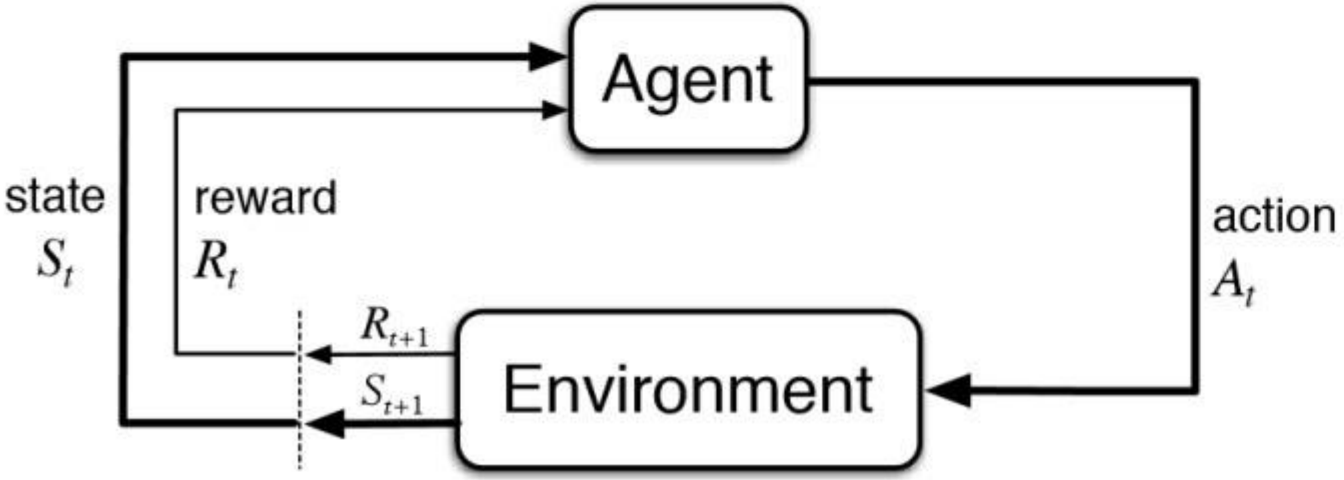
Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning

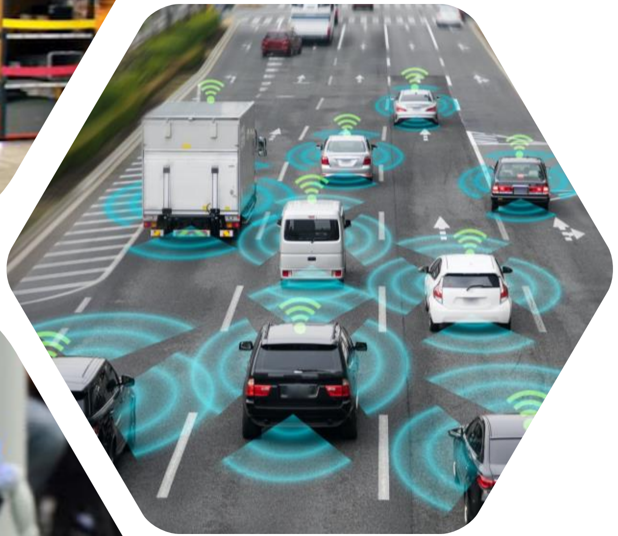
Filippos Christianos, Lukas Schäfer, Stefano V. Albrecht

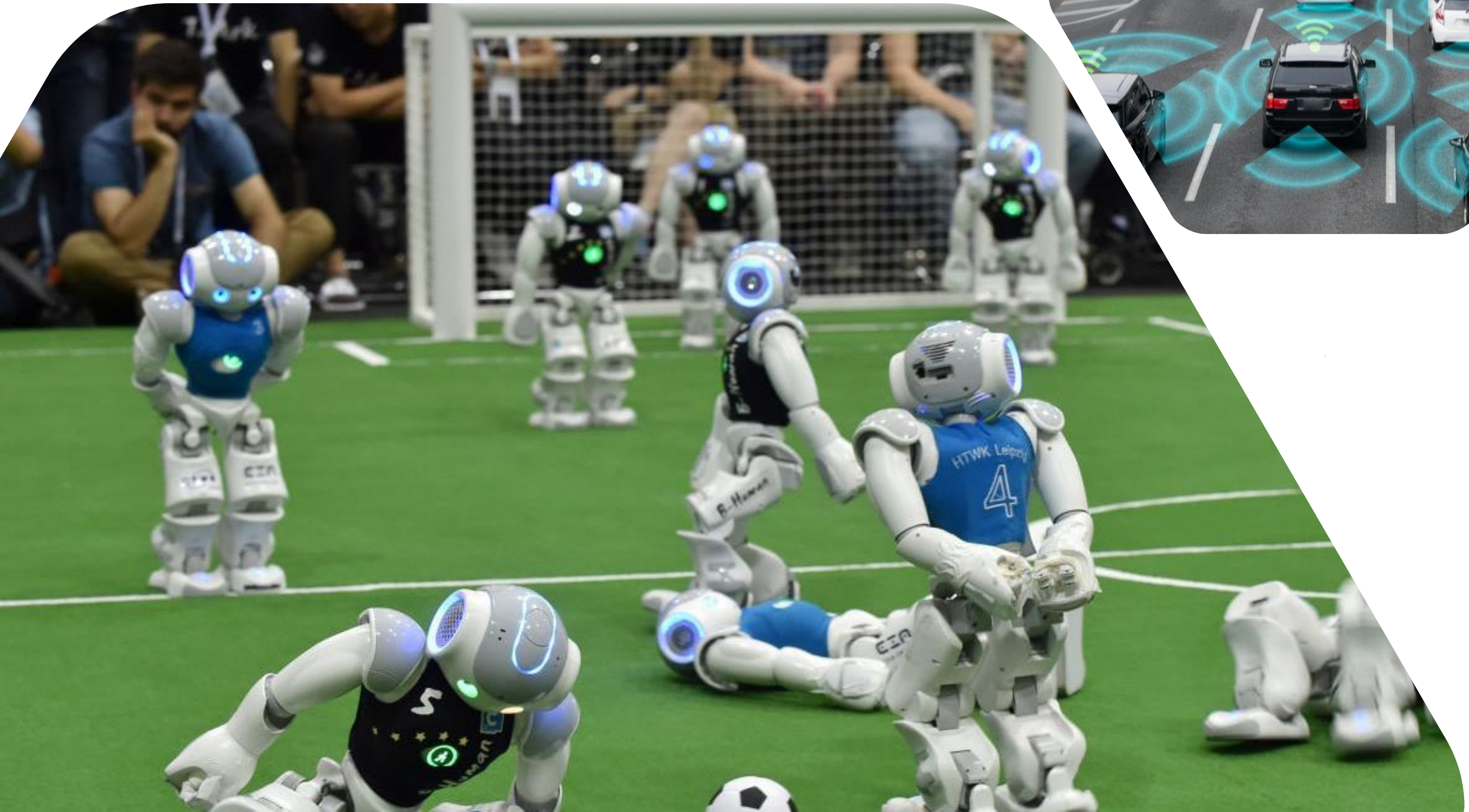


In 34th Conference on Neural Information Processing Systems 2020.

