Conditionally Optimistic Exploration for Cooperative Deep Multi-Agent Reinforcement Learning

Xutong Zhao\textsuperscript{1,2}, Yangchen Pan\textsuperscript{3}, Chenjun Xiao\textsuperscript{4}, Sarath Chandar\textsuperscript{1,2,6}, Janarthanan Rajendran\textsuperscript{1,5}

\textsuperscript{1} Mila - Quebec AI Institute
\textsuperscript{2} École Polytechnique de Montréal
\textsuperscript{3} University of Oxford
\textsuperscript{4} University of Alberta
\textsuperscript{5} Université de Montréal

Abstract

Efficient exploration is critical in cooperative deep Multi-Agent Reinforcement Learning (MARL). In this work, we propose an exploration method that effectively encourages cooperative exploration based on the idea of sequential action-computation scheme. The high-level intuition is that to perform optimism-based exploration, agents would explore cooperative strategies if each agent’s optimism estimate captures a structured dependency relationship with other agents. Assuming agents compute actions following a sequential order at each environment timestep, we provide a perspective to view MARL as tree search iterations by considering agents as nodes at different depths of the search tree. Inspired by the theoretically justified tree search algorithm UCT (Upper Confidence bounds applied to Trees), we develop a method called Conditionally Optimistic Exploration (COE). COE augments each agent’s state-action value estimate with an action-conditioned optimistic bonus derived from the visitation count of the global state and joint actions of preceding agents. COE is performed during training and disabled at deployment, making it compatible with 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 [Hüttenrauch et al., 2017] and autonomous driving [Cao et al., 2012, Shalev-Shwartz et al., 2016]. Cooperative MARL is concerned with a special learning setting with common rewards shared across all agents, where agents must coordinate their strategies to achieve a common goal. There are several major challenges posed by this setting, such as credit assignment, scalability, non-stationarity, and partial observability. To address these 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 agent’s 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, which learns a centralized action-value function that can be factorized into the individual utility function (often referred to as individual Q-function) of each agent. The centralized value function performs implicit credit assignment to determine each agent’s contribution towards common returns, and learns implicit inter-dependencies to encourage cooperation. 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 single-agent exploration challenge. In cooperative MARL, agents need to coordinate to explore the large joint state-action space as high-performing joint strategies 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 meth-
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 remain when MI-based methods are used. First, optimizing the MI quantity for every pair of agents is not scalable because the required computation to optimize all MI losses grows quadratically as the number of agents. Second, agents could learn different types of cooperative strategies, and one particular type may not lead to high performance. As pointed out by Li et al. [2022], 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 achieve.

In this work, we seek to explicitly leverage inter-agent dependencies to drive cooperative exploration. Our insight is simple: as a complement to implicit dependencies learned by centralized training, if each agent’s optimism estimate explicitly encodes a structured dependency relationship with other agents, by performing optimism-based exploration, agents would be guided to effectively explore cooperative strategies. It is worth noting that centralized training (CT) only requires the joint policy to output the joint actions of all agents at the same time, without posing restrictions on how the controlling algorithm performs internal calculations or what information is allowed to be shared across agents. During CT, assuming at each environment timestep agents compute actions according to a sequential pre-determined order before executing them simultaneously, we can view the action computation sequence as a path from the root to a leaf of a tree. At each node of the tree, we consider the preceding agent’s action as the parent node of the current agent. We revisit the idea of UCT exploration [Kocsis and Szepesvári, 2006] proposed in the perfect-information game setting, where the game state is accessible at all nodes, and take inspiration from it to encourage cooperative exploration in MARL. We develop a method called Conditionally Optimistic Exploration (COE). Essentially, COE performs optimism-based exploration by computing the upper confidence bounds of each action for the current agent, conditioned on the visitation count of its parent node (i.e., preceding agents’ actions). To obtain decentralized agents in the decentralized execution (or deployment) phase, COE is not applied, i.e., we disable exploration by removing the optimistic bonus terms.

