Actor-Attention-Critic for Multi-Agent Reinforcement Learning

Jack (Jianxiang) Xu

Feb 26, 2022

[1] S. Iqbal and F. Sha, “Actor-Attention-Critic for Multi-Agent Reinforcement Learning,” Sep. 2018, Accessed: Feb. 16, 2022. [Online]. Available: https://openreview.net/forum?id=HJx7l309Fm

INTRODUCTION

Multi-Agent Benefits

- Learn faster and better with experience sharing through communication
- Exploitation with a decentralized structure of the task in parallel
- Inherently robust in case of failures of one or more agents
- Scalability

UNIVERSITY OF WATERLOO

Multi-Agent Challenges

- Curse of dimensionality
- A good objectives in general stochastic game is challenging
- Poor Stability
- Exploration-exploitation Trade-off

UNIVERSITY OF WATERLOO

BACKGROUND

Multi-Agent Objectives

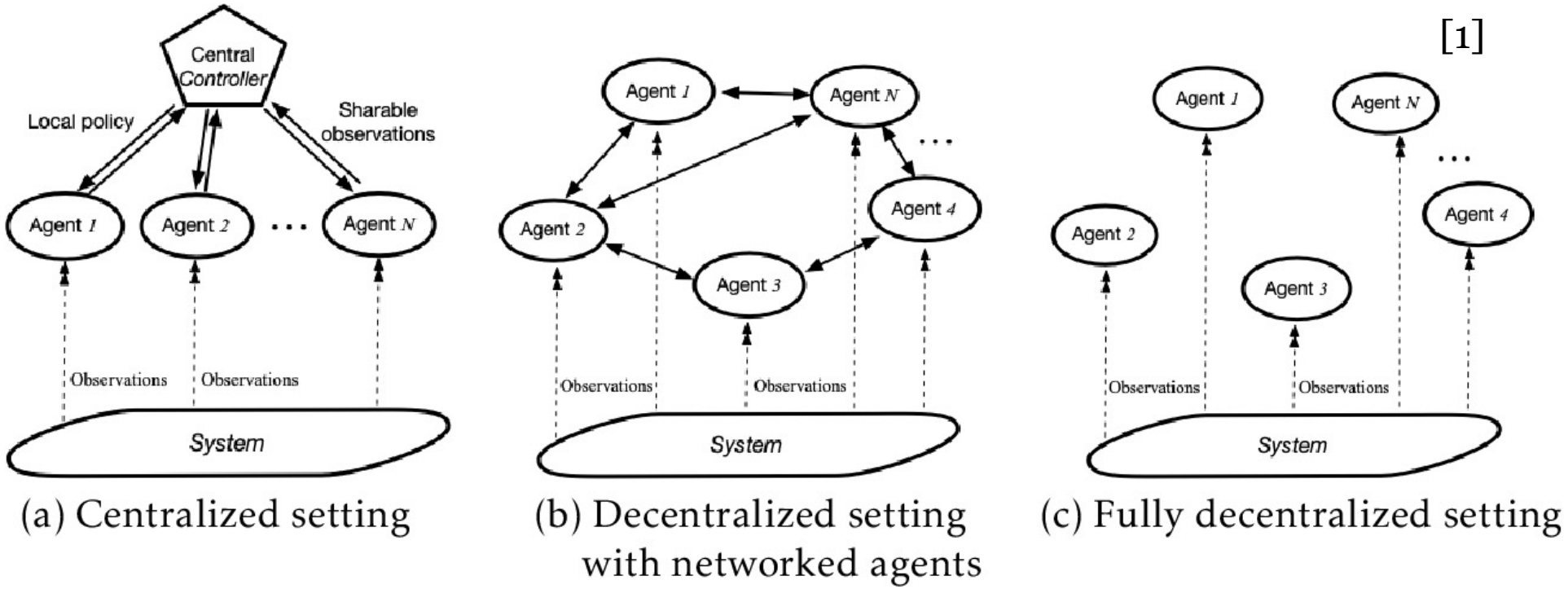
Cooperative

Competitive

Mixed (General sum)

UNIVERSITY OF WATERLOO

Multi-Agent Information Structure



[1]

(a) Centralized setting

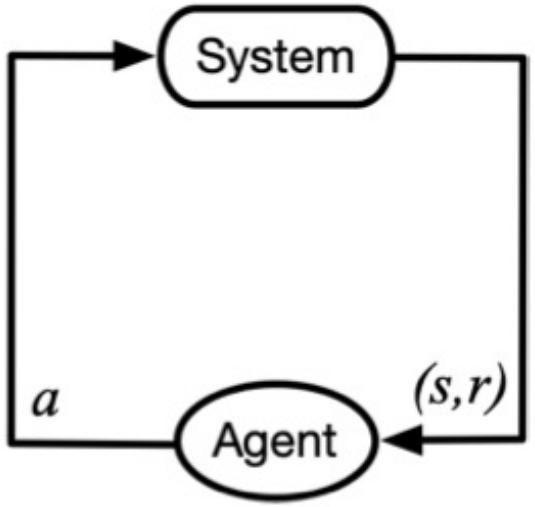
(b) Decentralized setting with networked agents

(c) Fully decentralized setting

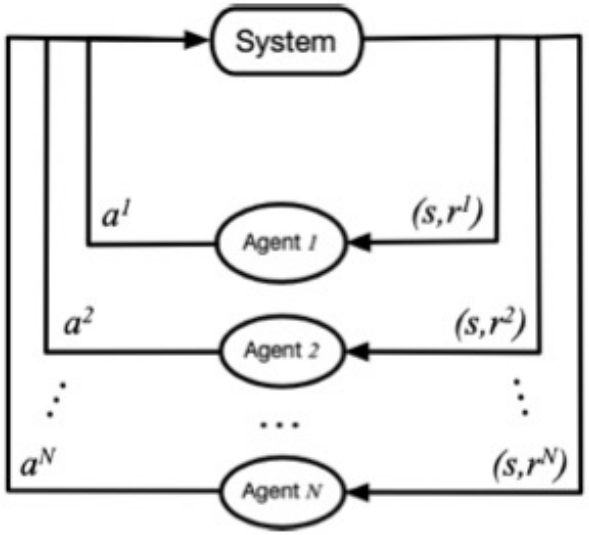
[1] Zhang, K., Yang, Z., & Başar, T. (2019). Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. *ArXiv, abs/1911.10635*.

