Multi-Agent Reinforcement Learning: Centralized VS Decentralized

Shusen Wang

http://wangshusen.github.io/

Architectures

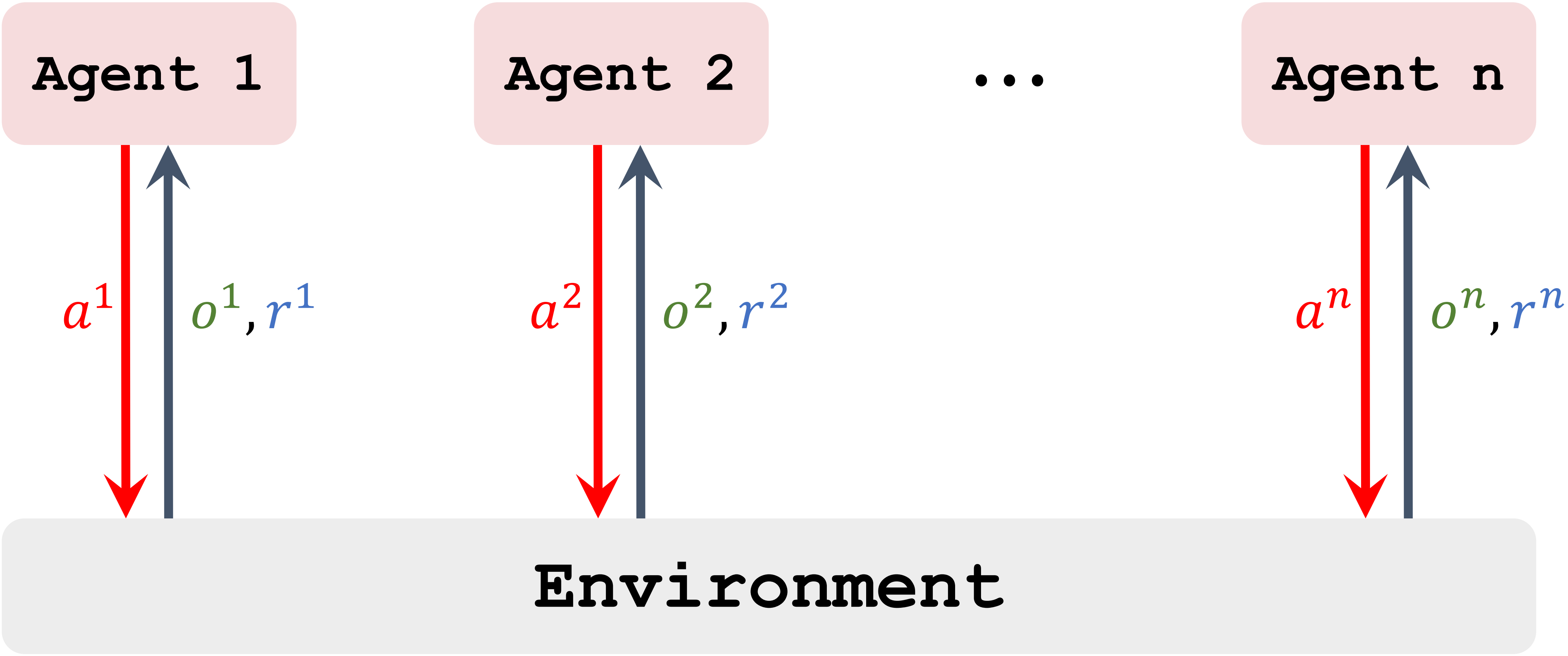
- **Fully decentralized:** Every agent uses its own observations and rewards to learn its policy. Agents do not communicate.
- **Fully centralized:** The agents send everything to the central controller. The controller makes decisions for all the agents.
- **Centralized training with decentralized execution:** A central controller is used during training. The controller is disabled after training.