In the subsequent sections, we first review the background on MARL and the UCT algorithm. We then describe how conditional optimism can be applied to the MARL setting to encourage cooperative exploration. We build COE on commonly used value decomposition methods, and utilize the hash-based counting technique [Fang 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 challenging exploration tasks, and matches baseline performance in general MARL tasks. Our source code is available at https://github.com/chandar-lab/COE

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 $G = (S, A, F, R, \Omega, O, n, \gamma)$, where $S$ is the global state space, $A$ is the action space, $\Omega$ 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 $s \in S$ each agent $i \in \mathcal{N} \equiv \{1, \ldots, n\}$ takes an action $a_i \in A$. The joint action $a = [a_i]_{i=1}^n \in \mathcal{A} \equiv \mathcal{A}^n$ leads to the next state $s'$ sampled from the transition probability $P(s' | s, a) : S \times A \times S \rightarrow [0, 1]$, and obtains a global reward $r$ according to the reward function $R(s, a) : S \times A \rightarrow \mathbb{R}$ shared across all agents. Each agent $i$ has a local policy $\pi_i(a_i | s) : S \times A \rightarrow [0, 1]$. Based on the joint policy $\pi \equiv [\pi_i]_{i=1}^n$, the joint action-value function is defined as $Q_\pi(s, a) = \mathbb{E}_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r^{(t+k)} | s^{(t)} = s, a^{(t)} = a \right]$. The objective is to find a joint policy that maximizes the action-value function.

We consider the partially observable setting, where each
When the trajectory is completed, a reward is received at

Amin et al. [2021] provide a thorough literature survey of

...where

X

2.2 UCT

is to capture some notion of uncertainty (often referred to as

optimism in the face of uncertainty (OFU) principle, the high-level idea

is to capture some notion of uncertainty (often referred to as

novelty or curiosity), and augment the extrinsic reward the

environment emits with an intrinsic reward that quantifies

the uncertainty. For instance, the count-based method [Belle-
mare et al., 2016, Ostrovski et al., 2017, Tang et al., 2017] measures novelty through the number of times the agent ob-

serves a state-action tuple. The learning progress methods

(such as [Houthooft et al., 2016, Pathak et al., 2017, Burda

et al., 2018, Pathak et al., 2019]) capture the curiosity

through the progress of the agent’s knowledge of the

environment. Despite their state-of-the-art performance

on hard exploration tasks in the Atari benchmark [Taiga

et al., 2019], naively applying intrinsic reward methods to

MARL may not work well due to the multi-agent credit

assignment challenge. As the same global reward signal is

shared across all agents, training local policies still relies

on the centralized value/critic function to perform implicit

credit assignment; hence augmenting an intrinsic reward

may still be inefficient to learn structured exploration and

cooperation. Other successful exploration approaches like

BootstrappedDQN [Osband et al., 2016] or Go-Explore

[Ecoffet et al., 2021] are unscaable in MARL due to the

exponentially large state-action space.

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:

\[ B_i = X_i + c \sqrt{\frac{2 \log(p)}{n_i}}, \]

(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 previously taken

actions in the trajectory, 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 original 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 cate-

gory 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 (often referred to as

novelty or curiosity), and augment the extrinsic reward the

Multi-agent exploration. A recent branch of research

proposes to drive multi-agent exploration by promoting col-

laboration among agents through the maximization of the

correlation or influence among agents. The correlation is

commonly realized by the mutual information (MI) of quan-

tities that define or reflect agents’ behaviour, such as the

trajectory history of each agent. For instance, MAVEN [Ma-
hajan 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 col-

laboration by maximizing pairwise MI between every two

agents in the form of intrinsic rewards. For instance, EITI

[Wang et al., 2019] maximizes the MI between one agent’s

state 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

quadratically with the number of agents. Moreover, as

mentioned in the previous paragraph about single-agent explo-

ration, the same multi-agent credit assignment challenge

persists in intrinsic reward MARL methods in general given

a centralized value/critic function is used. Li et al. [2022b]

claims one other important downside of MI-based methods

is the fact that a strong correlation does not necessarily cor-

respond to high-return collaboration, especially when there

exist multiple sub-optimal highly-cooperative strategies as-

associated with the given task. Aside from MI-based meth-

ods, there are other approaches based on different intuitions.

VACL [Chen et al., 2021] leverages variational inference and

automatic curriculum learning to solve sparse-reward coop-

erative MARL challenges. EMC [Zheng et al., 2021] utilizes
We first formulate action computation of MARL as a tree search procedure at each environment timestep. We consider an action sequence of agents \( \{a_1, \ldots, a_n\} \). When agent \( i > 1 \) chooses its action, the computation 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 CONDITIONALLY OPTIMISTIC EXPLORATION

In this section, we introduce our method Conditionally Optimistic Exploration (COE) that effectively drives exploration in cooperative deep MARL. We describe how we can view cooperative MARL as a sequence of tree search iterations. We then discuss how we take inspiration from the tree search method UCT, as well as the challenges to directly applying its idea to MARL. We finally present approaches to address these challenges and the details of our proposed COE method.