UNIVERSITY OF WATERLOO

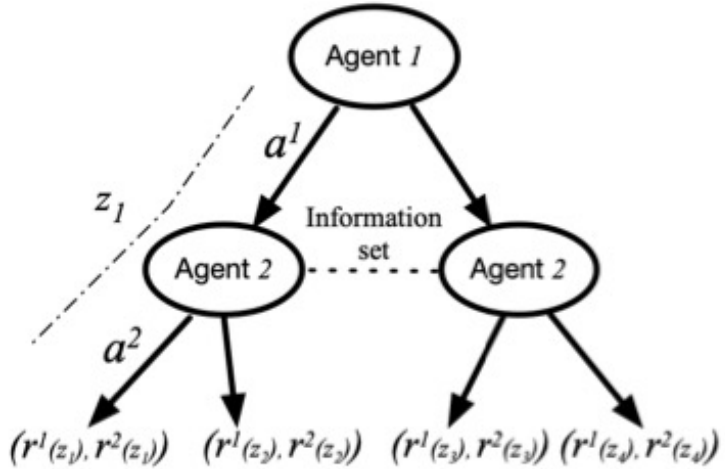
RL Frameworks



(a) Markov decision process



(b) Markov game



(c) Extensive-form game

UNIVERSITY OF WATERLOO

Timeline of Related Works

Learning Communication in Cooperative Agents
(Tan, 1993; Fischer et al., 2004)

Optimal Play in Competitive
(Littman, 1994)
Markov Games as a Framework for MARL
(Littman, 1994)

Deep MARL
(Tampuu et al., 2017; Gupta et al., 2017),

Attention is All You Need !!!
(Vaswani et al., 2017)

Attention in Fully Centralized MARL
(Choi et al., 2017)

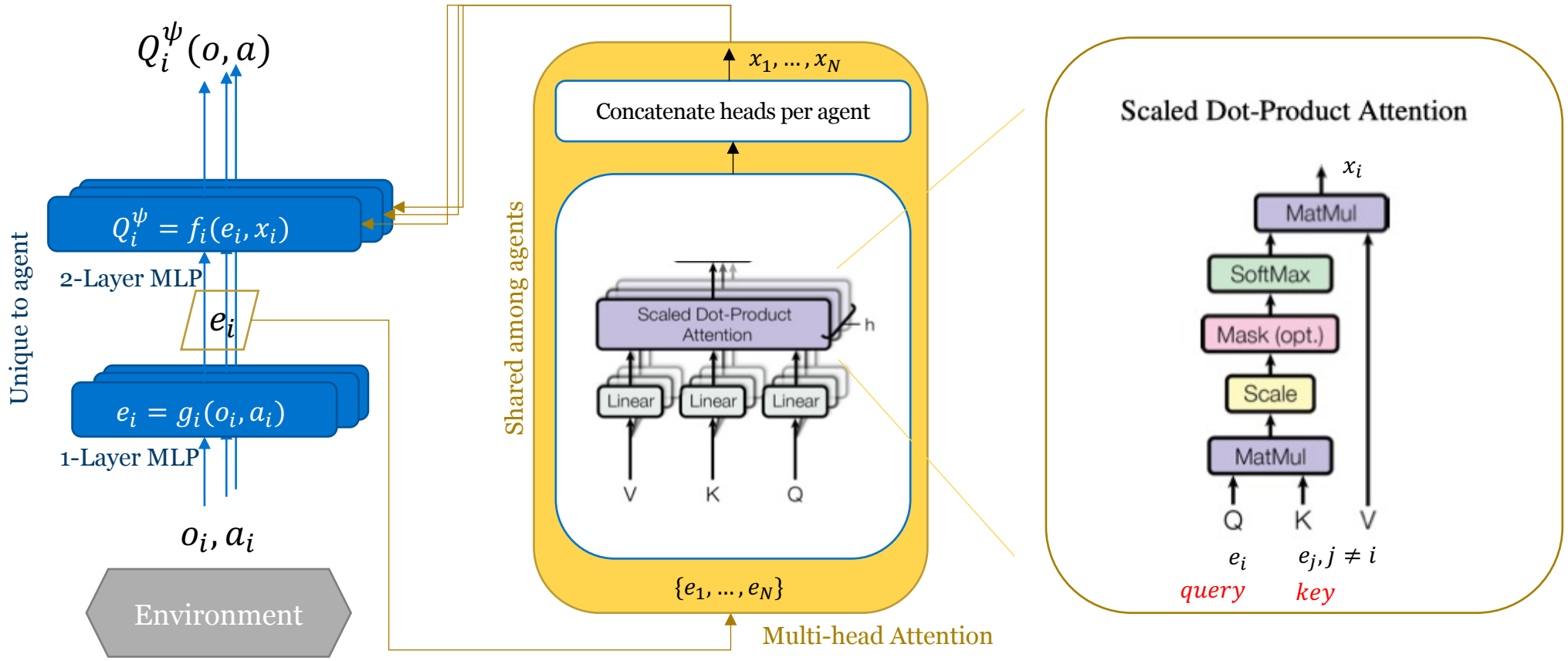
Attention-based Actor Critic
(Jiang & Lu, 2018)

Actor-Attention Critic MARL
(Ours, 2019)