Reinforcement Learning

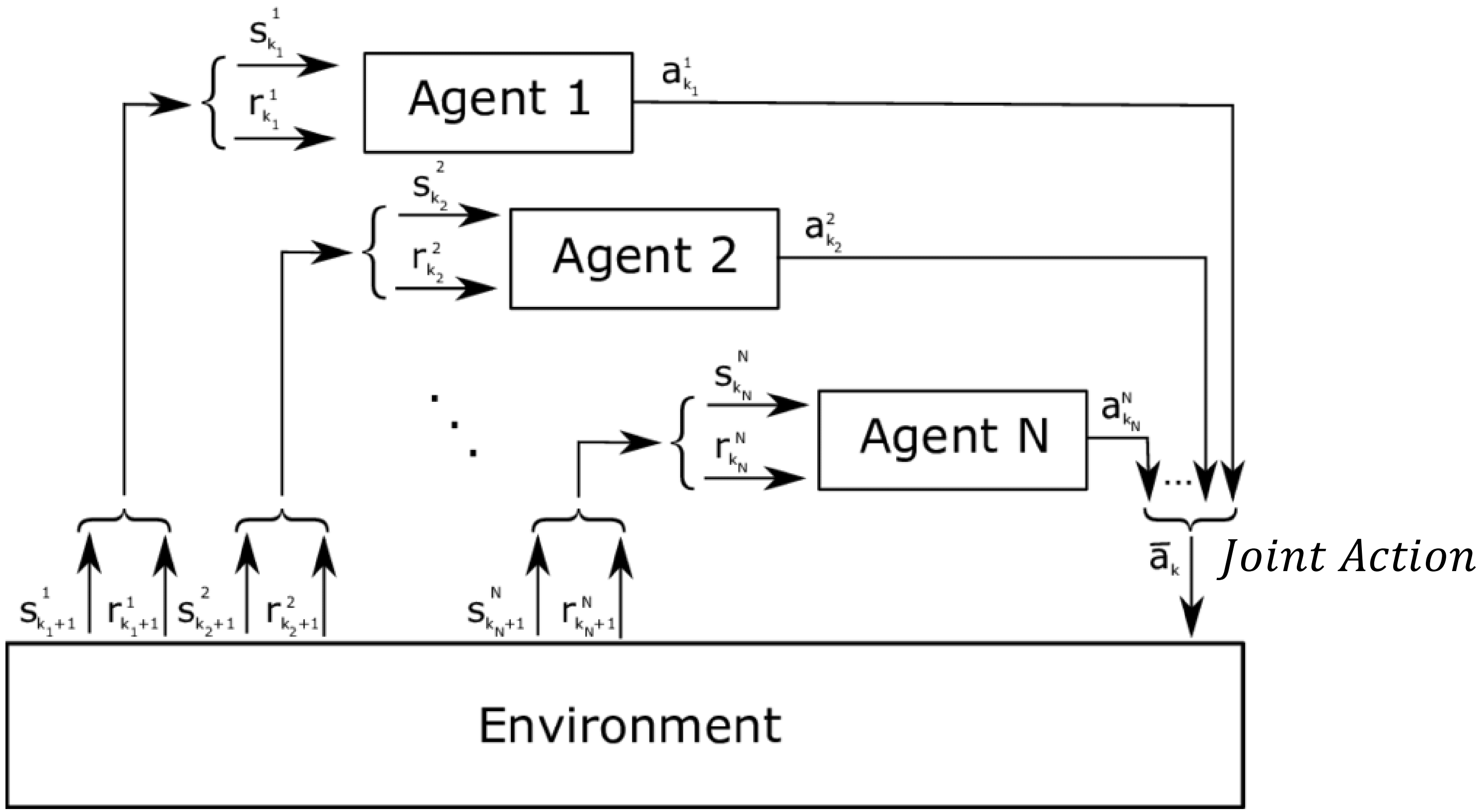




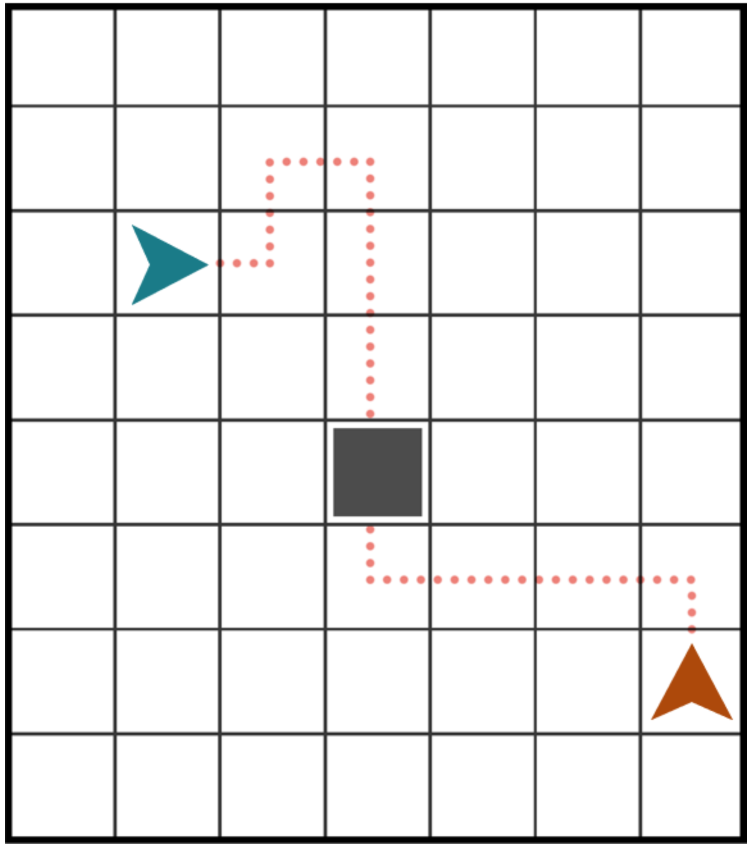


Multi-Agent Systems

Multi-Agent Reinforcement Learning

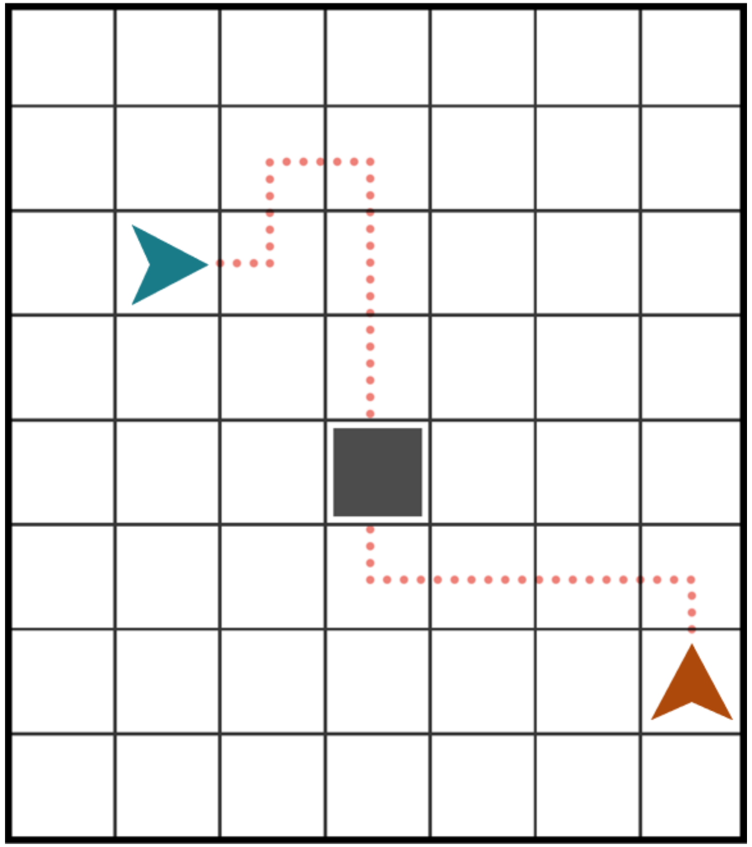


Motivational Example



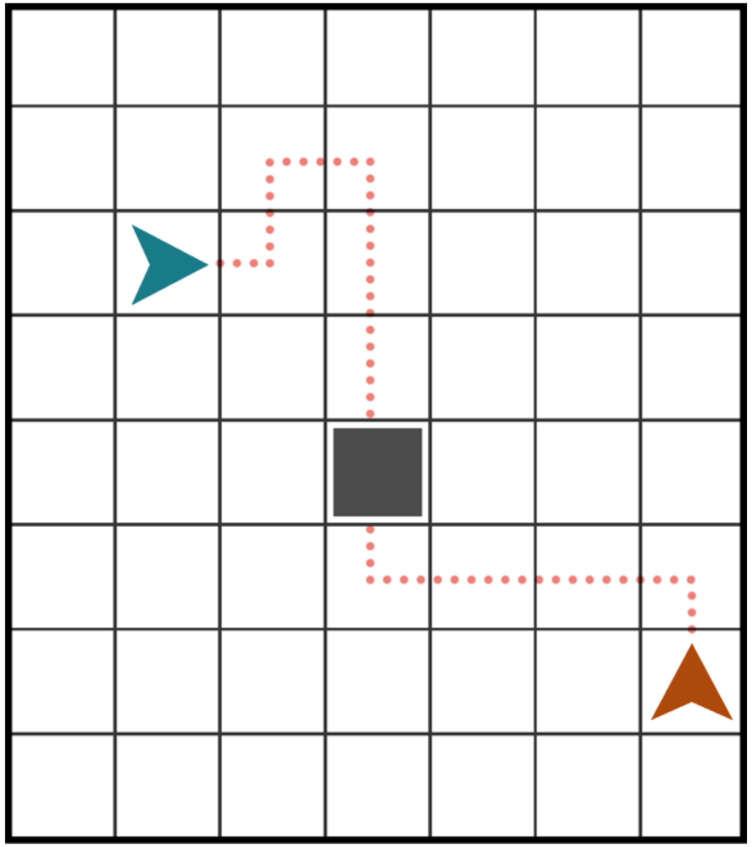
Motivational Example

- Both agents must reach goal simultaneously
  - ➤ Sparse reward signal

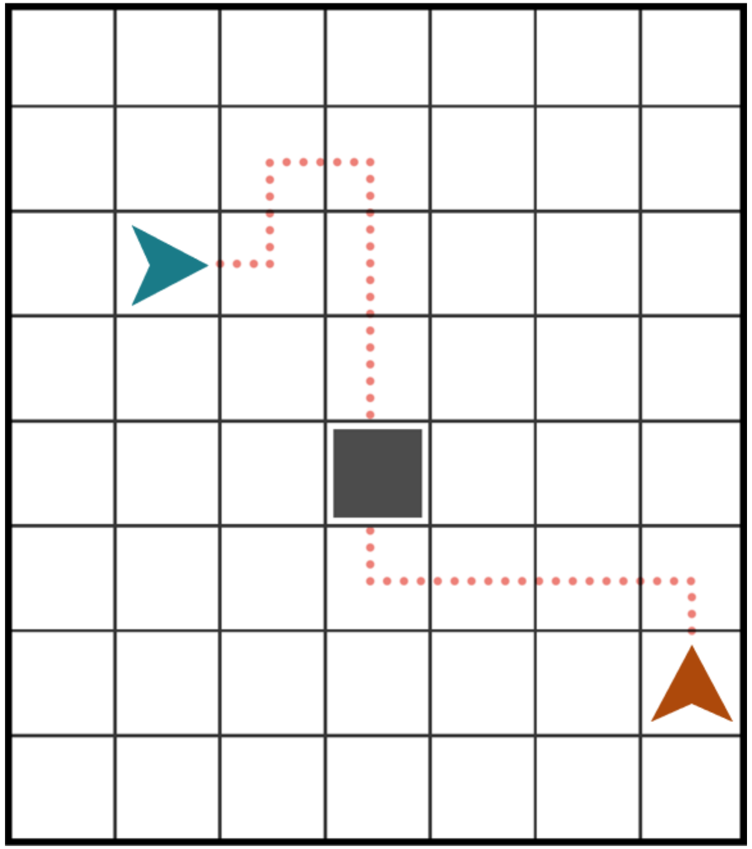


Motivational Example

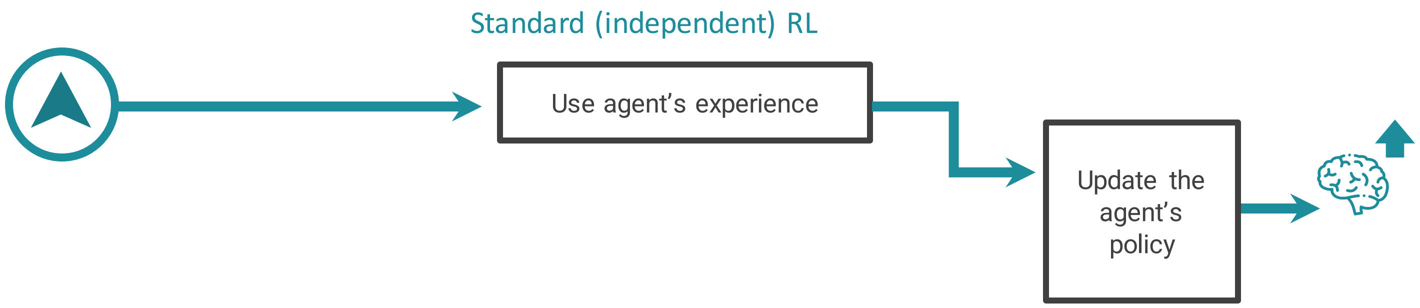
- Both agents must reach goal simultaneously
  - ➤ Sparse reward signal
- **Idea**: Make us of both agents’ exploration
  - ➤ Share experience of agents

