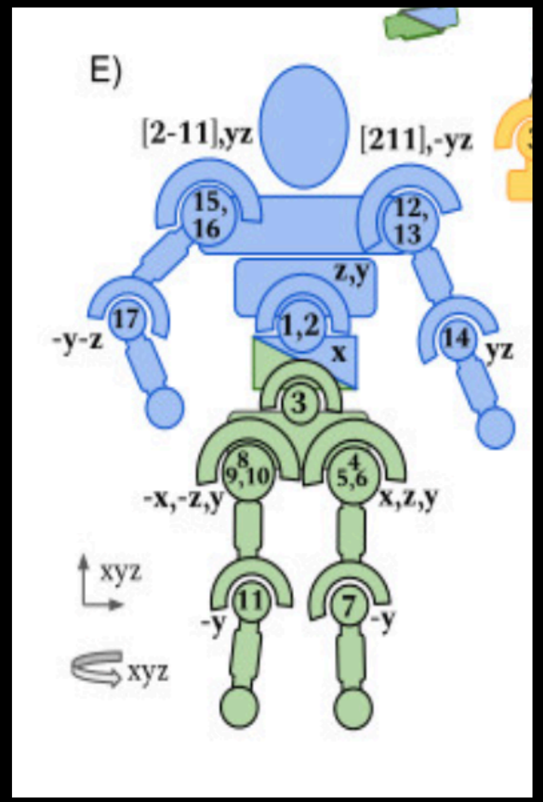
Benchmark design: what's next?