UNIVERSITY OF WATERLOO

PROPOSED METHOD

Proposed Solution – Actor Attention Critic



Multi-Agent Objectives

Cooperative

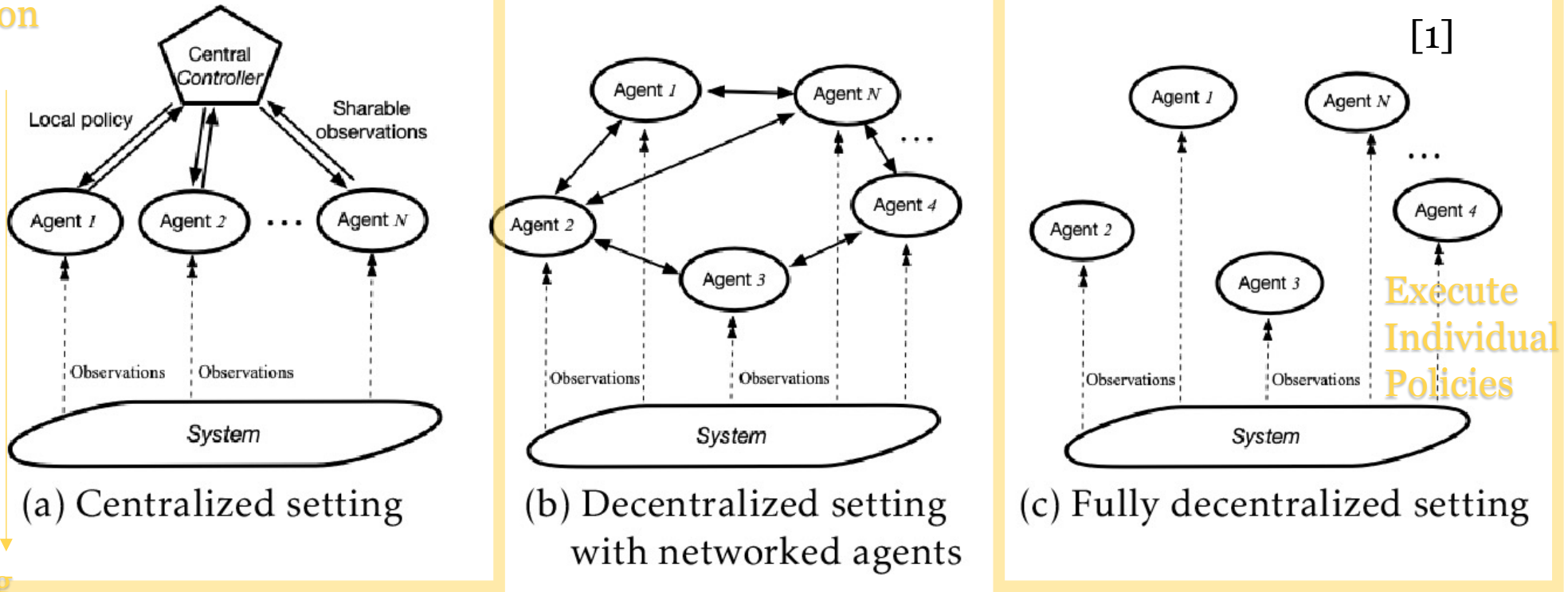
Competitive

Mixed (General sum)

UNIVERSITY OF WATERLOO

Multi-Agent Information Structure

Learning Attention Critic



Learning Individual Policies (Actors)

Execute Individual Policies

[1] Zhang, K., Yang, Z., & Başar, T. (2019). Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. *ArXiv, abs/1911.10635*.

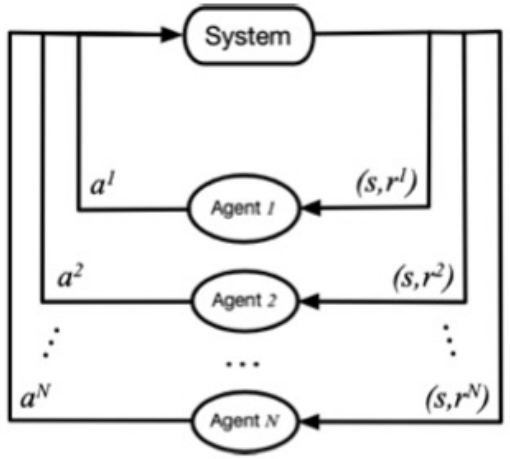
UNIVERSITY OF WATERLOO

Multi-agent Markov Game Framework (Notation)

- A set of states: S
- A set of agents: $\{1, \dots, N\}$
- Action sets for each of N agents: $\{A_1, \dots, A_N\}$
- Replay buffer: $(\mathrm{s}, \mathrm{a}, \mathrm{r}, s') \sim D$

- State Transition Function: $T: S \times A_1 \times \cdots \times A_N \rightarrow P(S)$
- Reward Function: $R_i: S \times A_1 \times \cdots \times A_N \rightarrow \mathbb{R}$

- Partially Observable Variant:
  - o_i : observation of agent i
  - π_i: $O_i \rightarrow P(A_i)$
  - Objective: $J_i(\pi_i) = \mathbb{E}_{a_1 \sim \pi_1, \dots, a_N \sim \pi_N \mid s \sim T} \left[\sum_{t=0}^{\infty} \gamma^t r_{it}(s_t, a_{1t}, \dots a_{Nt})\right]$



(b) Markov game

UNIVERSITY OF WATERLOO

Actor-Critic

Policy Gradient

$$\nabla_\theta J(\pi_\theta) = \nabla_\theta \log(\pi_\theta(a_t|s_t)) \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}(s_{t'}, a_{t'})$$

Actor-Critic

$$Q_\psi(s_t, a_t) = \mathbb{E}[\textstyle\sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}(s_{t'}, a_{t'})]$$

Replay Buffer

$$\mathcal{L}_Q(\psi) = \mathbb{E}_{(s,a,r,s')\sim D}\left[(Q_\psi(s,a) - y)^2\right]$$

$$\text{where } y = r(s,a) + \gamma \mathbb{E}_{a'\sim\pi(s')}\left[Q_{\bar{\psi}}(s', a')\right]$$

Exponential Moving Average of Past Q

Soft Actor-Critic (entropy)

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim D, a\sim\pi}[\nabla_\theta \log(\pi_\theta(a|s))(-\alpha \log(\pi_\theta(a|s)) + Q_\psi(s,a) - b(s))]$$

