Cautiously-Optimistic Knowledge Sharing for Cooperative Multi-Agent Reinforcement Learning

Yanwen Ba[1], Xuan Liu[1*], Xinning Chen[1], Hao Wang[1], Yang Xu[1], Kenli Li[1], Shigeng Zhang[2]

[1]College of Computer Science and Electronic Engineering, Hunan University, Changsha, China
[2]School of Computer Science and Engineering, Central South University, Changsha, China
{yanwenba, xuan_liu, chenxinning, wonhow, xuyangcs, lkl}@hnu.edu.cn, sgzhang@csu.edu.cn

Abstract

While decentralized training is attractive in multi-agent reinforcement learning (MARL) for its excellent scalability and robustness, its inherent coordination challenges in collaborative tasks result in numerous interactions for agents to learn good policies. To alleviate this problem, action advising methods make experienced agents share their knowledge about what to do, while less experienced agents strictly follow the received advice. However, this method of sharing and utilizing knowledge may hinder the team's exploration of better states, as agents can be unduly influenced by suboptimal or even adverse advice, especially in the early stages of learning. Inspired by the fact that humans can learn not only from the success but also from the failure of others, this paper proposes a novel knowledge sharing framework called *Cautiously-Optimistic kNowledge Sharing* (CONS). CONS enables each agent to share both positive and negative knowledge and cautiously assimilate knowledge from others, thereby enhancing the efficiency of early-stage exploration and the agents' robustness to adverse advice. Moreover, considering the continuous improvement of policies, agents value negative knowledge more in the early stages of learning and shift their focus to positive knowledge in the later stages. Our framework can be easily integrated into existing Q-learning based methods without introducing additional training costs. We evaluate CONS in several challenging multi-agent tasks and find it excels in environments where optimal behavioral patterns are difficult to discover, surpassing the baselines in terms of convergence rate and final performance.

Introduction

Cooperative multi-agent reinforcement learning (MARL) has attracted much attention in recent years due to its ability to solve complex real-world problems, such as multi-robot control (Willemsen, Coppola, and de Croon 2021) and traffic scheduling (Zhang et al. 2019). Most of the currently proposed MARL algorithms follow the paradigm of *centralized training and decentralized execution* (CTDE) (Lowe et al. 2017; Rashid et al. 2018; Son et al. 2019; Peng et al. 2021), where a centralized critic collects information from all agents during the training phase to learn decentralized agent policies. However, this paradigm struggles with the huge joint state-action spaces that grow exponentially with the number of agents, and the ideal conditions for deploying centralized critics are often lacking in reality. In contrast, the paradigm of *decentralized training and decentralized execution* (DTDE) (Tan 1993; Tampuu et al. 2017) is more scalable and robust, and more adaptable to harsh real-world conditions, as it does not require a centralized critic.

While the DTDE paradigm has many advantages, it inevitably faces coordination difficulties in collaborative tasks due to the lack of an explicit centralized coordinator and partial observability. Agent teams that follow the DTDE paradigm often need to spend a lot of time exploring to develop good strategies. To alleviate this problem, some communication-based MARL methods focus on allowing proper exchange of local information about observations among agents while following the DTDE paradigm. This information can be regarded as the perceptual-level knowledge, allowing agents to make decisions from a broader perspective. It is usually fed directly into the receiver's network (Jiang and Lu 2018; Singh, Jain, and Sukhbaatar 2019; Ding, Huang, and Lu 2020), which expands the local policy spaces of agents and significantly increases the training burden. Furthermore, high-dimensional information in the network has uncertainty and uninterpretability.

Unlike general communication-based methods, action advising shares policy-level rather than perceptual-level knowledge in a more direct and explainable manner, where more experienced agents advise less experienced agents on the best actions and the suggested actions will be executed directly by the advisees. This scheme speeds up coordination among agents and resembles the common way humans communicate—we often provide helpful advice based on our knowledge and beliefs rather than simply providing our own information. However, agents may give many suboptimal or even poor suggestions, especially in the early stages of learning. These inappropriate suggestions not only waste precious communication budget, but also hinder the team from further exploring better states, diminishing the advantages of action advising in complex environments that require extensive exploration. Furthermore, the suggestions provided are also one-sided.

To make knowledge sharing more comprehensive and mitigate the negative impacts of inappropriate knowledge, we propose a novel knowledge sharing framework

*Corresponding Author.

called ***C**autiously-**O**ptimistic k**N**owledge **S**haring* (CONS) [1]. Specifically, inspired by the fact that humans can learn not only from the success of others but also from their failure, CONS agents share experiences of both failure and success (i.e., negative and positive knowledge) simultaneously, whereas previous works only involve the latter. Unlike agents in other action advising methods, CONS agents do not simply adopt the suggested actions after receiving knowledge from others. Instead, they incorporate the received knowledge by softly updating their action probabilities, thereby forming new policy. Subsequently, CONS agents conduct targeted exploration based on their new policy and their confidence. They are optimistic due to the belief that sharing and learning from negative knowledge is beneficial; at the same time, they are cautious due to not blindly following the acquired knowledge. It should to be emphasized that CONS only affects the action selection of the agent, not the training process of the underlying algorithm. Therefore, CONS can be easily integrated into existing Q-learning based methods without introducing additional training overhead. Experimental results show that CONS performs well in environments where the optimal behavioral patterns are harder to discover compared to suboptimal ones, surpassing the baselines in terms of convergence rate and final performance.