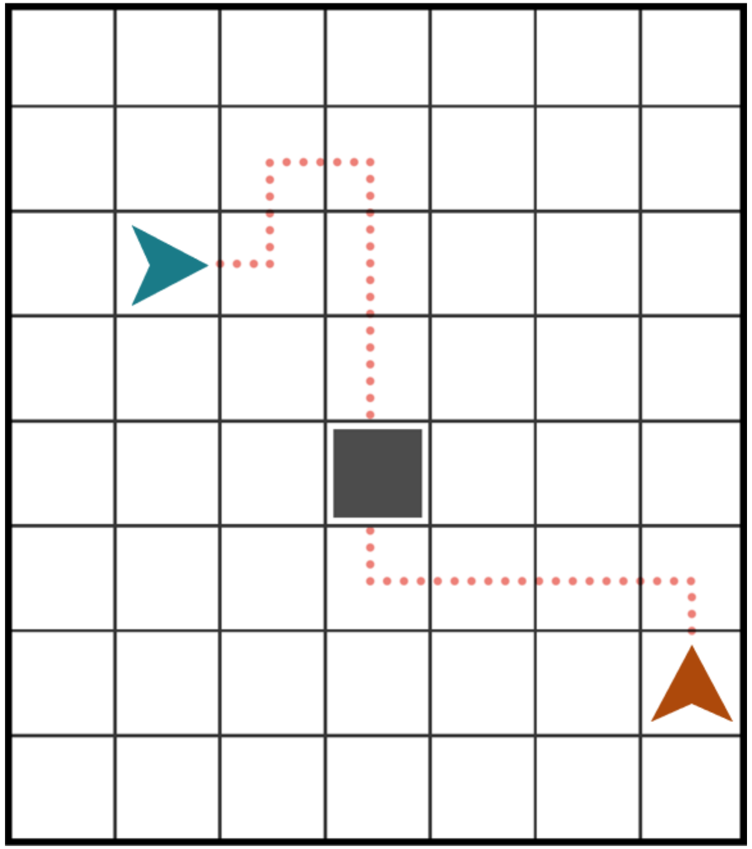


SHARED EXPERIENCE ACTOR-CRITIC

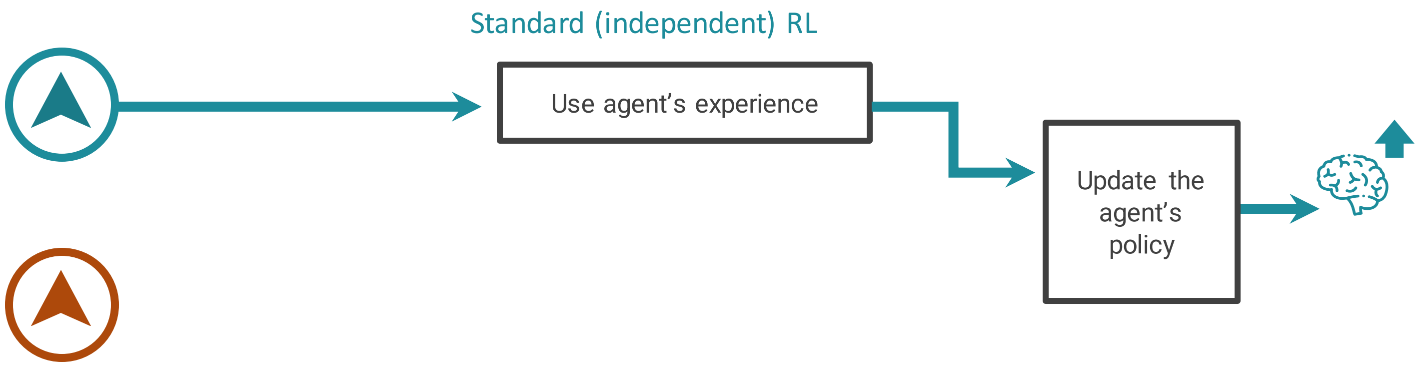


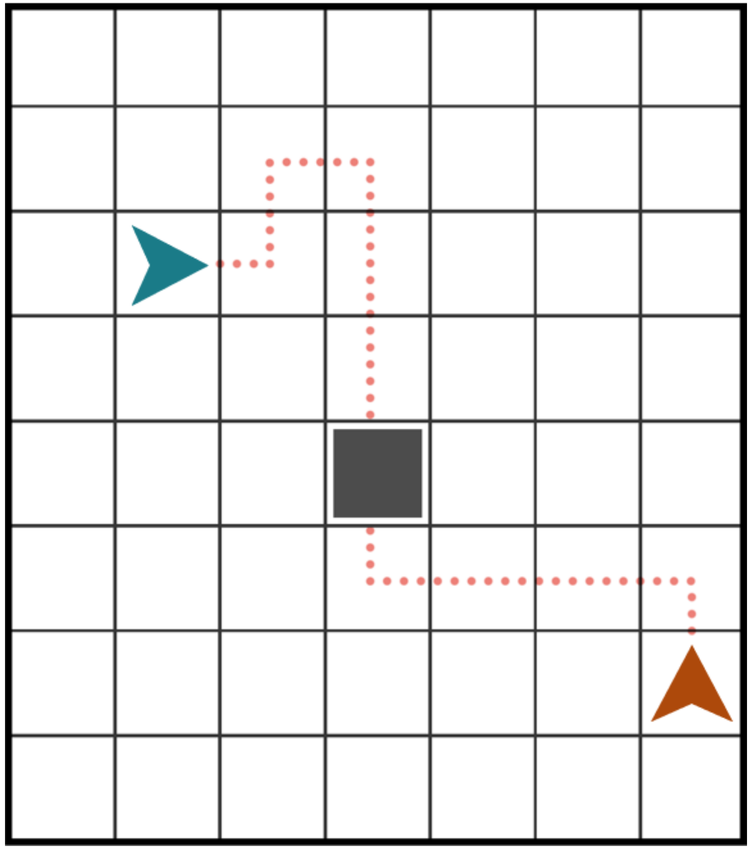
SHARED EXPERIENCE ACTOR-CRITIC



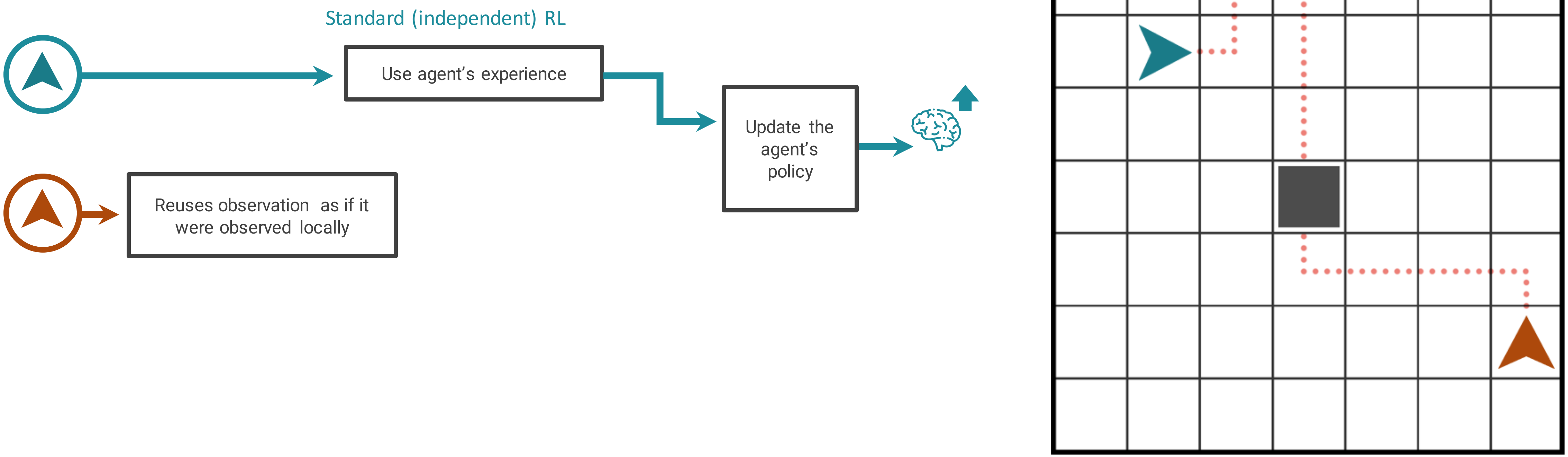


SHARED EXPERIENCE ACTOR-CRITIC

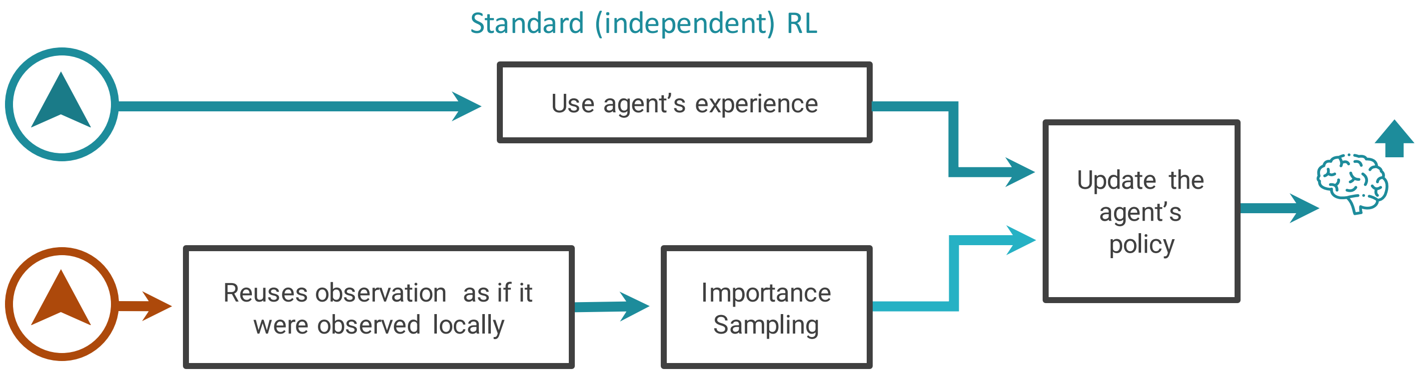


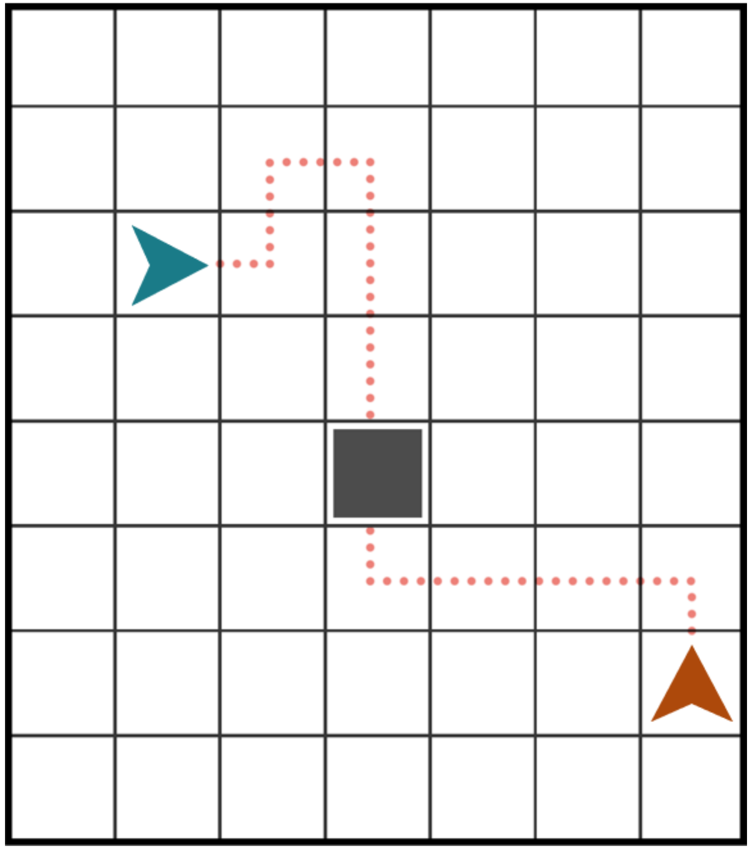


SHARED EXPERIENCE ACTOR-CRITIC

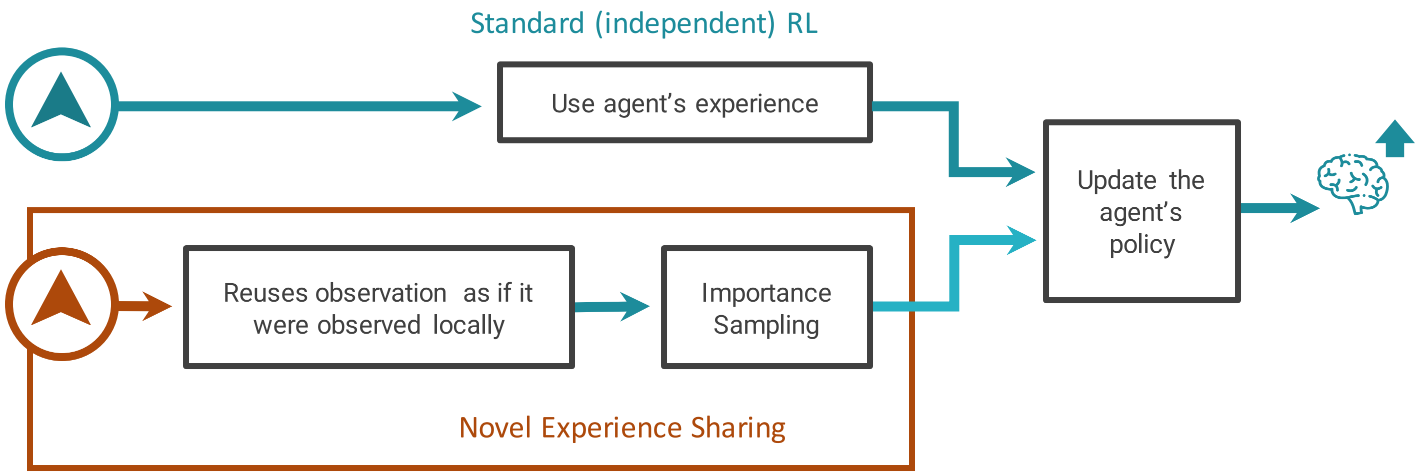


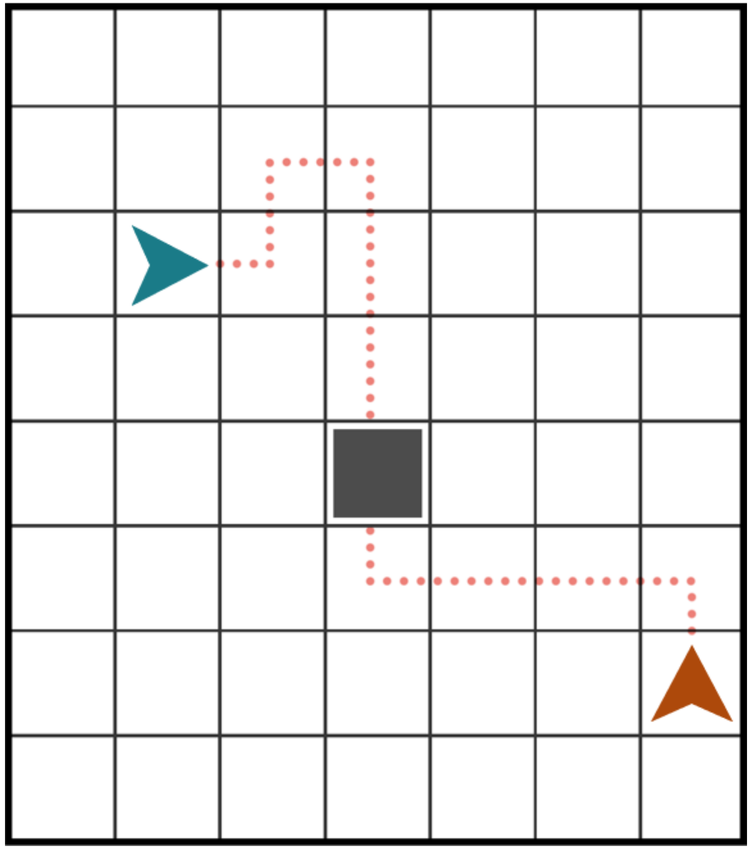
SHARED EXPERIENCE ACTOR-CRITIC



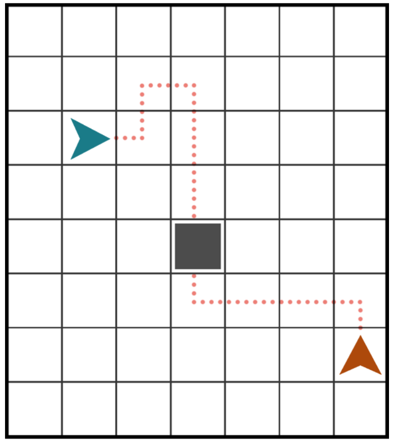


SHARED EXPERIENCE ACTOR-CRITIC



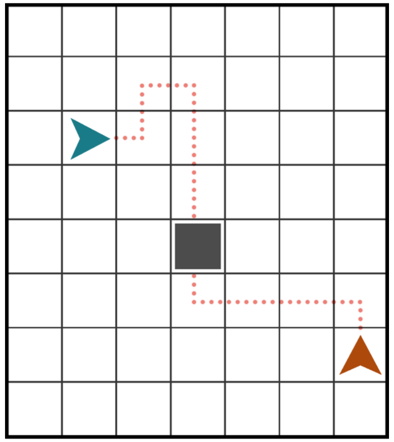


SHARED EXPERIENCE ACTOR-CRITIC



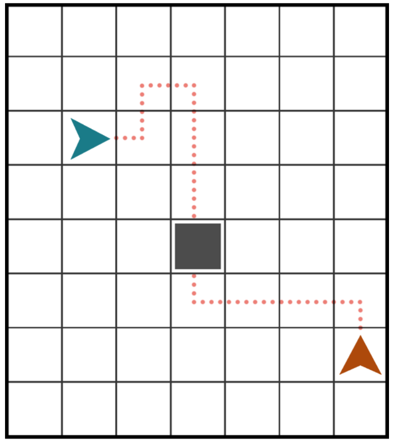
$$\mathcal{L}(\phi_i) = \boxed{-\log \pi\left(a_t^i \middle| o_t^i; \phi_i\right) \left(r_t^i + \gamma V\left(o_{t+1}^i; \theta_i\right) - V\left(o_t^i; \theta_i\right)\right)}$$

SHARED EXPERIENCE ACTOR-CRITIC



$$\mathcal{L}(\phi_i) = \boxed{-\log \pi\left(a_t^i \middle| o_t^i; \phi_i\right) \left(\underbrace{r_t^i + \gamma V\left(o_{t+1}^i; \theta_i\right)}_{\text{Estimate of Returns}} - V\left(o_t^i; \theta_i\right)\right)}$$

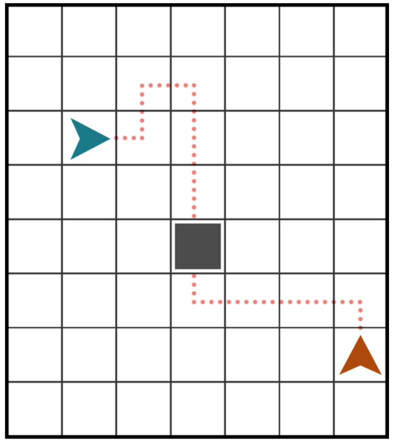
SHARED EXPERIENCE ACTOR-CRITIC



$$\mathcal{L}(\phi_i) = \boxed{-\log \pi\left(a_t^i \middle| o_t^i; \phi_i\right) \left(r_t^i + \gamma V\left(o_{t+1}^i; \theta_i\right) - V\left(o_t^i; \theta_i\right)\right)}$$

