# Knowledge Transfer with Attention for Multi-Agent Team Learning


## Abstract

Multi-agent reinforcement learning is a standard framework for modeling multi-agent interactions applied in real-world scenarios. Inspired by the mode of experience sharing in human groups, learning knowledge reusing between agents can potentially promote team learning performance, especially in multi-task environments. When all agents interact with the environment and learn simultaneously, how each independent agent selectively learns from other agents' behavior knowledge is a problem that we need to solve. This paper proposes a novel knowledge transfer framework in MARL, ATAC. We design two acting modes in ATAC, student mode and individual learning mode. Each agent in our approach trains a decentralized student actor-critic to determine its acting mode at each time step. When agents are unfamiliar with the environment, the shared attention mechanism in student mode effectively selects learning knowledge from other agents to decide agents' actions. ATAC outperforms state-of-the-art empirical evaluation results against the prior advising approaches. Our approach not only significantly improves team learning rate and global performance, but also is flexible and transferable to be applied in various multi-agent systems.


## Introduction

Knowledge transfer is a common method in the general learning process of a new task or in a new environment. The educational behavior in human society is an advanced form of knowledge transfer. In a multi-agent team, when an agent in an unfamiliar environment and learn to get more reward, the knowledge from other experienced agents is beneficial to the agent. Reinforcement Learning (Kaelbling, Littman, and Moore 1996), as a popular framework, has been employed in sequential decision-making problems. And Transfer Learning (Taylor and Stone 2009) always aims to improve learning through the learning experience from a related task, which also means knowledge reusing. In Reinforcement Learning framework, Transfer Learning is used to accelerate agents' learning via sharing knowledge, and the source of informative knowledge varies from experienced agents(experts) to human guidance. In this paper, we work on applying knowledge transfer method in agents' behavior transfer for multi-agent team.



In a system without the existence of experts, similar to the sharing of knowledge and experience in human society, if agents who learn the same or similar task in the shared environment simultaneously can share local learning experience and knowledge with each other, it would likely improve team-wide learning performance. Cooperative Multi-agent Reinforcement Learning (MARL) is a standard framework for modeling coordination problem in Multi-agent team, which has been applied in a series of meaningful problems such as multi-robot control (Matignon, Jeanpierre, and Mouaddib 2012) and team-game playing (Le et al. 2017). In Cooperative MARL, if an individual agent's learning is simply seen as independent RL with partial observations, the interactions between agents and the non-stationary environment will cause significant difficulties. To accelerate the team-wide learning efficiency and maximize the advantage of knowledge-transfer in the multi-agent domain, this paper targets the problem of optimizing knowledge-transfer between agents in Cooperative MARL under local constraints with a joint task or multiple tasks.

Our work is different from prior works that study inter-agent communication mechanisms in cooperative MARL (Lowe et al. 2017; Sukhbaatar, Fergus, and others 2016; Foerster et al. 2016). These works focus on communication and require a centralized critic training. Considering the scale of an agent team in the real scenarios, centralized training is challenging considering training stability. Jiang & Lu and Das et al. both proposed attention based communication in MARL (Jiang and Lu 2018; Das et al. 2018), but these approachs did not consider agents' behavior knowledge. We try to find a method to reuse behavior information in a agent team with less cost for the goal of improving team-wide learning. Our work concerns behavior knowledge reusing between cooperative agents with a teacher-student framework in order to improve team-wide performance in learning process. This paper proposes a new knowledge transfer method with high efficiency in MARL framework.

Regarding privacy constraints and communication cost in a multi-agent team, the main issues to be concerned are knowledge transfer decision, knowledge selection, and knowledge utilization. Invalid or confusing messages from other agents may cause a negative impact on agents' indi-

vidual learning. Also, in an environment where information interaction is frequent, there is a danger of risk contagion, resulting in poor performance of the entire team. For instance, the phenomenon of "over-advising" mentioned in previous studies, has increased team-wide learning instability, especially in the team with more than two agents. Hence, our new work supports a more robust and reliable knowledge-transfer in multi-agent system with no limit on number of agents.

In a novel teacher-student framework, we here state the based settings:

- All agents simultaneously learn in an environment and make decisions with interactions with the environment and other agents.
- There is no optimal expert (good-enough agent) in the multi-agent team in the initial state.
- For all agents, their local role of student or teacher is not fixed. An agent can use knowledge from other agents as a student and provide its own behavior knowledge as a teacher for students.
- All agents are learning for their local reward. But agents in the team are friendly to share knowledge. Our goal is to maximize the team-wide reward.