Melting Pot

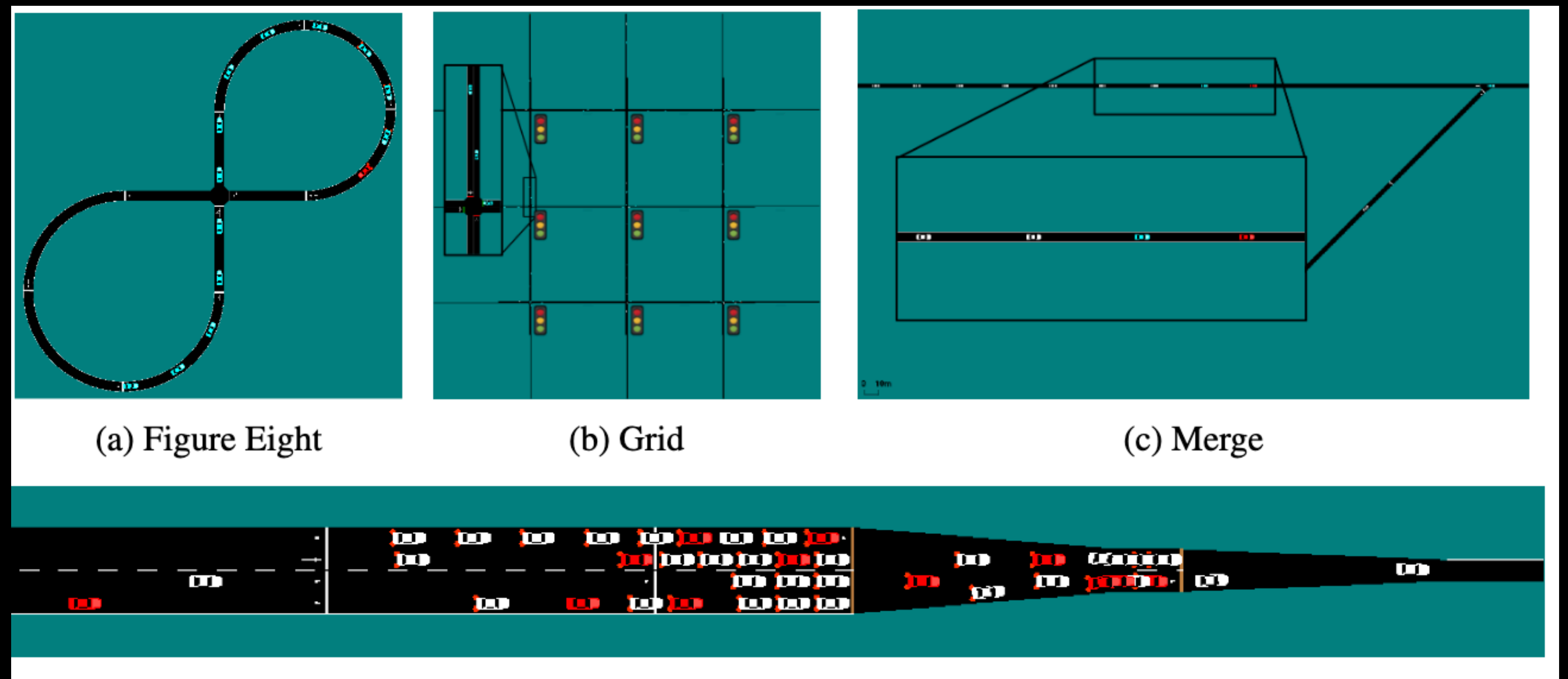


Google Football

MA Mujoco



FLOW

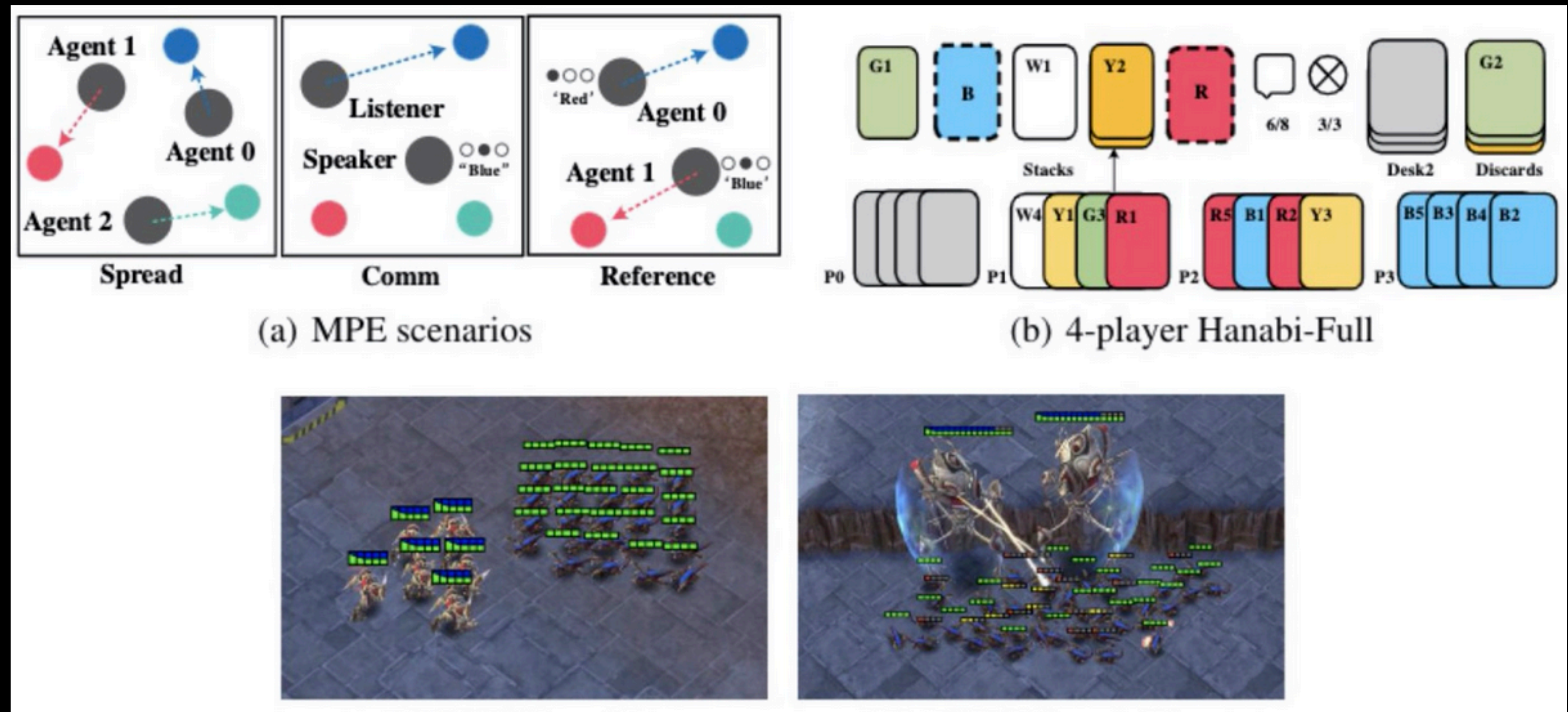


(a) Figure Eight
(b) Grid
(c) Merge
(d) Bottleneck

Overcooked

Poker

Multi-Particle Envs, Hanabi, Decentralized Starcraft



