Decentralized Multi-Agents by Imitiation of a Centralized Controller

Speaker: Alex Tong Lin
Co-authors: Mark J. Debord, Katia Estabridis, Gary Hewer, Guido Montúfar, Stanley Osher

MSML21

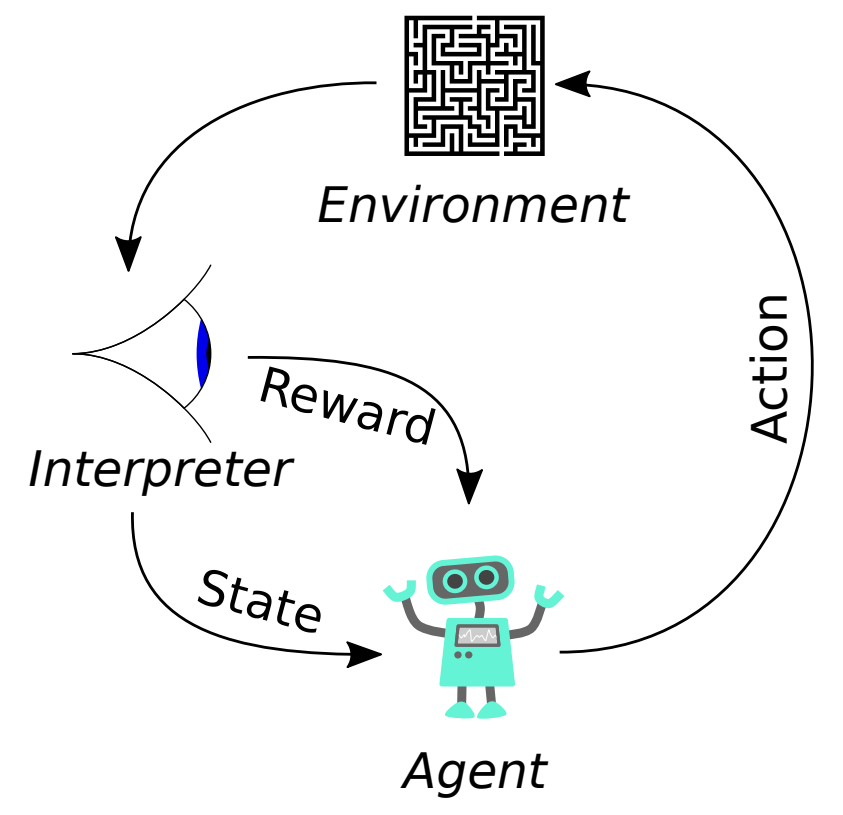
Outline

1. Multi-Agent Reinforcement Learning
2. The Bottom-Up Approach
3. The Top-Down Approach and CESMA
4. Numerical Experiments

Multi-Agent Reinforcement Learning

Reinforcement Learning

Reinforcement Learning (RL) is the study of determining how an agent should take actions in an environment in order to maximize a cumulative reward.