Related Work

Decentralized Training and Decentralized Execution (DTDE)

With excellent scalability and robustness, DTDE is a promising paradigm for using MARL to solve real-world problems. There is no centralized critic in DTDE paradigm so agents only use local information to make decisions during both training and execution. The most straightforward way to use the DTDE paradigm in MARL is to make each agent run a single-agent reinforcement learning algorithm independently (Tan 1993), such as independent Q-learning (IQL) (Tampuu et al. 2017) and independent PPO (IPPO) (de Witt et al. 2020). In addition to the aforementioned intuitive algorithms, some works also focus on other aspects of decentralized training. Hysteretic Q-learning (Matignon, Laurent, and Le Fort-Piat 2007) and lenient Q-learning (Palmer et al. 2018) let agents be optimistic and appropriately ignore value penalties, thereby promoting team cooperation. Ideal independent Q-learning (I2Q) (Jiang and Lu 2022) alleviates environmental non-stationarity by having each agent model an ideal transition function and perform independent Q-learning on it. Please note that this paper focuses on knowledge sharing rather than decentralized algorithms, so we directly use deep recurrent Q-network (DRQN) (Hausknecht and Stone 2015) as an instance of the underlying algorithm to implement CONS.

Knowledge Sharing

Knowledge sharing speeds up learning, fosters coordination among agents, and has various forms.

[1]We provide open-source implementations of CONS in https://github.com/byw0919/CONS

Communication. Communication-based methods usually share local observations (or observation embeddings) of agents. ATOC (Jiang and Lu 2018) agent uses an attention unit to decide whether to communicate or not, and if so, selects several collaborators in its observable field to communicate. IC3Net (Singh, Jain, and Sukhbaatar 2019) extends CommNet (Sukhbaatar, Szlam, and Fergus 2016) by using the gating mechanism to determine whether to broadcast messages on a common channel. I2C (Ding, Huang, and Lu 2020) agent learns prior knowledge for agent-agent communication through causal effect to capture the necessity of communication. GA-Comm (Liu et al. 2020) employs attention to decide which pair of agents can communicate, thereby learning a shared undirected communication graph. However, these methods all expand local policy spaces of agents and make learning more difficult.

Experience Sharing. Agents in experience sharing methods like SEAC and SEQL (Christianos, Schäfer, and Albrecht 2020) acquire the trajectories of others as off-policy data to train their own networks, without increasing learning complexity. However, the strong assumption that agents have access to others' private trajectory data and the huge amount of information exchanged make it less attractive.

Advising Mechanism. Advising mechanism shares policy-level knowledge, where less experienced agents can take good actions without making decisions themselves. Unfortunately, many methods based on advising mechanism assume the teacher has a well-trained policy (Ilhan, Gow, and Perez Liebana 2021; Anand et al. 2021; Guo et al. 2023), or have a centralized information structure (Omidshafiei et al. 2019; Kim et al. 2020; Gupta et al. 2021), or increase training overhead (Ilhan, Gow, and Perez Liebana 2021), or are limited to two agents (Omidshafiei et al. 2019; Kim et al. 2020). Two works that are similar to ours are AdHocTD (da Silva, Glatt, and Costa 2017) and PSAF (Zhu et al. 2021), but both are based on tabular Q-learning and lack robustness to suboptimal advice. Unlike the mentioned knowledge sharing methods, CONS agents modify their action probabilities according to the received policy-level knowledge, and then explore in a targeted manner based on the modified probabilities, avoiding increased training overhead and being robust to suboptimal advice. Moreover, CONS agents learn from scratch and can act as both teachers and students during the learning process.

Background

Problem Formulation

A general cooperative multi-agent reinforcement learning problem can be typically modeled as a partially observable Markov games for n agents (Littman 1994), which is defined by the tuple $(\mathcal{N}, \mathcal{S}, \mathcal{O}, \mathcal{A}, \Omega, \mathcal{P}, \{\mathcal{R}^i\}_{i\in\mathcal{N}}, \gamma)$. Here $\mathcal{N} = \{1, \ldots, n\}$ is the set of agents; $\mathcal{S}$ is the state space; $\mathcal{O} = O^1 \times \ldots \times O^n$ is the joint observation space; $\mathcal{A} = A^1 \times \ldots \times A^n$ is the joint action space. At each time step, each agent $i \in \mathcal{N}$ can only access local observations $o^i \in O^i$ drawn from the observation function $\Omega(s, i)$ where $s \in \mathcal{S}$, and then choose an action $a^i \in A^i$ according to

its policy $\pi_i(a^i|o^i) : O^i \mapsto A^i$. The transition function $\mathcal{P}(s'|s, \boldsymbol{a}) : \mathcal{S} \times \mathcal{A} \mapsto \Delta(\mathcal{S})$ returns a distribution over successor states s' given state s and joint action $\boldsymbol{a}$. After that, each agent i receives an individual reward r_t^i based on its reward function $\mathcal{R}^i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ at time step t. The purpose of each agent i is to find a policy π_i that maximize its expected discounted return $\mathbb{E}[\sum_{t=0}^{H} \gamma^t r_t^i|\pi_i]$ over horizon H, where γ is the discount factor.

In this work, we focus on general cooperative tasks that do not require agents to share the same reward at each time step, but share the same behavior patterns that lead to rewards. Besides, we assume $O^1 = \ldots = O^n = O$ and $A^1 = \ldots = A^n = A$.