4.1 MARL ACTION COMPUTATION AS A TREE

We first formulate action computation of MARL as a tree search procedure at each environment timestep. We consider a value decomposition method to implicitly capture influence among agents and uses the prediction errors of individual Q-value functions as intrinsic rewards. EMC achieves state-of-the-art performance on multiple challenging tasks in the StarCraft Multi-Agent Challenge [Samvelyan et al., 2019] benchmark. Our method also tries to capture agent-wise dependency to guide exploration. Different from MI maximization or EMC, our method captures structured interdependency through each agent’s conditional optimism estimate and performs optimism-based exploration.

**Action Conditioned Learning.** As the learning objective in CTDE is to obtain decentralized agents for execution, previous works commonly assume agents both compute and take actions simultaneously, even during the centralized training phase. A few recent works explicitly consider interdependency and cooperation learned through sequential action computation, where at each timestep 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.

Figure 1: Modelling of MARL as Tree Search Procedure.
network function approximators, which avoids maintaining an empirical reward estimate at every node in every tree. At depth $i$ of each tree, agent $i$ uses the same Q-value function approximator to select action, no matter which subtree the corresponding node is in.

It should be noted that directly applying tree-based algorithms is incompatible with the cooperative MARL setting due to its distinctions with the conventional tree search problem setting. Tree search methods, especially those that rely on visitation count like UCT, typically assume the global state information is accessible at all nodes, whereas the Dec-POMDP setting assumes partial observability of each agent. Accessing full state information enables agents to estimate action values based on the same global state and predecessors’ actions, while in Dec-POMDP agents cannot have such estimates. CTDE also requires agents to act independently at execution time, without conditioning policies on other agents’ actions. To tackle these challenges, we (1) develop an approximate implementation by building conditional optimistic exploration, (2) disable exploration after training to obtain decentralized agents. We present details of our approach in the next subsection, and empirically evaluate it in Section 5.

### 4.2 COE ALGORITHM

We first briefly describe the value decomposition learning paradigm [Sunehag et al. 2017, Rashid et al. 2018]. We then present how we utilize conditional counts on value decomposition to drive optimistic exploration.

Each agent $i$ has an independent Q-network $Q_{idp}^i(\tau_i, a_i; \phi_i)$ parameterized by $\phi_i$. It is important to note that the superscript $idp$ indicates that the $Q_i$ is independent of other agents’ actions, as opposed to a $Q_i$ that is dependent on predecessors’ actions if action computation follows a sequential order. The same naming rule also applies to joint Q-values. A mixing network $\text{Mixer}(\cdot; \theta)$ parameterized by $\theta$ is used to compute the joint Q-values from all individual Q-values:

$$Q_{joint}^i(\tau, a) = \text{Mixer} \left( \left[ Q_{idp}^i(\tau_i, a_i) \right]_{i=1}^n, s; \theta \right).$$

(2)

Individual agent’s action-value networks $Q_{idp}^i$ and the mixing network $\text{Mixer}(\cdot; \theta)$ are trained by minimizing the mean-squared temporal-difference error:

$$L_{idp}^i(\theta_{i=1}^n, \theta) = \mathbb{E}_D \left[ \left( Q_{joint}^i(\tau, a) - y_{idp} \right)^2 \right]$$

(3)

where $y_{idp} = \left( r + \gamma \max_{a'} (Q_{joint}^i(\tau', a')) \right)$ is the update target, and $D$ is the replay buffer containing trajectory data collected by all agents. It is worth noting that by IGM principle, the greedy actions selected by $Q_{idp}^i$'s are the same actions $Q_{joint}^i$ would have taken. As centralized training backpropagates the global reward signal to learn the individual utilities $Q_{idp}^i$'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 computation and learning during the centralized training (CT) phase. For action computation, each agent $i$ selects greedy actions with respect to its conditional optimistic action-value

$$a_i = \arg \max_{a_i} \left\{ Q_{idp}^i(\tau_i, a_i') + c_{act} \sqrt{\frac{2 \log(N(s, a_{<i}))}{N(s, a_{<i}, a_i')}} \right\},$$

(4)

where $c_{act} \in \mathbb{R}_+$ is a hyper-parameter controlling the scale of optimism, $N(\cdot)$ denotes the visitation count. Note that counting is performed in the global state space, thanks to centralized training. The learning framework of COE is illustrated in Figure [2].