Partial Observations

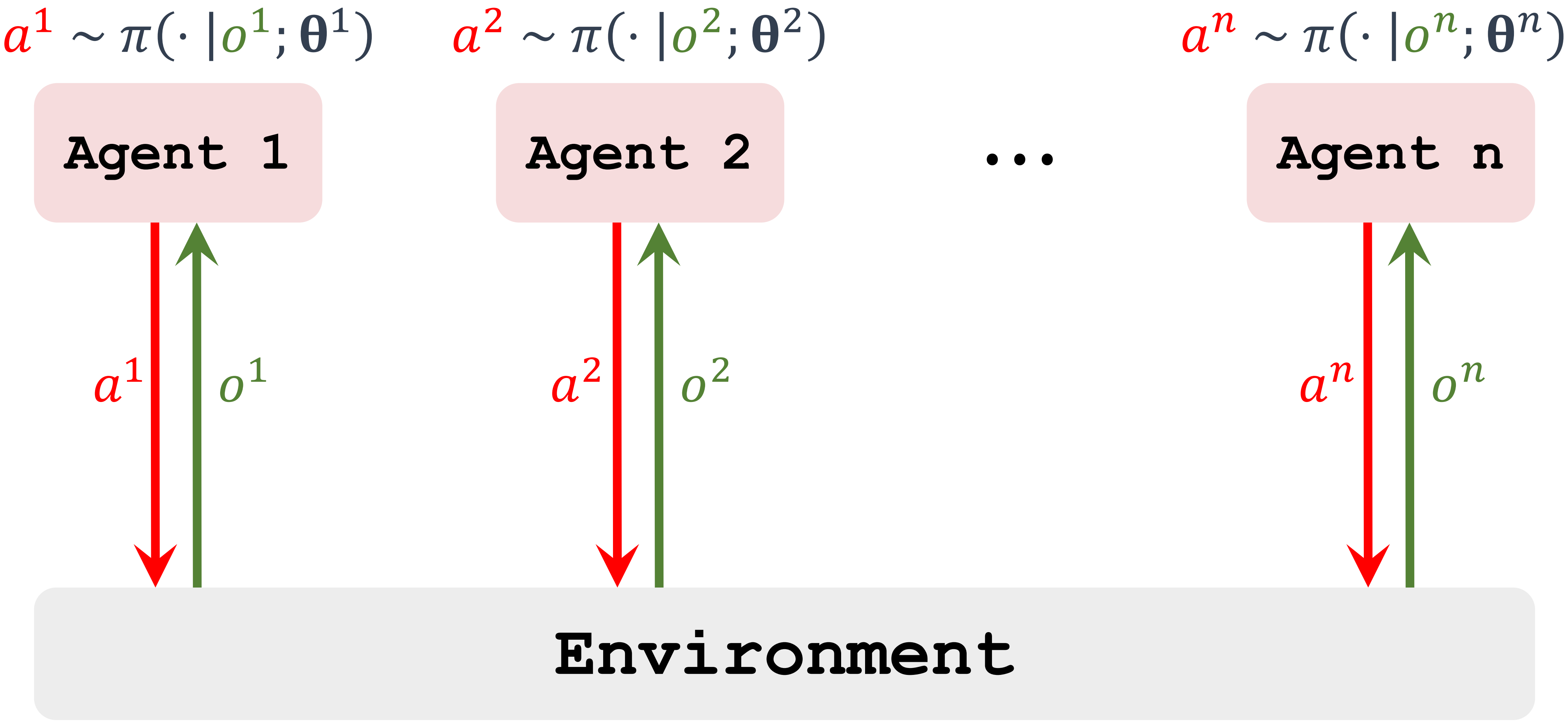
- An agent may or may not have full knowledge of the state, s.
- Let o^i be the i-th agent's observation.
- Partial observation: $o^i \neq s$.
- Full observation: $o^1 = \cdots = o^n = s$.

