Multi-Agent Reinforcement Learning: Concepts and Challenges

Shusen Wang

Settings

Settings

1. Fully cooperative.
2. Fully competitive.
3. Mixed Cooperative & competitive.
4. Self-interested.

Fully Cooperative Setting

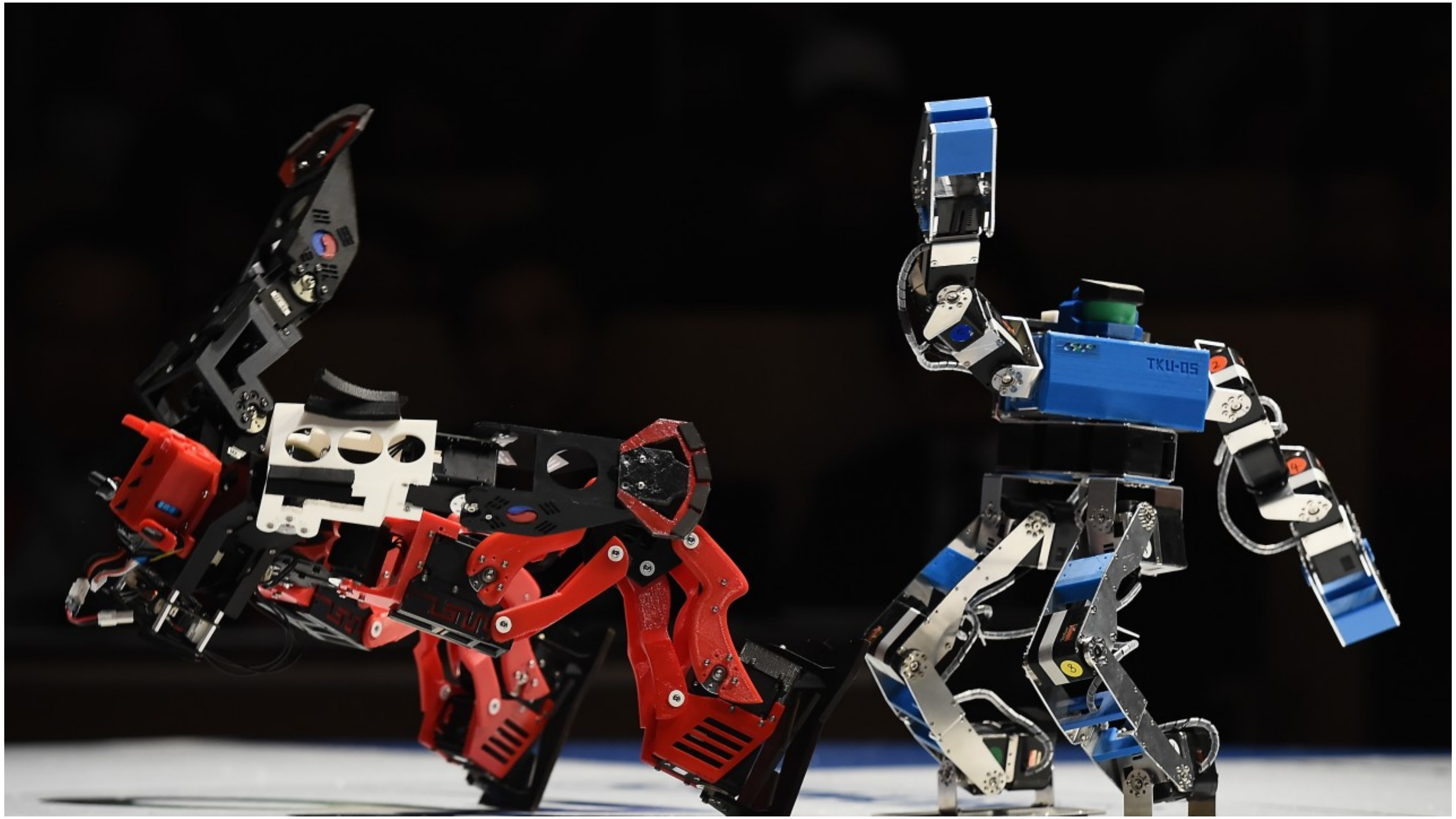
Agents collaborate to optimize a common return.