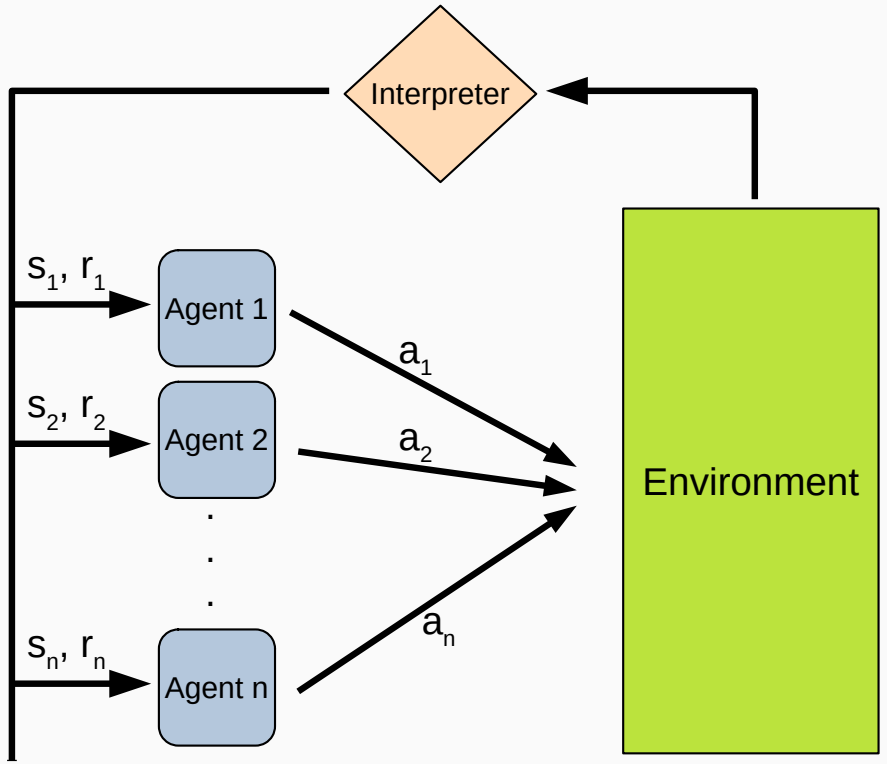


[1]

[1] (image credit: Wikipedia article for "Reinforcement Learning", author: Megajuice)

Multi-Agent Reinforcement Learning

Multi-Agent Reinforcement Learning (MARL) is the study of determining how multiple agents should take actions interacting with an environment and amongst themselves, in order to maximize their own cumulative reward. A key feature is that the agents observe and act in a *decentralized* fashion. And note from the perspective of each agent, the environment is non-Markovian.



Cooperative MARL

Cooperative Multi-Agent Reinforcement Learning is multi-agent reinforcement learning except now there is a single reward signal that is shared among all agents, i.e. $r = r_1 = \cdots = r_n$.

[2]

[2] (image credit: Charlee Riggio, https://executiveforum.net/ready-set-start-moving-mountains/)

RL and Cooperative MARL value functions

- Value function for RL

$$\max_{\pi} V(s_0) = \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t R(a_t, s_t) \mid s_0, \pi\right]$$

- Value function for Cooperative MARL

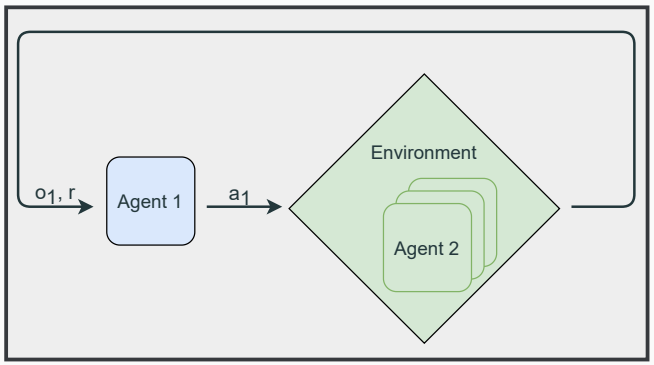
$$\max_{(\pi_1,\ldots,\pi_M)} V(\mathbf{o}_0) = \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t R(\mathbf{a}_t, \mathbf{o}_t) \mid \mathbf{o}_0, (\pi_1, \ldots, \pi_M)\right]$$

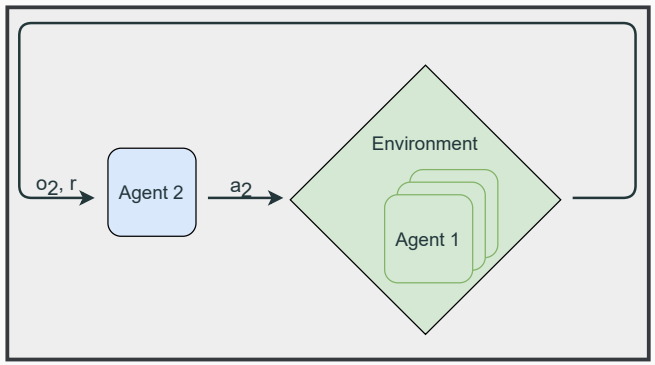
  where we now have *joint* observations and actions. Note the terms "observation" and "state" are synonymous, but observations is more widely used in MARL.