Fully Decentralized

Fully Decentralized Training



Fully Decentralized Execution

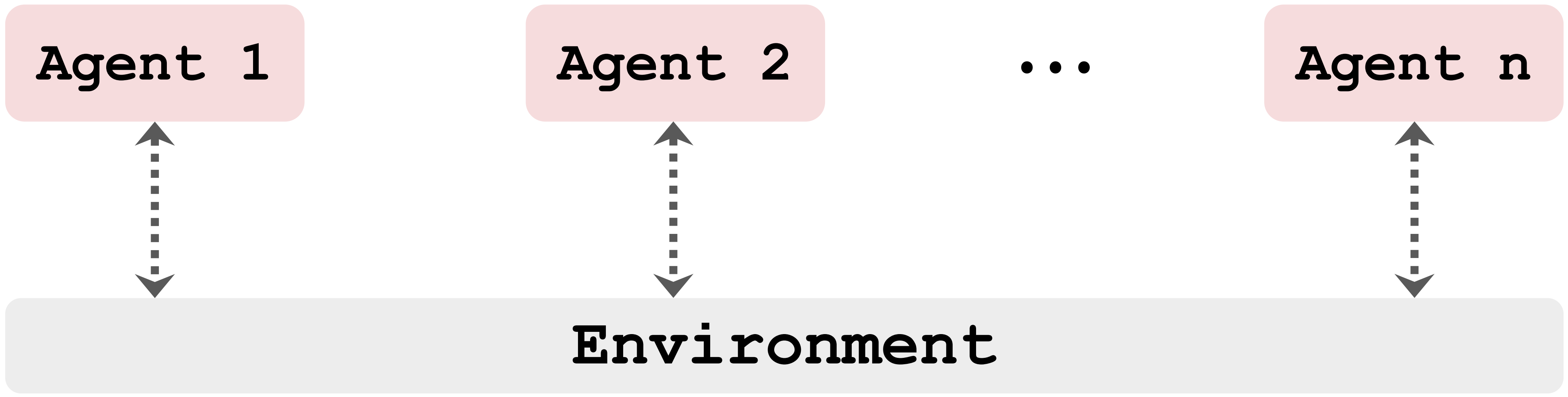


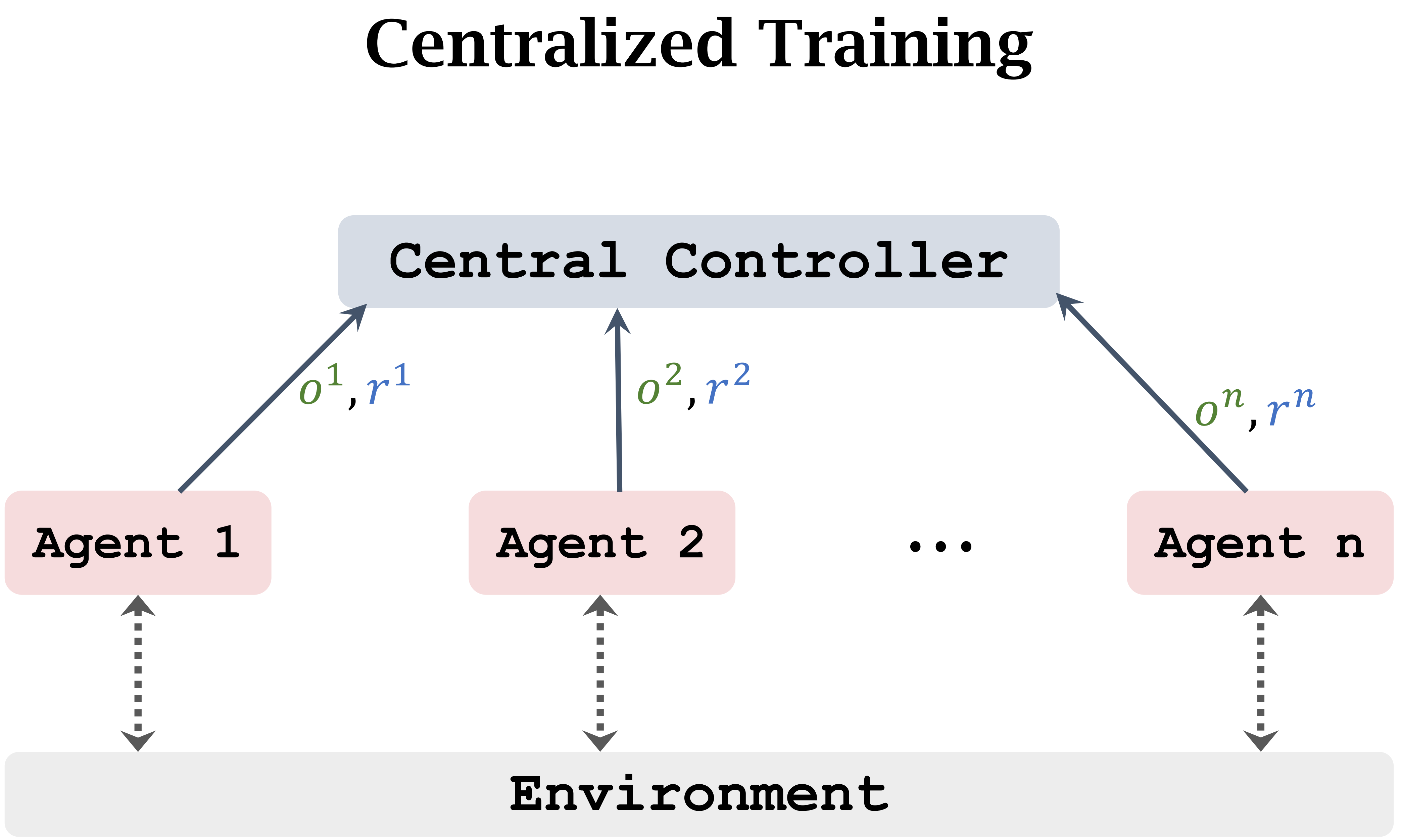
Fully Decentralized Actor-Critic Method

- The i-th agent has a policy network (actor): $\underline{\pi\left(a^i \middle| o^i; \boldsymbol{\theta}^i\right)}$.
- The i-th agent has a value network (critic): $\underline{q\left(o^i, a^i; \mathbf{w}^i\right)}$.
- Agents do not share observations and actions.
- Train the policy and value networks in the same way as the single-agent setting.
- This does not work well.