## Related Work

As a long-standing topic in the field of Reinforcement Learning, Multi-agent Reinforcement Learning (MARL) (Buşoniu, Babuška, and De Schutter 2010) track has a series of works in various ways to improve the performance and efficiency of team coordination. Deep Reinforcement Learning (Mnih et al. 2015) uses deep neural networks to approximate the policy and value functions of agents in the environment to address the problem of large-scale action-value space in RL. And Knowledge transfer method has been studied in several related fields including imitation learning, learning from demonstration (Le et al. 2017), and inverse reinforcement learning (Hadfield-Menell et al. 2016). Several works (Da Silva and Costa 2019) extended the source-target framework in transfer learning on reinforcement learning task. In the extensive teacher-student framework for transfer from expert policy to student policy, the student agent takes actions from the expert agent's advice. The extension works about this framework revolves around three questions of when to teach, what to teach, and how to learn from advice. Student-initiated approaches mainly concern with the students decision value, such as Ask Uncertain (Clouse 1997) and Ask Important (Amir et al. 2016). In teacher-initiated approaches, teachers decide when to teach based on the comparison between students and teachers learning experience, such as Importance Advising (Torrey and Taylor 2013), Early Correcting (Amir et al. 2016), and Correct Important (Torrey and Taylor 2013). Q-teaching (Fachantidis, Taylor, and Vlahavas 2018) designs teaching rewards to help teachers determine when to advise. Moreover, episode-sharing mechanism (Tan 1993) helps agents share individual successful episodes to accelerate learning.

However, all of the above works require an expert (all-knowing teacher) as the best agent to guide the learning of other agents. Zhan et al. (Zhan, Ammar, and others 2016) analyzed the case of negative transfer with the existence of a sub-optimal expert and present some theoretical results. At the same time, it also solves the advice selection problem from multiple teachers by the method of majority vote.

In this paper, we are concerned about the issue existing in multi-agent team learning framework with our settings. Recent works provide some solutions:

**AdHocVisit and AdHocTD** (Da Silva, Glatt, and Costa 2017) is an advisor-advisee framework without an expert in the multi-agent environment for agents learning simultaneously. The learning agents ask for advice and provide advising policy for other agents. Advisees use state visit counts to decide when to request advice and advisors evaluate their advice reliability through confidence metrics to decide when and what to provide for advisees. For advice selection, AdHocVisit and AdHocTD follow majority vote (Zhan, Ammar, and others 2016).

**LeCTR** (Omidshafiei et al. 2019) is a new teacher-student framework, which targets peer-to-peer teaching in order to solve advising-level problem. Each agent in system learns when and what to advise. In LeCTR, teacher-student (advising-level) policies are trained using the multi-agent actor-critic approach (MADDPG) (Lowe et al. 2017). LeCTR sets the advising-level policies as decentralized actor and uses a centralized action-value function as critic with advising-level reward. It is worth mentioning that LeCTR considers the communication cost in information exchange. LeCTR only works in two-player games.

## Preliminaries

In this work, we consider a decentralized multi-agent reinforcement learning scenario, multiple agents in cooperative team $\mathcal{G}$ simultaneously learn a joint task or multiple tasks. Our settings are formalized as a Decentralized POMDP (Dec-POMDP) in cooperative multi-agent system. All the agents in the environment receive local observation $o_t^i$ at each time step, and interact with the environment by executing local action. Agents then update their policy parameters according to the feedbacks (reward) given by the environment. The system is described as $(\mathcal{I}, S, A, T, R, \Omega, O, \gamma)$,

- $S$ is a set of states,
- $A$ is a set of joint actions, $\mathcal{A} = \times_i \mathcal{A}^i$,
- $T$ is a set of conditional transition probabilities $T(s'|s, a)$ between states,
- $R$: $S \times A \to R$ is the global reward function.
- $\Omega$ is a set of joint observations, $\boldsymbol{o} = \left\langle o^1, \ldots, o^n \right\rangle$
- $O$ is a set of conditional joint observation probability, $P(\boldsymbol{o}|s', \boldsymbol{a}) = \mathcal{O}(\boldsymbol{o}, s', \boldsymbol{a})$,
- $\gamma \in [0, 1]$ is the discount factor.

At each time period, the environment is in some state $s \in S$. Agents take a joint action $\mathcal{A} \in A$. Then each agent receives a local reward $r_t^i = \mathcal{R}^i\left(s_t, \boldsymbol{a}_t^i\right)$. The process repeats.

The goal is for all agents to take actions at each time step that maximize the global expected future discounted reward: $E\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$.

**Reinforcement Learning** (Kaelbling, Littman, and Moore 1996) is a standard framework to achieve the above goal of MDP (or POMDP). The process of value based reinforcement learning is to learn a policy which can maximize agents final reward. Through collecting experience from environment, agent updates its value function$v^{\pi}(s) = E_{\pi}[R_t|s_t = s]$and action value function$Q^{\pi}(s,a) = E_{\pi}[R_t|s_t = s, a_t = a]$.

**Deep Q Learning (DQN)** (Mnih et al. 2015) is a value-based Reinforcement Learning approach combined with deep neural networks, which learns the action value function (Q-value) in continuous environment using value function approximation. Q-Network updates by minimizing the loss:$L(\theta) = E[(r + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta)]$, and outputs the expected action value $Q(s, a; \theta)$.