Fully Cooperative Setting

Agents collaborate to optimize a common return.

Fully Competitive Setting

One agent's gain is the other agent's loss.



Fully Competitive Setting

One agent’s gain is the other agent’s loss.

Mixed Cooperative & Competitive

There are both cooperative setting and competitive setting.

Mixed Cooperative & Competitive

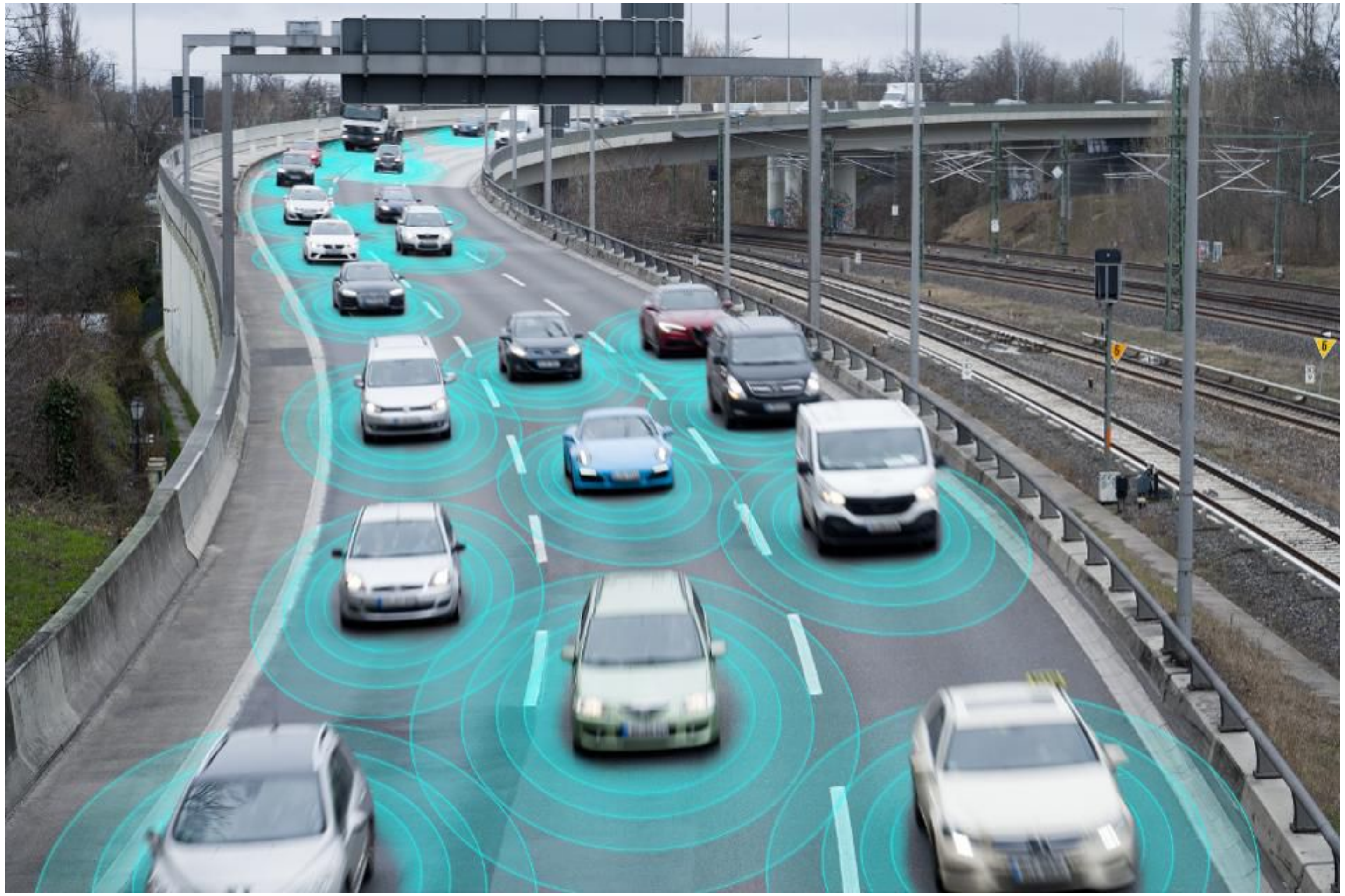
There are both cooperative setting and competitive setting.