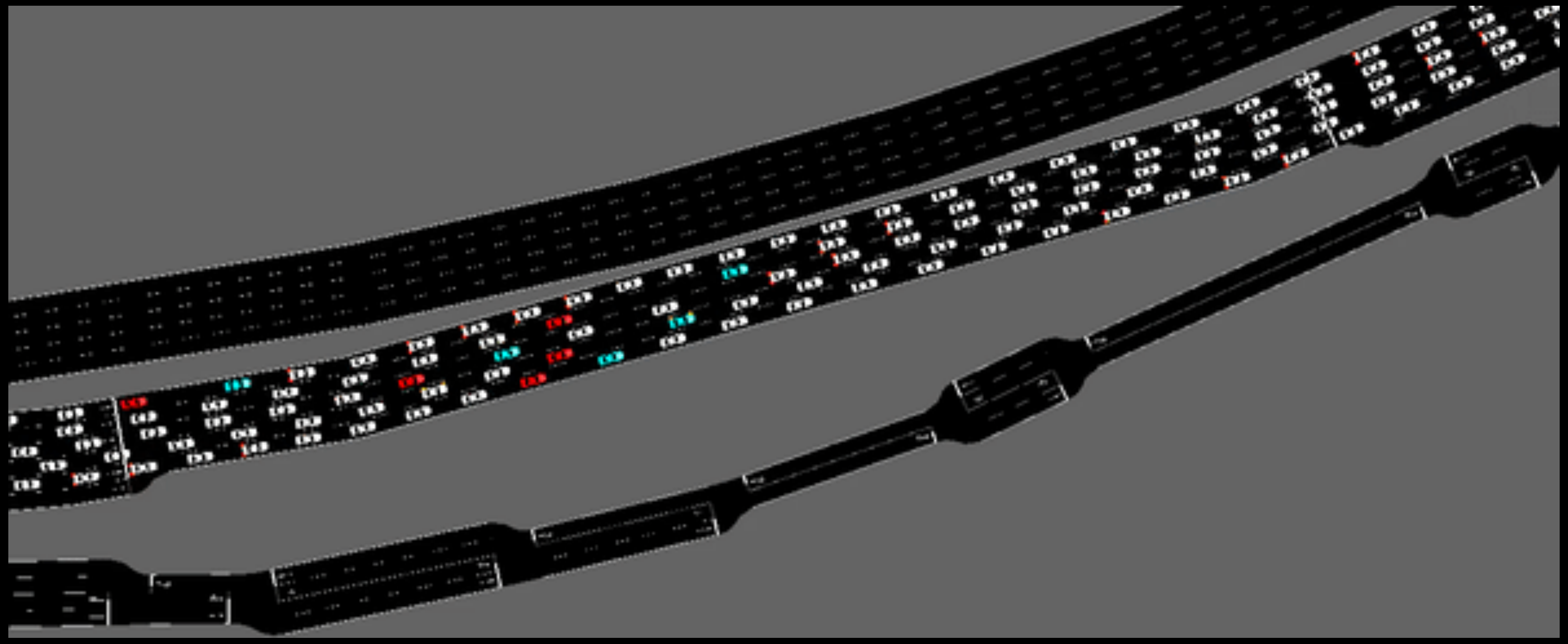
(a) MPE scenarios
(b) 4-player Hanabi-Full
(c) SMAC corridor
(d) SMAC 2c_vs_64zg

Diplomacy

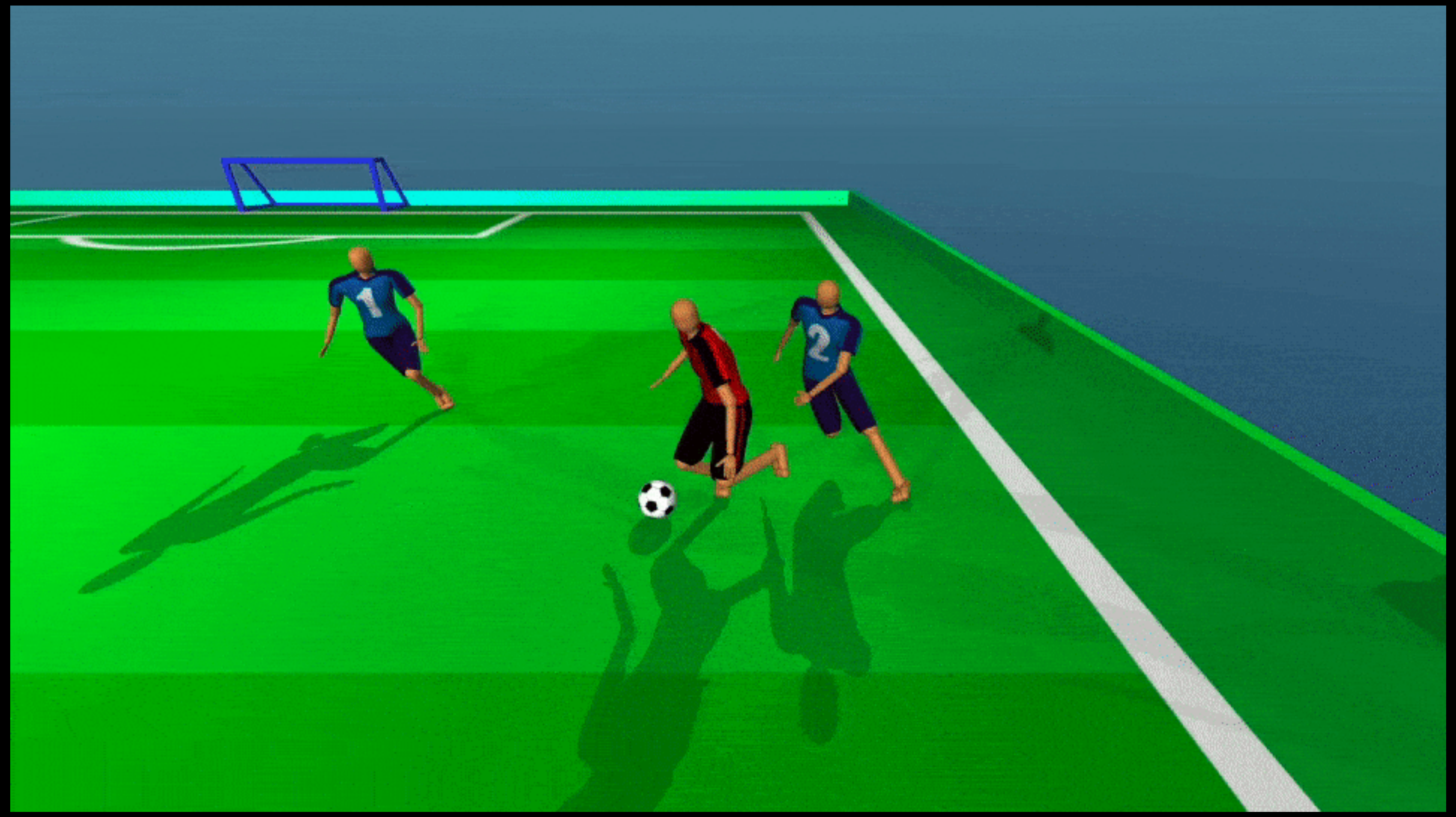
Benchmark design: what's next?