$$\underbrace{V\left(o_t^i; \theta_i\right)}_{\text{Baseline}}$$

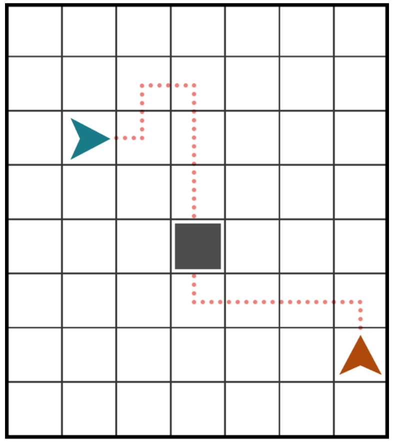
SHARED EXPERIENCE ACTOR-CRITIC



$$\mathcal{L}(\phi_i) = \boxed{-\log \pi\left(a_t^i \middle| o_t^i; \phi_i\right) \underbrace{\left(r_t^i + \gamma V\left(o_{t+1}^i; \theta_i\right) - V\left(o_t^i; \theta_i\right)\right)}_{\text{Advantage: high/low}}}$$

Action

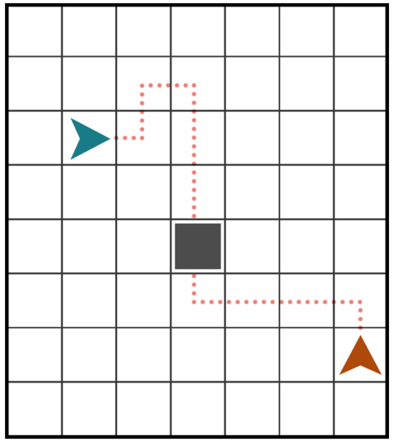
SHARED EXPERIENCE ACTOR-CRITIC



$$\mathcal{L}(\phi_i) = -\log \pi\left(a_t^i \middle| o_t^i; \phi_i\right) \left(r_t^i + \gamma V\left(o_{t+1}^i; \theta_i\right) - V\left(o_t^i; \theta_i\right)\right)$$

$$-\lambda \sum_{k \neq i} \log \pi\left(a_t^k \middle| o_t^k; \phi_i\right) \left(r_t^k + \gamma V\left(o_{t+1}^k; \theta_i\right) - V\left(o_t^k; \theta_i\right)\right)$$

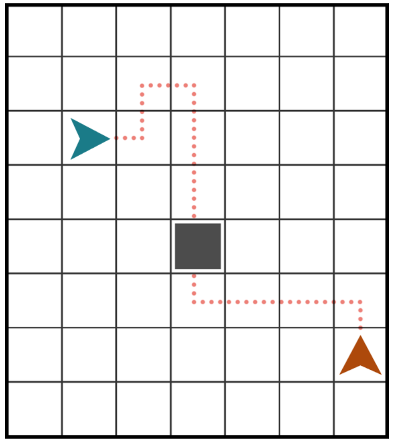
SHARED EXPERIENCE ACTOR-CRITIC



$$\mathcal{L}(\phi_i) = \boxed{-\log \pi\left(a_t^i \middle| o_t^i; \phi_i\right) \left(r_t^i + \gamma V\left(o_{t+1}^i; \theta_i\right) - V\left(o_t^i; \theta_i\right)\right)}$$

$$\boxed{-\lambda \sum_{k \neq i} \boxed{\frac{\pi(a_t^k | o_t^k; \phi_i)}{\pi(a_t^k | o_t^k; \phi_k)}} \log \pi\left(a_t^k \middle| o_t^k; \phi_i\right) \left(r_t^k + \gamma V\left(o_{t+1}^k; \theta_i\right) - V\left(o_t^k; \theta_i\right)\right)}$$