**Deterministic Policy Gradient (DPG)** (Silver et al. 2014) is a policy-based Reinforcement Learning approach as an extension of policy gradient (PG) (Sutton et al. 2000), which optimize policy by update policy parameters $\theta$ along the gradient direction $\nabla_{\theta} J(\theta), \nabla_{\theta} J(\theta) = E_{s\sim p^{\pi}, a\sim\pi_{\theta}}[\nabla_{\theta} \log \pi_{\theta}(a|s) Q^{\pi}(s, a; \theta)]$.
Then, DPG is extended to **Deep Deterministic Policy Gradient (DDPG)**(Lillicrap et al. 2015), with $\nabla_{\theta} J(\theta) = E_{s\sim\mathcal{D}}[.\nabla_{\theta}\mu_{\theta}(a|s)\nabla_{a} Q(s, a; \mu)|_{a=\mu_{\theta}(s)}]$ using deterministic policy when the variance of action probability distribution approaches 0.

**Attention Mechanism** is one of the most influential models which can be broadly interpreted as a vector of importance weights. It has recently been applied in reinforcement learning (Oh et al. 2016; Iqbal and Sha 2018).

## Overview

Our work targets knowledge reusing between agents in cooperative MARL, where all the agents in the environment are not good enough. In this section, we provide a story-level overview of our main idea. The overview of our motivating scenario is presented in Figure 1.

Considering our settings, all the agents act in the environment and update their self policy parameters with local rewards from the environment. The actions executed by the agents is dictated by their self policy parameters. Now, we explore a novel knowledge transfer framework, ATAC. In our framework, each agent has two acting modes, student mode and self-learning mode. Before each agent takes action, agents' student actor-critic modules decide agents' acting modes with agents' hidden states. It is not an ad-hoc design for specific domain. Agents should learn to ideally learn from other agents.

In self-learning mode (individual learning), agents take action based on their independent learned behavioral knowledge, which is represented as agents' behavior policy parameters. In this mode, agents' actions are independent of other agents' behavioral knowledge. All agents in self-learning mode is trained in an individual end-to-end manner using Deep Deterministic Policy Gradient (Lillicrap et al. 2015) algorithm with an actor network and a critic network.

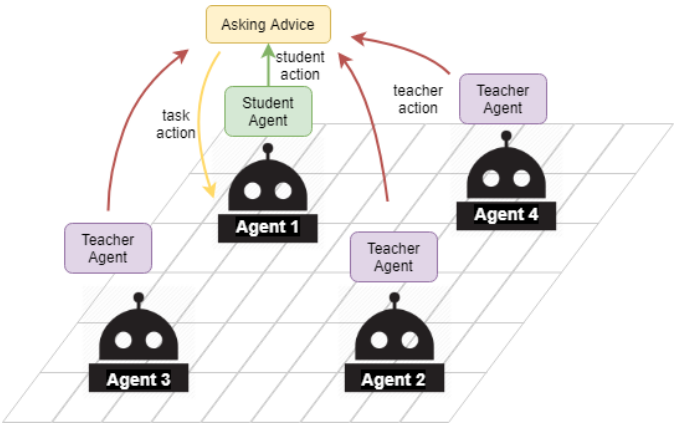



Figure 1: Overview

In student mode, if there are more than two agents in the system, a student agent receives multiple advice from other agents. We now refer to a new problem, teacher selecting, because not all teachers' knowledge is useful for the student agent. However, existing frameworks try to dodge this problem and have key limitation in the scenario where the number of agents is large, which also causes difficult in model transfer. We apply a soft attention mechanism in our work to select teachers' knowledge. The Attention Teacher Selector solves the problem by selecting contextual information in teachers' learning information and computing weights of teachers' knowledge. Considering from a different angle, our attentional module selectively transform the learning information from teachers with the target of solving student's problem. Our attentional selecting approach is effective both in multi-task scenarios and joint task scenarios.

We make a few assumptions on agents' identities to support our framework. When an agent chooses student mode in $t$ time step, other agents in the environment automatically become its teachers and provide their behavior policy and learning knowledge to the student agent. At the same time step, the agent in student mode can also be a teacher of the other student agents because agents' learning experience is different. When an agent is unfamiliar with its hidden state, it may have confidence in the other states and its behavior knowledge can help the other student agents. Our teacher selector module is designed to determine the appropriate teachers and transform teachers' local behavior knowledge into student's advising action. Moreover, our attention mechanism quantifies the reliability of teachers, so our scenarios do not need good-enough agents (experts).

In a high-level summary, because of agents' different learning experience, agents in a cooperative team are good at different tasks or different parts of a joint task. Knowledge Transfer is a framework to help agents solve unfamiliar task with experienced agents' learning knowledge.

ATAC's training and architecture details are presented in the next section.

## Attention based Knowledge-Transfer Architecture

This section introduces our knowledge transfer approach with more design details of the whole structure and all training protocols in our framework. In our framework, each agent has two actor-critic model and an attention mechanism to support two acting modes.

### Acting Mode

Different from original individual agent learning, after receiving observation from the environment, agents in our framework need to choose their action mode before taking action.