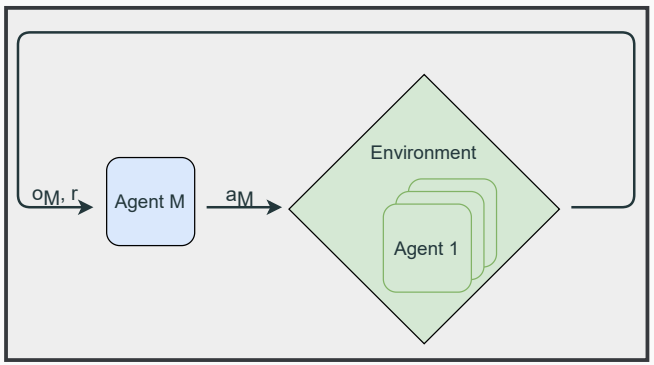
The Bottom-Up Approach

Independent Q-Learning

- The most straightforward way to extend RL to the multi-agent setting is to have each agent learn independently.
- From the perspective of each agent, all other agents are part of the environment.
- Each agent will have their own value function that they try to maximize, disregarding the cooperative aspect.

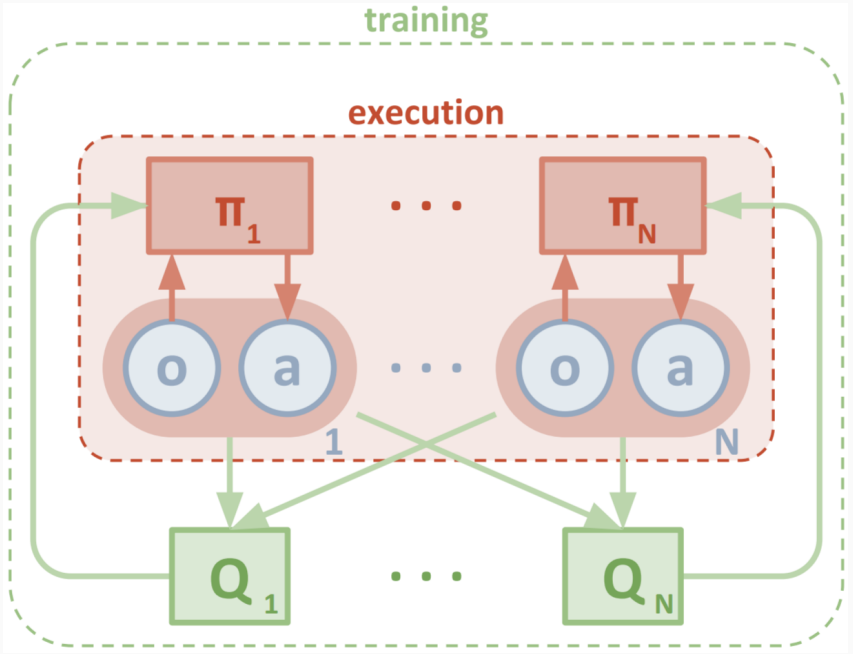






Multi-Agent Deep Deterministic Policy Gradient (MADDPG)

- Multi-Agent Deep Deterministic Policy Gradient (MADDPG) takes the Actor-Critic model and expands it into the multi-agent setting.
- It uses the framework of centralized learning, but decentralized execution.
- The critic observes the joint states and actions, while the actor only observes the local state.



[3]

[3] (image credit: Multi-Agent Deep Deterministic Policy Gradient, Lowe et al.)

The Bottom-Up Approach

- These methods are examples of the "bottom-up" approach to obtain decentralized multi-agents.
- They build up the solution while simultaneously solving the environment.

