Conditionally Optimistic Exploration for Cooperative Deep Multi-Agent Reinforcement Learning

Abstract

Efficient exploration is critical in cooperative deep Multi-Agent Reinforcement Learning (MARL). In this paper, we propose an exploration method that efficiently encourages cooperative exploration based on the idea of the theoretically justified tree search algorithm UCT (Upper Confidence bounds applied to Trees). The high-level intuition is that to perform optimism-based exploration, agents would achieve cooperative strategies if each agent's optimism estimate captures a structured dependency relationship with other agents. At each node (i.e., action) of the search tree, UCT performs optimism-based exploration using a bonus derived by conditioning on the visitation count of its parent node. We provide a perspective to view MARL as tree search iterations and develop a method called Conditionally Optimistic Exploration (COE). We assume agents take actions following a sequential order, and consider nodes at the same depth of the search tree as actions of one individual agent. COE computes each agent's state-action value estimate with an optimistic bonus derived from the visitation count of the state and joint actions taken by agents up to the current agent. COE is adaptable to any value decomposition method for centralized training with decentralized execution. Experiments across various cooperative MARL benchmarks show that COE outperforms current state-of-the-art exploration methods on hard-exploration tasks.

1 INTRODUCTION

In recent years multi-agent reinforcement learning (MARL) has drawn much attention and has shown high potential to be applied to various real-world scenarios, such as transportation [Seow et al., 2009], robotics [Perrusquía et al., 2021], and autonomous driving [Shalev-Shwartz et al., 2016]. Cooperative MARL is a multi-agent learning setting where the objective is to train multiple agents that can cooperate to maximize the expected return defined by the same reward function shared across all agents. There are several major challenges posed by this setting, such as credit assignment, scalability, non-stationarity, and partial observability. To address those challenges, Bernstein et al. [2002] propose the Centralized Training with Decentralized Execution (CTDE) learning paradigm. In this paradigm, information is shared across agents during training, guiding the learning of individual agents' policies and promoting cooperation during training, while agents still being able to run independently during decentralized execution.

One important line of research in CTDE is value decomposition, a general approach upon which many cooperative exploration methods build. Value decomposition learns a centralized action-value function that can be factorized into the individual utility function (i.e., individual Q-function) of each agent. To ensure the centralized policy is aligned with individual policies, Son et al. [2019] propose the Individual-Global-Max (IGM) principle that guarantees consistency between global and local greedy actions. A common approach to value decomposition is to learn a mixing network that computes the centralized action value from the utilities of all agents. Depending on the specific way to satisfy IGM, different methods have been introduced, including VDN [Sunehag et al., 2017], QMIX [Rashid et al., 2018], QTRAN [Son et al., 2019], and QPLEX [Wang et al., 2020].

Cooperative exploration adds another level of difficulty to the exploration challenge. In cooperative MARL, agents need to cooperatively explore the joint state-action space as the optimal joint policy may require a high degree of collaboration among them. In addition, there may exist different types of cooperative strategies associated with a task. Sound cooperative exploration methods should be able to identify the optimal strategy from potentially many sub-optimalities. For instance, if the task for a group of robots is to move

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

desks and chairs to another location, the optimal strategy is that at each time several agents together lift a heavy desk that one single agent cannot lift, meanwhile each other spare agent carries one chair. In this task delivering items either only collectively or only separately is sub-optimal, even though either way agents achieve a cooperative strategy. Therefore cooperative exploration is challenging in cooperative MARL, especially when the reward signals are sparse. Although directed exploration strategies have been widely studied in multi-armed bandit and single-agent RL settings, they fail to account for cooperation among agents. Moreover, it is not straightforward to adopt single-agent methods to cooperative MARL, due to the exponentially large state-action space and multi-agent credit assignment. The popular ε-greedy strategy has been shown to be ineffective in complex MARL coordination tasks [Wang et al., 2020].

Some recent works encourage cooperative exploration in MARL settings by maximizing the correlation among agents' behaviour, which trains each agent's policy to account for influences from other agents, hence agents achieve effective collaborative exploration behaviour. Correlation maximization is often realized by maximizing the mutual information (MI) between some quantities that can determine or reflect agents' behaviours, such as the trajectory history of each agent. Utilizing this idea, some works have been proposed and empirically outperformed the value decomposition baselines across various benchmark tasks [Jaques et al., 2019, Mahajan et al., 2019, Wang et al., 2019, Kim et al., 2020, Li et al., 2021]. However, two major issues may prohibit agents from learning the optimal joint strategy. First, optimizing the MI quantity for every pair of agents is not scalable because the required computation to optimize all MI losses grows as the number of agents increases. Second, agents could learn different types of cooperative strategies, therefore one particular high degree of collaboration does not guarantee high performance. As pointed out by Li et al. [2022b], simply maximizing the MI may not lead to high returns because agents may learn sub-optimal joint strategy, regardless of how strong the correlation they have.