An increasing focus on the real world (I hope)

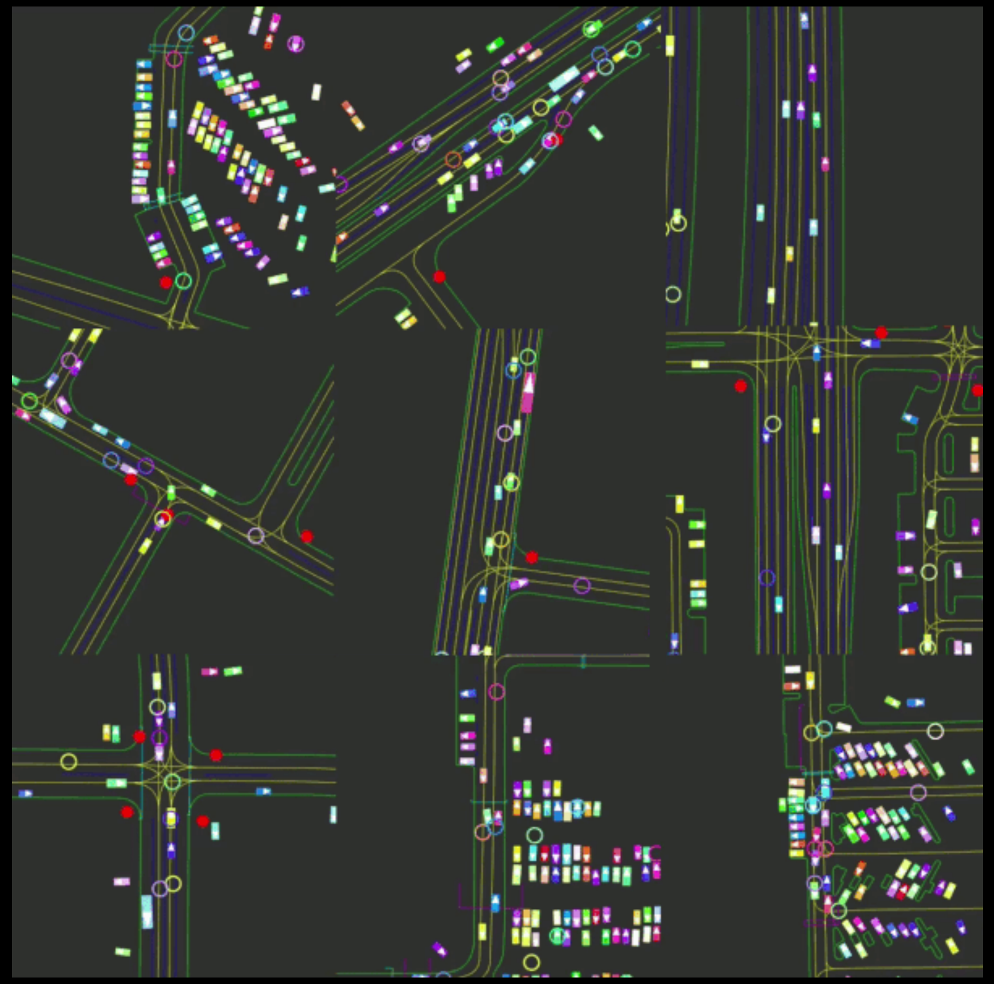
Learning cooperative cruise controllers



Learning sports teams to explore new strategies



Self-driving?