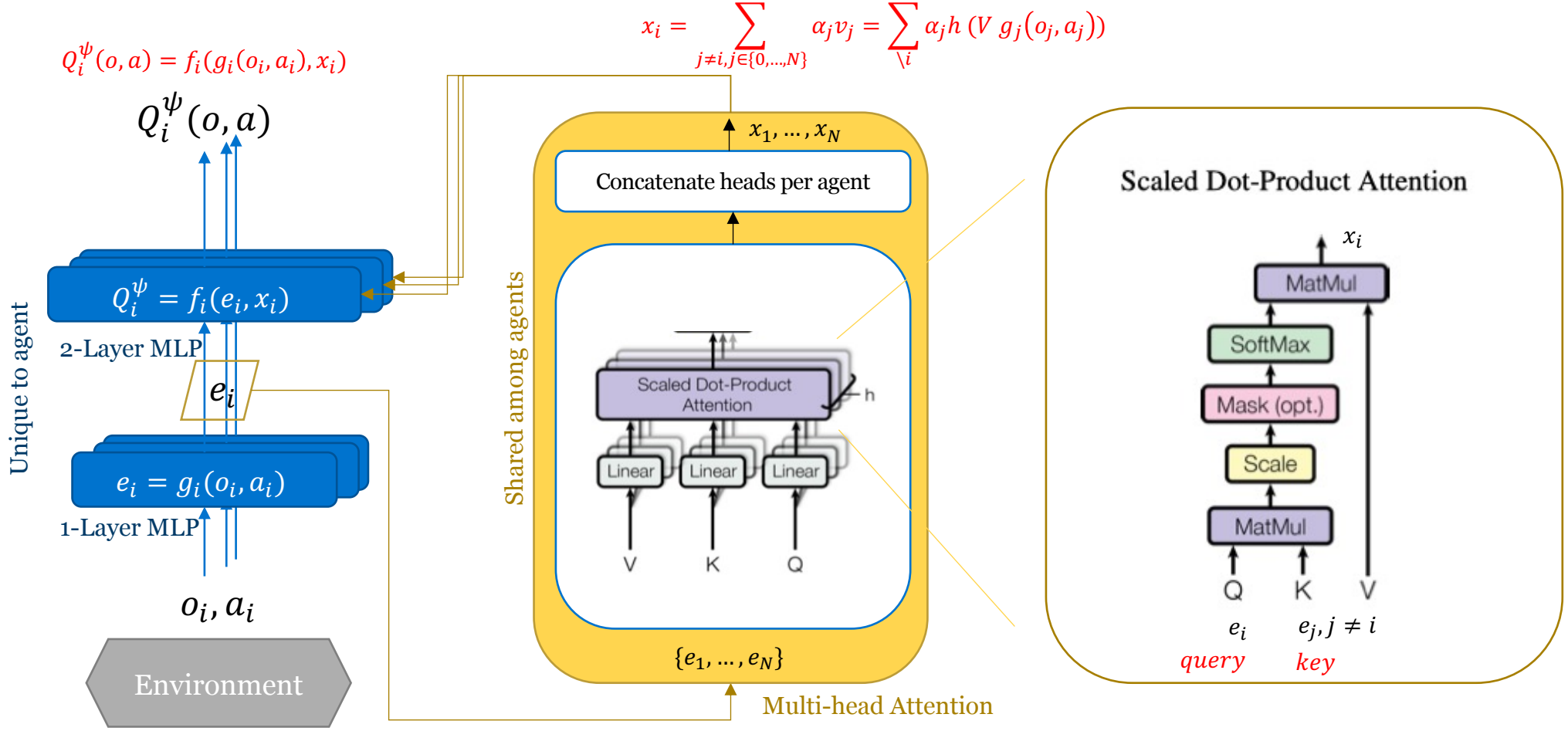
$$y = r(s,a) + \gamma \mathbb{E}_{a'\sim\pi(s')}[Q_{\bar{\psi}}(s', a') - \alpha \log(\pi_{\bar{\theta}}(a'|s'))]$$

Advantage Function

Temperature

UNIVERSITY OF WATERLOO

Attention Module



Learning with Attentive Critics

- Shared Central Critics (minimize a joint regression loss function):
  - $\mathcal{L}_Q(\psi) = \sum_{i=1}^{N} \mathbb{E}_{(o,a,r,o')\sim D} \left[\left(Q_i^{\psi}(o,a) - y_i \right)^2 \right]$
  - $y_i = r_i + \gamma \mathbb{E}_{a' \sim \pi_\theta(o')} \left[Q_i^{\bar{\psi}}(o', a') - \alpha \log\left(\pi_{\overline{\theta_i}}(a_i'|o_i') \right) \right]$
- Individual Policies (actor):
  - $\nabla_{\theta_i} \mathrm{J}(\pi_\theta) = \mathbb{E}_{o\sim D, a\sim\pi} \left[\nabla_{\theta_i} \log\left(\pi_{\theta_i}(a_i|o_i) \right) \left(-\alpha \log\left(\pi_{\theta_i}(a_i|o_i) \right) + A_i(o,a) \right) \right]$
- Baseline: $\mathrm{b}(o, a_{\setminus i}) = \mathbb{E}_{a_i \sim \pi_i(o_i)} \left[Q_i^{\psi} \left(o, (a_i, a_{\setminus i}) \right) \right] = \sum_{a_i' \in A_i} \pi(a_i'|o_i)\, Q_i \left(o, (a_i', a_{\setminus i}) \right)$
- Advantage: $A_i(o,a) = Q_i^{\psi}(o,a) - b(o, a_{\setminus i})$

UNIVERSITY OF WATERLOO

PSEUDO-CODE

Algorithm

Algorithm 1.1: Seudo implementation of Actor-Attention-Critic

AAC()

Initialize E parallel environments with N agents

Initialize replay buffer D

Loop forever (for each episode $k = 1, \ldots$)

Reset E environments, and initialize $o^e_{i,0}$ for each agent i

Loop through each episode (for each time step $n = 1, \ldots$)

Select $a^e_{i,n} \sim \pi_i(\cdot|o^e_i)$ for each agent i in each environment e

Execute $a^e_{i,n}$ to all parallel environments, and observe $o^e_{i,n+1}, r^e_{i,n}, \forall i \in N$

Store transitions for all environments in D

Every C steps, do

Update critic N_c times:

Sample Mini-batch from buffer, $(o^B_{1\ldots N}, a^B_{1\ldots N}, r^B_{1\ldots N}, o'^B_{1\ldots N}) \in B \sim D$

Compute $Q^{\psi}_i(o^B_{1\ldots N}, a^B_{1\ldots N}), \forall i \in 1, \ldots, N$ (in parallel)

Compute $a'^B_i \sim \pi^{\bar{\theta}}_i(o'^B_i), \forall i \in 1, \ldots, N$

Compute $Q^{\bar{\psi}}_i(o'^B_{1\ldots N}, a'^B_{1\ldots N}), \forall i \in 1, \ldots, N$ (in parallel)

Update critic with Adam: $\nabla \mathscr{L}_Q(\psi)$

Update policy N_p times:

Sample $(o_{1\ldots N}) \sim D$

Compute $a'^B_i \sim \pi^{\bar{\theta}}_i(o'^B_i), \forall i \in 1, \ldots, N$

Compute $Q^{\bar{\psi}}_i(o'^B_{1\ldots N}, a'^B_{1\ldots N}), \forall i \in 1, \ldots, N$ (in parallel)

Update individual policies with Adam: $\nabla_{\theta_i} J(\pi_\theta)$

Update target parameters:

$\bar{\phi} \leftarrow \tau\bar{\phi} + (1-\tau)\phi, \bar{\theta} \leftarrow \tau\bar{\theta} + (1-\tau)\theta$

UNIVERSITY OF WATERLOO

Algorithm (Simplified)

Algorithm 1.1: Seudo implementation of Actor-Attention-Critic

AAC()

Initialize E parallel environments with N agents

Initialize replay buffer D

Loop forever (for each episode $k = 1, \ldots$)

Reset E environments, and initialize $o^e_{i,0}$ for each agent i

Loop through each episode (for each time step $n = 1, \ldots$)

Select $a^e_{i,n} \sim \pi_i(\cdot|o^e_i)$ for each agent i in each environment e

Execute $a^e_{i,n}$ to all parallel environments, and observe $o^e_{i,n+1}, r^e_{i,n}, \forall i \in N$

Store transitions for all environments in D

Sample Mini-batch from buffer, $(o^B_{1\ldots N}, a^B_{1\ldots N}, r^B_{1\ldots N}, o'^B_{1\ldots N}) \in B \sim D$

Compute $Q^{\psi}_i(o^B_{1\ldots N}, a^B_{1\ldots N}), \forall i \in 1, \ldots, N$ (in parallel)

Compute $a'^B_i \sim \pi^{\bar{\theta}}_i(o'^B_i), \forall i \in 1, \ldots, N$

Compute $Q^{\bar{\psi}}_i(o'^B_{1\ldots N}, a'^B_{1\ldots N}), \forall i \in 1, \ldots, N$ (in parallel)

Update critic with Adam: $\nabla \mathscr{L}_Q(\psi)$

Update individual policies with Adam: $\nabla_{\theta_i} J(\pi_\theta)$

UNIVERSITY OF WATERLOO

RESULTS

Comparison

Table 1. Comparison of various methods for multi-agent RL

	Base Algorithm	How to incorporate other agents	Number of Critics	Multi-task Learning of Critics	Multi-Agent Advantage
MAAC (ours)	SAC[‡]	Attention	N	✓	✓
MAAC (Uniform) (ours)	SAC	Uniform Atttention	N	✓	✓
COMA[*]	Actor-Critic (On-Policy)	Global State + Action Concatenation	1		✓
MADDPG[†]	DDPG[**]	Observation and Action Concatenation	N		
COMA+SAC	SAC	Global State + Action Concatenation	1		✓
MADDPG+SAC	SAC	Observation and Action Concatenation	N		✓

Heading Explanation *How to incorporate other agents*: method by which the centralized critic(s) incorporates observations and/or actions from other agents (MADDPG: concatenating all information together. COMA: a global state instead of concatenating observations; however, when the global state is not available, all observations must be included.) *Number of Critics*: number of separate networks used for predicting Q_i for all N agents. *Multi-task Learning of Critics*: all agents' estimates of Q_i share information in intermediate layers, benefiting from multi-task learning. *Multi-Agent Advantage*: cf. Sec 3.2 for details.
Citations: [*](Foerster et al., 2018), [†](Lowe et al., 2017), [‡](Haarnoja et al., 2018), [**](Lillicrap et al., 2016)

UNIVERSITY OF WATERLOO

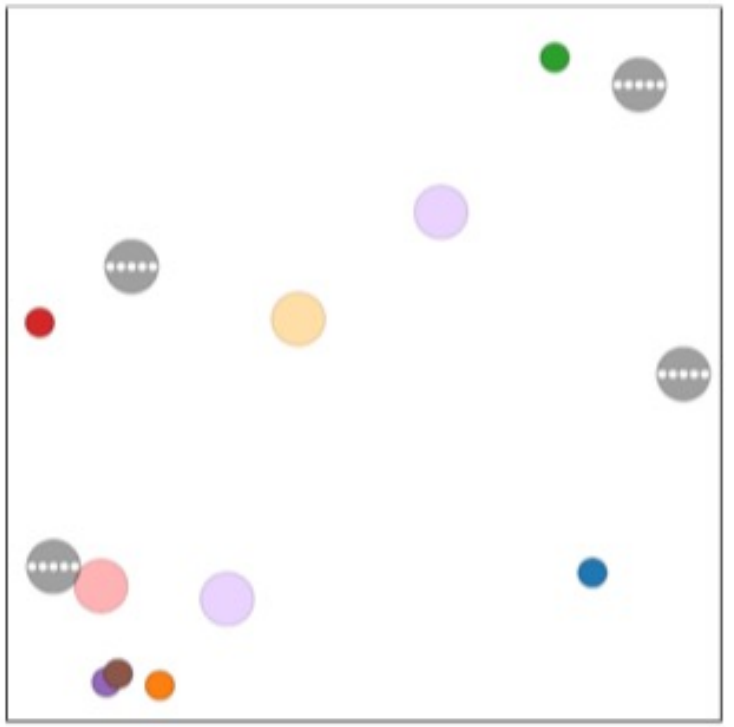
Setup

- 12 parallel rollouts
- 100 steps per episode
- 1024 mini batch size
- Adam, $\eta = 0.001$
- Discount, $\gamma = 0.99$
- Update rate, $\tau = 0.005$
- Hidden dimension, 128
- ReLU
- 4 Attention Heads

UNIVERSITY OF WATERLOO

Setup

(a) Cooperative Treasure Collection. The small grey agents are "hunters" who collect the colored treasure, and deposit them with the correctly colored large "bank" agents.



(b) Rover-Tower. Each grey "Tower" is paired with a "Rover" and a destination (color of rover corresponds to its destination). Their goal is to communicate with the "Rover" such that it moves toward the destination.