In this work, we revisit the idea of UCT exploration [Kocsis and Szepesvári, 2006] proposed in a perfect-information game setting, where the game state is accessible at all nodes, and introduce it to encourage cooperative exploration in MARL. Our insight is simple: if each agent's optimism estimate encodes a structured dependency relationship with other agents, by performing optimism-based exploration, agents would be guided to explore cooperative strategies. Assuming at each timestep agents take actions according to a sequential order, the action execution sequence can be viewed as a path from the root to a leaf of a tree. At each node of the tree, the preceding agent's taken action can be considered as the parent node of the current agent. Then we can perform optimism-based exploration by computing the upper confidence bounds of each action for the current agent, conditioned on its parent node's visitation count. We disable exploration after training to obtain decentralized agents for execution. We first review the basic background in the MARL setting and the UCT algorithm. Then, we describe how the UCT exploration can be applied to the MARL setting to encourage cooperative exploration. We build COE on commonly used value decomposition methods, and used the hash-based counting technique [Tang et al., 2017] to enable counting the visitations in continuous state-action domains. Our empirical results on various benchmark domains show that our method is more effective than well-known baselines in exploration-challenging tasks, and matches baseline performance in general MARL tasks.

2 BACKGROUND

2.1 DEC-POMDP

We model the cooperative multi-agent task as a Dec-POMDP (Decentralized Partially Observable Markov Decision Process) [Oliehoek and Amato, 2016], which is formally defined as a tuple $G = \langle \mathcal{S}, \mathcal{A}, P, R, \Omega, O, n, \gamma \rangle$, where $\mathcal{S}$ is the global state space, $\mathcal{A}$ is the action space, Ω is the observation space, n is the number of agents in the environment, and $\gamma \in [0, 1]$ is the discount factor. At each timestep t, on state $s \in \mathcal{S}$ each agent $i \in \mathcal{N} \equiv \{1, \ldots, n\}$ takes an action $a_i \in \mathcal{A}$. The joint action $\mathbf{a} = [a_i]_{i=1}^n \in \boldsymbol{\mathcal{A}} \equiv \mathcal{A}^n$ leads to the next state s' sampled from the transition probability $P(s'|s, \mathbf{a}) : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0, 1]$, and obtains a global reward r according to the reward function $R(s, \mathbf{a}) : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$ shared across all agents. Each agent i has a local policy $\pi_i(a_i|s) : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Based on the joint policy $\boldsymbol{\pi} \equiv [\pi_i]_{i=1}^n$, the joint action-value function is defined as $Q_{\boldsymbol{\pi}}(s, \mathbf{a}) = \mathbb{E}_{\boldsymbol{\pi}}[\sum_{t=0}^{\infty} \gamma^t r_t | s, \mathbf{a}]$. The objective is to find a joint policy that maximizes the action-value function.

We consider the partially observable setting, where each agent i does not observe the global state s, instead only has access to a local observation $o_i \in \Omega$ drawn from the observation function $O(s, i) : \mathcal{S} \times \mathcal{N} \to \Omega$. Hence each agent i maintains its action-observation history $\tau_i \in T \equiv (\Omega \times \mathcal{A})^*$, on which it can condition its policy $\pi_i(a_i|\tau_i) : T \times \mathcal{A} \to [0, 1]$. With agent i observing the next observation o_i', the updated next history is represented by $\tau_i' = \tau_i \cup \{o_i'\}$. We denote the joint history by $\boldsymbol{\tau} \equiv [\tau]_{i=1}^n \in \mathbf{T} \equiv T^n$, and similarly joint next history by $\boldsymbol{\tau}' \equiv [\tau']_{i=1}^n$.

2.2 UCT

UCT (Upper Confidence bounds applied to Trees) [Kocsis and Szepesvári, 2006] is a tree search algorithm commonly used in Monte-Carlo Tree Search for perfect-information games. In UCT, node selection is treated as a multi-armed bandit problem, where at each node its children nodes corre-

spond to the arms, and the Upper Confidence Bound (UCB) bandit algorithm [Auer et al., 2002] is used to select the child node with the highest upper confidence. In particular, consider a sequence of node selections from the root to a leaf of a search tree as a trajectory at one timestep, at each depth the child node i with the highest upper confidence bound is selected:

$$B_i = X_i + c\sqrt{\frac{2\log(p)}{n_i}}, \tag{1}$$

where X_i is the empirical mean of the rewards that have been obtained by trajectories going through node i, c is a constant controlling the scale of exploration, n_i and p are the number of times node i and its parent node have been visited, respectively. Intuitively, conditioned on preceding agents' actions, at the current node actions that have been taken fewer times will have a higher exploration bonus, hence UCT tends to take action combinations that are under-explored or promising actions with higher reward estimates. When the trajectory is completed, a reward is received at the leaf. The visitation count and reward estimate of each selected node are updated accordingly. The UCT paper provides a regret analysis of the UCT algorithm, proving that its expected regret is upper bounded by $O(\log t)$, where t is the number of trajectories/timesteps.

3 RELATED WORK

Single-agent exploration. Exploration strategies have been extensively studied in single-agent deep RL settings. [Amin et al., 2021] provide a thorough literature survey of advanced exploration methods. In recent years, the category of bonus-based methods has been commonly applied to solve hard exploration tasks. Based on the Optimism in the Face of Uncertainty (OFU) principle, the high-level idea is to capture some notion of uncertainty or novelty, and augment the extrinsic reward the environment emits with an intrinsic reward that quantifies the uncertainty. For instance, the count-based method [Bellemare et al., 2016, Ostrovski et al., 2017, Tang et al., 2017] measures novelty through the number of times the agent observes a state-action tuple. Houthooft et al. [2016] propose an intrinsic bonus based on the maximization of information gain about the agent's belief of environment dynamics. Despite their recent success, simply applying bonus-based methods to MARL does not guarantee effective exploration. As the reward signal is shared across all agents, adding one additional centralized intrinsic reward may still be inefficient to learn structured cooperation due to the multi-agent credit-assignment challenge. Other successful exploration approaches like BootstrappedDQN [Osband et al., 2016] are not scalable in MARL because of the exponentially large state-action space.