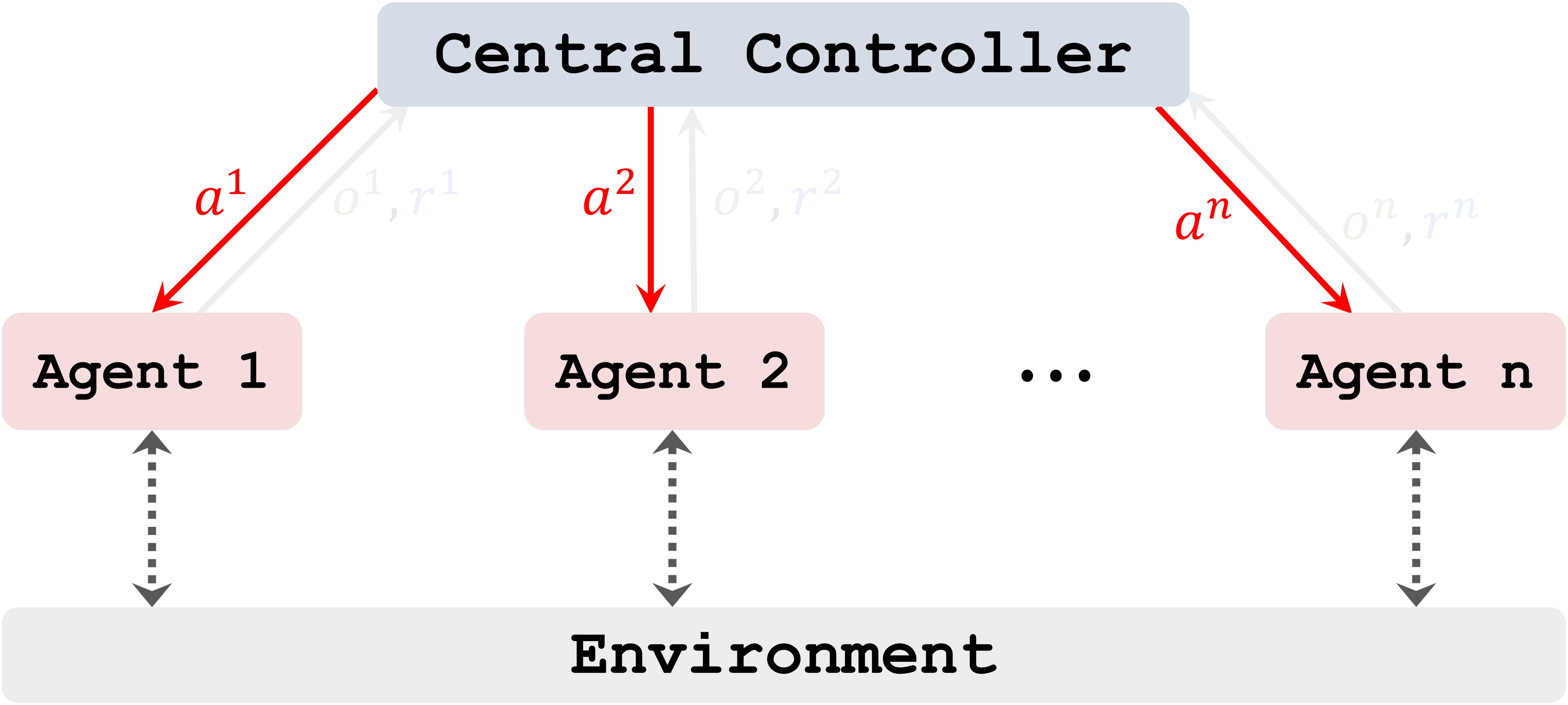
Fully Centralized

Centralized Training



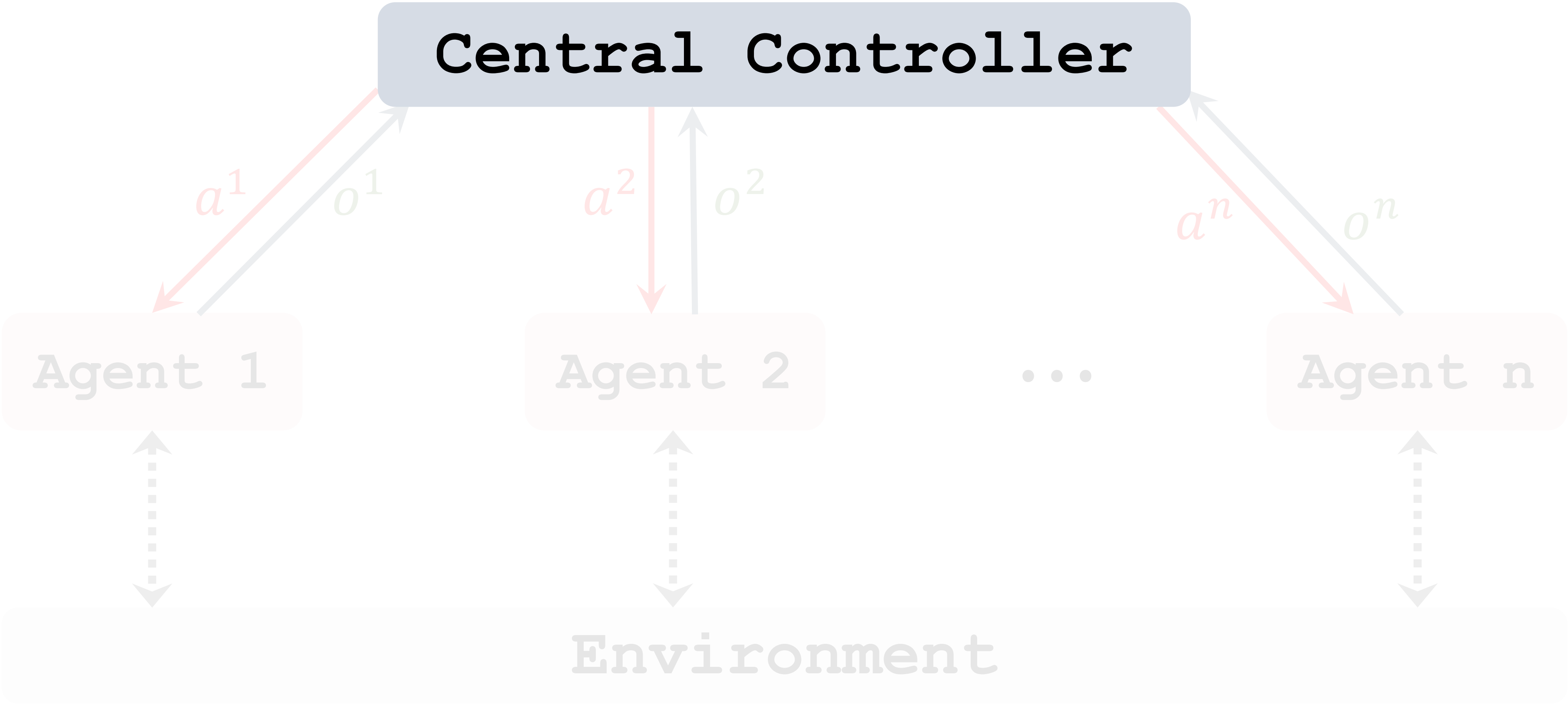

Centralized Training
Central Controller
o^1, r^1
o^2, r^2
o^n, r^n
Agent 1
Agent 2
...
Agent n
Environment