Self-Interested Setting

Agents are self-interested. Their rewards may or may not conflict.

Self-Interested Setting

Agents are self-interested. Their rewards may or may not conflict.



Terminologies

State, Action, State Transition

- There are n agents.
- Let S be the state.
- Let A^i be the i-th agent's action.
- State transition:

$$p(s' | s, a^1, \cdots, a^n) = \mathbb{P}(S' = s' | S = s, A^1 = a^1, \cdots, A^n = a^n).$$

- The next state, S', depends on all the agents' actions.

Rewards

- Let R^i be the reward received by the i-th agent.
- Fully cooperative: $R^1 = R^2 = \cdots = R^n$.
- Fully competitive: $R^1 \propto -R^2$.
- R^i depends on A^i as well as all the other agents' actions $\{A^j\}_{j \neq i}$.

Returns

- Let R_t^i be the **reward** received by the i-th agent at time t.
- **Return** (of the i-th agent):

$$U_t^i = R_t^i + R_{t+1}^i + R_{t+2}^i + R_{t+3}^i + \cdots$$

- **Discounted return** (of the i-th agent):

$$U_t^i = R_t^i + \gamma \cdot R_{t+1}^i + \gamma^2 \cdot R_{t+2}^i + \gamma^3 \cdot R_{t+3}^i + \cdots$$

Here, $\gamma \in [0, 1]$ is the discount rate.

Policy Network

- Each agent has its own policy network: $\pi(a^i \mid s; \boldsymbol{\theta}^i)$.
- Policy networks can be exchangeable: $\boldsymbol{\theta}^1 = \boldsymbol{\theta}^2 = \cdots = \boldsymbol{\theta}^n$.
  - Self-driving cars can have the same policy.
- Policy networks can be nonexchangeable: $\boldsymbol{\theta}^i \neq \boldsymbol{\theta}^j$.
  - Soccer players have different roles, e.g., striker, defender, goalkeeper.

Uncertainty in the Return

- The reward R_t^i depends on S_t and $A_t^1, A_t^2, \cdots, A_t^n$.
- Uncertainty in S_t is from the state transition, p.
- Uncertainty in A_t^i is from the policy network, $\pi(\cdot \mid s_t; \boldsymbol{\theta}^i)$.
- The return, $U_t^i = \sum_{k=0}^{\infty} \gamma^k \cdot R_{t+k}^i$, depends on:
  - all the future states: $\{S_t, S_{t+1}, S_{t+2}, \cdots\}$;
  - all the future actions: $\{A_t^i, A_{t+1}^i, A_{t+2}^i, \cdots\}$, for all $i = 1, \cdots, n$.

State-Value Function

- State-value of the i-th agent:

$$V^i(s_t; \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n) = \mathbb{E}\left[U_t^i \mid S_t = s_t\right].$$

- The expectation is taken w.r.t. all the future actions and states except S_t.

- Randomness in actions: $A_t^j \sim \pi\left(\cdot \mid s_t; \boldsymbol{\theta}^j\right)$, for all $j = 1, \cdots, n$.
(That is why the state-value V^i depends on $\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n$.)

State-Value Function

- One agent's state-value, $V^i(s; \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n)$, depends on all the agents' policies.
- If any agent changes its policy, then all of $V^1, \cdots, V^n$ can change.
- Example: soccer game.
    - A striker improves his policy, while everyone else's policies are fixed.
    - His teammates' state-values all increase.
    - The opposing players' state-values all decrease.