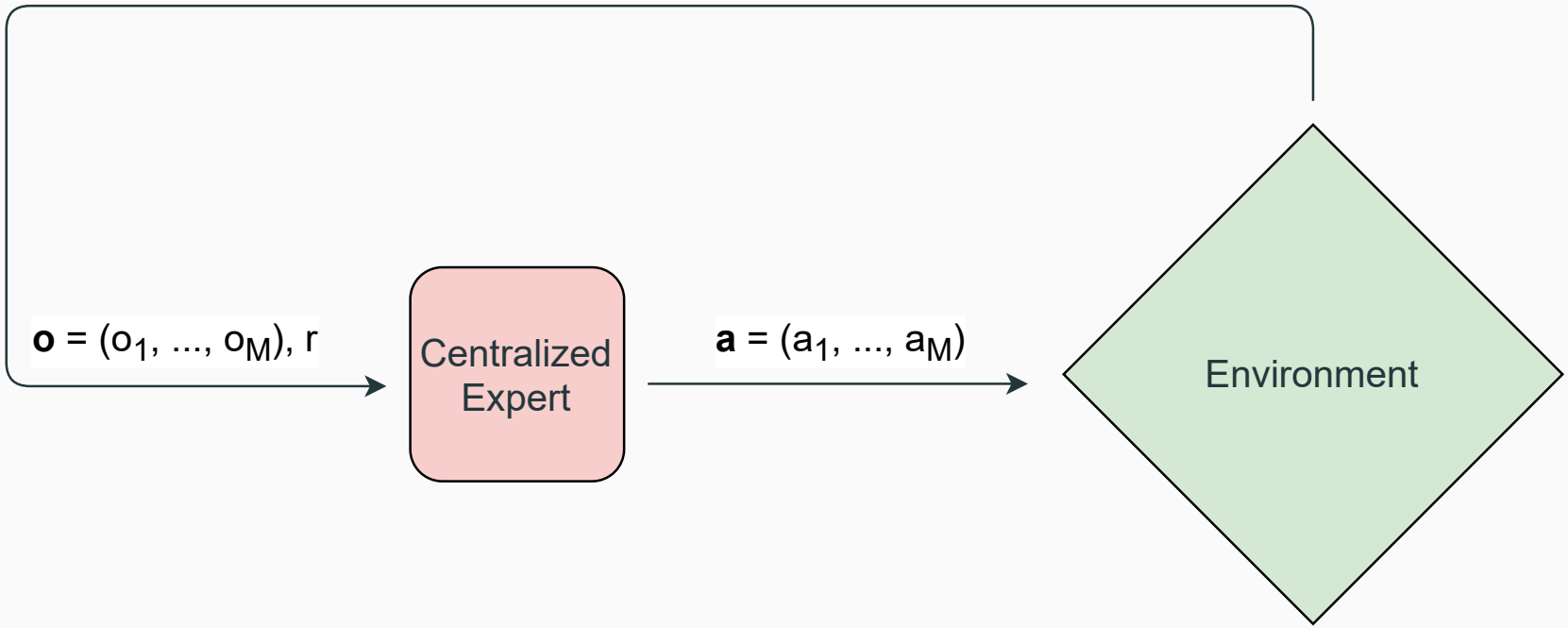
The Top-Down Approach and CESMA

The Top-Down Approach and CESMA

- Centralized Expert Supervises Multi-Agents (CESMA) first trains a centralized expert to solve the environment, then we use imitation learning to decentralize the solution to obtain multi-agents.
- To train the centralized expert, we can use any single-agent reinforcement learning algorithm.
- To decentralize the expert, we can use any imitation learning algorithm.
- Also centralized learning, but decentralized execution.