Centralized Training



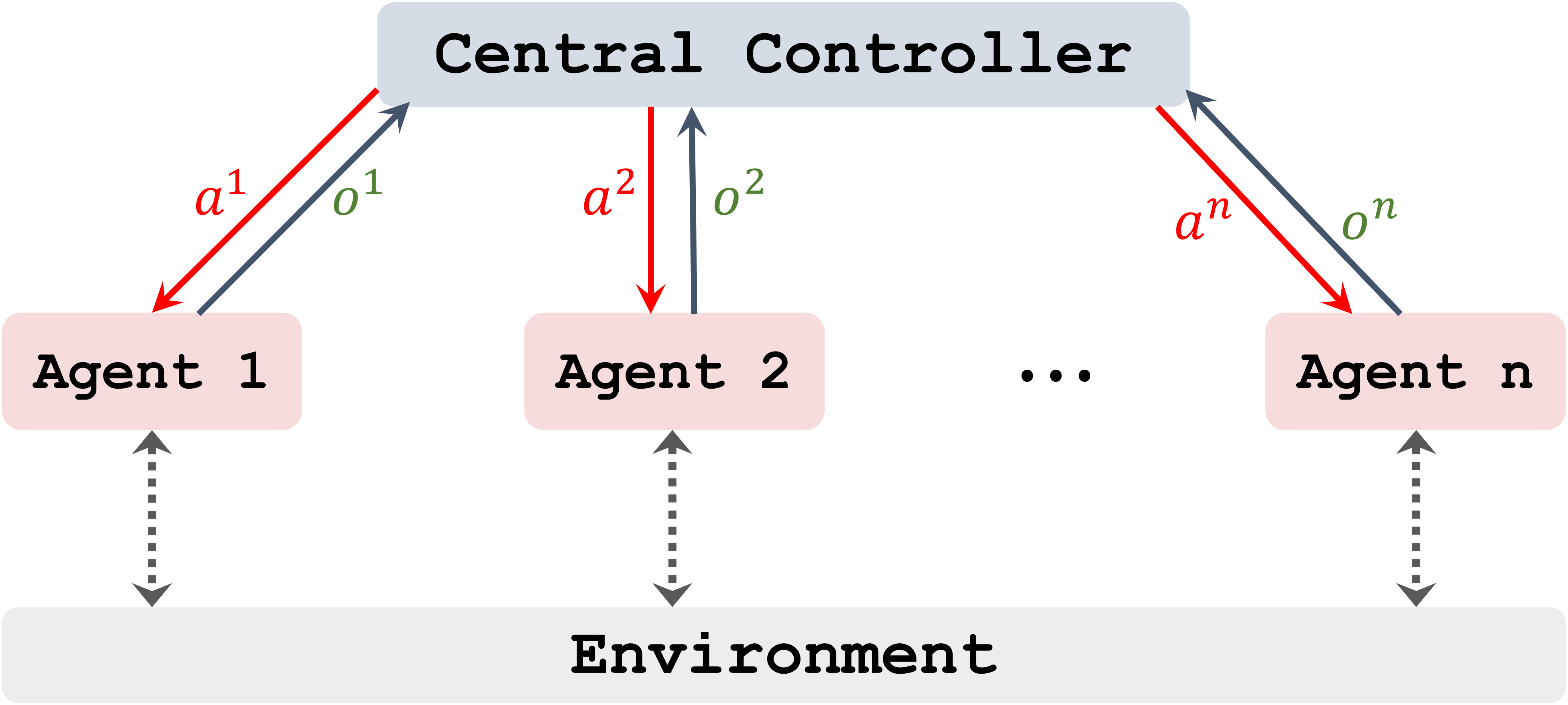
Centralized Execution

$\pi\left(a^i \middle| o^1, \cdots, o^n ; \boldsymbol{\theta}^i\right)$, for all $i = 1, 2, \cdots, n$.



Centralized Execution

$$\pi\left(a^{i} \middle| o^{1}, \cdots, o^{n}; \boldsymbol{\theta}^{i}\right), \text{ for all } i = 1, 2, \cdots, n.$$



Centralized Actor-Critic Method

- Let $\mathbf{a} = [a^1, a^2, \cdots, a^n]$ contain all the agents' actions.
- Let $\mathbf{o} = [o^1, o^2, \cdots, o^n]$ contain all the agents' observations.
- The central controller knows $\mathbf{a}$, $\mathbf{o}$, and all the rewards.
- The controller has n policy networks and n value networks:
    - Policy network (actor) for the i-th agent: $\underline{\pi\left(a^i \middle| \mathbf{o}\,; \boldsymbol{\theta}^i\right)}$.
    - Value network (critic) for the i-th agent: $\underline{q\left(\mathbf{o}, \mathbf{a}\,; \mathbf{w}^i\right)}$.

Centralized Actor-Critic Method

- Centralized Training: Training is performed by the controller.
  - The controller knows all the observations, actions, and rewards.
  - Train $\pi\left(a^i \middle| \mathbf{o}\,;\boldsymbol{\theta}^i\right)$ using policy gradient.
  - Train $q\left(\mathbf{o}, \mathbf{a}\,;\mathbf{w}^i\right)$ using TD algorithm.
- Centralized Execution: Decisions are made by the controller.
  - For all i, the i-th agent sends its observation, o^i, to the controller.
  - The controller knows $\mathbf{o} = [o^1, o^2, \cdots, o^n]$.
  - For all i, the controller samples action by $a^i \sim \pi\left(\cdot \middle| \mathbf{o}\,;\boldsymbol{\theta}^i\right)$ and sends a^i to the i-th agent.

Shortcoming: Slow during Execution

- All the agents send their observations to the central controller.
- The central controller makes decisions, $\mathbf{a} = [a^1, a^2, \cdots, a^n]$, and sends a^i to the i-th agent.
- Communication and synchronization cost time.
- Real-time decision is impossible.