Empirical Results

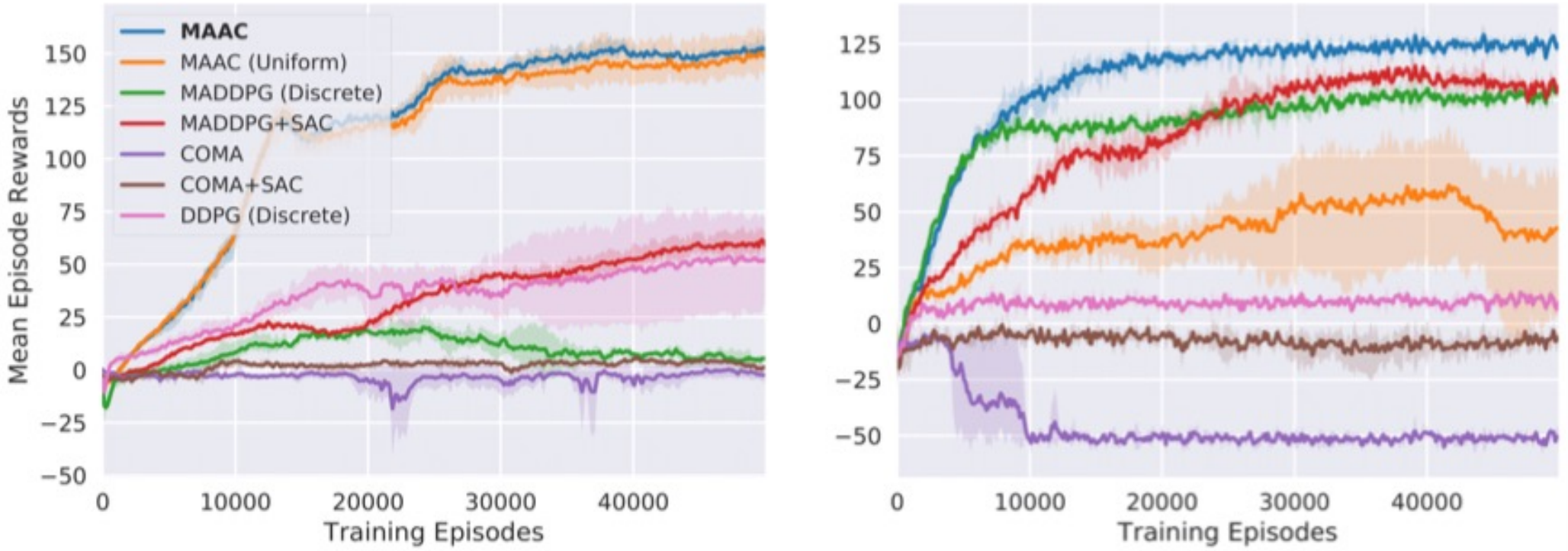


Figure 3. (Left) Average Rewards on Cooperative Treasure Collection. (Right) Average Rewards on Rover-Tower. Our model (MAAC) is competitive in both environments. Error bars are a 95% confidence interval across 6 runs.

UNIVERSITY OF WATERLOO

Empirical Results

Table 3. MAAC improvement over MADDPG+SAC in CTC

# Agents	4	8	12
% Improvement	17	98	208

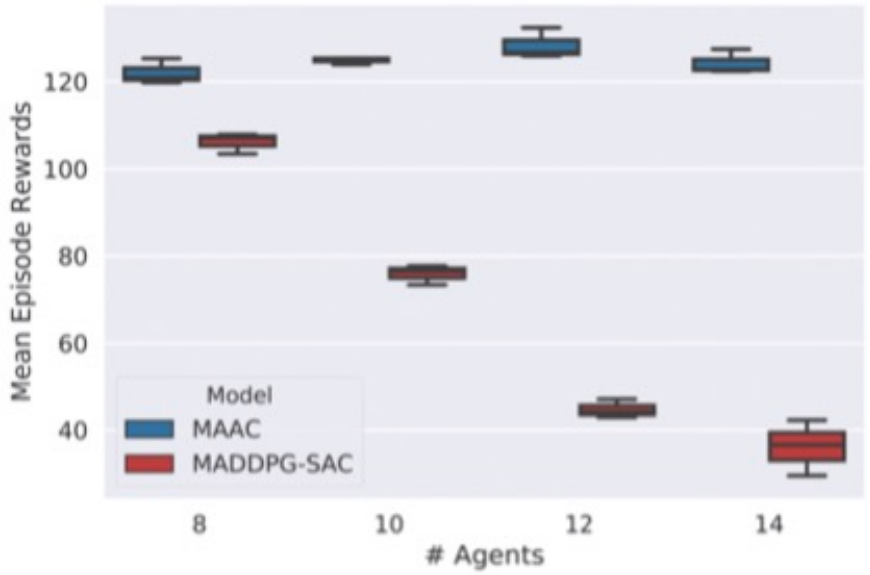


Figure 4. Scalability in the Rover-Tower task. Note that the performance of MAAC does not deteriorate as agents are added.