At $t$ time step, agent $i$ reprocesses the observation from environment with a hidden LSTM (or RNN) unit, which integrates information (observation) from $i$'s observation history. The LSTM unit $l^i$ outputs agent's observation encoding $m_t^i$, which represents agent's hidden state.

$$l^i : (o_{t-m}^i, a_{t-m}^i, ..., o_t^i) \rightarrow m_t^i \tag{1}$$

Next, based on this step's memorized observation, $m_t^i$, agent $i$'s student actor network takes this step's memorized observation, $m_t^i$, as input and output agent $i$'s acting mode. Considering the efficiency of information exchange and communication cost, student actor is used to deciding agent $i$' confidence in $t$ time step. If $i$ has enough confidence with $m_t^i$, student actor chooses self-learning mode. Conversely, student actor chooses student mode and send advice request to other agents.

Student actor and student critic is a deep deterministic policy gradient model. The action of student actor $w$ is the probability of choosing student mode.

Student actor and student critic represents the acting mode choosing model which determine whether agent $i$ become a student and ask teacher agents for advice. We train student actor-critic using a student reward $\widetilde{r}_t^i$.

$$\widetilde{r}_t^i = V\left(m_t^i; \theta_t'^i\right) - V\left(m_t^i; \theta_t^i\right) \tag{2}$$

$\theta_t'^i$ and $\theta_t^i$, are agent $i$ policy parameters in student mode and self-learning mode. The student reward measures gain in agent learning performance from student mode. The sharing of student actor-critic network parameters allows this module learning effectively in environment and easily extending to other settings.

In our experiments, student actor-critic is trained with the trained Attention Teacher Selector. Student critic is updated to minimized the student loss function:

$$\begin{aligned} \mathcal{L}\left(\theta^{\tilde{Q}}\right) &= E_{m,w,\tilde{r},m'}\left[\left(\tilde{y} - \widetilde{Q}\left(m, w|\theta^{\tilde{Q}}\right)^2\right)\right], \\ \tilde{y} &= \tilde{r} + \gamma \widetilde{Q}\left(m', w'|\theta^{\widetilde{Q}'}\right)\Big|_{w'=\widetilde{\mu}'\left(m'|\theta^{w'}\right)} \end{aligned} \tag{3}$$

Student policy network is updated by ascent with the following gradient:

$$\nabla_{\theta^{\widetilde{\mu}}} J = E_{m,w\sim\widetilde{\mathcal{R}}}\left[\nabla_w \widetilde{Q}\left(m, w|\theta^{\widetilde{Q}}\right)\Big|_{w=\widetilde{\mu}(m)} \nabla_{\theta^{\widetilde{\mu}}} \widetilde{\mu}\left(m|\theta^{\widetilde{\mu}}\right)\right] \tag{4}$$

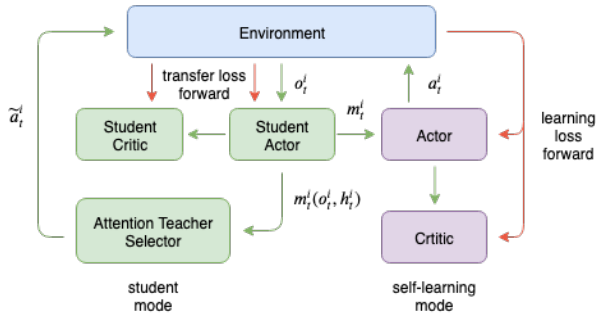



Figure 2: ATAC Architecture

Here, $\widetilde{\mu}$ is agent's student policy, which is parameterized by $\widetilde{\theta}$

### Student Mode

**Attention Teacher Selector** Inspired by the similarity between source task and target task in transfer learning, we use attention mechanism to evaluate the task similarity between student and teachers and teachers' confidence of student's state. Therefore, each agent's Attention Teacher Selector in student mode is used to select advice from teachers based on their similarity and confidence. The main idea behind our knowledge transfer approach is to learn the student mode by selectively paying attention to policy advice from other agents in the cooperative team. Figure 3 illustrates the main components of our attention mechanism.

We now describe the Attention Teacher Selector mechanism in agent student mode. The Attention Teacher Selector (ATS) is a soft attention mechanism as a differentiable query-key-value model (Graves, Wayne, and Danihelka 2014; Oh et al. 2016). After the student actor of student agent $i \in \mathcal{G}$ compute the memorized observation at $t$ time step and choose student mode, ATS receives the encoding hidden state $m_t^i$. Then, from other agents in the team as teacher agents, ATS receives the teachers' encoding learning history $h_t^j = l^j(o_1^j, a_1^j, ..., o_t^j)$ and encoding policy parameter $\theta^j$.