Advising Mechanism

Advising mechanism is an efficient, scalable knowledge sharing paradigm, where experienced agents (acting as teachers) give advice to less experienced agents (acting as students) on what to do based on their positive knowledge. CONS follows the teacher-student relationship in the advising mechanism, but innovates in the content of advice and the way of using advice. To make our work easy to understand, here we introduce a promising advising framework, AdHocTD (da Silva, Glatt, and Costa 2017), which focuses on when to ask for and when to give advice within two budgets: b_{ask} and b_{give}. To encourage agents to engage in these behaviors only in critical states, the probabilities for them to ask for and give advice can be calculated as

$$P_{ask}(s) = (1 + \upsilon_a)^{-f(s)} \tag{1}$$

and

$$P_{give}(s) = 1 - (1 + \upsilon_g)^{-g(s)}, \tag{2}$$

respectively. $f(s) = \sqrt{n_{visit}(s)}$ is the confidence function of asking for advice for the current state s, where $n_{visit}(s)$ is the number of times the agent has visited s. $g(s) = f(s)|\max_a Q(s, a) - \min_a Q(s, a)|$ is the confidence function of giving advice for the state s, where Q is the agent's Q-network and $|max_a Q(s, a) - min_a Q(s, a)|$ measures the importance of state s. In the above two equations, υ_a and υ_g are pre-defined scaling variables. When receiving one advice, an AdHocTD agent follows it exactly; when receiving more than one advice, it selects the executed action through a majority vote.

Independent Q-learning

Independent Q-learning (IQL) (Tampuu et al. 2017) combines deep Q-network (DQN) (Mnih et al. 2015) with independent learning (Tan 1993), where each agent runs the DQN algorithm independently. The Q-function of each agent i that estimates the value of each state-action pair is $Q^\pi(s, a) = \mathbb{E}[R|s^t = s, a^t = a]$ (the superscript i is omitted for simplicity), where π is its policy, R is its total discounted return, s is the current state and a is the action it chooses. The optimal Q-function $Q^*(s, a) = \max_\pi Q^\pi(s, a)$ obeys the Bellman optimality equation $Q^*(s, a) = \mathbb{E}_{s'}[r + \gamma \max_{a'} Q^*(s', a')]$. DQNs are optimized by minimizing

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}[y - Q(s, a; \theta))^2], \tag{3}$$

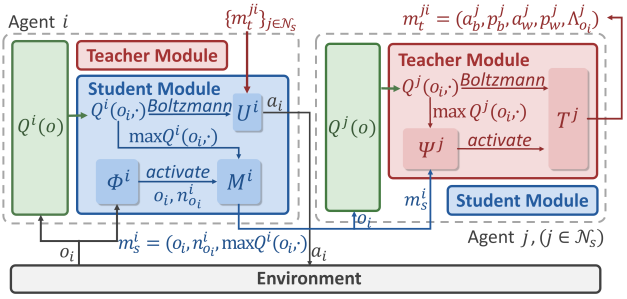


Figure 1: An overview of CONS. Student: Φ^i receives o_i, sends it with $n_{o_i}^i$ to activate M^i with the probability $P_{ask}(o_i)$. If M^i is activated, it assembles o_i, $n_{o_i}^i$ and $\max Q^i(o_i, \cdot)$ into a student request message m_s^i and broadcasts it. Upon receiving messages m_t^{ji} from the teachers $j \in \mathcal{N}_s$ ($\mathcal{N}_s$ is the set of agents sharing knowledge), U^i modifies the original $\pi^i(\cdot|o_i)$ that derived from $Q^i(o_i, \cdot)$ according to the messages and then samples an action a_i from the modified policy. Teacher: Ψ^j decides whether to share knowledge with agent i according to m_s^i, and if so, activates module T^i. Module T^i extracts positive knowledge a_b^j and p_b^j as well as negative knowledge a_w^j and p_w^j from its policy $\pi^j(\cdot|o_i)$, and then combines them with its prestige $\Lambda_{o_i}^j$ to form a teacher message m_t^{ji} to reply to agent i.

where $\mathcal{D}$ is experience replay buffer and $y = r + \gamma \max_{a'} Q(s', a'; \bar{\theta})$. The parameters $\bar{\theta}$ of the agent's target network are periodically copied from θ and remain constant for a certain number of iterations. Other variants of DQN, such as DRQN (Hausknecht and Stone 2015), can also be combined with independent learning while keeping the loss function unchanged in form.

Method

In this section, we propose CONS, a novel knowledge sharing framework that leverages two types of knowledge and reduces the negative effects of suboptimal knowledge on agents. We first introduce policy confidence to quantify the level of certainty of the agent's policy and then provide a detailed description of the three stages of CONS. Please note that knowledge sharing is initiated only after agents have interacted with the environment for a short period to avoid ineffective sharing in the very early stages. Without loss of generality, we assume that in the following, agent i takes the role of student, while the other agents may take the role of teachers (uniformly represented as j). Figure 1 shows how CONS works after sharing is initiated.

Policy Confidence

Policy confidence should assess the certainty of agent policy under certain observation o across different $|A|$. So we define it as the min-max normalized value of the standard deviation of the action probability distribution:

$$\Gamma_o = \frac{\sigma_o - \min(\sigma_o)}{\max(\sigma_o) - 0} = \frac{|A|\sigma_o}{\sqrt{|A| - 1}}. \tag{4}$$

In the above equation, σ_o denotes the standard deviation of the action probability distribution $\pi(\cdot|o)$ conditioned on observation o and A denotes the agent's action space. σ_o reaches its maximum value when only one action has the maximum probability of 1, while its minimum value occurs when the probabilities of all actions are equal to $\frac{1}{|A|}$. $\pi(\cdot|o)$ is derived from the Boltzmann distribution

$$\pi\left(a \mid o\right)=\frac{e^{Q(o,a)/T}}{\sum_k e^{Q(o,a_k)/T}}=p_a, \tag{5}$$

where p_a is the probability of action a and T is the temperature parameter used to adjust the randomness of decisions and we set it to 1.