UNIVERSITY OF WATERLOO

Key Highlights

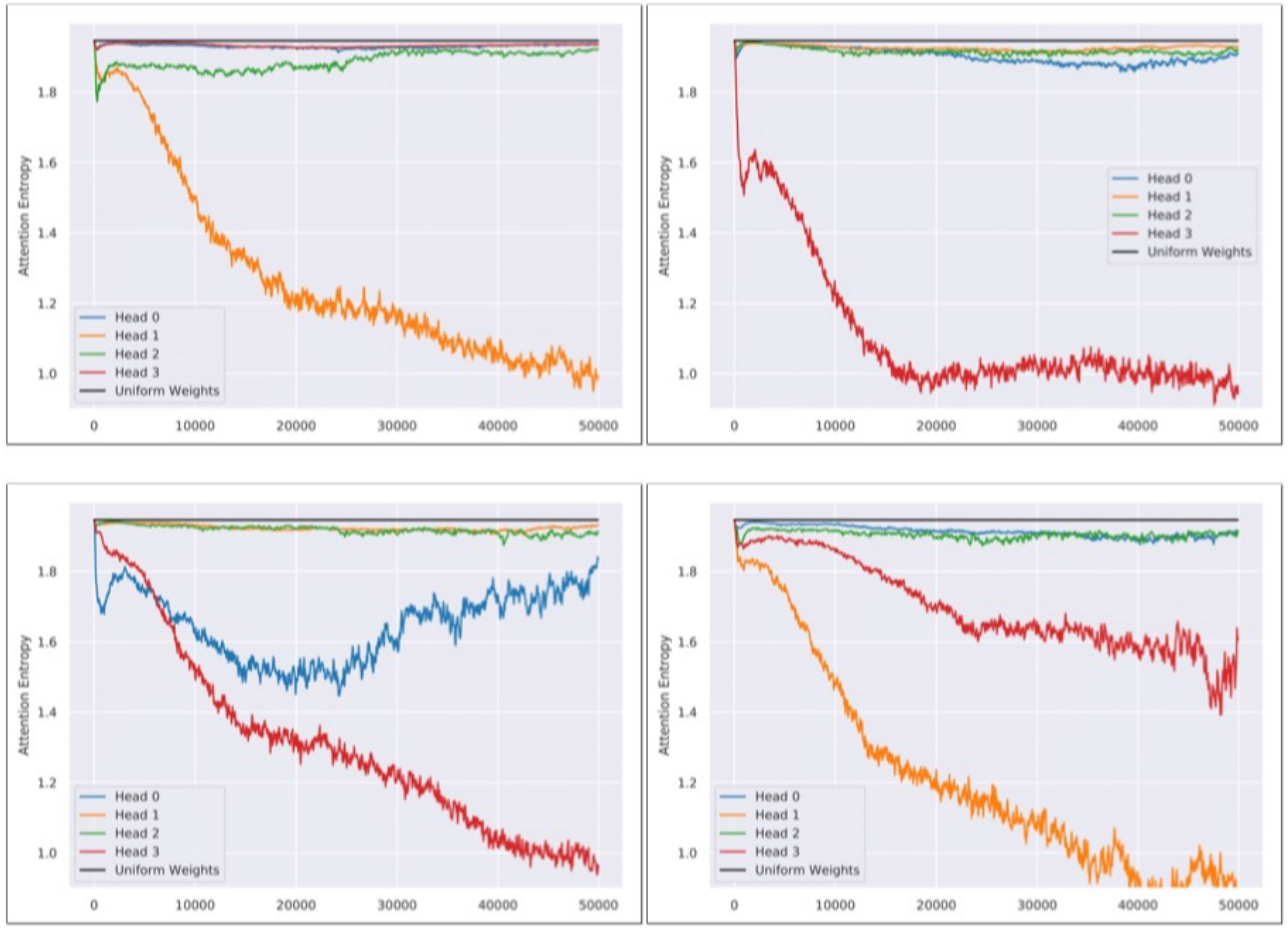


Figure 6. Attention "entropy" for each head over the course of training for the four rovers in the Rover-Tower environment

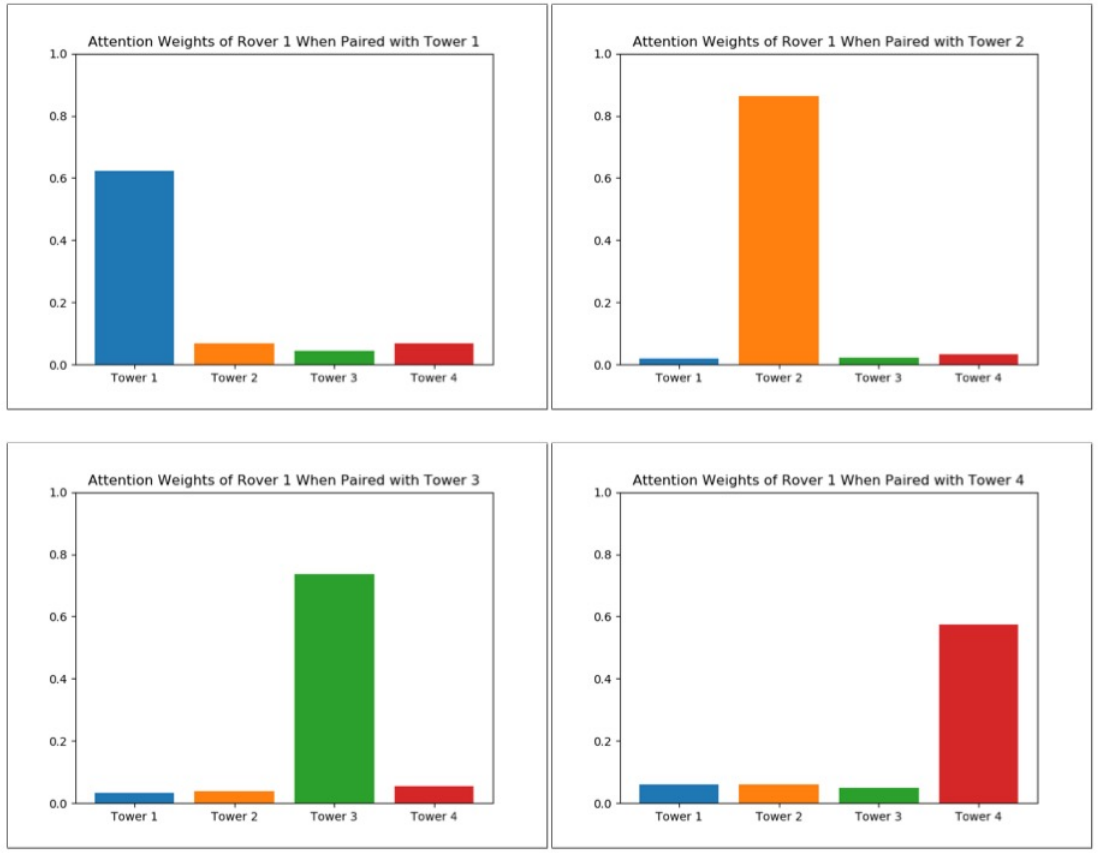


Figure 8. Attention weights when subjected to different Tower pairings for Rover 1 in Rover-Tower environment