SHARED EXPERIENCE ACTOR-CRITIC

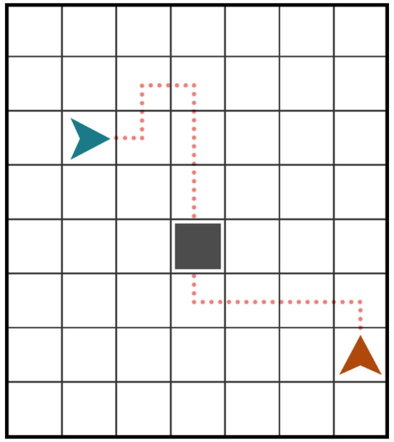


Policy Gradient Actor Loss:

$$\mathcal{L}(\phi_i) = \boxed{-\log \pi\left(a_t^i \middle| o_t^i; \phi_i\right) \left(r_t^i + \gamma V\left(o_{t+1}^i; \theta_i\right) - V\left(o_t^i; \theta_i\right)\right)}$$

$$\boxed{-\lambda \sum_{k \neq i} \frac{\pi(a_t^k | o_t^k; \phi_i)}{\pi(a_t^k | o_t^k; \phi_k)} \log \pi\left(a_t^k \middle| o_t^k; \phi_i\right) \left(r_t^k + \gamma V\left(o_{t+1}^k; \theta_i\right) - V\left(o_t^k; \theta_i\right)\right)}$$

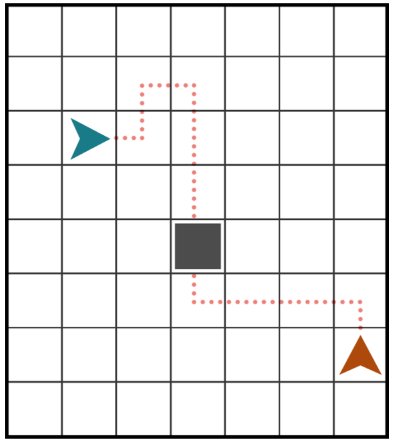
SHARED EXPERIENCE ACTOR-CRITIC



$$\mathcal{L}(\theta_i) = \boxed{\left|\left|V\left(o_t^i;\theta_i\right) - y_i^i\right|\right|^2}$$

$$y_k^i = r_t^k + \gamma V\left(o_{t+1}^k;\theta_i\right)$$

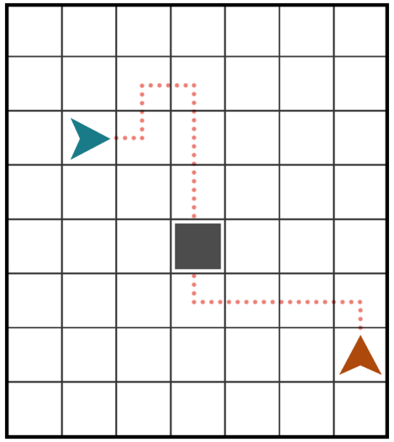
SHARED EXPERIENCE ACTOR-CRITIC



$$\mathcal{L}(\theta_i) = \left|\left|V(o_t^i;\theta_i) - y_i^i\right|\right|^2 + \lambda \sum_{k \neq i} \left|\left|V(o_t^k;\theta_i) - y_k^i\right|\right|^2$$

$$y_k^i = r_t^k + \gamma V(o_{t+1}^k;\theta_i)$$

SHARED EXPERIENCE ACTOR-CRITIC

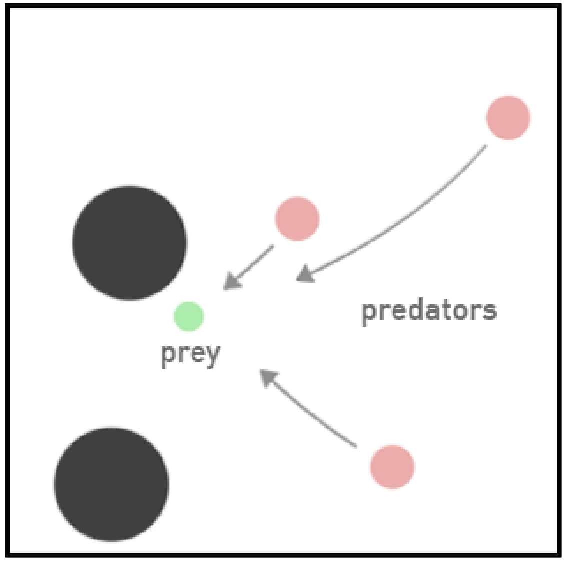


Value Function Critic Loss:

$$\mathcal{L}(\theta_i) = \boxed{\left|\left|V\left(o_t^i;\theta_i\right) - y_i^i\right|\right|^2} + \boxed{\lambda \sum_{k \neq i} \frac{\pi(a_t^k | o_t^k; \phi_i)}{\pi(a_t^k | o_t^k; \phi_k)} \left|\left|V\left(o_t^k;\theta_i\right) - y_k^i\right|\right|^2}$$

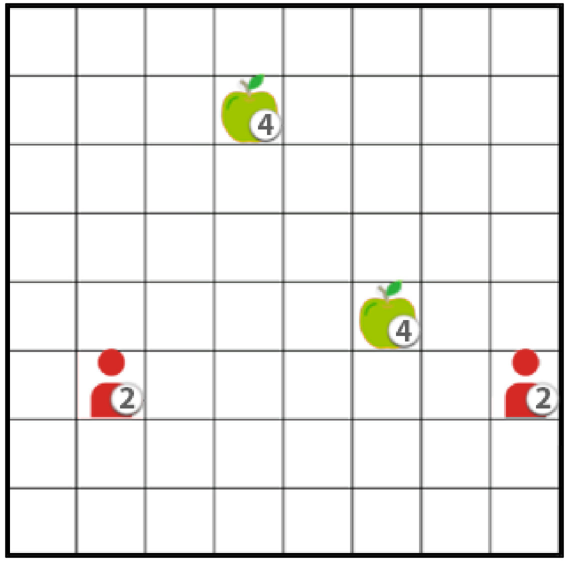
$$y_k^i = r_t^k + \gamma V\left(o_{t+1}^k;\theta_i\right)$$

Evaluation – Domains

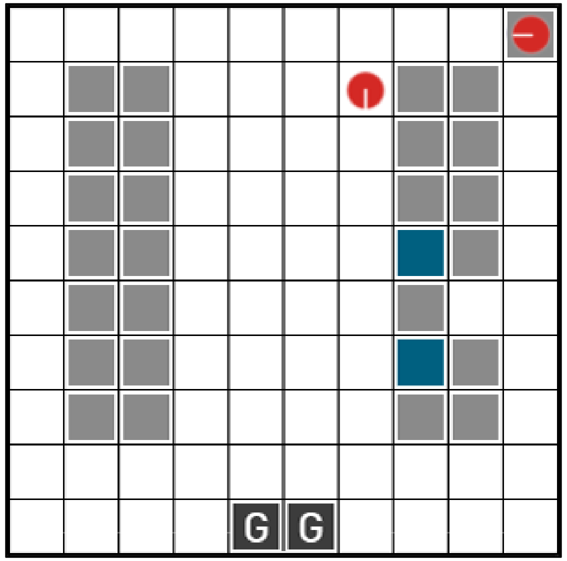


Predator Prey (sparse)

SMAC – 3m (sparse)



Level-Based Foraging (LBF)



Multi-Robot Warehouse (RWARE)

Evaluation – Baselines

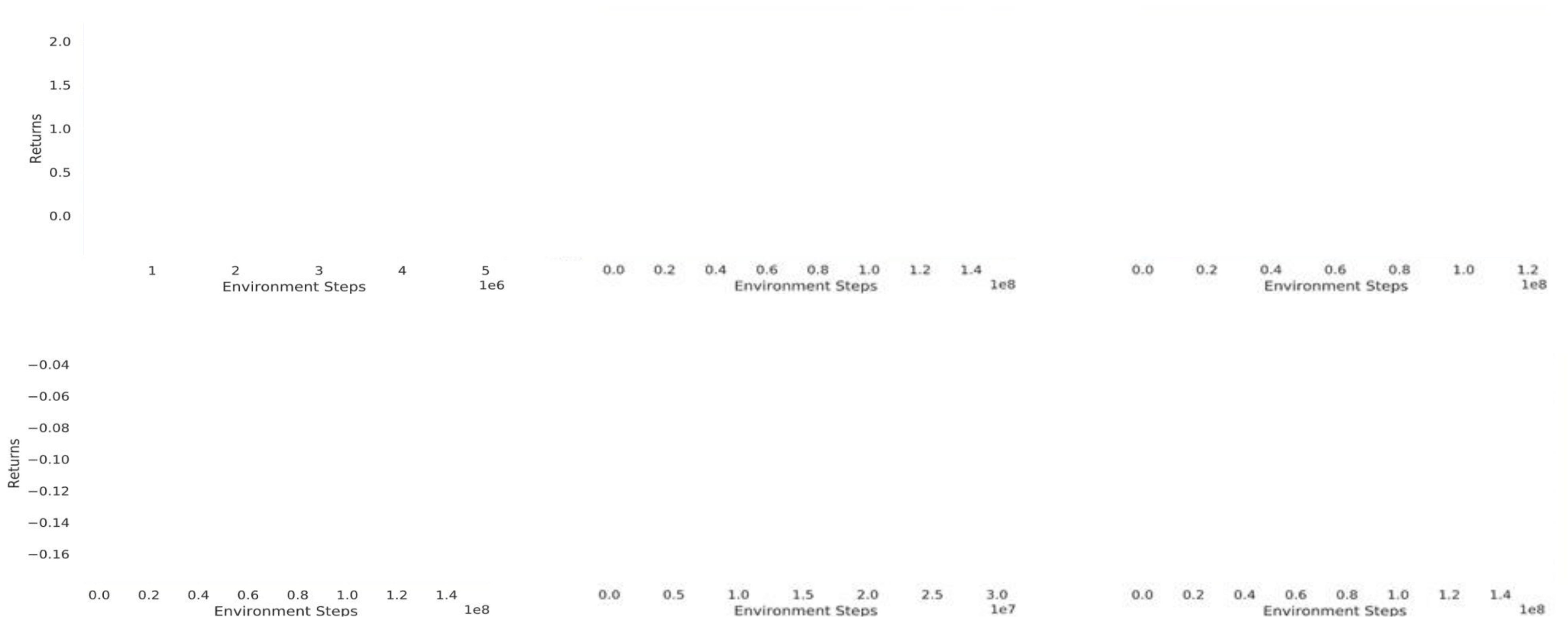
Baselines:

(1) Independent Actor-Critic (IAC)
(2) Shared Network Actor-Critic (SNAC)

State-of-the-art MARL:

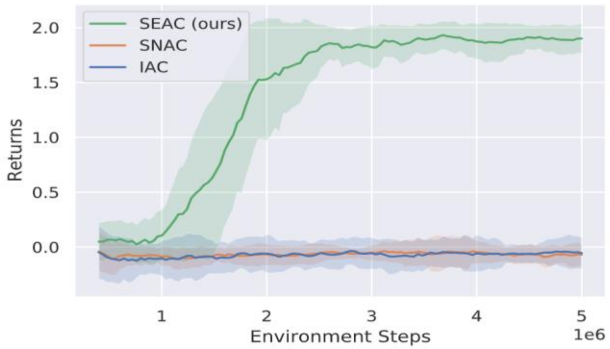
(1) MADDPG
(2) QMIX
(3) ROMA

Results

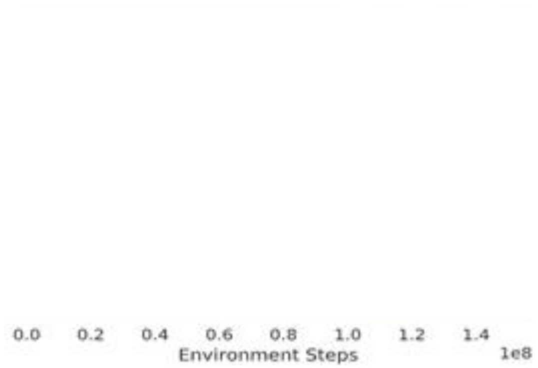


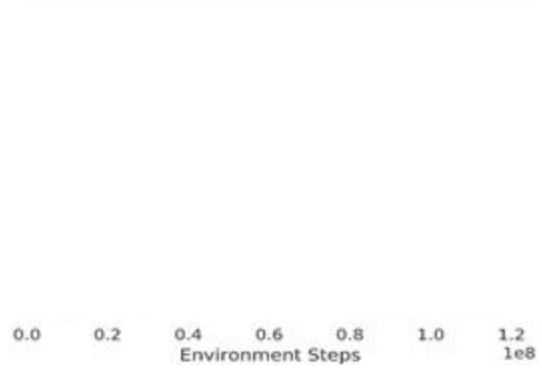
Returns
2.0
1.5
1.0
0.5
0.0
1
2
3
4
5
1e6
Environment Steps
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e8
Environment Steps
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1e8
Environment Steps
Returns
−0.04
−0.06
−0.08
−0.10
−0.12
−0.14
−0.16
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e8
Environment Steps
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e7
Environment Steps
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e8
Environment Steps

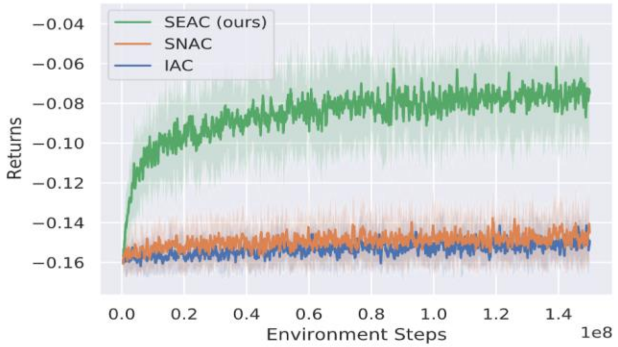
Results



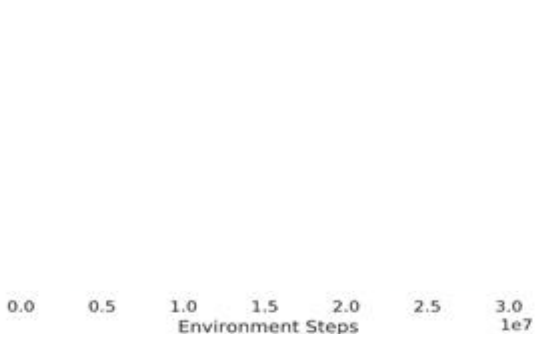
(a) PP, sparse rewards

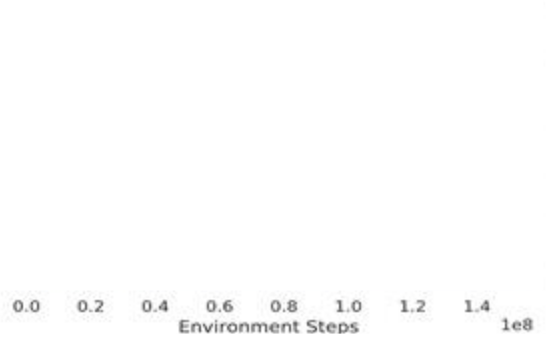




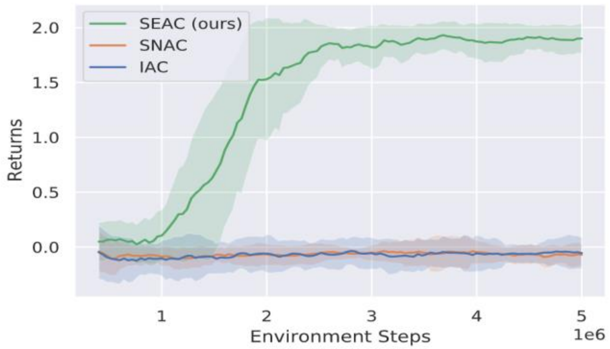


(b) SMAC with three marines, sparse rewards

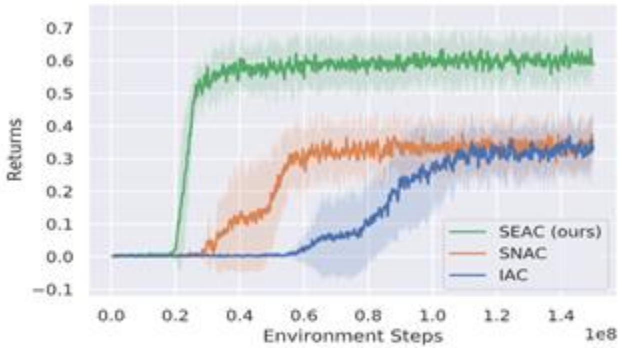




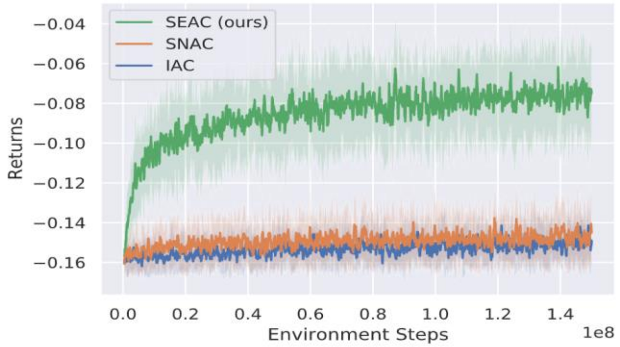
Results



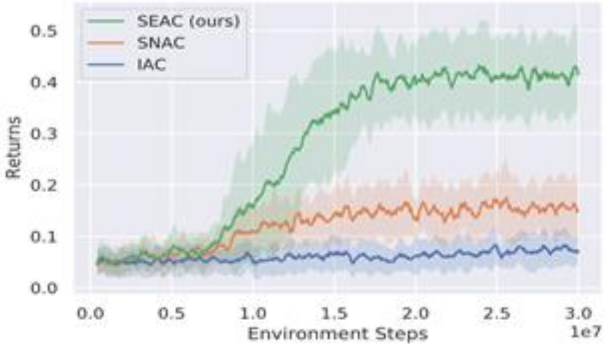
(a) PP, sparse rewards



(d) LBF: (8 × 8), two agents, two foods, cooperative

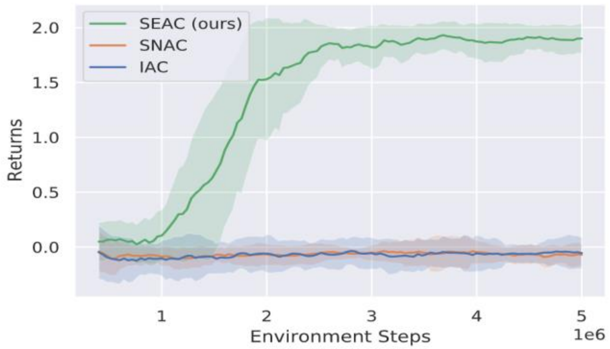


(b) SMAC with three marines, sparse rewards

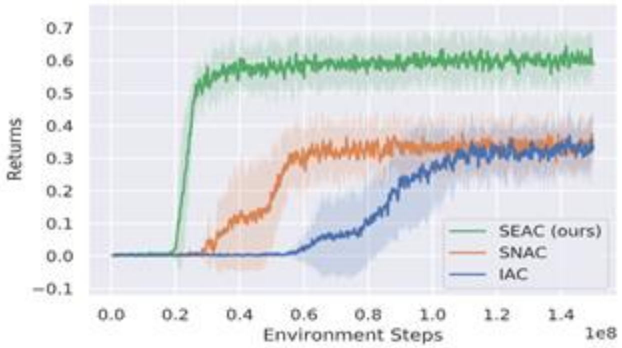


(c) LBF: (15 × 15), three agents, four food

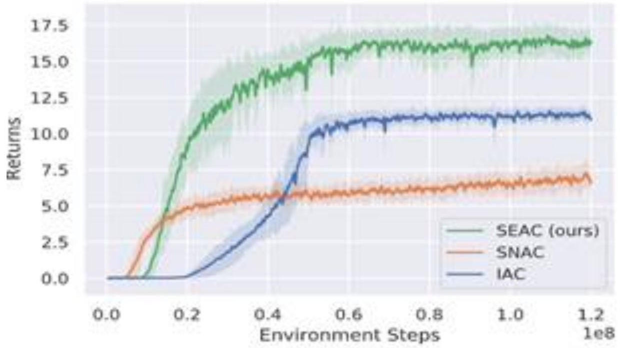
Results



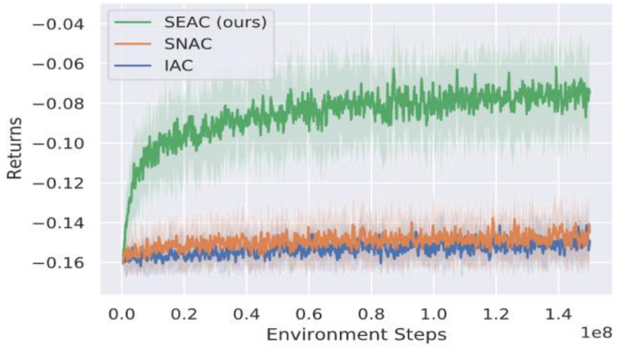
(a) PP, sparse rewards



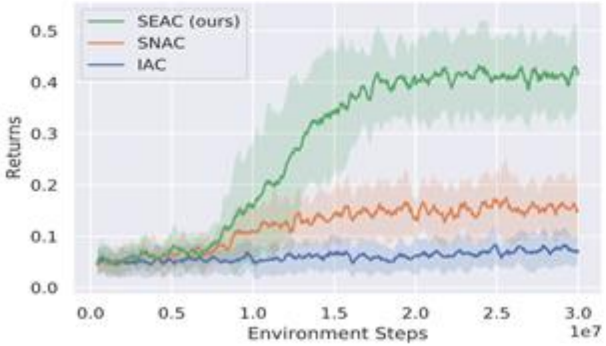
(d) LBF: (8 × 8), two agents, two foods, cooperative