Stage 1: Student Sends Request

After knowledge sharing is initiated, agent i checks its budget b^i_{ask}. If not exhausted, it broadcasts a student request message m^i_s with the probability of $P_{ask}(o_i)$ calculated by Eq. 1, where o_i is its current observation; otherwise, it samples an action from its own policy. In addition to o_i, m^i_s also includes $n^i_{o_i}$ and $\max Q^i(o_i,\cdot)$, representing the number of times agent i has observed o_i and its corresponding maximum Q-value respectively.

Stage 2: Teacher Shares Knowledge

Teachers in CONS share both positive and negative knowledge regarding o_i with the student. Upon receiving $m^i_s = (o_i, n^i_{o_i}, \max Q^i(o_i,\cdot))$, agent j first checks its budget b^j_{give}. If the budget is exhausted, no response will be provided; otherwise, agent j decides whether to share policy-level knowledge about o_i with agent i based on m^i_s, $n^j_{o_i}$ and $\max Q^j(o_i,\cdot)$. CONS agents are well-intentioned, aiming to share knowledge only at appropriate times, thereby avoiding any potential misinformation. Specifically, the module T^j in Figure 1 is activated for knowledge extraction only when agent j has more or better experience compared to agent i with respect to observation o_i. This activation condition, which also helps reduce unnecessary communication overhead, can be expressed as

$$\mathbb{1}_{n^j_{o_i}>n^i_{o_i}}+\mathbb{1}_{\max Q^j(o_i,\cdot)>\max Q^i(o_i,\cdot)}>0, \tag{6}$$

where $\mathbb{1}$ is the indicator function. If this inequality holds, it implies that agent j either has observed o_i more frequently or taken more valuable actions under the observation o_i compared to agent i. If T^j is activated, it extracts the knowledge to be shared from the policy distribution $\pi^j(\cdot|o_i)$ derived from Eq. 5. Along with decision-related knowledge, agent j also shares its local information with agent i so that agent i can calculate the weights of each responding teacher. Specifically, the teacher message replied by agent j is $m^j_t = (a^j_b, p^j_b, a^j_w, p^j_w, \Lambda^j_{o_i})$, where a^j_b and p^j_b represent the best action and its probability, a^j_w and p^j_w represent the worst action and its probability, and $\Lambda^j_{o_i}$ represents the prestige of agent j. This observation-specific prestige should reflect agent j's familiarity with o_i and confidence in making decisions under o_i, which can be defined as

$$\Lambda^j_{o_i}=\sqrt{n^j_{o_i}}\times\Gamma^j_{o_i}, \tag{7}$$

where $\Gamma^j_{o_i}$ is the policy confidence of agent j under o_i derived from Eq. 4.

Stage 3: Student Utilizes the Acquired Knowledge

CONS agents are optimistic—they believe that the teacher's sharing is well-intentioned, and their knowledge, whether positive or negative, can be beneficial to themselves. However, CONS agents are also cautious—they do not blindly trust that the teachers' knowledge is always correct. Therefore, upon receiving knowledge from teachers, CONS agents carefully adjust their action probabilities and conduct targeted exploration based on their new policies. This process of absorbing and utilizing knowledge involves several specific details, which we will describe below.

The Changing Weights of Positive Knowledge and Negative Knowledge. In challenging tasks, agents initially face failure and gain more success as their policies improve. Therefore, negative knowledge is valuable in the early learning period as it helps agents narrow down their exploration space, while positive knowledge becomes more valuable later on as it as it enables agents to accomplish tasks more effectively. CONS adjusts weights for positive and negative knowledge, denoted as w_p and w_n respectively, increasing the former and decreasing the latter progressively during learning. The sum of the two weights is always equal to 1. Specifically, we use

$$h(x)=\frac{1}{\frac{1-\mathrm{a}}{e_i}\cdot x+\mathrm{a}} \tag{8}$$

to generate w_n for the x^{th} episode, then w_p can be obtained by $w_p = 1 - w_n$. In the above equation, e_i is the episode when knowledge sharing is initiated and a is an hyperparameter that used to adjust the descent rate of w_n. After knowledge sharing is initiated, w_n first decreases rapidly from 1, followed by a progressively slower decline. We avoid using a linear function because agents should quickly shift their focus to positive knowledge, which aligns with the intuition that negative knowledge is more important than positive knowledge only in the early stages of learning.

Soft Updating of Action Probabilities. The CONS agents modify their action probabilities according to the received teachers' knowledge. They regard the probabilities within each teacher's knowledge as the update targets for their corresponding actions, and perform multi-objective soft updates while taking into account the weights assigned to positive knowledge, negative knowledge and individual teachers. Assume that agent i has acquired positive knowledge and negative knowledge about action a_m from teachers in set $\mathcal{N}^b_m$ and set $\mathcal{N}^w_m$ respectively, that is, $a_m = a^k_b = a^l_w \left(k \in \mathcal{N}^b_m, l \in \mathcal{N}^w_m\right)$. Then agent i modifies the original probability p^i_m of a_m using the following equation:

$$\begin{aligned}\tilde{p}^i_m = p^i_m &+ w_p \sum_k w_k \cdot \tau\left(p^k_b - p^i_m\right)\cdot \mathbb{1}_{p^k_b > p^i_m} \\ &+ w_n \sum_l w_l \cdot \tau\left(p^l_w - p^i_m\right)\cdot \mathbb{1}_{p^l_w < p^i_m},\end{aligned} \tag{9}$$

Algorithm 1: Sample an action a_i to be executed through targeted exploration.