Playing with real humans in spoken language

Some Open Directions (super biased)
Algorithmic Questions

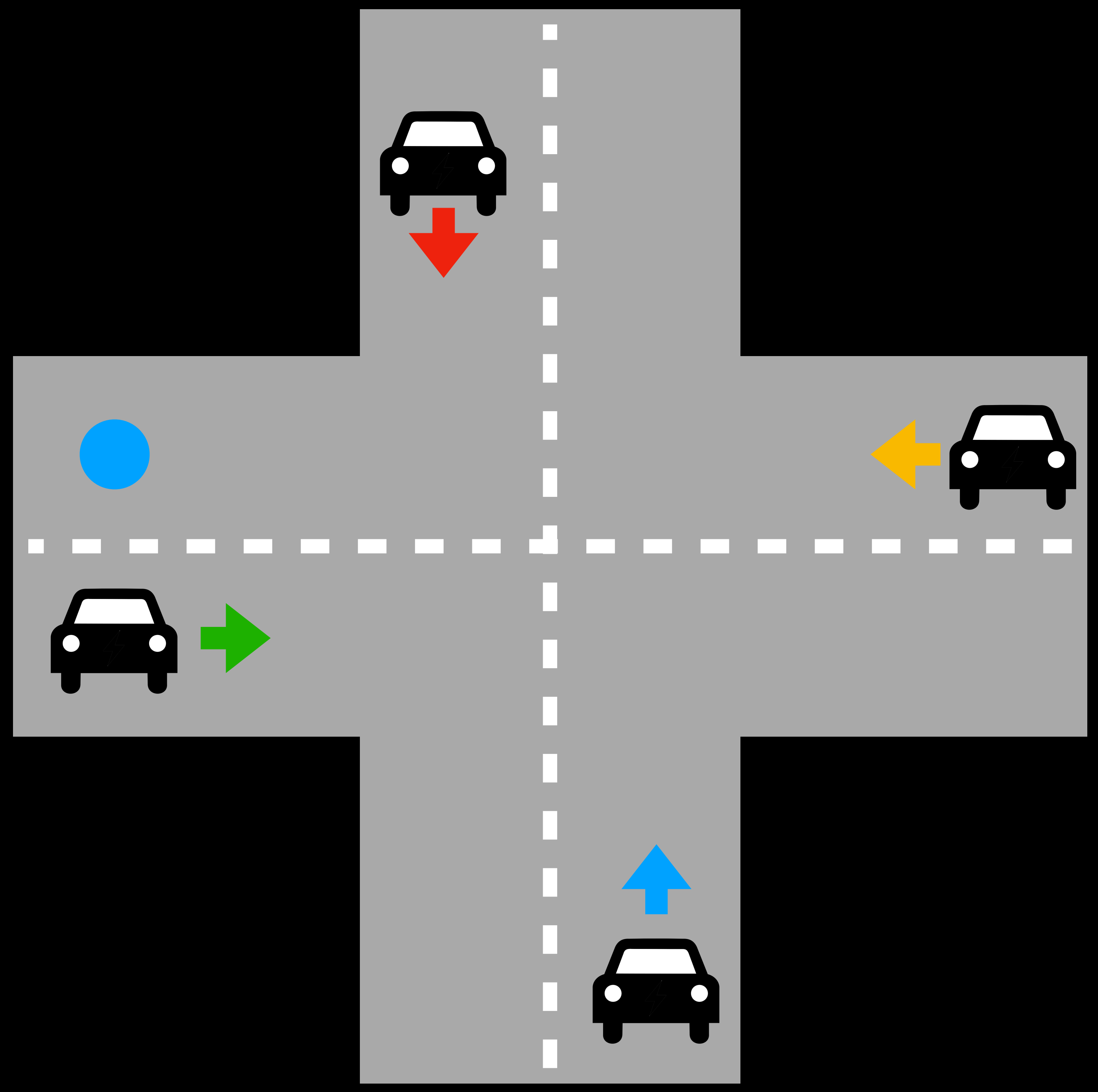
- Does centralized training - decentralized execution help?
    - At convergence, centralized and decentralized Q functions should give same predictions[1]
    - Before convergence, having centralized Q function might speed up learning
    - Empirical evidence is mixed[2,3]
- If they don't help, why?

[1] Lyu, Xueguang, et al. "Contrasting centralized and decentralized critics in multi-agent reinforcement learning." *arXiv preprint arXiv:2102.04402* (2021).
[2] "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games", Yu, Velu, **Vinitsky** et.al.
[3] de Witt, Christian Schroeder, et al. "Is independent learning all you need in the starcraft multi-agent challenge?." *arXiv preprint arXiv:2011.09533* (2020).

Some active areas of research (super biased)

How do we find human-compatible equilibria

- There isn't just one equilibrium in many-player games and non-zero-sum games!
- Imagine the cars have blinkers
- They might learn to use blinkers like "2-left blinks 1-right" means I'm going first
- No human could drive with such cars