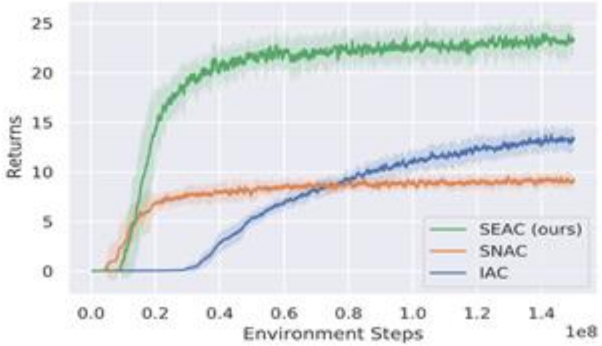
(c) RWARE: (10 × 11), two agents, hard



(b) SMAC with three marines, sparse rewards



(c) LBF: (15 × 15), three agents, four food

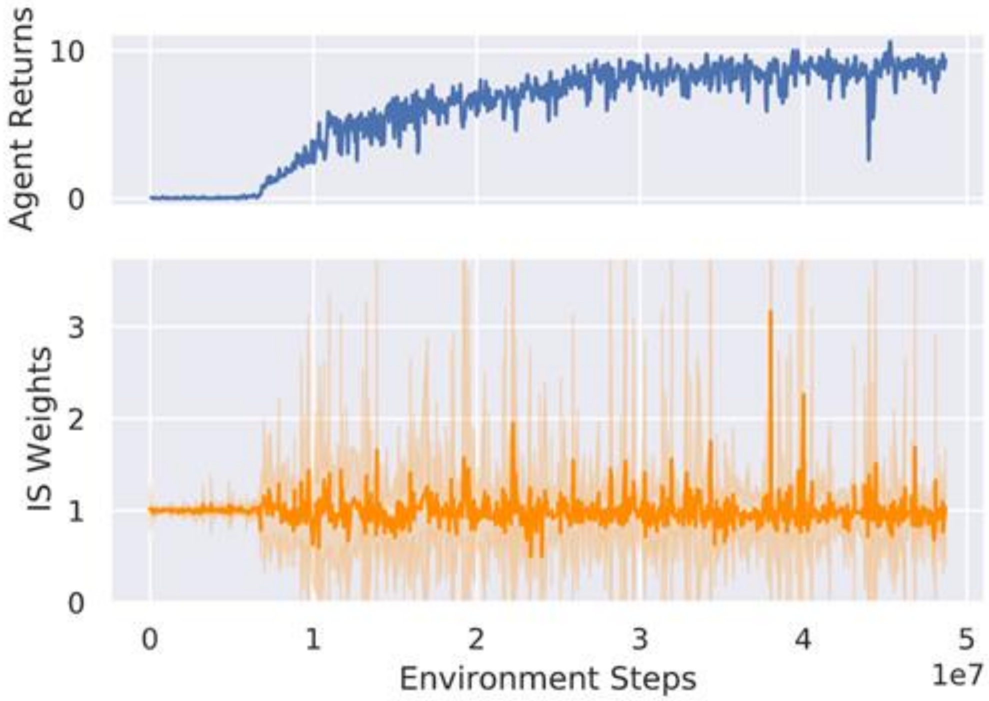


(d) RWARE: (10 × 20), four agents

Results

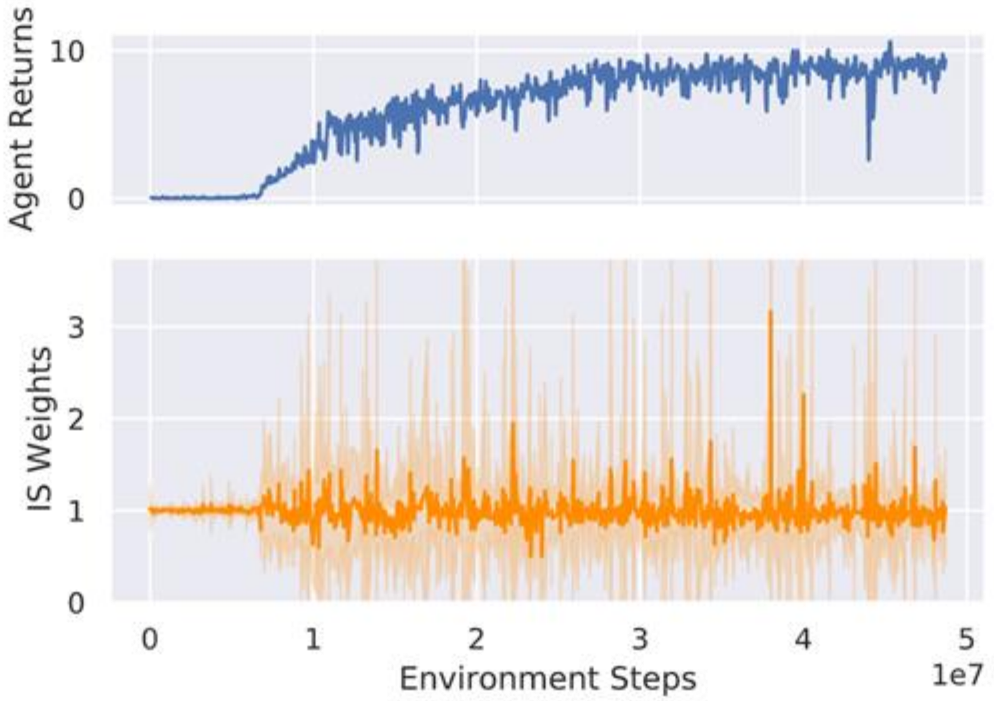
	IAC	SNAC	SEAC (ours)	QMIX	MADDPG	ROMA
PP (sparse)	-0.04 ±0.13	-0.04 ±0.1	**1.93 ±0.13**	0.05 ±0.07	**2.04 ±0.08**	0.04 ±0.07
SMAC-3m (sparse)	-0.13 ±0.01	-0.14 ±0.02	**-0.03 ±0.03**	**0.00 ±0.00**	**-0.01 ±0.01**	**0.00 ±0.00**
LBF-(15x15)-3ag-4f	0.13 ±0.04	0.18 ±0.08	**0.43 ±0.09**	0.03 ±0.01	0.01 ±0.02	0.03 ±0.02
LBF-(8x8)-2ag-2f-coop	0.37 ±0.10	0.38 ±0.10	**0.64 ±0.08**	**0.79 ±0.31**	0.01 ±0.02	0.01 ±0.02
RWARE-(10x20)-4ag	13.75 ±1.26	9.53 ±0.83	**23.96 ±1.92**	0.00 ±0.00	0.00 ±0.00	0.00 ±0.00
RWARE-(10x11)-4ag	**40.10 ±5.60**	36.79 ±2.36	**45.11 ±2.90**	0.00 ±0.00	0.00 ±0.00	0.01 ±0.01

Analysis (1)



Importance weights of one SEAC agent in RWARE, (10x11), two agents, hard

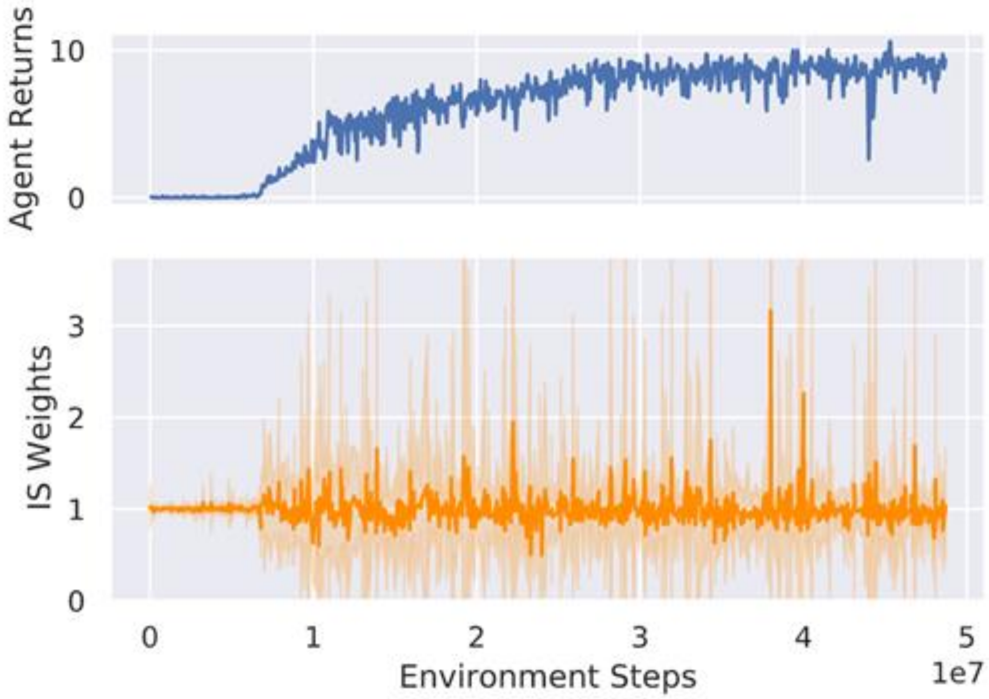
Analysis (1)



Importance weights of one SEAC agent in RWARE, (10x11), two agents, hard

- Agents learn similar, but not identical policies which improves coordination

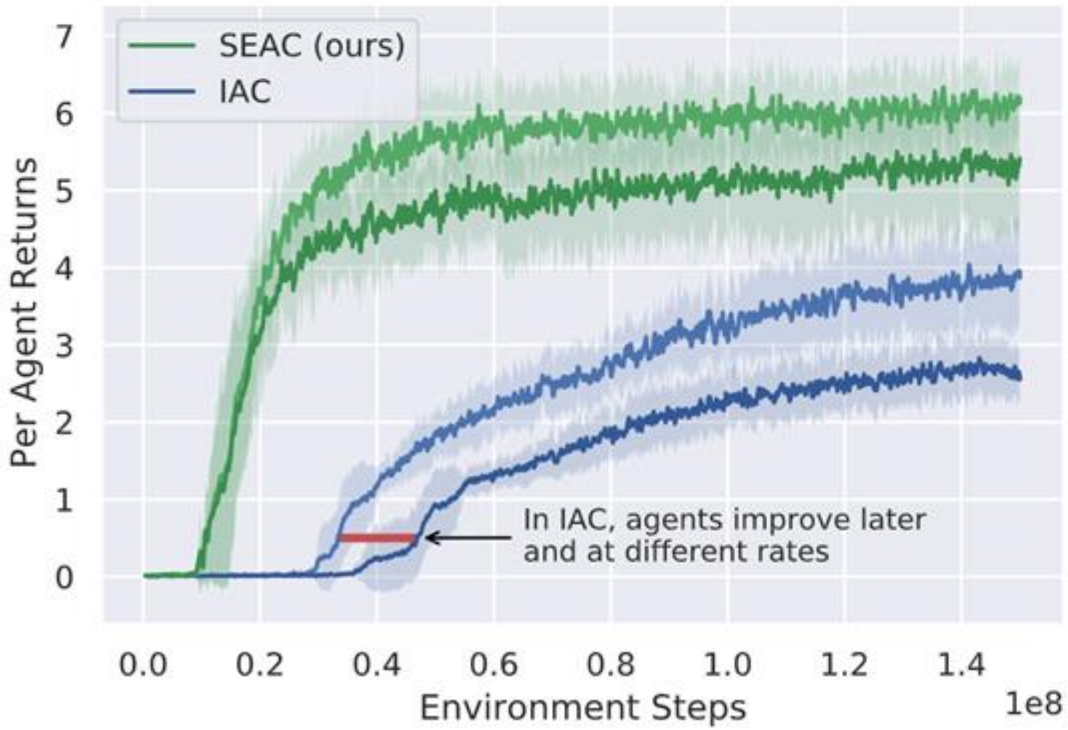
Analysis (1)