Centralized Training with Decentralized Execution

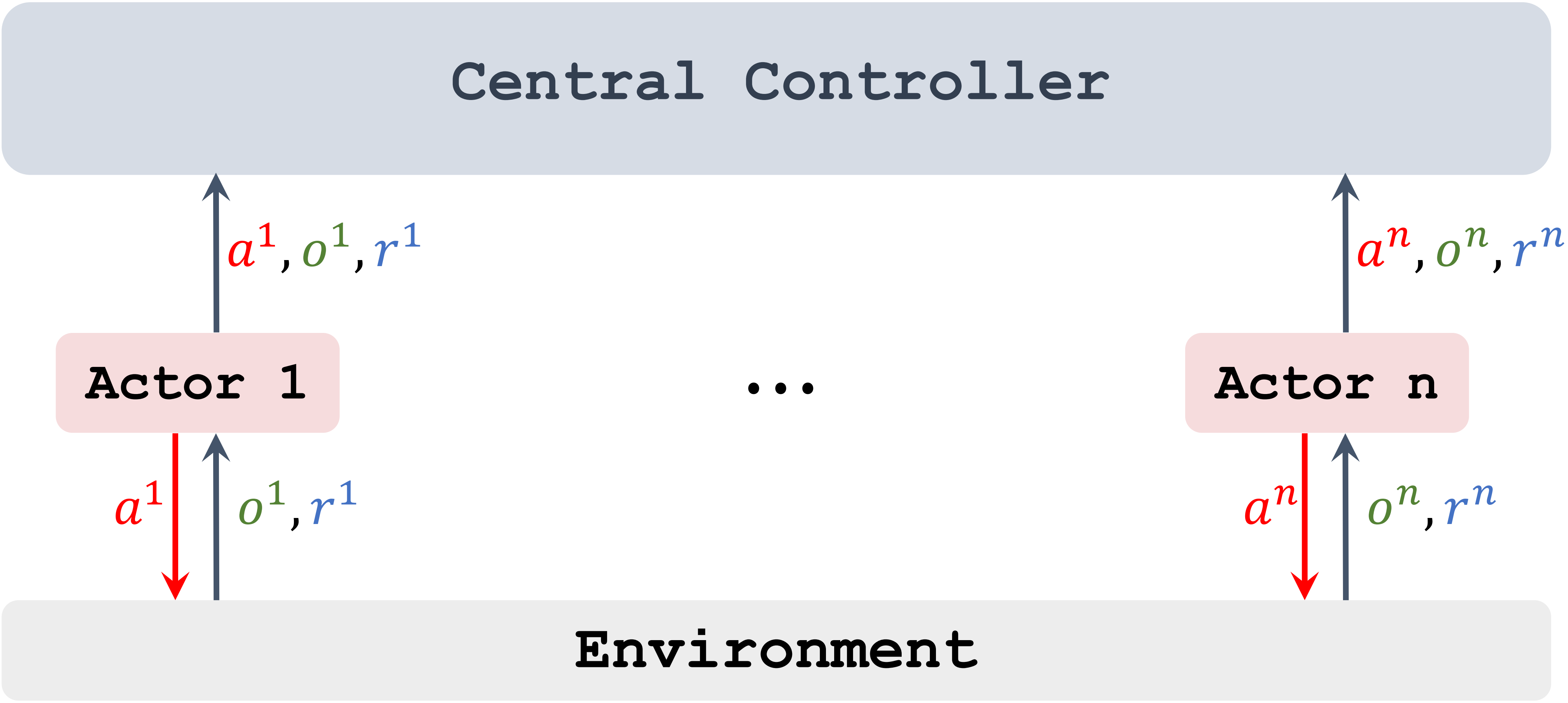
Centralized Training with Decentralized Execution

- Each agent has its own policy network (actor): $\pi(a^i | o^i \,; \boldsymbol{\theta}^i)$.
- The central controller has n value networks (critics): $q(\mathbf{o}, \mathbf{a} \,; \mathbf{w}^i)$.
- **Centralized Training:** During training, the central controller knows all the agents' observations, actions, and rewards.
- **Decentralized Execution:** During execution, the central controller and its value networks are not used.

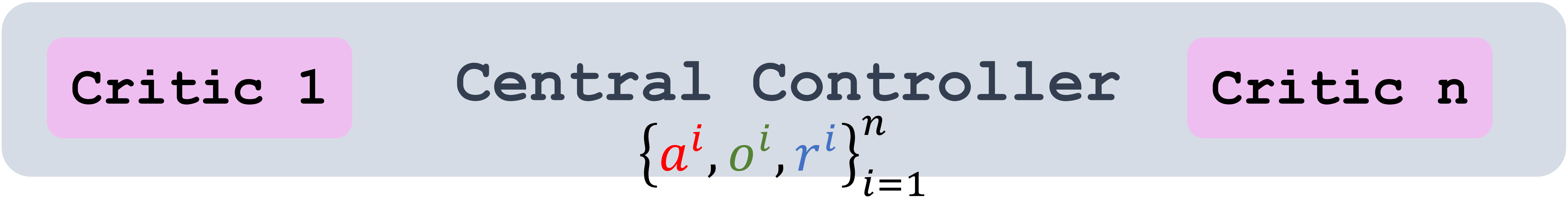
Reference:

1. Lowe et al. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NIPS*, 2017.
2. Foerster et al. Counterfactual multi-agent policy gradients. In *AAAI*, 2018.