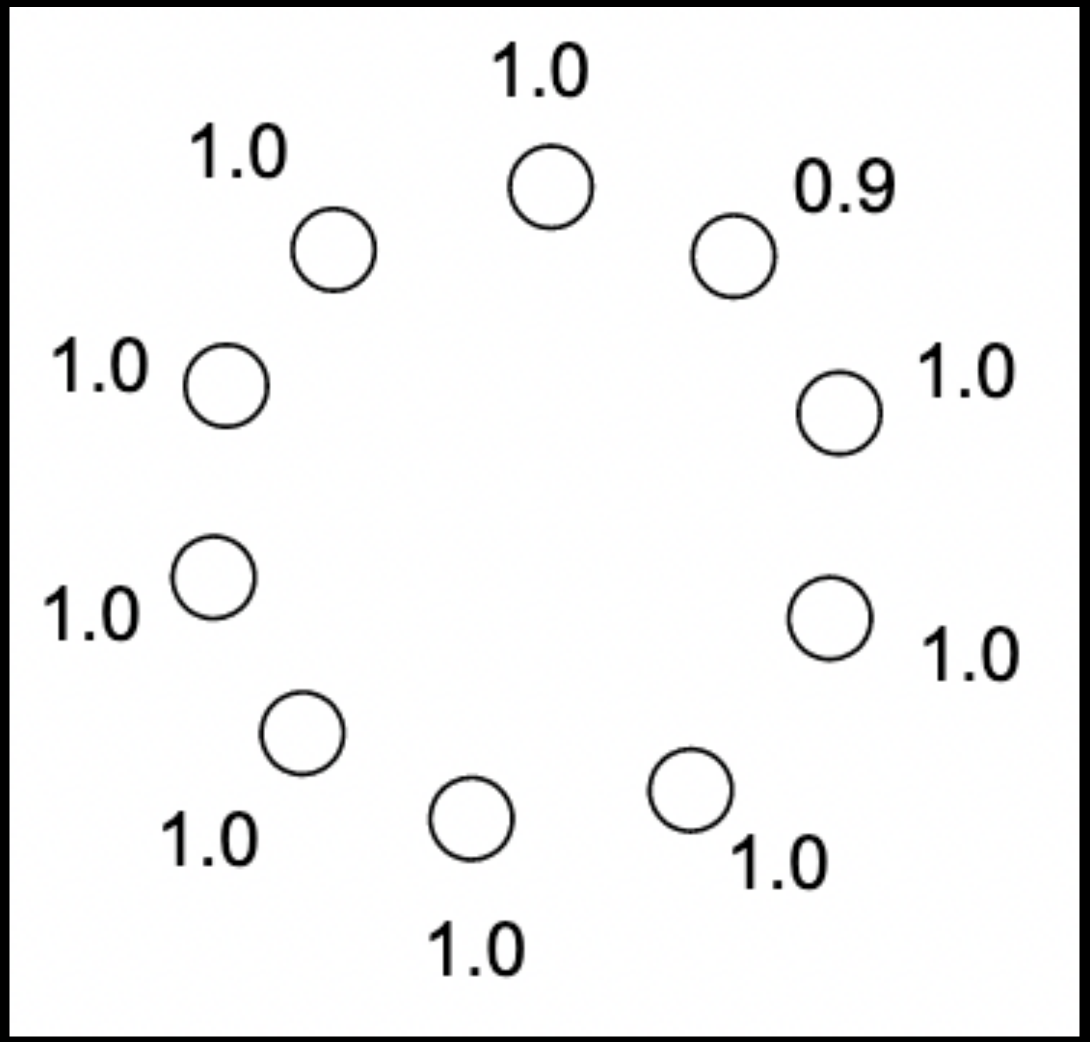


Some active areas of research (super biased)

How do we find human-compatible equilibria

- Agents can evolve non-robust conventions
  - Using uninterpretable nonsense
  - Knowing exactly what the other agent will do
- Lots of work on forcing sensible conventions:
  - Other-play
  - Off-belief learning
  - Incorporation of human data

The Lever Game[1]



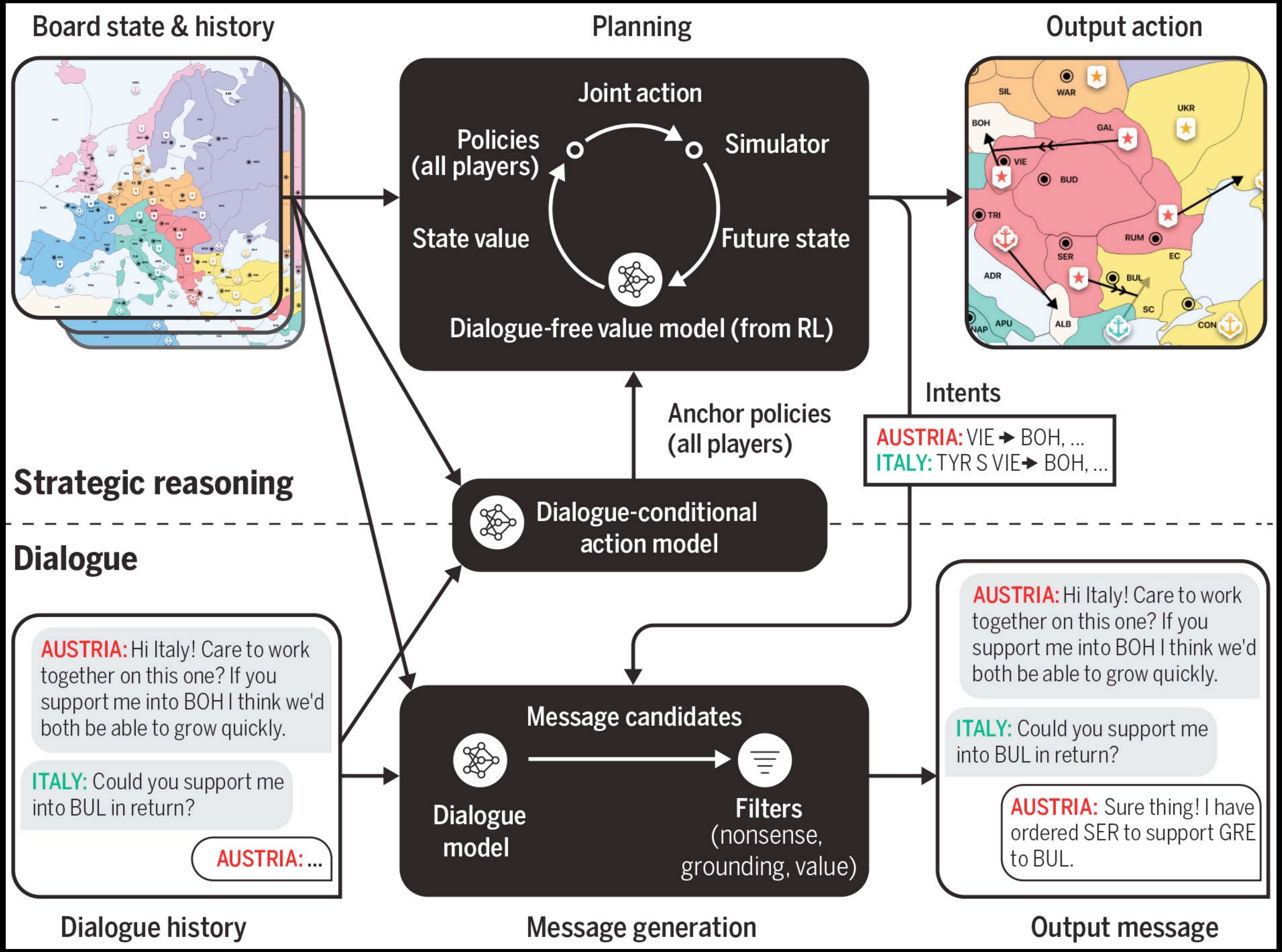
Hu, H., Lerer, A., Peysakhovich, A., & Foerster, J. (2020, November). “Other-Play” for Zero-Shot Coordination. In *International Conference on Machine Learning* (pp. 4399-4410). PMLR.