Require: The new policy $\tilde{\pi}^i(\cdot|o_i)$ and its confidence $\tilde{\Gamma}$.

With probability $\tilde{\Gamma}$ **do**
  $a_i \leftarrow \arg\max_a \tilde{\pi}^i(a|o_i)$ ▷ Sample the best action
With probability $1 - \tilde{\Gamma}$ **do**
  Divide $[0, 1]$ into $|A| - 1$ equal intervals
  if $\tilde{\Gamma}$ is in the q^{th} interval in ascending order **then**
    Remove the worst q actions from $\tilde{\pi}^i$.
    ▷ $q \in 1, 2, \ldots, (|A| - 1)$
    Normalize the remaining action probabilities to policy Π to be sampled.
    $a_i \leftarrow sample(\Pi)$
return a_i

where $\tilde{p}^i_m$ is the modified intermediate probability of action a_m for agent i to be subsequently normalized through softmax. $\tau \in (0, 1)$ controls the update rate, and the indicator function is used to mask inappropriate modifications. w_k and w_l represent the weights of teacher k and l respectively, which are calculated based on the prestige Λ^k and Λ^l (the subscript o_i is omitted for simplicity) using the following equation:

$$w_k = \frac{e^{\Lambda^k}}{\sum_k e^{\Lambda^k}}, w_l = \frac{e^{\Lambda^l}}{\sum_k e^{\Lambda^l}}. \tag{10}$$

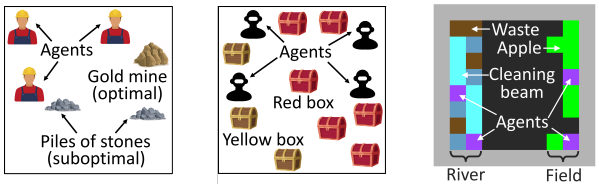
The probabilities of other actions in A that are considered best or worst by teachers can be modified in the same way.

Sample An Action. After completing all necessary probability modifications, agent i obtains a new policy $\tilde{\pi}^i(\cdot|o_i)$ by performing softmax normalization on all probabilities, and then calculates its new policy confidence $\tilde{\Gamma}$ (the superscript i and the subscript o_i are omitted for simplicity). The policy $\tilde{\pi}^i(\cdot|o_i)$ is derived by cautiously absorbing the knowledge from all teachers, which integrates their experiences. Based on $\tilde{\pi}^i$ and $\tilde{\Gamma}$, agent i performs targeted exploration to sample an action a_i to be executed as algorithm 1 shows.

Why do CONS agents explore after excluding several actions? (i) Exploration (rather than taking the best action) is to gain a comprehensive understanding of the task and avoid getting stuck in a suboptimal solution; (ii) Excluding some low-probability actions can improve their exploration efficiency. In addition, the way CONS agents choose actions also conforms the following intuitions. A small value of $\tilde{\Gamma}$ indicates that the probabilities of each action are similar, thus agent i should prioritize exploration. Interestingly, at this juncture, there is a low probability of directly sampling the best action, and an action will be sampled from a larger subset of A. A large value of $\tilde{\Gamma}$ indicates the exact opposite, i.e., agent i has a high probability of taking the best action directly, and its exploration is more limited.

Experiments

In this section, we evaluate the effectiveness of CONS in three cooperative multi-agent tasks: *patient gold miner*, *find the treasure* and *cleanup*. Additionally, we study ablations to further demonstrate the significance of negative knowledge sharing, cautious absorption of knowledge and targeted exploration on the team learning efficiency. Lastly, we discuss the limitations of CONS. We mainly compare CONS with I2Q (Jiang and Lu 2022), SEQL (Christianos, Schäfer, and Albrecht 2020), GA-Comm (Liu et al. 2020), AdHocTD (da Silva, Glatt, and Costa 2017) and IQL (Tampuu et al. 2017). The diversity exploration method MAVEN (Mahajan et al. 2019) is also included in the evaluation of the *find the treasure* task, where agents receive global rewards. For all experiments, unless otherwise stated, we run 10 evaluation episodes without any sharing or exploration every 10k episodes. More training details and environment settings can be found in Appendix (Ba et al. 2023).



(a) (b) (c)

Figure 2: Illustrations of three environments. (a) Patient gold miner (PGM). (b) Find the treasure (FT). (c) Cleanup.

Patient Gold Miner

Task Settings. In the patient gold miner (PGM) environment, depicted in Figure 2(a), a group of n agents act as miners aiming to maximize their gold collection. To obtain a piece of gold and receive an individual reward of r_g, an agent must spend T_d time steps at a gold mine. However, each step incurs an individual penalty of -1. In addition to the N_g gold mines, agents can also get rewards from N_p stone piles without any additional penalties. Each agent can gather one stone per step and receive an individual reward of 0.3. They can obtain a maximum of T_s stones from a stone pile and one piece of gold from a gold mine. Collecting stones is an easy-to-learn suboptimal behavior, while mining gold is an optimal yet highly risky behavior due to rewards being deeply hidden behind penalties. We conduct experiments under two different settings, as detailed in Table **??**. Note that the task difficulty is determined by the ratio of T_d to the episode length L, not by n. A larger ratio significantly reduces the probability of agents finding the optimal strategy, making PGM-3ag more challenging than PGM-6ag.