Importance weights of one SEAC agent in RWARE, (10x11), two agents, hard

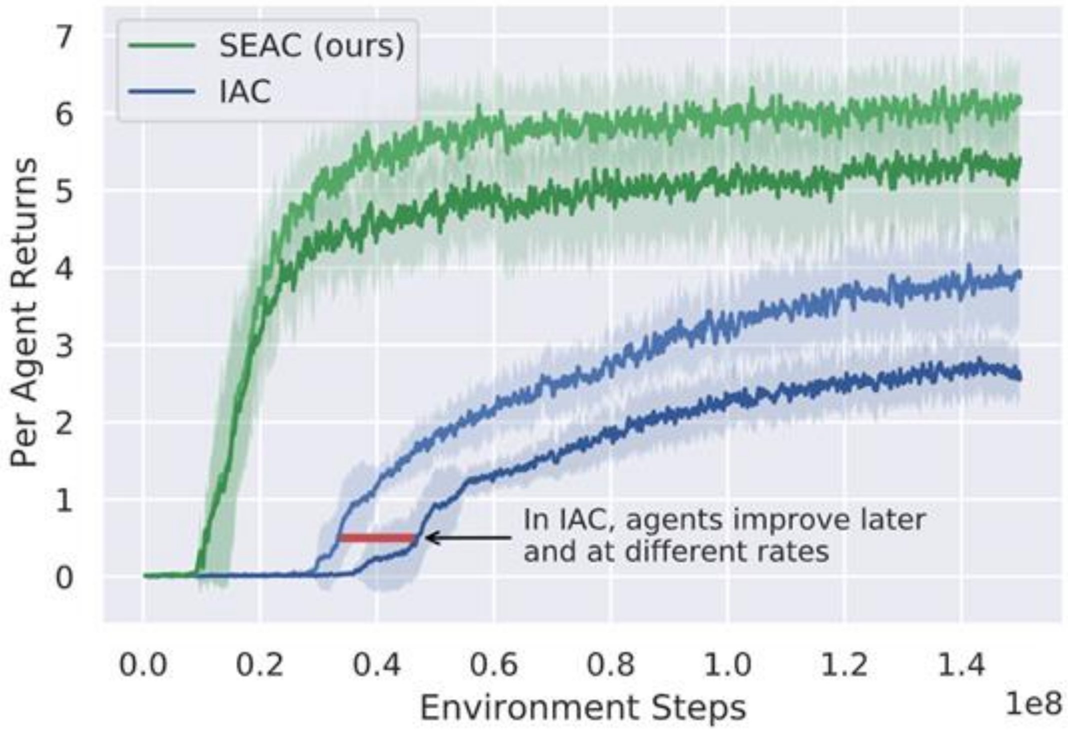
- Agents learn similar, but not identical policies which improves coordination
- Policies diverge because of …
  1. Random network initialization
  2. Entropy regularization term in final policy loss (based on own policy)

Analysis (2)



Best vs. Worst performing agents on RWARE, (10x20), four agents

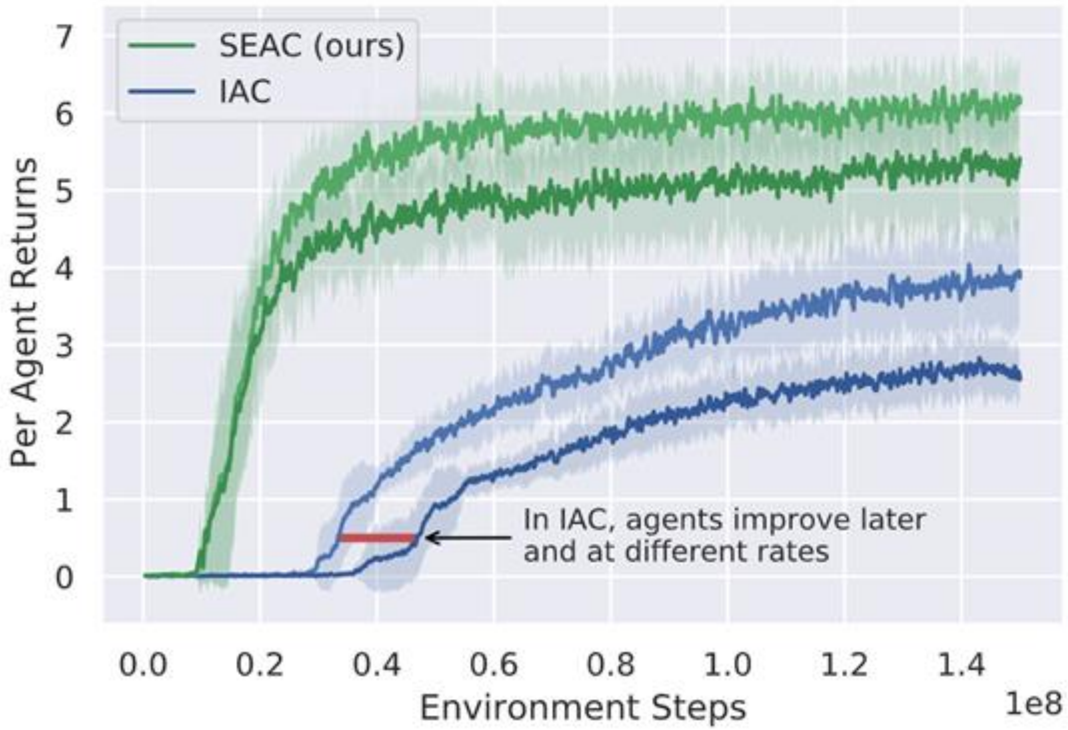
Analysis (2)



Best vs. Worst performing agents on RWARE, (10x20), four agents

- Agents learn simultaneously which helps in exploring promising joint actions more

Analysis (2)



Best vs. Worst performing agents on RWARE, (10x20), four agents

- Agents learn simultaneously which helps in exploring promising joint actions more
- Synchronise training progress of agents

Contributions

We propose a novel experience sharing method (Shared Experience Actor-Critic or SEAC) that combines gradients of multiple agents to share experience between agents.

- Evaluated in four sparse-reward multi-agent environments
- Consistently outperforms baselines and three state-of-the-art MARL algorithms (MADDPG, QMIX, ROMA)
- SEAC learns in fewer steps and converges to higher returns
- In harder tasks, sharing experience makes the difference between not learning at all and learning

Conclusion

- Using our method, agents learn similar – but not identical policies.
  - Facilitates coordination between agents
- Exploration is improved:
  - Agents tend to pick-up behaviors concurrently: more promising joint actions are explored more
- Simple and general method (can be used to extend any on- and even off-policy algorithms)

Future Work

- Relax assumptions about the task required to share experience
- Learn lambda for agents (which agents share experience with whom)

Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning

Filippos Christianos, Lukas Schäfer, Stefano V. Albrecht

Links to code:	SEAC:	https://github.com/uoe-agents/seac
	RWARE:	https://github.com/uoe-agents/robotic-warehouse
	LBF:	https://github.com/uoe-agents/lb-foraging

Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning

Filippos Christianos, Lukas Schäfer, Stefano V. Albrecht

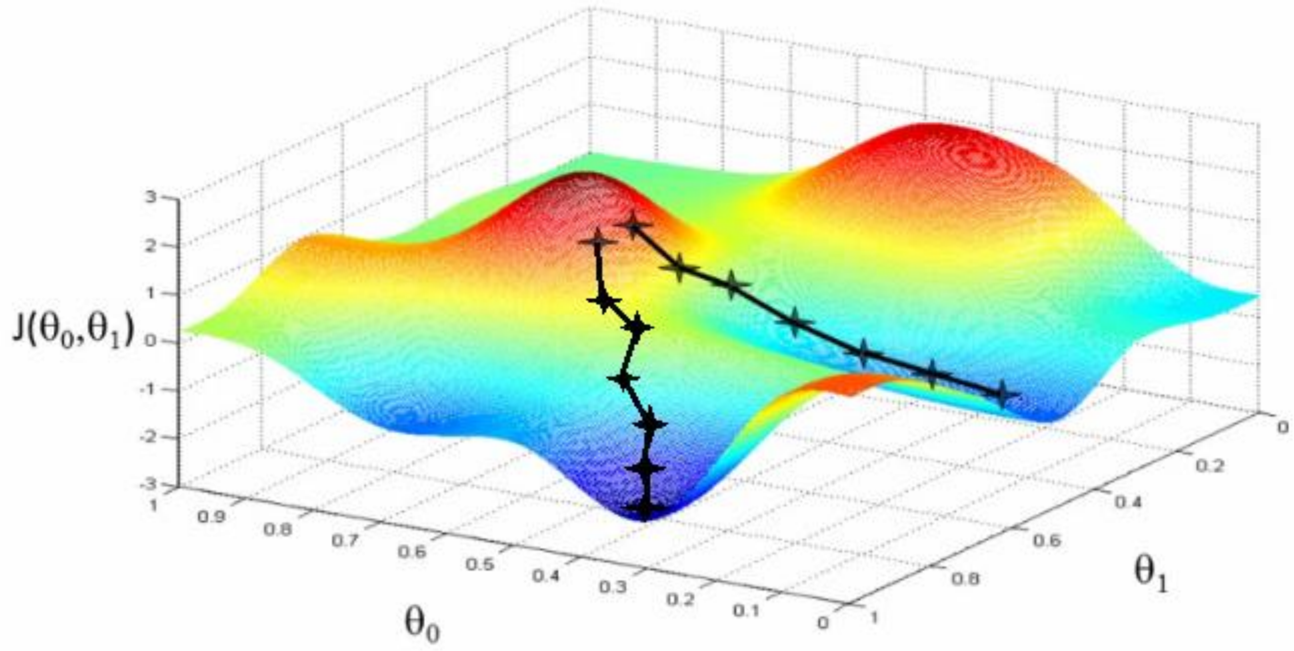
Arxiv: https://arxiv.org/pdf/2006.07169.pdf
Contact: f.christianos@ed.ac.uk

Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. "Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning." In *34th Conference on Neural Information Processing Systems* 2020.

Thank you for your Attention!
Any Questions?

Filippos Christianos, Lukas Schäfer, Stefano V. Albrecht

NeurIPS	Poster: **Poster Session 5** on Thu, Dec 10th, 2020 @ 18:00 – 20:00 CET



$J(\theta_0,\theta_1)$
3
2
1
0
-1
-2
-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
θ_0
0
0.2
0.4
0.6
0.8
1
θ_1