Some active areas of research (super biased)

How do we find human-compatible equilibria

- Using language

- Using human data as a regularizer



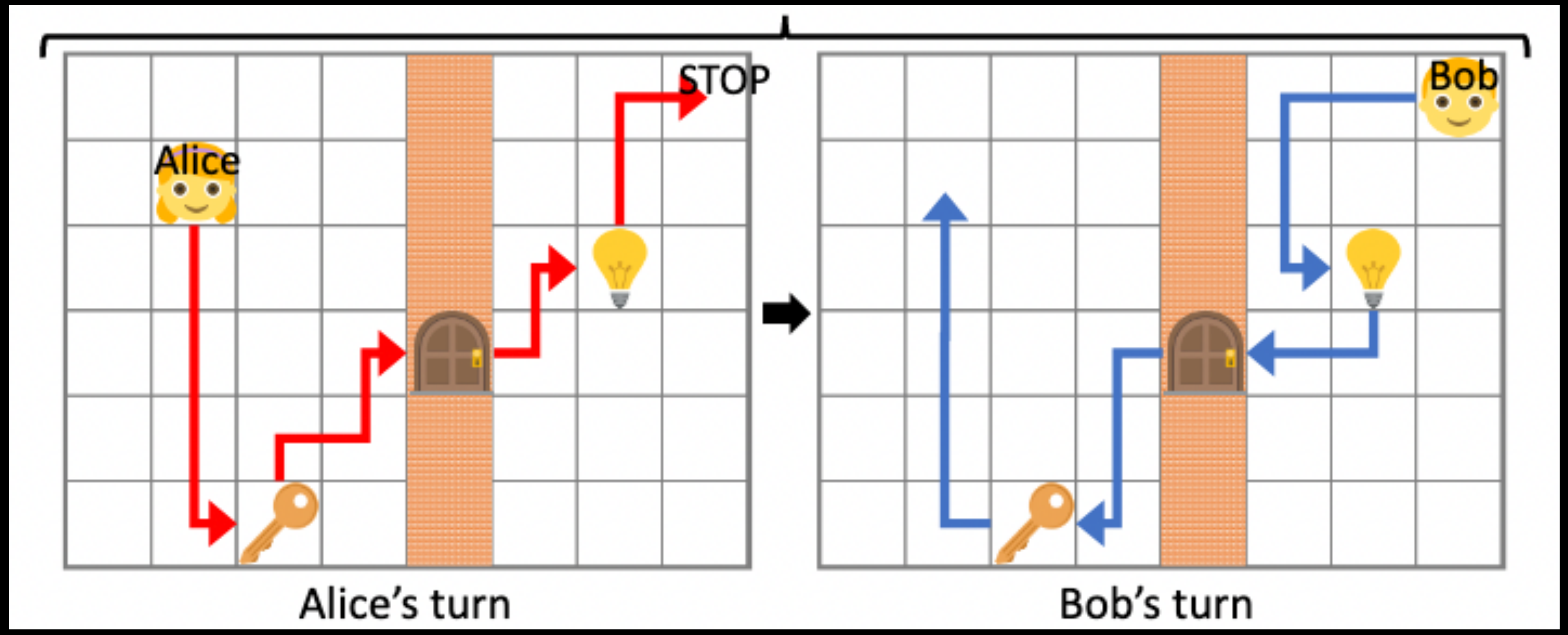
Meta Fundamental AI Research Diplomacy Team (FAIR)†, Bakhtin, A., Brown, N., Dinan, E., Farina, G., Flaherty, C., ... & Zijlstra, M. (2022). Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*