Results. Figure 3(a) shows that CONS outperforms other baselines both in sample efficiency and final performance. It achieves performance equivalent to AdHocTD and IQL in only half the number of episodes and eventually surpasses them. The IQL agents must independently explore the environment to discern the higher value of gold mines. For AdHocTD, the preference of experienced agents towards suboptimal behavior is propagated throughout the entire team, leading to a suboptimal outcome. I2Q spends a lot of time

Exp name	Grid size	Agent view	N	N_g	T_d	R_g	N_p	T_s	L	T_d/L
PGM-6ag **(easier)**	12x12	5x5	6	2	10	30	3	10	50	20%
PGM-3ag **(harder)**	8x9	3x5	3	1	8	20	2	8	25	32%

Table 1: Two settings for Patient Gold Miner (PGM) environment

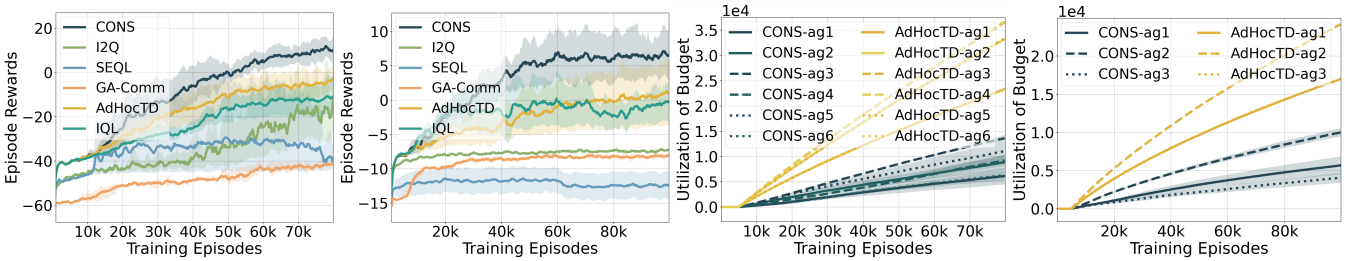


Figure 3: Experimental results of PGM task. (a-b) Mean evaluating episode returns for the whole team across 5 seeds on PGM-6ag and PGM-3ag with the 95% confidence interval shaded. (c-d) The utilization of requesting budget by CONS agents and AdHocTD agents in PGM-6ag and PGM-3ag.

learning the ideal transition model in the early stages, resulting in slower performance improvement. SEQL agents hardly benefit from others' trajectories due to the rarity of high-value trajectories in this task. Instead, they are overly influenced by numerous suboptimal trajectories, making the discovery of high-value states even more difficult. Despite the provision of richer information for agents' decision-making, the communication messages in GA-Comm have limited impact on facilitating the transition from suboptimal behavioral patterns to optimal ones. Compared to the methods above, CONS agents cautiously assimilate the acquired positive and negative knowledge to update their policy and conduct targeted exploration, thereby achieving a higher learning rate and better performance. Figure 3(b) shows that CONS still outperforms other baselines on this harder task, and its advantage is more pronounced. All baselines exhibit a relative performance decrease, indicating that the aforementioned issue worsens with an increase in environmental difficulty. Figure 3(c) and (d) depict the utilization of requesting budget b_{ask} by CONS and AdHocTD agents throughout the entire learning process. In both settings, CONS agents consistently exhibit significantly lower average budget utilization compared to AdHocTD agents, at 30.1% and 33.5% of the budget used by AdHocTD agents, respectively. This is primarily because CONS avoids many inappropriate knowledge sharing through Eq. 6. The learning curves and budget utilization indicate that CONS can achieve better performance with less knowledge sharing.

Find the Treasure

Task Settings. In find the treasure (FT) environment depicted in Figure 2(b), 4 agents must collaborate to search for a single treasure hidden within one of the 6 red boxes. Collecting this treasure yields a team reward of +15. There are also 3 yellow boxes in this environment, each containing a coin that brings a team reward of +2. Each agent has an action space of [up, down, right, left, open, pick up, stay].To open a box, both agents must perform the *open* action on it simultaneously. The items inside the box can only be collected when at least one agent performs the *pick up* action at the opened box. Opening a yellow box incurs a team cost of -1, while opening a red box incurs a team cost of -2.

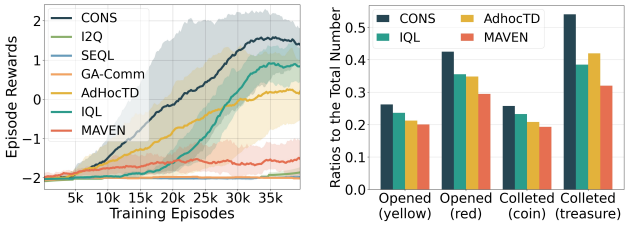


Figure 4: FT task: (a) Mean evaluating episode returns for the whole team across 4 seeds with the 95% confidence interval shaded. (b) The ratios of opened boxes and collected items to the total number of corresponding entities within 400 evaluation episodes.

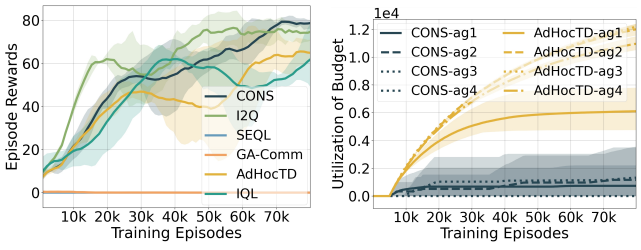
Results. As shown in Figure 4(a), the I2Q, SEQL and GA-Comm agents completely failed in this sparse reward task. The diversity exploration method MAVEN also has unsatisfactory performance, suggesting that blind exploration can be counterproductive. IQL and AdHocTD perform relatively well, surpassing MAVEN. However, CONS learns faster and achieves higher rewards compared to them. CONS agents exhibit better coordination than IQL agents due to knowledge sharing, and demonstrate greater robustness to suboptimal knowledge compared to AdHocTD agents due to the cautious absorption and rational utilization of knowledge. We run 400 evaluation episodes for each algorithm after training and record the counts of opened boxes (red and yel-

low) and collected items (coins and treasures). Figure 4(b) shows the ratios of opened boxes and collected items to the total number of corresponding entities for CONS and baselines (I2Q and SEQL are omitted due to poor performance). The CONS agents have the best grasp of the environment—they effectively balance the objectives of opening red boxes for treasure search and yellow boxes for coin collection, thus achieving excellent performance.