Multi-agent exploration. A recent branch of research proposes to drive multi-agent exploration by promoting collaboration among agents through the maximization of the correlation or influence of agents. The correlation is commonly realized by the mutual information (MI) of quantities that define or reflect agents' behaviour, such as the trajectory history of each agent. For instance, MAVEN [Mahajan et al., 2019] learns a hierarchical policy to produce a latent variable that encodes the information about the joint policy, and maximizes the MI between this latent variable and the joint trajectories to encourage the correlation of agents' behaviour. Some other methods try to promote collaboration by maximizing pairwise MI between every two agents. For instance, EITI [Wang et al., 2019] maximizes the MI between one agent's transition and the other's state-action. VM3-AC [Kim et al., 2020] maximizes the MI between two agents' policy distributions. Pairwise MI is hard to scale to scenarios with a large number of agents, because the computation grows with the number of agents. Li et al. [2022b] claims one important downside of MI-based methods is the fact that a strong correlation does not necessarily correspond to high-return collaboration, especially when there exist multiple sub-optimal highly-cooperative strategies associated with the given task. Aside from MI-based methods, there are other approaches based on different intuitions. VACL [Chen et al., 2021] leverages variational inference and automatic curriculum learning to solve sparse-reward cooperative MARL challenges. Since this method only aims to solve goal-conditioned problems, it is not generally applicable. EMC [Zheng et al., 2021] utilizes a value decomposition method to implicitly capture influence among agents and uses the prediction errors of individual Q-value functions as intrinsic rewards. Our method also tries to capture agent-wise dependency to guide exploration. Different from MI maximization or EMC, our method captures structured inter-dependency through each agent's conditional optimism estimate and performs optimism-based exploration.

Action Conditioned Training. As the learning objective in CTDE is to obtain decentralized agents for execution, previous works commonly assume agents take actions independently and simultaneously, even during the centralized training phase. A few recent works explicitly consider inter-dependency and cooperation learned through sequential action selection, where each agent's policy is conditioned on preceding agents' joint action. MACPF [Wang et al., 2022] learns a dependent joint policy and its independent counterpart by maximum-entropy RL [Ziebart, 2010]. ACE [Li et al., 2022a] is a Q-learning method that models the multi-agent MDP into a single-agent MDP by making the bootstrap target dependent on subsequent agents' actions. Leveraging the multi-agent advantage decomposition theorem [Kuba et al., 2021], Multi-Agent Transformer (MAT) [Wen et al., 2022] casts MARL into a sequence modeling problem and uses a transformer architecture to map agents'

observation sequences to agents' optimal action sequences. These methods consider action conditioning to increase the expressiveness of the joint policy, hence improving its performance. Our method leverages action conditioning from a different perspective: predecessors' actions reflect dependency among agents, therefore can be used to adjust the optimism level to achieve efficient cooperative exploration.

4 METHOD: CONDITIONALLY OPTIMISTIC EXPLORATION (COE)

In this section, we introduce our UCT-inspired method Conditionally Optimistic Exploration (COE) to effectively drive exploration in cooperative deep MARL. We describe how we can view cooperative MARL as a sequence of tree search iterations. We then discuss the challenges to directly applying UCT to MARL. Then we present approaches to address these issues, which concludes the details of our COE method.

4.1 MULTI-AGENT EXPLORATION AS UCT

We first formulate action selection at each timestep of MARL as a bandit-based tree search procedure. We consider the following sequential decision-making scheme: at each timestep t all n agents take actions sequentially following some arbitrary but fixed pre-determined order. Without loss of generality, we use the identities of agents as the order, i.e., agent $i \in \mathcal{N} \equiv \{1, \ldots, n\}$ is the i-th agent to select its action. Figure 1 depicts the formulation of MARL as UCT at each timestep t, where the tree is a partially shown binary tree for the simplicity of illustration. To construct the tree structure, first the state s^t is the root node. Each agent i has k actions, corresponding to the k children nodes of the parent node at depth $i-1$. Each node at depth i represents an intermediate stage to which the action sequence of agents $\{1, \ldots, i\}$ sequentially transitions. When agent n takes its action, the action sequence reaches a leaf node, where the environment emits a reward to all agents, and transitions to the next state s^{t+1}, which is the root node of the next tree.

By applying the UCT tree search algorithm to each tree, we model the cooperative MARL exploration as a sequence of UCT procedures. For action selection, conditioned on predecessors' joint action, denoted by $a_{<i}$, each agent i estimates an action-value function $Q_i(s, a_i|a_{<i})$. Augmenting the Q-value by an optimistic bonus conditioned on prior agents' actions, we have the upper confidence bound, denoted as $B_i(s, a_i|a_{<i})$. It is worth noting that the Q-value estimate is generalized across states and preceding agents' joint actions using neural network function approximators, which removes the need to maintain an empirical reward estimate at every node in every tree. At depth i, agent i uses the same Q-value function to take action, no matter which subtree the corresponding node is in.

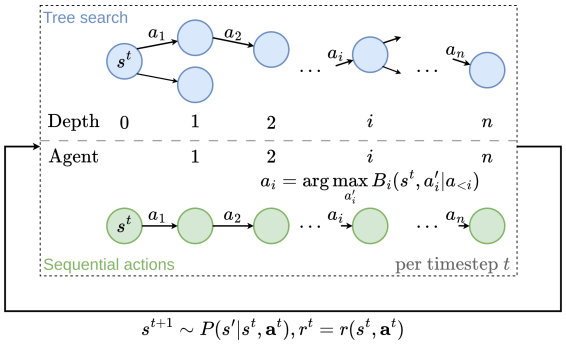

Figure 1: Modelling of MARL as Tree Search Procedure and Application of the UCT Algorithm.