Centralized Training



Centralized Training



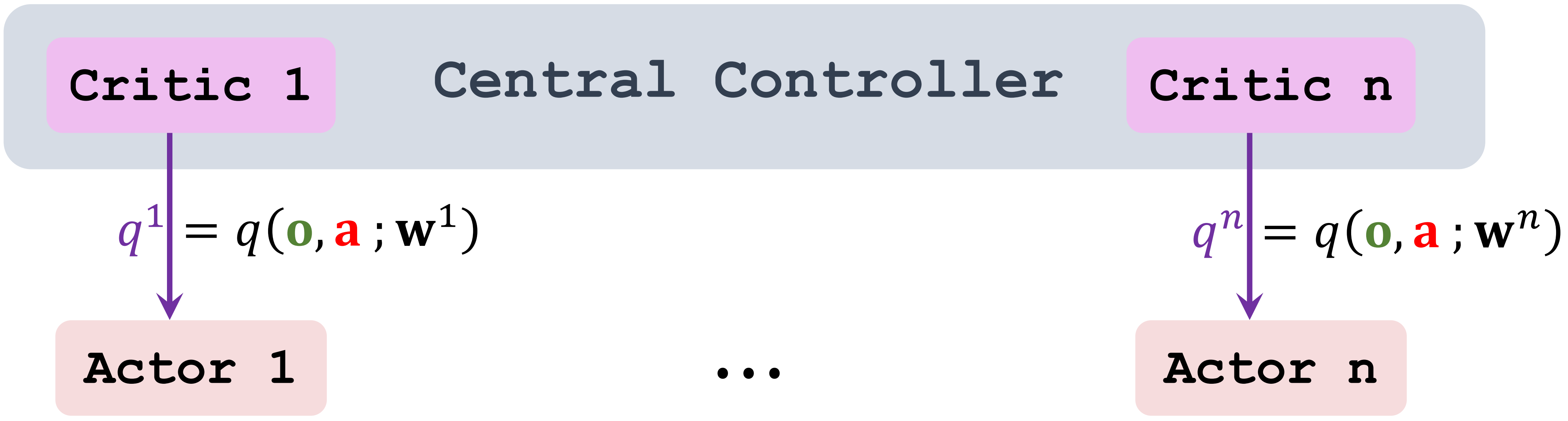
Centralized Training

Critic 1 | Central Controller | Critic n

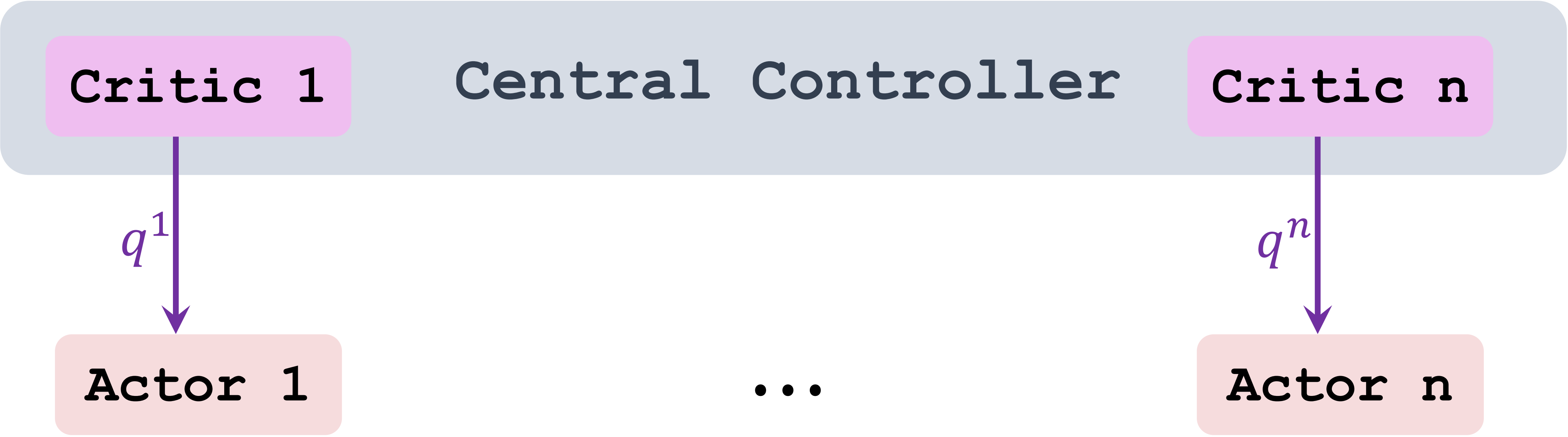
$$\{a^i, o^i, r^i\}_{i=1}^n$$

- The central controller trains the critics, $\underline{q(\mathbf{o}, \mathbf{a}\,; \mathbf{w}^i)}$, for all i.
- To update $\mathbf{w}^i$, TD algorithm takes as inputs:
  - All the actions: $\mathbf{a} = [a^1, a^2, \cdots, a^n]$.
  - All the observations: $\mathbf{o} = [o^1, o^2, \cdots, o^n]$.
  - The i-th reward: r^i.

Centralized Training



Centralized Training



- Each agent locally trains the actor, $\pi(a^i | o^i ; \boldsymbol{\theta}^i)$, using policy gradient.
- To update $\boldsymbol{\theta}^i$, the policy gradient algorithm takes as input (a^i, o^i, q^i).

Decentralized Execution

Parameter Sharing

Parameter Sharing?

- Policy networks: $\pi\left(a^i \middle| o^i ; \boldsymbol{\theta}^i\right)$, for $i = 1, 2, \cdots, n$.
- Value networks: $q\left(\mathbf{o}, \mathbf{a} ; \mathbf{w}^i\right)$, for $i = 1, 2, \cdots, n$.
- Trainable parameters: $\left\{\boldsymbol{\theta}^i, \mathbf{w}^i\right\}_{i=1}^n$
- Parameter sharing: $\boldsymbol{\theta}^i = \boldsymbol{\theta}^j$ and $\mathbf{w}^i = \mathbf{w}^j$, for some i and j.