Cleanup

Task Settings. Cleanup (Yang et al. 2020) is a classic public goods game where agents can earn rewards by collecting apples whose growth rate is negatively correlated with the amount of waste in the river. Waste is generated uniformly in the river with a probability of 0.5 per time step until 40% of the river is covered, at which point apples will not grow either. All agents can fire the cleaning beam, which can clean the waste within three cells above the agent. Figure 2(c) illustrates the cleanup task where 4 agents collaborate to collect apples in an 8×8 grid world. Obviously, the agents in this task need a well-coordinated division of labor so that the team can obatain more rewards.

Results. Cleanup without suboptimal interferences is a simpler task compared to the previous two. However, GA-Comm and SEQL fail completely, as shown in Figure 5. Despite achieving acceptable final convergence results, both IQL and AdHocTD exhibit significant performance decline during the middle stage of learning. This decline can be attributed to agents oscillating between the roles of cleaner and collector, leading to inefficient waste cleaning. Notably, AdHocTD shows a more prominent performance drop as action advice worsens this oscillation. Compared with I2Q that performs greedy experience exploitation, the mechanisms in CONS may slightly slow down the learning speed in the early stage. However, CONS efficiently utilizes shared knowledge to fully explore the environment and eventually outperforms I2Q. In a word, CONS benefits from sharing two types of knowledge and avoids the drawbacks of traditional advising mechanism, resulting in good performance. Furthermore, Figure 5(b) shows that CONS agents only use 8% of the requesting budget used by AdHocTD agents, again demonstrating the superior performance of CONS.



(a) (b)

Figure 5: (a) Mean evaluating episode returns for the whole team across 3 seeds with the 95% confidence interval shaded. (b) The utilization of requesting budget by CONS agents and AdHocTD agents.

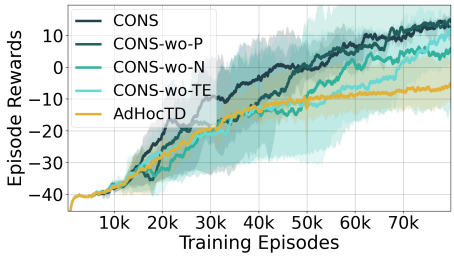


Figure 6: Ablation studies on PGM-6ag. Mean team training rewards are plotted with the 95% confidence interval shaded.

Discussions

Ablations. To understand the outstanding performance of CONS, we conduct ablation studies on PGM-6ag to evaluate the contribution of its key innovations: negative knowledge sharing, cautious knowledge absorption and targeted exploration. We denote CONS without negative knowledge sharing as CONS-wo-N, CONS without positive knowledge sharing as CONS-wo-P, and CONS without targeted exploration (i.e., agents sample actions directly from the modified probability distribution) as CONS-wo-TE. Additionally, the original CONS and AdHocTD are also included in the ablation study. As shown in Figure 6, CONS-wo-P outperforms CONS-wo-N, and CONS outperforms CONS-wo-P, indicating that negative knowledge may be more crucial than positive knowledge in challenging tasks, but the presence of positive knowledge can further enhance performance. Additionally, CONS-wo-TE performs similarly to AdHocTD in the early stages of learning but eventually surpasses it significantly. This indicates that cautious absorption of others' knowledge indeed enhances the agents' robustness to suboptimal advice, thus avoiding falling into suboptimal solutions. The ablation study results show the effectiveness of negative knowledge sharing, cautious knowledge absorption and targeted exploration.

Limitations. CONS is ideal for complex tasks that require extensive exploration or where agents are prone to getting trapped in suboptimal solutions. However, when the optimal strategy is evident or no suboptimal solutions exist, strictly following advice would be better as the targeted exploration in CONS may slightly slow down the learning process. In addition, CONS currently relies on observation counters and value-based underlying algorithms, making it suitable only for discrete tasks for now.

Conclusion and Future Work

In this paper, we propose CONS to maximize the benefits of knowledge sharing for agents. The CONS agents share both positive and negative knowledge optimistically and absorb others' knowledge cautiously. Experimental results show that CONS can significantly improve learning speed and final performance in challenging tasks. For future work, we will extend the underlying idea of CONS to continuous tasks by designing additional networks to replace counters and utilizing action discretization technologies.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC3400404, the National Natural Science Foundation of China under Grant 62172154, 62372473, 62321003, 62002113 and 62272154, the Hunan Provincial Key Research and Development Program of China under Grant 2022GK2004, the Hunan Provincial Natural Science Foundation of China under Grant 2023JJ30702. The authors are grateful for resources from the High Performance Computing Center of Central South University. Prof. Xuan Liu is the corresponding author of the paper.

References

Anand, D.; Gupta, V.; Paruchuri, P.; and Ravindran, B. 2021. An Enhanced Advising Model in Teacher-Student Framework using State Categorization. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 35(8): 6653–6660.