It should be noted that there is a distinction between the conventional tree search problem setting and the cooperative MARL setting. In UCT, the game state information is accessible at all nodes, but our Dec-POMDP setting assumes partial observability. Accessing full state information enables agents to estimate action values based on the same global state and predecessors' actions, while in our setting we cannot have such estimates. CTDE also requires agents to act independently at execution time, without following any sequential order or conditioning policies on other agents' actions. As an approximate implementation, we build the conditional count module on value decomposition methods, and disable exploration after training to obtain decentralized agents. We empirically test QMIX with conditional optimism in Section 5.

4.2 COE ALGORITHM

We first briefly describe a common value decomposition learning paradigm. Then we present how we utilize conditional counts to drive optimistic exploration.

Each agent i has an independent Q-network $Q_i^{ind}(\tau_i, a_i; \phi_i)$ parameterized by ϕ_i. A mixing network $\text{Mixer}(\cdot; \theta)$ parameterized by θ is used to compute the joint Q-values:

$$Q_{jt}^{ind}(\boldsymbol{\tau}, \mathbf{a}) = \text{Mixer}([Q_i^{ind}(\tau_i, a_i)]_{i=1}^N, s; \theta). \quad (2)$$

Individual agent's action-value networks Q_i^{ind} and the mixing network Mixer are trained by minimizing the mean-squared temporal-difference error:

$$\mathcal{L}^{ind}([\phi]_{i=1}^N, \theta) = \mathbb{E}_{\mathcal{D}}[(Q_{jt}^{ind}(\boldsymbol{\tau}, \mathbf{a}) - y^{ind})^2] \quad (3)$$

where $y^{ind} = (r + \gamma \max_{\mathbf{a}'}(Q_{jt}^{ind}(\boldsymbol{\tau}', \mathbf{a}')))$ is the update target, and $\mathcal{D}$ is the replay buffer containing trajectory data collected by Q_i^{ind}'s. It is worth noting that by IGM principle the greedy actions selected by Q_i^{ind}'s are the same actions Q_{jt}^{ind} would have taken. As centralized training backpropagates the global reward signal to learn the individual utilities

Q_i^{ind}'s, value factorization implements an implicit multi-agent credit assignment that enables each agent to grasp the inter-dependency among all utilities.

Building on top of the value decomposition skeleton, we incorporate count-based optimism in both action selection and learning. During action selection, each agent i takes greedy actions with respect to its optimistic action-value

$$a_i = \arg\max_{a_i'} \left\{ Q_i(\tau_i, a_i') + c_{\text{act}} \sqrt{\frac{2\log(N(s, a_{<i}))}{N(s, a_{<i}, a_i')}} \right\}, \quad (4)$$

where $a_{<i}$ represents the joint actions taken by agents prior to agent i, and $c_{\text{act}} \in \mathbb{R}_+$ is a hyper-parameter controlling the scale of optimism. Note that counting is performed in the global state space thanks to centralized training.

Moreover, we augment the global reward and the bootstrapped target each with a bonus term, such that the update target becomes

$$y = \left(r(s, \mathbf{a}) + \frac{c_{\text{rew}}}{\sqrt{N(s, \mathbf{a})}} \right) + \gamma \max_{\mathbf{a}'} \text{Mixer}\left(\left[Q_i(\tau_i', a_i') + \frac{c_{\text{boot}}}{\sqrt{N(s', a_{<i}', a_i')}} \right]_{i=1}^{N} \right), \quad (5)$$

where $c_{\text{rew}}, c_{\text{boot}} \in \mathbb{R}_+$ are hyper-parameters controlling the scale of the optimistic bias in reward and bootstrapped target, respectively. These two bonus terms are added for two major reasons. First, we intend to maintain long-term optimism in the Q-functions. The acting-time optimism decreases as agents take actions, but unlike bandit or tabular MDP methods, COE's Q-value estimate is updated at a relatively slower rate due to the nature of gradient updates of neural networks. To encourage COE to explore persistently, the augmentation to the bootstrap target allows the Q-value itself to encode optimism through TD loss update. Second, since the bootstrap target is defined based on the Q-value estimates of the next states, the optimistic bootstrap target also captures uncertainty from subsequent agents and future timesteps. The idea of learning optimistic Q-values is originally proposed by Jin et al. [2018, 2020], Yang et al. [2020] and extended to deep RL by Rashid et al. [2020].

With the count-based optimism introduced, the complete learning algorithm is presented in Algorithm 1. During decentralized execution, the optimistic bonuses, although decayed to negligible magnitude, are disabled, and agents take independent actions according to Q_i^{ind}'s only.

To apply COE to deep MARL tasks, we need to approximate counts in high-dimensional or continuous state space. In our experiments we use the SimHash method [Tang et al., 2017] that projects states to a lower-dimensional feature space before counting. We record the visitation count for the tuple of the state s and all agents' joint action $\mathbf{a}$, denoted by $N(s, \mathbf{a})$. For each agent i, the count up to its action a_i satisfies $N(s, a_{<i}, a_i) = \sum_{a_{i+1}} N(s, a_{<i}, a_i, a_{i+1}) = \sum_{a_{>i}} N(s, a_{<i}, a_i, a_{>i})$, where $a_{<i}$ and $a_{>i}$ denote the joint actions taken by preceding and subsequent agents of i, respectively. This relationship shows that we can obtain any count up to a_i by summing up the counts of joint actions that overlap $a_{<i}$ at state s. Details about SimHash counting are presented in Appendix A.