Convergence

Single-Agent Policy Learning

- Policy network: $\pi(a \mid s; \boldsymbol{\theta})$.
- State-value function: $V(s; \boldsymbol{\theta})$.
- $J(\boldsymbol{\theta}) = \mathbb{E}_S[V(S; \boldsymbol{\theta})]$ evaluates how good the policy is.
- Learn the policy network's parameter, $\boldsymbol{\theta}$, by

$$\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}).$$

- **Convergence:** $J(\boldsymbol{\theta})$ stops increasing.

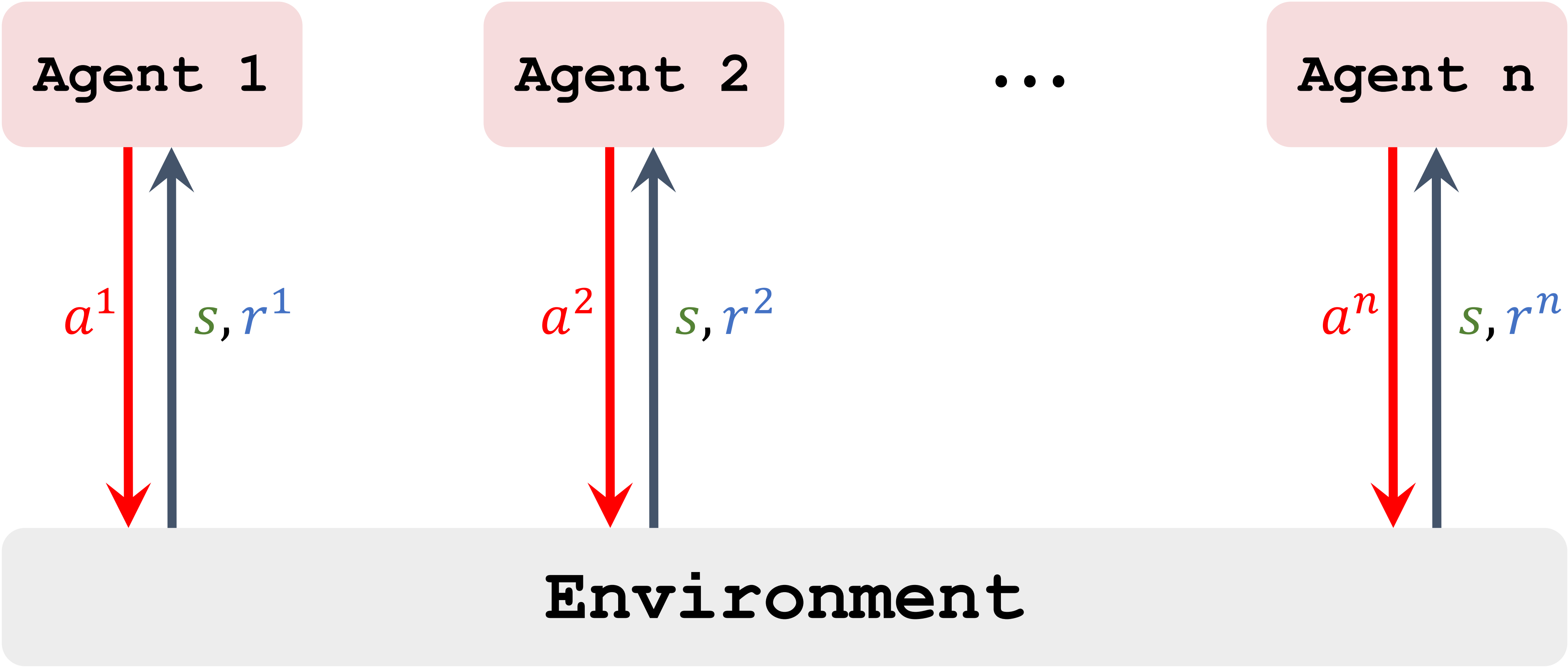
Multi-Agent Policy Learning

Nash Equilibrium

- While all the other agents' policy remain the same, the i-th agent cannot get better expected return by changing its own policy.
- Every agent is playing a best-response to the other agents' policies.
- Nash equilibrium indicates convergence because no one has any incentive to deviate.