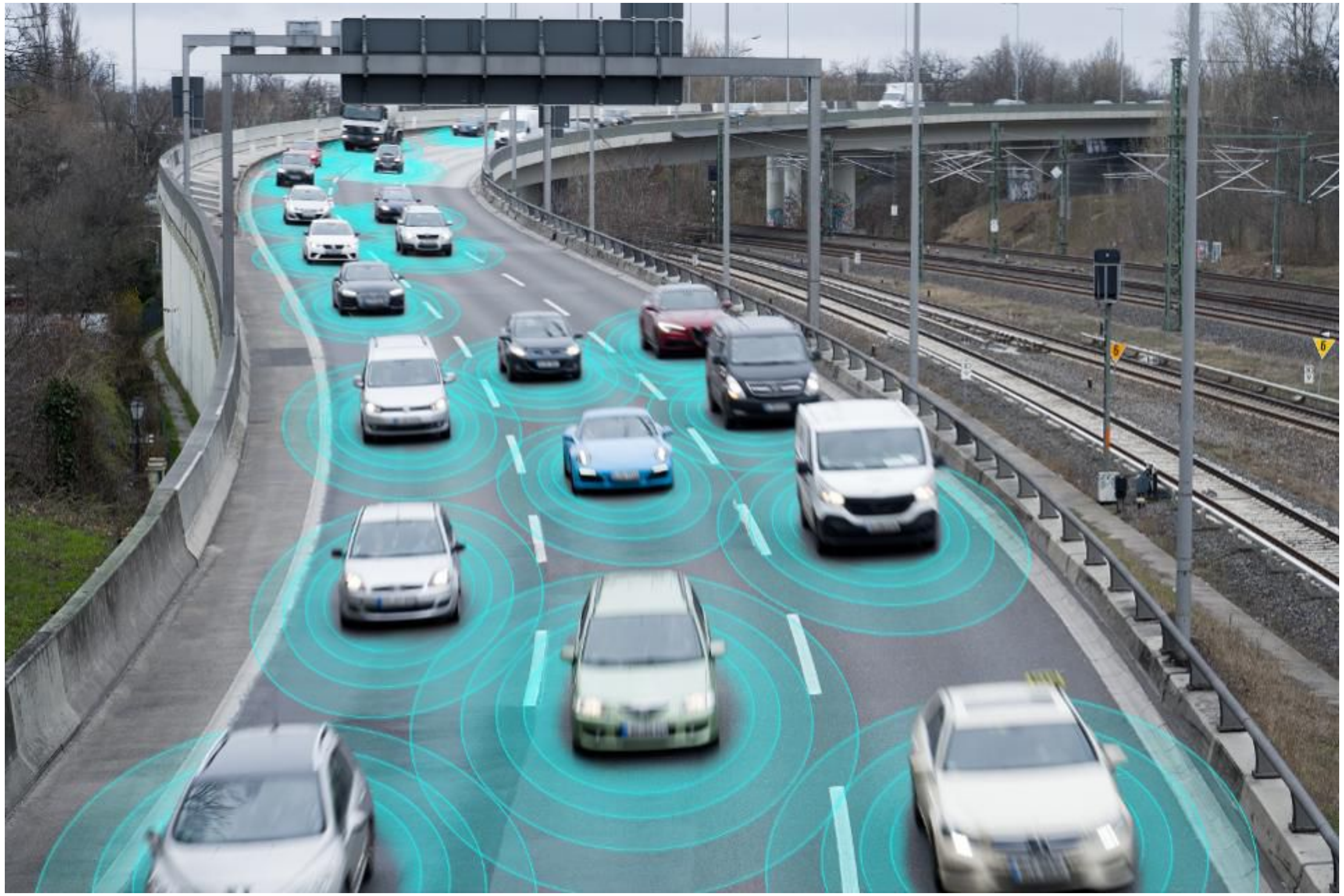
Question: Shall the networks share parameters?

Parameter Sharing?

Do not share parameters if the agents are non-exchangeable.

Parameter Sharing?

Share parameters if the agents are exchangeable.



Summary

Fully Decentralized

- The agents are independent.
- One agent is unaware of the other agents’ observations and actions.
- Train every agent in the same way as single-agent RL.
- This does not work well.

Fully Centralized

- All the policy and value networks are in the central controller.
- Agents send everything to the controller.
- The controller makes decisions based on all the agents’ observations. Agents do not make decisions.
- The controller tells every agent what to do.

Centralized Training, Decentralized Execution

- Each agent has its own policy network.
- The central controller has all the value networks.
- The central controller helps with the training; it is disabled during execution.

	Policy (Actor)	Value (Critic)
Fully Decentralized	$\pi\left(a^i \middle| o^i; \boldsymbol{\theta}^i\right)$	$q\left(o^i, a^i; \mathbf{w}^i\right)$

	Policy (Actor)	Value (Critic)
Fully Decentralized	$\pi\left(a^i \middle\vert o^i; \boldsymbol{\theta}^i\right)$	$q\left(o^i, a^i; \mathbf{w}^i\right)$
Fully Centralized	$\pi\left(a^i \middle\vert \mathbf{o}; \boldsymbol{\theta}^i\right)$	$q\left(\mathbf{o}, \mathbf{a}; \mathbf{w}^i\right)$

	Policy (Actor)	Value (Critic)
Fully Decentralized	$\pi\left(a^i \middle| o^i; \boldsymbol{\theta}^i\right)$	$q\left(o^i, a^i; \mathbf{w}^i\right)$
Fully Centralized	$\pi\left(a^i \middle| \mathbf{o}; \boldsymbol{\theta}^i\right)$	$q\left(\mathbf{o}, \mathbf{a}; \mathbf{w}^i\right)$
Centralized Training, Decentralized Execution	$\pi\left(a^i \middle| o^i; \boldsymbol{\theta}^i\right)$	$q\left(\mathbf{o}, \mathbf{a}; \mathbf{w}^i\right)$

Thank you!

Recommended Survey Papers

1. Zhang, Yang, & Başar. Multi-agent reinforcement learning: a selective overview of theories and algorithms. *arXiv*, 2019.
2. François-Lavet et al. An Introduction to Deep Reinforcement Learning. *Foundations and Trends in Machine Learning*, 2018.
3. Hernandez-Leal et al. A survey of learning in multiagent environments: dealing with non-stationarity. *arXiv*, 2017.
4. Nguyen, Nguyen, & Nahavandi. Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications. *IEEE Transactions on Cybernetics,* 2020.
5. Li. Deep reinforcement learning. *arXiv*, 2018.