Algorithm 1 Conditionally Optimistic Exploration

```
Initialize parameters φ, θ
Visitation count N(s, a) ← 0, ∀(s, a) ∈ S × A
Replay buffer D ← {}
for each episode m = 1, ..., M do
  for each environment timestep t = 1, ..., T do
    for agent i = 1, ..., n do
      Select action a_i^t according to Equation (4)
    end for
    N(s^t, a^t) ← N(s^t, a^t) + 1
    s^{t+1} ~ P(s'|s^t, a^t), r^t = r(s^t, a^t)
    D ← D ∪ {(s^t, a^t, r^t, s^{t+1})}
    Perform a gradient update on Equation (3)
  end for
end for
```

5 EXPERIMENTS

In this section, we evaluate COE on two sets of cooperative MARL tasks across three commonly used benchmarks: 1) sparse-reward tasks that specifically pose the cooperative exploration challenge, and 2) tasks that generally assess MARL methods' ability for effective coordination. Empirical results show that COE achieves higher sample efficiency and performance than other state-of-the-art approaches in sparse-reward tasks, and matches their performance in general cooperative tasks. We also present ablation studies to demonstrate the effectiveness of conditional optimism and COE's compatibility with common MARL methods. As a sanity check, we examine conditional optimism in a didactic multi-agent bandit problem.

5.1 EVALUATION SETUP

We evaluate all algorithms on nine tasks over three benchmark environments. The tasks can be categorized into two sets according to their challenges: (1) Challenging sparse-reward tasks focused on efficient exploration. This includes *SparseTag* and *Sparse Spread* from the Multi-agent Particle Environments (MPE) [Lowe et al., 2017, Mordatch and Abbeel, 2018], and four tasks with different configurations from the Level-Based Foraging (LBF) environments [Albrecht and Ramamoorthy, 2015, Christianos et al., 2020, Papoudakis et al., 2020]. (2) Tasks that generally assess

Table 1: Average Returns and 95% Confidence Interval for All Four Algorithms, and Average Win-rates for SMAC Tasks.

		Tasks \Algs.	COE	EMC	MAVEN	QMIX
MPE		Adversary	17.77 ± 0.71	16.73 ± 0.83	$\mathbf{19.57 \pm 0.51}$	18.20 ± 0.56
		Sparse Tag	$\mathbf{0.65 \pm 0.09}$	0.43 ± 0.06	0.01 ± 0.00	0.40 ± 0.05
		Sparse Spread	$\mathbf{0.79 \pm 0.09}$	0.41 ± 0.18	0.10 ± 0.14	0.29 ± 0.05
LBF		**10x10-3p-3f**	$\mathbf{0.71 \pm 0.05}$	0.68 ± 0.03*	0.16 ± 0.06	0.49 ± 0.01
		15x15-3p-5f	$\mathbf{0.20 \pm 0.02}$	0.12 ± 0.02	0.03 ± 0.00	0.08 ± 0.01
		15x15-4p-3f	$\mathbf{0.41 \pm 0.06}$	0.25 ± 0.07	0.04 ± 0.01	0.19 ± 0.02
		15x15-4p-5f	$\mathbf{0.30 \pm 0.02}$	0.23 ± 0.04	0.04 ± 0.00	0.15 ± 0.02
SMAC	ret	2s-vs-1sc	17.83 ± 0.16*	17.88 ± 0.74*	17.78 ± 1.26*	$\mathbf{18.21 \pm 0.39}$
		3s-vs-5z	$\mathbf{16.03 \pm 1.58}$	9.66 ± 2.62	14.11 ± 2.36*	11.74 ± 1.87
	win	2s-vs-1sc	0.79 ± 0.01	$\mathbf{0.83 \pm 0.04}$	0.82 ± 0.08*	$\mathbf{0.83 \pm 0.02}$
		3s-vs-5z	$\mathbf{0.45 \pm 0.09}$	0.08 ± 0.14	0.29 ± 0.12*	0.13 ± 0.11

multi-agent coordination. This includes *Adversary* in MPE, and an easy task *2s-vs-1sc* and a hard task *3s-vs-5z* in StarCraft Multi-Agent Challenge (SMAC) [Samvelyan et al., 2019]. Note that LBF tasks and *Adversary* are fully observable, whereas SMAC and other MPE domains are partially observable environments. More detailed descriptions of the environments and the evaluation protocol can be found in Appendix B and Appendix C, respectively.

To promote fair comparisons, we build all methods on the QMIX agents with the same agent and mixer network architectures. For the same reason we implement the canonical version of all methods where the only additional component is the exploration module unless otherwise specified. We follow the same protocol presented by Papoudakis et al. [2020] to optimize hyperparameters. Specifically we sweep hyperparameters on one task of each environment with three random seeds, and run the best configuration for all tasks in the respective environment with five seeds for the final experiments. Appendix F explains hyperparameter optimization in more detail.