Difficulty of MARL

Single-Agent Policy Gradient for MARL



Single-Agent Policy Gradient for MARL

- The i-th agent's policy network: $\pi\left(a^i \mid s; \boldsymbol{\theta}^i\right)$.
- The i-th agent's state-value function: $V^i(s; \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n)$.
- Objective function: $J^i(\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n) = \mathbb{E}_S\left[V^i(S; \boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n)\right]$.
- Learn the policy network's parameter, $\boldsymbol{\theta}^i$, by

$$\max_{\boldsymbol{\theta}^i} J^i(\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n).$$

Single-Agent Policy Gradient for MARL

- The 1^{st} agent solves: $\max_{\boldsymbol{\theta}^1} J^1(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \boldsymbol{\theta}^n)$.
- The 2^{nd} agent solves: $\max_{\boldsymbol{\theta}^2} J^2(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \boldsymbol{\theta}^n)$.

⋮

- The n^{th} agent solves: $\max_{\boldsymbol{\theta}^n} J^n(\boldsymbol{\theta}^1, \boldsymbol{\theta}^2, \cdots, \boldsymbol{\theta}^n)$.

It may not converge…

Single-Agent Policy Gradient for MARL

What is wrong?

- The i-th agent found $\boldsymbol{\theta}_\star^i = \underset{\boldsymbol{\theta}^i}{\operatorname{argmax}}\, J^i(\boldsymbol{\theta}^1, \cdots, \boldsymbol{\theta}^n)$.
- Now, another agent changes its policy.
- So $\boldsymbol{\theta}_\star^i$ is no longer the best policy of the i-th agent. The i-th agent has to find a new $\boldsymbol{\theta}^i$.
- The other agents' objective functions will change, and therefore they will change their policies…

Summary

Multi-Agent Reinforcement Learning (MARL)

- There are $n > 1$ agents in the system.
- The agents are usually not independent.
  - Every agent's action can affect the next state.
  - Thus, every agent can affect all the other agents.
- Unless the agents are independent of each other, single-agent RL methods do not work well for MARL.

Settings of MARL

1. **Fully cooperative**, e.g., industrial robots.
2. **Fully competitive**, e.g., predator and prey.
3. **Mixed cooperative & competitive**, e.g., robotic soccer.
4. **Self-interested**, e.g., automated trading systems.

Convergence

- **Convergence:** No agent can get better expected return by improving its own policy.
- If there is only one agent, convergence means the objective function does not increase any more.
- If there are multiple agents, Nash equilibrium means convergence.

Thank you!

Stationary VS Non-stationary

- Consider single-agent setting.
- Stationary environment requires state transition be fixed throughout.
  - State transition: $p(s'|s, a)$.
  - Given s and a, the probability distribution of the next state s' is always the same.
- All the single-agent RL methods we have learned so far require stationary environment.

Stationary VS Non-stationary

- Consider multi-agent setting.
- Stationary environment requires state transition be fixed throughout.
  - State transition: $p(s'|s, a^1, \cdots, a^n)$.
  - Given s and $a^1, \cdots, a^n$, the probability distribution of the next state s' is always the same.

Stationary VS Non-stationary

- Consider multi-agent setting.
- Stationary environment requires state transition be fixed throughout.
- The environment is typically stationary.
- However, from any single agent's perspective, the environment is non-stationary.
  - p depends not only on s and a^i, but also on the other agents' actions.
  - If the i-th agent knows only s and a^i, then from its perspective, the state transition is not fixed.

Stationary VS Non-stationary

- Consider multi-agent setting.
- Stationary environment requires state transition be fixed throughout.
- The environment is typically stationary.
- However, from any single agent's perspective, the environment is non-stationary.
- Thus, the single-agent RL method we have learned are not applicable.