Now, ATS computes a query $Q_t^i = W_Q m_t^i$ as student query vector, a key $K_t^j = W_K h_t^j$ as teacher key vector, and a value $V_t^j = W_V \theta^j$ as teacher policy value vector, where $W_K, W_Q$ and $W_V$ are attentional learning parameters. After ATS receives all key-value $(K^j, V^j)$ from all of teachers $j \in \mathcal{G}$, the attention weight $\alpha^{ij}$ is assigned by passing key vector from teacher and query vector from student into a softmax:

$$\alpha^{ij} = softmax\left(\frac{Q^i K^j}{\sqrt{D_K}}\right) \tag{5}$$

Here, $D_K$ is the dimension of teacher $j$'s key vector, which is used to resolve vanishing gradients (Vaswani et al. 2017). The final policy advice is a weight sum with a linear transformation:

$$v^i = W_T \sum_{j \neq i} \alpha^{ij} V^j \tag{6}$$

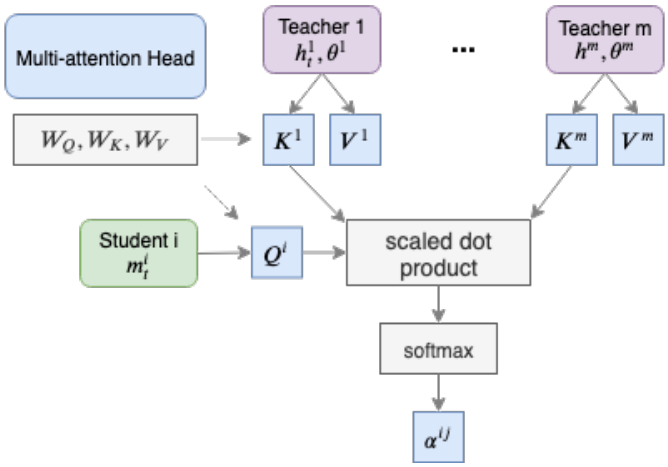



Figure 3: Attention based Knowledge Selection

Here, $W_T$ is a learning parameter for policy parameter decoding.

Behind the single attention head, we use a simple multi-attention head with a set of learning parameters $(W_K, W_Q, W_V)$ to aggregate all advice from different representation subplaces. Besides, attention head dropout is applied to improve the effectivity of our attention mechanism.

Finally, student agent $i$ obtains its action at this time with policy parameters from Attention Teacher Selector:

$$\widetilde{a}_t^i = v^i(m_t^i) \tag{7}$$

In our experiments, the attention parameters $(W_K, W_Q, W_V)$ are shared across all agents, because knowledge transfer process is similar in all pairs of student-teacher, but different observations introduce different teacher weight vector. This setting encourages our approach to learn more efficient and make our model easy to be extended in different settings, such as larger number of agents or a different environment.

In this work, we consider scenarios where other agents' learning experience is useful to a student agent. Feeding student's observation information and teacher's learning experience into our attention mechanism, which helps to select action with other agents' behavioral policy for the student agent. This module is an end-to-end knowledge transfer method without any decentralized learning parameter sharing.

## Self-learning Mode

If agent $i$'s student actor chooses self-learning mode, the student actor sends $i$'s encoding hidden state $m_t^i$ to the actor network. In self-learning mode, agents learn as a common individual agent. Each agent's policy in self-learning mode is independently trained by DDPG (Lillicrap et al. 2015) algorithm.

In games with discrete action space, in self-learning mode, agents' actor-critic networks can be replaced by an action-value network. Agents in self-learning mode take action based on action values (Q-values) (Mnih et al. 2015). Our framework is adapted for both continuous action space and discrete action space.

Agents critic network is updated by TD error:

$$\mathcal{L}\left(\theta^Q\right) = E_{m,a,r,m'}\left[\left(y - Q\left(m, a|\theta^Q\right)^2\right]\right.,$$
$$y = r + \gamma Q\left(m', a'|\theta^{Q'}\right)\Big|_{a'=\mu'\left(m'|\theta^{\mu'}\right)} \tag{8}$$

The policy gradient of agents actor network can be derived as:

$$\nabla_{\theta^\mu} J = E_{m,a\sim\mathcal{R}}\left[\nabla_a Q\left(m, a|\theta^Q\right)\big|_{a=\mu(m)} \nabla_{\theta^\mu}\mu\left(m|\theta^\mu\right)\right. \tag{9}$$

## Empirical evaluations

We construct three environments to test the team-wide performance of ATAC and existing advising methods in multi-tasks and joint task scenarios. Also, we compare the scalability of all the approaches with the increase of number of agents. Additionally, the transferability of ATAC is evaluated in different environments.