5.2 PERFORMANCE

We evaluate COE and compare it with the following state-of-the-art baselines in the experiments: (i) QMIX [Rashid et al., 2018]: ε-greedy QMIX with linearly annealed epsilon schedule; (ii) EMC [Zheng et al., 2021]; (iii) MAVEN [Mahajan et al., 2019]: combined with annealing ε-greedy. Empirical results show that COE outperforms all baselines in the sparse-rewards domains well-known for exploration challenges, and matches strong baseline performance in general multi-agent tasks.

Table 1 summarizes the average returns for the four algorithms in all nine tasks. We highlight the maximum average return in bold. We perform a two-sample t-test [Snedecor and Cochran, 1980] with a significance level 0.05 between the best performing algorithm and each other algorithm in each task. We mark the return values with an asterisk if the corresponding algorithm achieves a performance level that is not statistically significantly different from the highest performance. Difficult exploration tasks are shown in bold. The same table also reports the average win-rates in SMAC tasks as it is a common practice in MARL literature. The table summarizing maximum returns over training is presented in Appendix D.

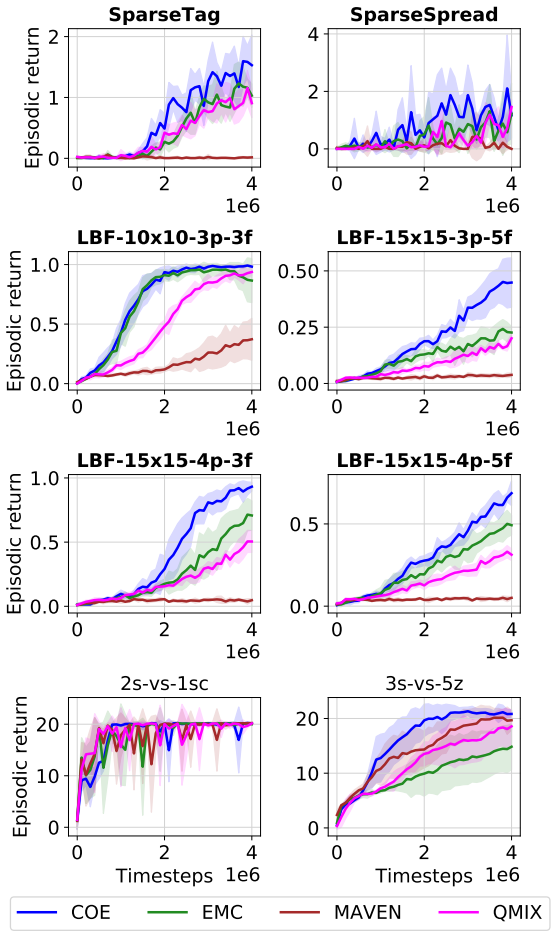

Figure 2: Episodic Returns and 95% Confidence Interval for All Algorithms in All Tasks except *Adversary*, with Sparse-Reward Tasks Marked Bold.

The results in Table 1 and Figure 2 show that COE signif-

icantly outperforms other baselines in sparse-reward tasks that require efficient exploration. Particularly COE has higher sample efficiency in difficult LBF domains. In the early exploration stage, all algorithms gain performance slowly, resulting in indistinguishable learning curves. As time progresses, COE makes improvements much faster than the baselines. The sample efficiency improvement leads to higher final and overall return values. In relatively easier exploration tasks *SparseTag*, *SparseSpread*, and *Foraging-10x10-3p-3f*, COE's outperformance is not as large as it is in the hard tasks. Some other baselines also learn strong policies in these tasks. Since all algorithms are built on the same QMIX agent, overall the results in sparse-reward domains demonstrate the effectiveness of conditional-optimism-guided exploration.

In the general MARL coordination tasks, COE has similar performance as the baselines. *Adversary* is evidently the easiest task among all tested tasks, where all algorithms quickly converge to the optimal policy at almost identical speed. In the hard *3s-vs-5z* task in SMAC, COE shows better sample efficiency and final performance in terms of the mean episodic returns. This trend is similar to the trends we observe in sparse-reward tasks, although in this task the outperformance is not statistically significant. These results indicate that COE is not only an effective approach to hard exploration tasks; it is also a strong algorithm generally applicable to common MARL domains.

5.3 ABLATIONS

COE consists of two major components, namely the independent Q-value functions learned through centralized training, and the conditional optimism. In order to have a better understanding of COE, we test several ablation variants to evaluate these two components' contribution to performance gain. Results suggest that conditional optimism plays a dominant role in performance improvement. Compared to dependent Q-values conditioned on predecessors' actions, independent Q-values learned through value decomposition also work well with conditional optimism despite partial observability.

To evaluate the contributions of *independent* Q-values in COE, we test the following ablation variants that learn *conditional* Q-values: (1) COE-Cond-IQ: We apply conditional optimism to IQL. Each agent simultaneously learns an independent Q-network and a dependent Q-network that takes in predecessors' actions as extra inputs without centralized training. The dependent network selects actions during training. Two nets are trained on separate TD losses using the same replay batches. After training the independent network is responsible for decision-making at execution time. This variant directly mimics UCT in MARL without considering each agent's partial observability issue. (2) COE-Cond-CQ: We add a QMIX mixer to COE-Cond-IQ to enable centralized Q-value training. The same mixer computes the centralized Q-value Q_{jt}^{ind} for independent networks and Q_{jt}^{dep} for dependent networks. This variant ignores the potential issue that within individual Q-values the inter-dependency among agents captured by centralized training and that captured by action conditioning may not be aligned with each other.