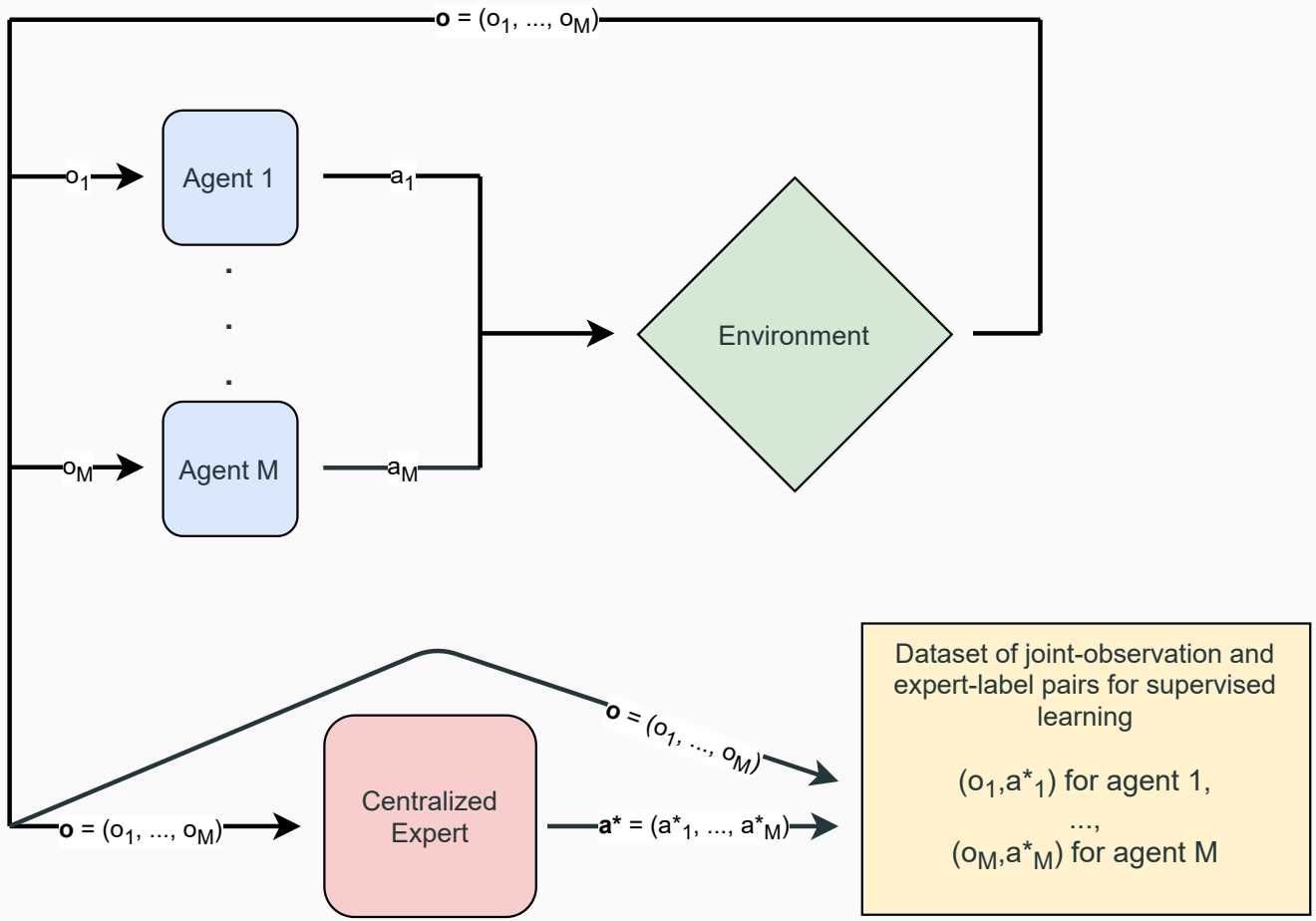
CESMA Phase 1

You can train the centralized expert using any single-agent reinforcement learning algorithm.



CESMA Phase 2

As agents run through environment, create a dataset of observation-label pairs, where the labels are expert actions. Then every k times the agents run the environment, use this dataset for supervised learning/imitation learning. Based on Dataset Aggregation (DAgger, Ross et al., 2011).