Key Highlights

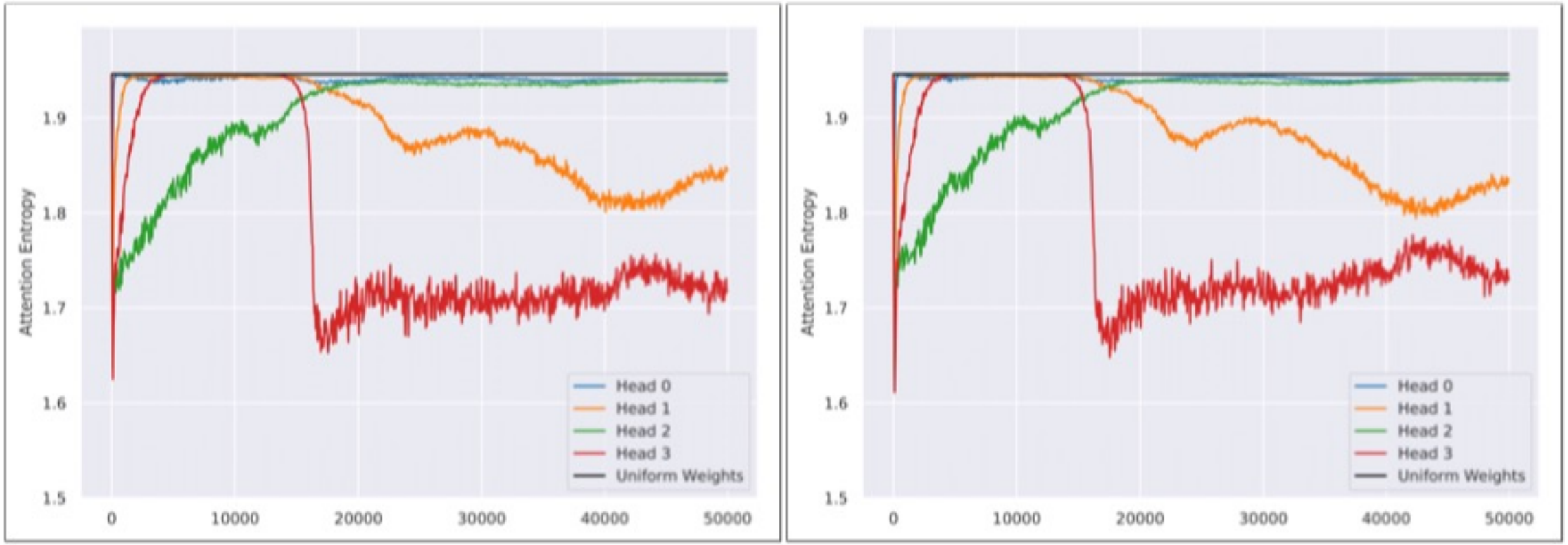


Figure 7. Attention "entropy" for each head over the course of training for two collectors in the Treasure Collection Environment

CONCLUSION

Key Points

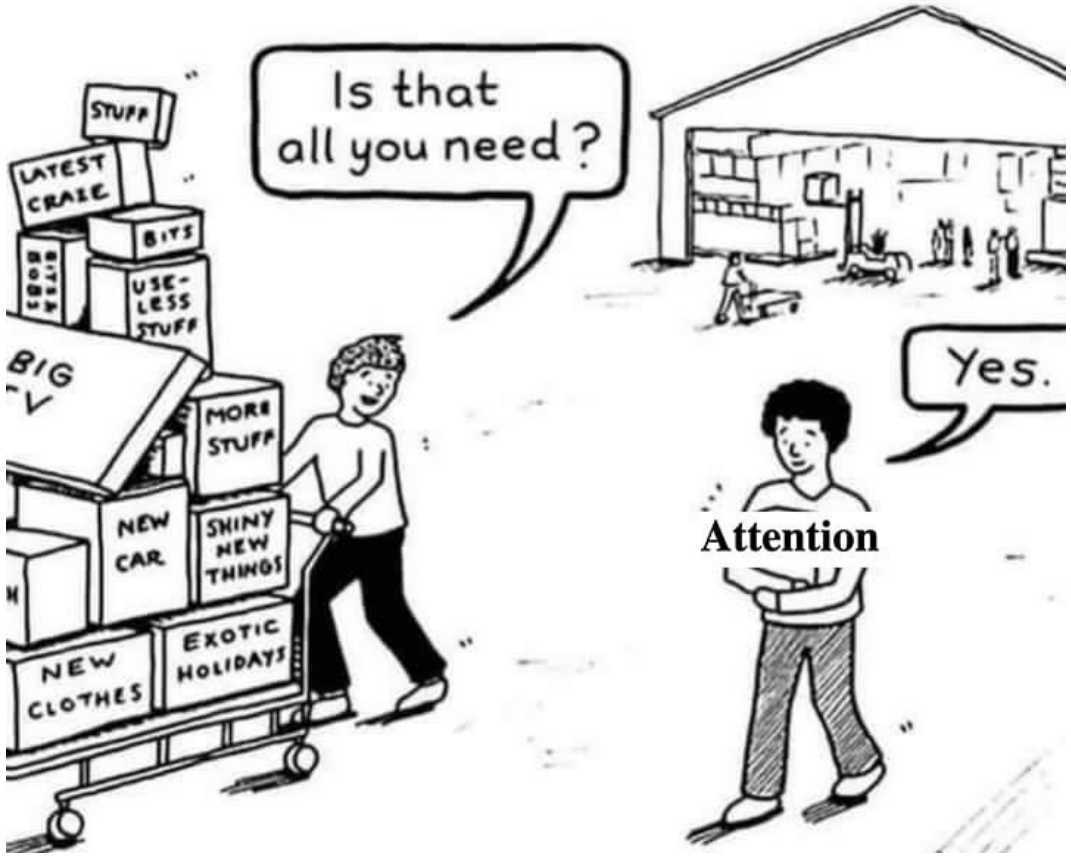
- A centralized learning and decentralized execution
- Training decentralized policies with Attention Mechanism in the Central Critics
- The key idea is to utilize attention in order to select relevant information for estimating critics.
- Performance of the proposed approach was evaluated with respect to:
  - the number of agents,
  - different configurations of rewards,
  - and the span of relevant observational information.
- Empirical results are promising
- Reduced input space
- Adaptability in a highly complicated and dynamic environment
- General purpose MARL algorithm with adaptive capability on (cooperative, competitive, and mixed environments)

Future Extensions

- Improve the scalability by sharing policies among agents, and performing attention on sub-groups of agent
- A more complicated environments with agents organized in clusters and sub-societies or even with overlapped or multiple interests

UNIVERSITY OF WATERLOO

Thanks for watching !



UNIVERSITY OF WATERLOO