To evaluate the contributions of *conditional* optimism, we propose the following ablation variants that use *non-conditional* optimism: (1) UCB-Ind: Similar to COE, each agent performs UCB-based exploration except that optimism is not conditioned on others' actions. (2) UCB-Cen: Agents receive UCB optimism only through the intrinsic reward $\frac{c_{\text{rew}}}{\sqrt{N(s,\mathbf{a})}}$ during centralized training.

We follow the same evaluation protocol described in Section 5.1 to conduct experiments. The average returns of the ablations are summarized in Table 2. Appendix D and Appendix E present the learning curves and a more detailed introduction to the ablations, respectively. Results show that COE has a similar performance as COE-Cond-CQ and COE-Cond-IQ in the majority of tested tasks. COE-Cond-IQ performs relatively worse in MPE tasks, but better in LBF tasks. This may be attributed to the partial observability issue: since LBF is fully observable, COE-Cond-IQ becomes a more legitimate adoption of UCT to cooperative MARL. COE-Cond-CQ matches COE's performance in MPE and SMAC. Although it underperforms COE in three LBF tasks, COE-Cond-CQ is still competitive and matches EMC's performance in LBF. These results suggest that conditional optimism boosts sample efficiency and overall performance with different Q-value estimation approaches.

On the other hand, UCB-Ind underperforms COE in hard LBF tasks and SMAC tasks. It also has a large variance across random seeds in SMAC tasks. UCB-Cen matches COE in half of the tasks, but it also suffers from large variances. Through these comparisons, we observe that conditional optimism guides more steady performance improvement.

5.4 DIDACTIC PROBLEM

A rigorous application of UCT in MARL requires learning each agent's state-action value conditioned on earlier agents' state-action pairs. Due to partial observability in MARL that prevents access to global states, we use value decomposition as an approximate implementation of conditional value estimation and empirically show its effectiveness in previous sections. In this section, we provide a sanity check on the bandit problem to re-demonstrate conditional optimism is important whereas conditional value estimation is unnecessary.

We consider the cooperative multi-agent multi-armed stochastic bandit problem, where the reward is based on the joint action of a group of agents. Suppose we have n

Table 2: Average Returns and 95% Confidence Interval for All Ablations, and Average Win-rates for SMAC Tasks.

		Tasks \Algs.	COE	COE-Cond-IQ	COE-Cond-CQ	UCB-Ind	UCB-Cen
MPE		Adversary	17.77 ± 0.71	15.46 ± 0.68	18.99 ± 0.26	17.70 ± 0.37	17.15 ± 0.82
		Sparse Tag	0.65 ± 0.09	0.07 ± 0.01	0.83 ± 0.17	0.52 ± 0.13	0.49 ± 0.13
		Sparse Spread	0.79 ± 0.09	0.36 ± 0.05	0.54 ± 0.19	0.59 ± 0.31	0.75 ± 0.16
LBF		**10x10-3p-3f**	0.71 ± 0.05	0.76 ± 0.04	0.68 ± 0.01	0.67 ± 0.07	0.64 ± 0.02
		15x15-3p-5f	0.20 ± 0.02	0.19 ± 0.02	0.12 ± 0.05	0.15 ± 0.03	0.13 ± 0.05
		15x15-4p-3f	0.41 ± 0.06	0.47 ± 0.06	0.24 ± 0.06	0.23 ± 0.05	0.16 ± 0.10
		15x15-4p-5f	0.30 ± 0.02	0.23 ± 0.04	0.14 ± 0.02	0.23 ± 0.07	0.27 ± 0.06
SMAC	ret	2s-vs-1sc	17.83 ± 0.16	16.67 ± 1.56	18.64 ± 0.48	11.09 ± 7.27	15.77 ± 1.45
		3s-vs-5z	16.03 ± 1.58	16.85 ± 1.55	17.01 ± 0.74	11.03 ± 3.03	13.36 ± 3.41
	win	2s-vs-1sc	0.79 ± 0.01	0.66 ± 0.13	0.87 ± 0.04	0.47 ± 0.35	0.66 ± 0.07
		3s-vs-5z	0.45 ± 0.09	0.56 ± 0.15	0.57 ± 0.06	0.19 ± 0.18	0.25 ± 0.21

agents, each has k actions. In our didactic Bernoulli bandit problem, only one out of k^n joint actions is optimal with distribution $\mathcal{B}(p = 0.9)$, and all other joint actions are sub-optimal with distribution $\mathcal{B}(p = p_0)$, where the sub-optimality value p_0 is an environment hyper-parameter. For each bandit instance, a uniformly sampled joint action from all combinations is set to be optimal.

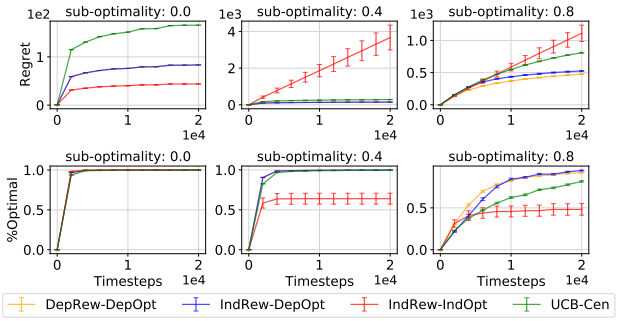

Figure 3: Mean Performance and Standard Error for Bandit Problem with 8 Agents and 3 Actions Each.