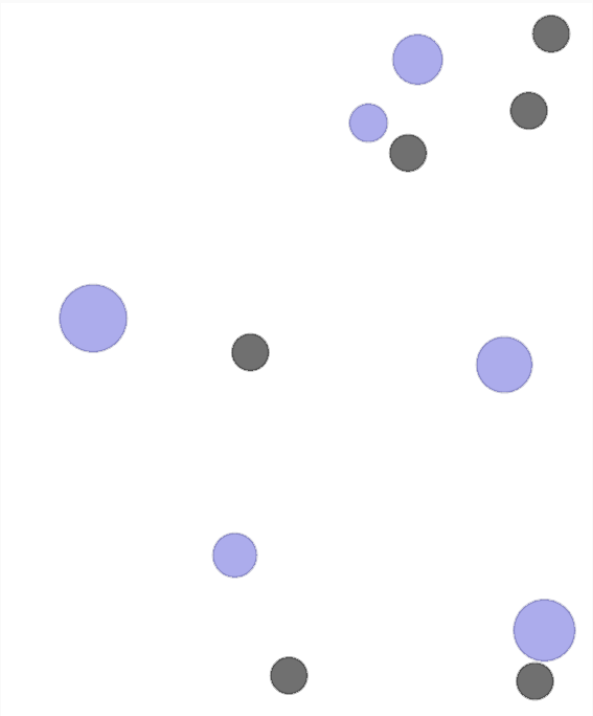


Numerical Experiments

Cooperative Navigation

- The goal of the agents (purple) is to cover the landmarks (grey).
- Reward calculation: letting $\{q_\ell\}_{\ell=1}^{L}$ be the landmark positions, and $\{p_i\}_{i=1}^{M}$ be the agent positions, then

$$\text{reward} = -\sum_{\ell=1}^{L} \min_{i=1,\ldots,M} \|q_\ell - p_i\| - (\#\text{ of collisions}) \qquad (1)$$



Cooperative Navigation

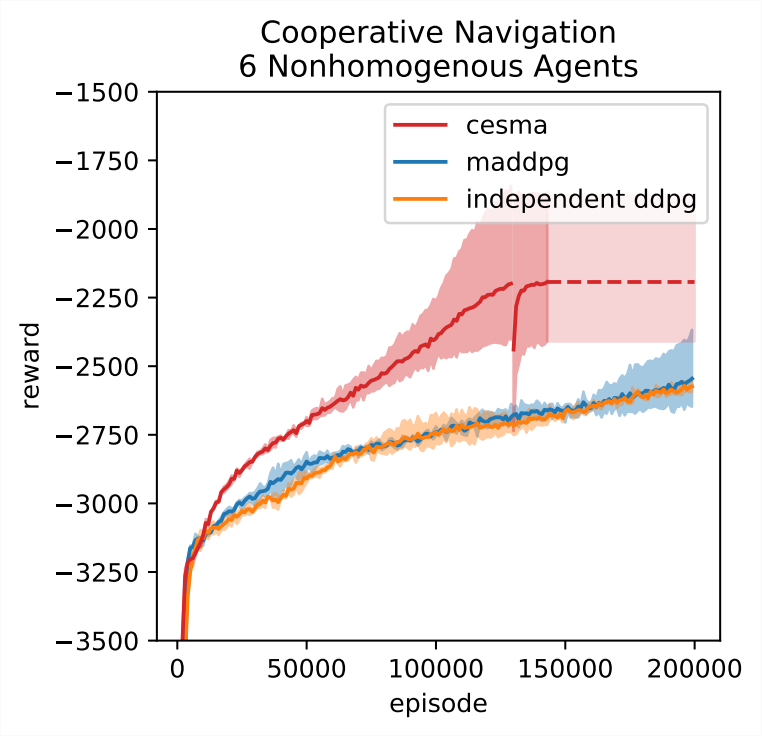


Figure 1: First part of red curve is centralized expert, second part is decentralization.