Some active areas of research (super biased)

Curricula from multi-agent

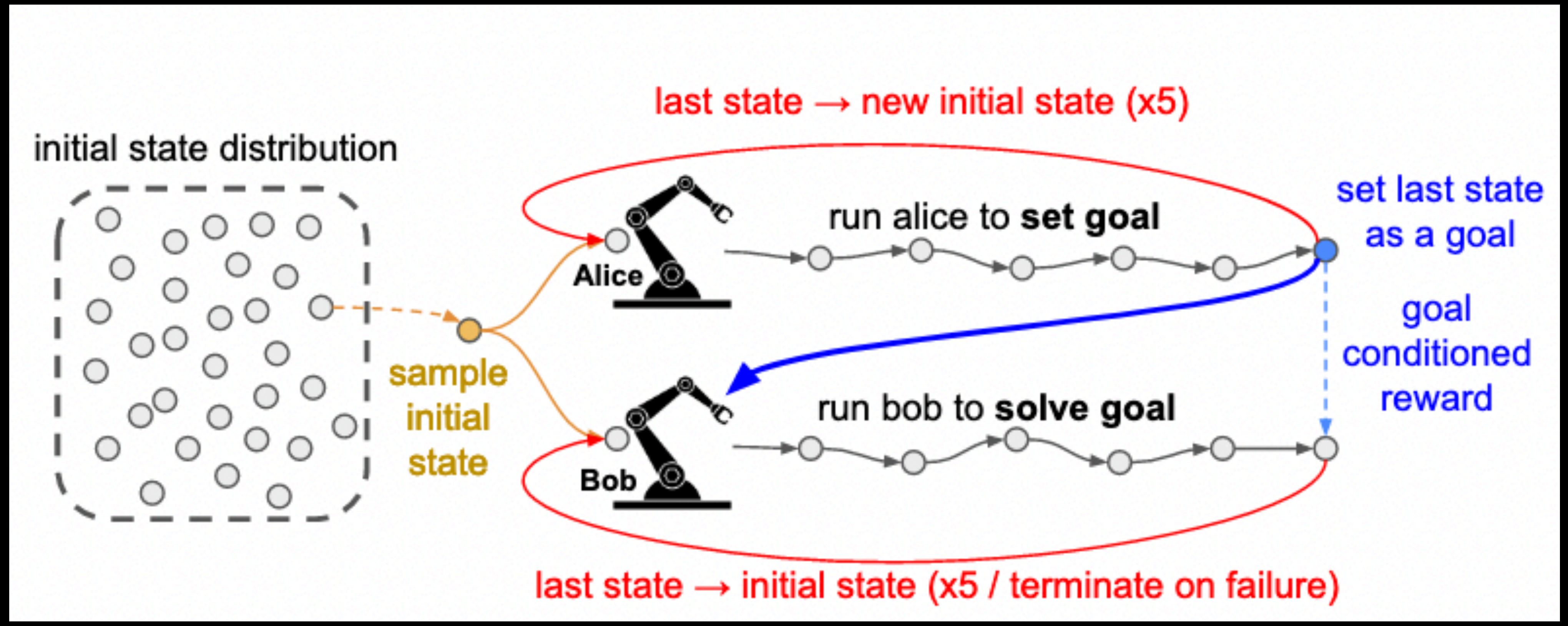
- Asymmetric self-play and friends
- Alice tries to set a hard goal for Bob. As Bob improves, Alice must improve.

Alice reaches a goal, Bob tries to reverse her path



Sukhbaatar, S., Lin, Z., Kostrikov, I., Synnaeve, G., Szlam, A., & Fergus, R. (2017). Intrinsic motivation and automatic curricula via asymmetric self-play. *arXiv preprint arXiv:1703.05407*.

Alice tries to achieve goals Bob can't achieve.



OpenAI, OpenAI, et al. "Asymmetric self-play for automatic goal discovery in robotic manipulation." *arXiv preprint arXiv:2101.04882* (2021).

Some active areas of research

A few more topics

- Scaling up search: search techniques have been critical but we run out of memory in big games
    - DeepNash
    - Monte-Carlo Tree Search
- Convergence Guarantees: can we understand which algorithms will converge to Nash?
- Avoiding the curse of dimensionality: can we find algorithms whose convergence is not exponential in agent number*?

&Jin, Chi, et al. "V-Learning--A Simple, Efficient, Decentralized Algorithm for Multiagent RL." *arXiv preprint arXiv:2110.14555* (2021).

Some useful resources for going further!

- Multi-agent Reinforcement Learning: A Selective Overview of Theories and Algorithms
- Papers linked throughout this talk

Thank you for your time!
Questions?