We test four UCB-based algorithm variants: (1) DepRew-DepOpt: it performs UCT exploration as in Equation (1), where both reward estimates and count-based optimism are dependent on prior agents' joint action; (2) IndRew-DepOpt: each agent maintains its reward estimates independently, but the optimism is dependent on predecessors; (3) IndRew-IndOpt: both reward estimates and optimism are independent; (4) UCB-Cen: one UCB learner whose action space is the cartesian product of all agents' action set. This variant is merely for performance comparison because it is not factorizable to decentralized agents.

We run experiments on a bandit problem with 8 agents and 3 actions of each agent over 50 seeds, and report the performance of all four algorithms for different sub-optimality settings in Figure 3. We evaluate algorithms with two metrics, the expected regret, which is preferably lower, and the percentage of optimal joint action being selected, which is preferably higher. Results show that both DepRew-DepOpt and IndRew-DepOpt quickly converge to the optimal policy across different sub-optimality settings. This suggests that in this bandit task, conditional optimism robustly drives efficient cooperative exploration, regardless of whether the reward estimates are learned independently. IndRew-IndOpt, on the other hand, is inefficient to identify the optimal joint action and has high variances across random seeds. These results highlight the significance of conditional count-based optimism, and its dominant role over the action-value estimates in coordinated exploration. In general, these results are consistent with results from the MARL ablations in Section 5.3.

6 CONCLUSIONS

In this paper, we draw the connection between cooperative multi-agent reinforcement learning (MARL) and tree search. Inspired by the tree search algorithm UCT, we propose a multi-agent exploration method Conditionally Optimistic Exploration (COE), that utilizes the sequential decision-making scheme and visitation count conditioned on previous agents' actions. Empirical results show that our method significantly outperforms state-of-the-art MARL baselines in sparse-reward hard-exploration tasks, and matches their performance in general coordination tasks.

One limitation of our method is that it may require a large amount of memory due to storing visitation counts of state-action tuples during training, which makes our method hard to scale to tasks with very large state-action space. An interesting future work is to utilize neural network density models to estimate pseudo-counts. Training such a model would require more computation but the model itself only occupies constant memory, which could eliminate potentially high memory usage.

References

Stefano V Albrecht and Subramanian Ramamoorthy. A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. *arXiv preprint arXiv:1506.01170*, 2015.

Susan Amin, Maziar Gomrokchi, Harsh Satija, Herke van

Hoof, and Doina Precup. A survey of exploration methods in reinforcement learning. *arXiv preprint arXiv:2109.00157*, 2021.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

Jiayu Chen, Yuanxin Zhang, Yuanfan Xu, Huimin Ma, Huazhong Yang, Jiaming Song, Yu Wang, and Yi Wu. Variational automatic curriculum learning for sparse-reward cooperative multi-agent problems. *Advances in Neural Information Processing Systems*, 34:9681–9693, 2021.

Filippos Christianos, Lukas Schäfer, and Stefano Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10707–10717, 2020.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pages 3040–3049. PMLR, 2019.

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pages 2137–2143. PMLR, 2020.

Woojun Kim, Whiyoung Jung, Myungsik Cho, and Youngchul Sung. A maximum mutual information framework for multi-agent reinforcement learning. *arXiv preprint arXiv:2006.02732*, 2020.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 282–293. Springer, 2006.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.

Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 3991–4002, 2021.

Chuming Li, Jie Liu, Yinmin Zhang, Yuhong Wei, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. Ace: Cooperative multi-agent q-learning with bidirectional action-dependency. *arXiv preprint arXiv:2211.16068*, 2022a.

Pengyi Li, Hongyao Tang, Tianpei Yang, Xiaotian Hao, Tong Sang, Yan Zheng, Jianye Hao, Matthew E Taylor, and Zhen Wang. Pmic: Improving multi-agent reinforcement learning with progressive mutual information collaboration. *arXiv preprint arXiv:2203.08553*, 2022b.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems*, 32, 2019.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*. Springer, 2016.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. *arXiv preprint arXiv:2006.07869*, 2020.

Adolfo Perrusquía, Wen Yu, and Xiaoou Li. Multi-agent reinforcement learning for redundant robot control in task-space. *International Journal of Machine Learning and Cybernetics*, 12:231–241, 2021.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International conference on machine learning*, pages 4295–4304. PMLR, 2018.

Tabish Rashid, Bei Peng, Wendelin Boehmer, and Shimon Whiteson. Optimistic exploration even with a pessimistic initialisation. *arXiv preprint arXiv:2002.12174*, 2020.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Kiam Tian Seow, Nam Hai Dang, and Der-Horng Lee. A collaborative multiagent taxi-dispatch system. *IEEE Transactions on Automation science and engineering*, 7(3): 607–616, 2009.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

George W Snedecor and William G Cochran. Statistical methods. iowa. *Iowa State University Press. Starkstein, SE, & Robinson, RG (1989). Affective disorders and cerebral vascular disease. The British Journal of Psychiatry*, 154:170–182, 1980.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pages 5887–5896. PMLR, 2019.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Jiangxing Wang, Deheng Ye, and Zongqing Lu. More centralized training, still decentralized execution: Multi-agent conditional policy factorization. *arXiv preprint arXiv:2209.12681*, 2022.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020.

Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512*, 2019.

Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *arXiv preprint arXiv:2205.14953*, 2022.

Zhuoran Yang, Chi Jin, Zhaoran Wang, Mengdi Wang, and Michael I Jordan. On function approximation in reinforcement learning: Optimism in the face of large state spaces. *arXiv preprint arXiv:2011.04622*, 2020.

Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34:3757–3769, 2021.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.