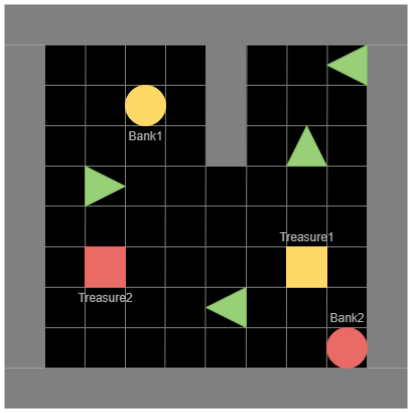
### Setup

Empirical evaluations are performed on three cooperative multi-agent environments: Grid Treasure Collection, Moving Treasure Collection, Predator-Prey, and. We implement Grid Treasure Collection, a standard grid world environment. Moving Treasure Collection and Predator-Prey are implemented based on Multi-Agent Particle Environment (Mordatch and Abbeel 2018; Lowe et al. 2017) where agents move around in a 2D space and involve interaction between agents. We briefly describe the three environments below:

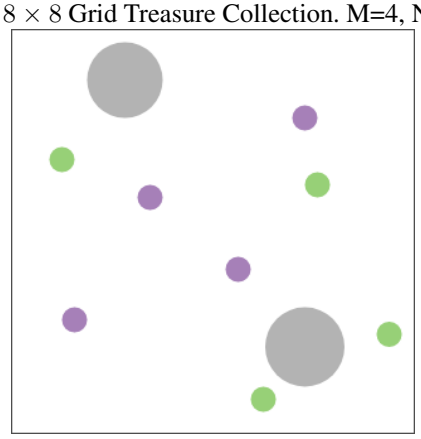
**Grid Treasure Collection** : There are M agents and N treasure grids and N treasure banks in the grid maze(see Figure 4(a)). Each treasure is corresponding to a treasure bank. When agents collect treasures in treasure grids, agents get small rewards, and then return treasures to its corresponding bank, agents receives big rewards. Each treasure grid has M treasures and agents can only obtain one treasure from each treasure grid. The ability of agents to carry treasures is not limited. But when agents return wrong treasures to treasure banks, agents receive big penalties.

**Moving Treasure Collection** : The game rule in Moving Treasure Collection is similar to the above game, but in this environment, agents are green and treasures and treasure banks are moving randomly. And all objects are moving in a 2D open ground. Obstacles (large black circles) block the way in the environment.

**Cooperative Navigation** : There are M agents (green) and M landmarks (purple) in this environment. Agents are rewarded based on how far any agent is from each landmark. Agents are required to position themselves covering all the landmarks. When an agent covers a landmark, it gets a local reward. Obstacles (large black circles) block the way.




(a) $8 \times 8$ Grid Treasure Collection. M=4, N=2



(b) Cooperative Navigation. M=4

Figure 4: Environments

To simplify our approach training, we set discrete action spaces in all the environments, allowing agents to move up, down, left, right, or stay. All agents receive partial observation at each step, and get local feedback from environments. We plan to evaluate our approach in both multi-task environment (Treasure Collection) and joint task environment

## Baselines

We compare our approach, ATAC with implementation of Individual Deep Q-Learning (IQN) (Mnih et al. 2015), AdHocTD (Da Silva, Glatt, and Costa 2017) and LeCTR (Omidshafiei et al. 2019). IQN, as a baseline, is a reinforcement learning algorithm for single agent, which is trained independently for each agent with partial observation in our environments. AdHocTD, LeCTR and our ATAC methods rely on action advising without any other information communication, which means each agent is unaware of the observations and rewards of other agents in the environment. All the detailed parametrization and training procedures are presented in Supplemental Material.

We evaluate all the models on two different agent teams , with M=4 agents and M=8 agents. When the number of agents is 8, the number of treasures and banks increase to 4. Average step length per episode (Treasure Collection games' max episode length = 1000), success rate of covering all

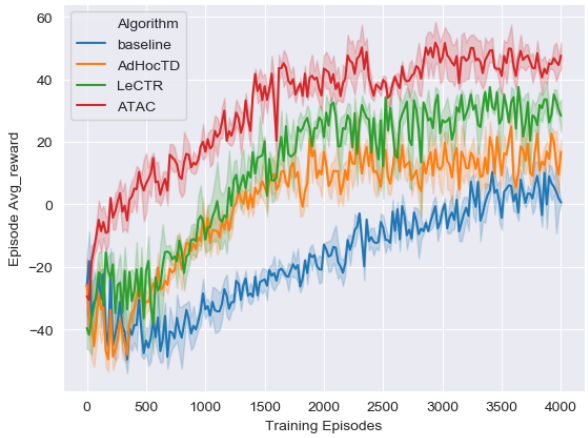



Figure 5: Average reward per episode in Grid Treasure Collection

landmarks in Cooperative Navigation, and average sum of rewards per episode are three indicators used to evaluate performance of all models.

## Results and Analysis

Figure 5 displays average team rewards per episode in Grid Treasure Collection. A thorough evaluation result is summarized in Table 1.

In 4-agent Grid Treasure Collection (GTC) and Moving Treasure Collection (MTC) game, ATAC outperforms all models with a higher average reward after convergence. In Comparison of all approachs in average episode length (Ave_step), ATAC has no obvious advantage compared with LeCTR, but ATAC greatly improves average team reward per episode (Avg_reward). Attention mechanism performs well and significantly fights out agents with useful experience (have successfully collected treasures), which can correctly guide unfamiliar agents to take more beneficial actions. AdHocTD selects teachers using agents' number of visiting the current state. As a result, the phenomenon of over-advising appears in AdHocTD after trained over 2000 episodes, which means that students may take advice from teachers with more bad experience. It contributes to AdHocTD worse final reward performance than LeCTR and ATAC. LeCTR learns how to teach by centralized training and decentralized executing, causing the learning instability in early episodes. Observing the results, ATAC successfully avoids this problem by training a decentralized student actor-critic network for the decision of acting mode.