Ba, Y.; Liu, X.; Chen, X.; Wang, H.; Xu, Y.; Li, K.; and Zhang, S. 2023. Cautiously-Optimistic Knowledge Sharing for Cooperative Multi-Agent Reinforcement Learning. arXiv:2312.12095.

Christianos, F.; Schäfer, L.; and Albrecht, S. 2020. Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 10707–10717. Curran Associates, Inc.

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 (AAMAS)*, 1100–1108. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

de Witt, C. S.; Gupta, T.; Makoviichuk, D.; Makoviychuk, V.; Torr, P. H. S.; Sun, M.; and Whiteson, S. 2020. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge? *arXiv preprint arXiv:2011.09533*.

Ding, Z.; Huang, T.; and Lu, Z. 2020. Learning Individually Inferred Communication for Multi-Agent Cooperation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 22069–22079. Curran Associates, Inc.

Guo, Y.; Campbell, J.; Stepputtis, S.; Li, R.; Hughes, D.; Fang, F.; and Sycara, K. 2023. Explainable Action Advising for Multi-Agent Reinforcement Learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 5515–5521.

Gupta, N.; Srinivasaraghavan, G.; Mohalik, S. K.; and Taylor, M. E. 2021. Hammer: Multi-level coordination of reinforcement learning agents via learned messaging. *arXiv preprint arXiv:2102.00824*.

Hausknecht, M.; and Stone, P. 2015. Deep recurrent q-learning for partially observable mdps. In *2015 aaai fall symposium series*.

Ilhan, E.; Gow, J.; and Perez Liebana, D. 2021. Action Advising with Advice Imitation in Deep Reinforcement Learning. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 629–637. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Jiang, J.; and Lu, Z. 2018. Learning Attentional Communication for Multi-Agent Cooperation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31. Curran Associates, Inc.

Jiang, J.; and Lu, Z. 2022. I2Q: A Fully Decentralized Q-Learning Algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 20469–20481. Curran Associates, Inc.

Kim, D.-K.; Liu, M.; Omidshafiei, S.; Lopez-Cot, S.; Riemer, M.; Habibi, G.; Tesauro, G.; Mourad, S.; Campbell, M.; and How, J. P. 2020. Learning Hierarchical Teaching Policies for Cooperative Agents. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 620–628. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*, 157–163. San Francisco (CA): Morgan Kaufmann.

Liu, Y.; Wang, W.; Hu, Y.; Hao, J.; Chen, X.; and Gao, Y. 2020. Multi-Agent Game Abstraction via Graph Attention Neural Network. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 34(05): 7211–7218.

Lowe, R.; WU, Y.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30. Curran Associates, Inc.

Mahajan, A.; Rashid, T.; Samvelyan, M.; and Whiteson, S. 2019. MAVEN: Multi-Agent Variational Exploration. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32. Curran Associates, Inc.

Matignon, L.; Laurent, G. J.; and Le Fort-Piat, N. 2007. Hysteretic Q-learning : an algorithm for Decentralized Reinforcement Learning in Cooperative Multi-Agent Teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 64–69.

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): 529–533.

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. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 33(01): 6128–6136.

Palmer, G.; Tuyls, K.; Bloembergen, D.; and Savani, R. 2018. Lenient Multi-Agent Deep Reinforcement Learning. In *Proceedings of the 17th International Conference*

on Autonomous Agents and MultiAgent Systems (AAMAS), 443–451. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Peng, B.; Rashid, T.; Schroeder de Witt, C.; Kamienny, P.-A.; Torr, P.; Boehmer, W.; and Whiteson, S. 2021. FACMAC: Factored Multi-Agent Centralised Policy Gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 12208–12221. Curran Associates, Inc.

Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, 4295–4304. PMLR.

Singh, A.; Jain, T.; and Sukhbaatar, S. 2019. Individualized Controlled Continuous Communication Model for Multiagent Cooperative and Competitive Tasks. In *International Conference on Learning Representations (ICLR)*.

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D. E.; and Yi, Y. 2019. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, 5887–5896. PMLR.

Sukhbaatar, S.; Szlam, A.; and Fergus, R. 2016. Learning Multiagent Communication with Backpropagation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NeurIPS)*, 2252–2260. Red Hook, NY, USA: Curran Associates Inc.

Tampuu, A.; Matiisen, T.; Kodelja, D.; Kuzovkin, I.; Korjus, K.; Aru, J.; Aru, J.; and Vicente, R. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PLOS ONE*, 12(4): 1–15.

Tan, M. 1993. Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents. In *Proceedings of the 10th International Conference on Machine Learning (ICML)*, 330–337.

Willemsen, D.; Coppola, M.; and de Croon, G. C. 2021. MAMBPO: Sample-efficient multi-robot reinforcement learning using learned world models. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5635–5640.

Yang, J.; Li, A.; Farajtabar, M.; Sunehag, P.; Hughes, E.; and Zha, H. 2020. Learning to Incentivize Other Learning Agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 15208–15219. Curran Associates, Inc.

Zhang, H.; Feng, S.; Liu, C.; Ding, Y.; Zhu, Y.; Zhou, Z.; Zhang, W.; Yu, Y.; Jin, H.; and Li, Z. 2019. CityFlow: A Multi-Agent Reinforcement Learning Environment for Large Scale City Traffic Scenario. In *The World Wide Web Conference (WWW)*, 3620–3624. New York, NY, USA: Association for Computing Machinery.

Zhu, C.; Leung, H.-F.; Hu, S.; and Cai, Y. 2021. A Q-Values Sharing Framework for Multi-Agent Reinforcement Learning under Budget Constraint. *ACM Trans. Auton. Adapt. Syst.*, 15(2).