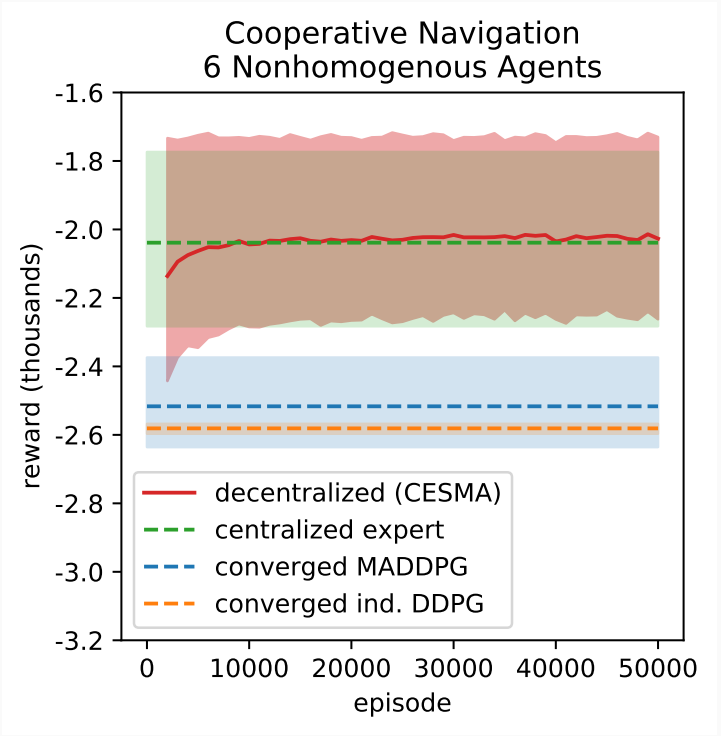
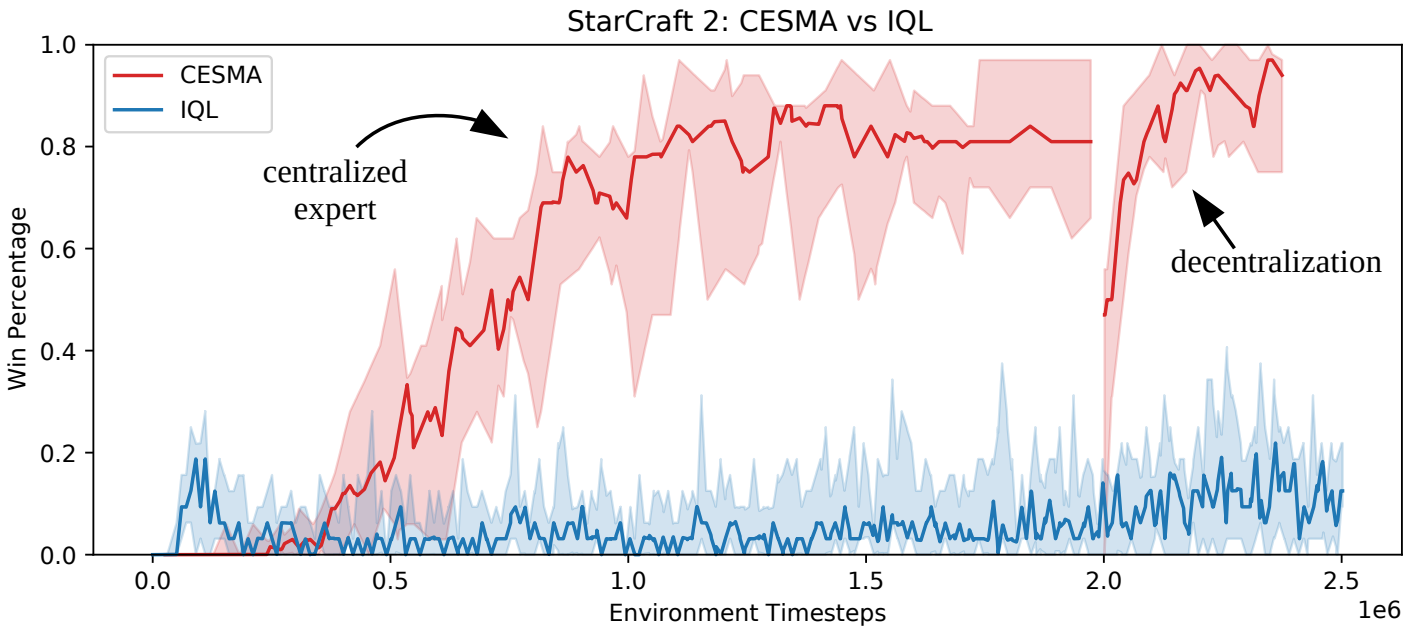


Figure 2: CESMA achieves better optimum than MADDPG and independent DDPG.

StarCraft II

- StarCraft II is a strategy computer game where players micromanage armies in order to defeat their opponent.
- Here we use StarCraft II to test the capabilities of multi-agent algorithms (Samvelyan et al., 2019).
- Each blue unit below will have its own neural network. The red units will be controlled by computer AI that was programmed by the developers, under the very hard difficulty setting (the hardest one).

StarCraft II



Learn More!

Please check out the paper **Decentralized Multi-Agents by Imitation of a Centralized Controller** to learn more!

We implement communicative features to CESMA so the agents not only learn the correct actions but also the correct communications. We also have more experiments!

Thanks for listening!