In 4-agent Cooperative Navigation, ATAC surprisingly performs longer episode length and lower success rate than LeCTR. Attention mechanism in ATAC tends to imitate successful action from other agents, which helps student agent gain more local rewards. But it might cause a bad impact on team coordination in a joint task. For example, agents tend to cover the same landmarks with successful experience, but the team needs to cover all the landmarks. ATAC is hard to

Table 1: Evaluation Results. Results are computed using a Student's t-test with significance level $\alpha = 0.05$.

| TASK | APPROACH | M=4 Avg_step | Success % | Avg_reward | M=8 Avg_step | Success % | Avg_reward |
|---|---|---|---|---|---|---|---|
| Grid Treasure Collection | Baseline | $1000 \pm 0$ | - | $0.78 \pm 1.43$ | $2000 \pm 0$ | - | $-1.20 \pm 0.04$ |
| Grid Treasure Collection | AdHocTD | $876 \pm 15$ | - | $18.76 \pm 1.14$ | $1821 \pm 29$ | - | $12.56 \pm 0.34$ |
| Grid Treasure Collection | LeCTR | $812 \pm 39$ | - | $29.87 \pm 1.23$ | $1836 \pm 23$ | - | $11.68 \pm 2.24$ |
| Grid Treasure Collection | ATAC | $762 \pm 20$ | - | $45.65 \pm 2.32$ | $1157 \pm 37$ | - | $20.34 \pm 1.49$ |
| Moving Treasure Collection | Baseline | $1000 \pm 0$ | - | $-1.84 \pm 0.66$ | $2000 \pm 0$ | - | $-3.59 \pm 0.26$ |
| Moving Treasure Collection | AdHocTD | $1000 \pm 0$ | - | $8.22 \pm 1.79$ | $1987 \pm 67$ | - | $-3.34 \pm 0.21$ |
| Moving Treasure Collection | LeCTR | $877 \pm 46$ | - | $20.76 \pm 0.24$ | $1889 \pm 48$ | - | $-2.79 \pm 2.12$ |
| Moving Treasure Collection | ATAC | $854 \pm 23$ | - | $33.98 \pm 2.13$ | $1239 \pm 32$ | - | $-0.32 \pm 0.43$ |
| Cooperative Navigation | Baseline | $397 \pm 25$ | $52.2 \pm 2.0$ | $-1.78 \pm 0.03$ | $782 \pm 4$ | $32.7 \pm 4.7$ | $-4.68 \pm 0.05$ |
| Cooperative Navigation | AdHocTD | $320 \pm 28$ | $69.0 \pm 3.7$ | $-1.23 \pm 0.27$ | $547 \pm 29$ | $42.6 \pm 2.0$ | $-3.50 \pm 0.08$ |
| Cooperative Navigation | LeCTR | $278 \pm 25$ | $81.4 \pm 3.2$ | $-1.29 \pm 0.39$ | $672 \pm 14$ | $40.7 \pm 1.9$ | $-3.36 \pm 0.70$ |
| Cooperative Navigation | ATAC | $289 \pm 18$ | $80.2 \pm 3.2$ | $-0.45 \pm 0.07$ | $498 \pm 10$ | $59.2 \pm 1.3$ | $-2.67 \pm 0.02$ |

learn the team cooperative policy and gain more cooperative rewards. However, LeCTR uses a centralized critic network to calculate advising value concatenating all agents' observation, which exactly improves cooperation performance between agents. This result gives us a future direction that we should try to modify attention tendency in global reward for cooperative tasks.

**Scalability** When the number of agents is 8 in all the environments, because of more difficulty in selecting advice from a larger agent team, AdHocTD's advising probabilistic mechanism can not handle the problem in more information selecting, resulting in sub-optimal rewards. LeCTR, as an algorithm originally supporting two-player games, has low performance in a larger size of agent team. We suspect it due to LeCTR's centralized control of advising. With more agents in the environments, LeCTR's centralized advising-level critic performs poorly with limited training time. As expected, ATAC's attention selector effectively filtering knowledge from more teachers and maintain student mode's accuracy, so our approach has a larger advantage with the increase in number of agents. Our experimental results confirm our inference.

In summary, ATAC performs much better than all approaches and has a distinct advantage in average team-wide rewards in complex multi-task environments as expected. We also report ATAC's disadvantage in the joint-task scenario. ATAC scales better when agents are added in all the experiments, which shows that sharing attention mechanism is useful for information selecting in a large multi-agent team.

**Transferablility** Analysing the above results, the behavior knowledge transfer becomes more difficult when number of agents increases. We design a new experiment to explore our model parameters transfer performance. We test our approach on two environments with a different number of agents. First, Agents are trained in M=6 Treasure Collection environments. Then, the trained parameters of agents' attention mechanisms are transferred to an agent team of 8 agents. With trained parameters of Attention Teacher Selector, the team of 8 agents starts their student actor-critic network and self-learning network training in a new environment. We compare the transfer experimental results with above results in M=8 environments. The compared results are summarized in Table 2, which shows that our approach is able to transfer efficiently in different environments. The transfer results are also better than other approaches' results. Our shared attention selector can effectively solve new tasks based on related experience.

Table 2: Transfer Evaluations

| TASK | M=8 Avg_step | Avg_reward | Transfer Avg_step | Avg_reward |
|---|---|---|---|---|
| Grid | $1157 \pm 37$ | $20.34 \pm 1.49$ | $1230 \pm 9$ | $18.85 \pm 0.98$ |
| Moving | $1239 \pm 32$ | $3.32 \pm 0.43$ | $1389 \pm 24$ | $3.02 \pm 0.09$ |

## Conclusions and Future Work

We introduce a knowledge-transfer framework, ATAC for decentralized multi-agent reinforcement learning. Our key idea is designing two acting mode for agents and using a shared attention mechanism to select behavior knowledge from other agents to accelerate student agent learning. We empirically evaluate our proposed approach against all state-of-the-art advising or teaching methods in multi-agent environments. Results in experiments of scaling the number of agents and model transfer are also shown. Extending knowledge transfer in joint task learning and more complicated multi-agent systems is our future research direction.

## References


Amir, O.; Kamar, E.; Kolobov, A.; and Grosz, B. 2016. Interactive teaching strategies for agent training.

Buşoniu, L.; Babuška, R.; and De Schutter, B. 2010. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*. Springer. 183–221.

Clouse, J. A. 1997. On integrating apprentice learning and reinforcement learning.

Da Silva, F. L., and Costa, A. H. R. 2019. A survey on transfer learning for multiagent reinforcement learning systems. *Journal of Artificial Intelligence Research* 64:645–703.

Da Silva, F. L.; Glatt, R.; and Costa, A. H. R. 2017. Simultaneously learning and advising in multiagent reinforcement learning. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 1100–1108. International Foundation for Autonomous Agents and Multiagent Systems.

Das, A.; Gervet, T.; Romoff, J.; Batra, D.; Parikh, D.; Rabbat, M.; and Pineau, J. 2018. Tarmac: Targeted multi-agent communication. *arXiv preprint arXiv:1810.11187*.

Fachantidis, A.; Taylor, M.; and Vlahavas, I. 2018. Learning to teach reinforcement learning agents. *Machine Learning and Knowledge Extraction* 1(1):21–42.

Foerster, J.; Assael, I. A.; de Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2137–2145.

Graves, A.; Wayne, G.; and Danihelka, I. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Hadfield-Menell, D.; Russell, S. J.; Abbeel, P.; and Dragan, A. 2016. Cooperative inverse reinforcement learning. In *Advances in neural information processing systems*, 3909–3917.

Iqbal, S., and Sha, F. 2018. Actor-attention-critic for multi-agent reinforcement learning. *arXiv preprint arXiv:1810.02912*.

Jiang, J., and Lu, Z. 2018. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, 7254–7264.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research* 4:237–285.

Le, H. M.; Yue, Y.; Carr, P.; and Lucey, P. 2017. Coordinated multi-agent imitation learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1995–2003. JMLR. org.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, O. P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 6379–6390.

Matignon, L.; Jeanpierre, L.; and Mouaddib, A.-I. 2012. Coordinated multi-robot exploration under communication constraints using decentralized markov decision processes. In *Twenty-sixth AAAI conference on artificial intelligence*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529.

Mordatch, I., and Abbeel, P. 2018. Emergence of grounded compositional language in multi-agent populations. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Oh, J.; Chockalingam, V.; Singh, S.; and Lee, H. 2016. Control of memory, active perception, and action in minecraft. *arXiv preprint arXiv:1605.09128*.

Omidshafiei, S.; Kim, D.-K.; Liu, M.; Tesauro, G.; Riemer, M.; Amato, C.; Campbell, M.; and How, J. P. 2019. Learning to teach in cooperative multiagent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 6128–6136.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms.

Sukhbaatar, S.; Fergus, R.; et al. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, 2244–2252.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 1057–1063.

Tan, M. 1993. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, 330–337.

Taylor, M. E., and Stone, P. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10(Jul):1633–1685.

Torrey, L., and Taylor, M. 2013. Teaching on a budget: Agents advising agents in reinforcement learning. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, 1053–1060. International Foundation for Autonomous Agents and Multiagent Systems.

Zhan, Y.; Ammar, H. B.; et al. 2016. Theoretically-grounded policy advice from multiple teachers in reinforcement learning settings with applications to negative transfer. *arXiv preprint arXiv:1604.03986*.