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

$$y_{idp} = \left( r(s, a) + \frac{c_{rew}}{\sqrt{N(s, a)}} \right) + \gamma \max_{a} \left[ \text{Mixer} \left( \left[ Q_{idp}^i(\tau_i', a_i') + \frac{c_{boot}}{\sqrt{N(s', a_{<i}', a_i')}} \right]_{i=1}^n \right) \right].$$

(5)

where $c_{rew}, c_{boot} \in \mathbb{R}_+$ 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 the corresponding count is incremented, 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 state-actions, this optimistic bootstrap target also captures uncertainty from subsequent agents and future timesteps. The idea of learning optimistic Q-values originates from theoretical works such as [Jin et al. 2018, 2020, Yang et al. 2020], and has been extended to deep RL recently (e.g., 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 may have decayed to negligible magnitude, are removed, and agents take independent actions according to $Q_{idp}^i$'s only.

To apply COE to deep MARL tasks, we need to approxi-
Algorithm 1 Conditionally Optimistic Exploration

Initialize parameters $\phi, \theta$
Visitation count $N(s, a) \leftarrow 0, \forall (s, a) \in S \times A$
Replay buffer $D \leftarrow \{\}$

for each episode $m = 1, \ldots, M$ do
  for each environment timestep $t = 1, \ldots, T$ do
    for agent $i = 1, \ldots, n$ do
      Compute action $a^{(i)}(t)$ according to Equation (4)
    end for
    $s(t+1) \sim P(s'|s(t), a^{(i)})$, $r(t) = r(s(t), a^{(i)})$
    $N(s^{(i)}, a^{(i)}) \leftarrow N(s^{(i)}, a^{(i)}) + 1$
    $D \leftarrow D \cup \{(s^{(i)}, a^{(i)}, r^{(i)}, s^{(i+1)})\}$
    Perform a gradient update on Equation (x)
  end for
end for

mate 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 $a$, denoted by $N(s, a)$. For each agent $i$, the count up to its action $a_i$ satisfies $N(s, a_{<i}, a_i) = \sum_{a_{>i}} N(s, a_{<i}, a_i, a_{>i}) = \sum_{a_{>i}} N(s, a_{<i}, a_i, a_{>i})$, where $a_{<i}$ and $a_{>i}$ denote the joint actions computed 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 share the same $a_{<i}$ at state $s$. Details about SimHash counting are presented in Appendix A.

5 EXPERIMENTS

In this section, we evaluate COE on cooperative MARL tasks across three commonly used benchmarks: Multi-agent Particle Environments (MPE) [Lowe et al., 2017, Mor-datch and Abbeel, 2018], Level-Based Foraging (LBF) [Albrecht and Ramamoorthy, 2015, Christianos et al., 2020, Papoudakis et al., 2020], and StarCraft Multi-Agent Challenge (SMAC) [Samvelyan et al., 2019]. These tasks can be categorized to two sets based on challenges they exhibit: (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 repeated multi-player game problem.

5.1 EVALUATION SETUP

We perform evaluation 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 Sparse-Tag and Sparse Spread from MPE, and four tasks with different configurations from LBF. (2) Tasks that generally assess multi-agent coordination. This includes Adversary in MPE, and an easy task 2s-vs-1sc and a hard task 3s-vs-5z in SMAC. 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.

It is important to note that COE is applicable to any value decomposition approach. To promote fair comparisons in
Table 1: Average Returns and 95% Confidence Interval for All Four Algorithms, and Average Win-rates for SMAC Tasks.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Adversary</th>
<th>Sparse Tag</th>
<th>Sparse Spread</th>
<th>10x10-3p-3F</th>
<th>15x15-3p-5F</th>
<th>15x15-4p-3F</th>
<th>15x15-4p-5F</th>
<th>2s-vs-1sc</th>
<th>3s-vs-5z</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPE</td>
<td>17.77 ± 0.71</td>
<td>0.65 ± 0.09</td>
<td>0.79 ± 0.09</td>
<td>0.71 ± 0.05</td>
<td>0.20 ± 0.02</td>
<td>0.41 ± 0.06</td>
<td>0.30 ± 0.02</td>
<td>0.79 ± 0.01</td>
<td>0.45 ± 0.09</td>
</tr>
<tr>
<td>LBF</td>
<td>16.73 ± 0.83</td>
<td>0.43 ± 0.06</td>
<td>0.41 ± 0.18</td>
<td>0.68 ± 0.03*</td>
<td>0.12 ± 0.02</td>
<td>0.25 ± 0.07</td>
<td>0.23 ± 0.04</td>
<td>0.83 ± 0.04</td>
<td>0.08 ± 0.14</td>
</tr>
<tr>
<td>SMAC</td>
<td>10x10-3p-3F</td>
<td>17.88 ± 0.74*</td>
<td>10.63 ± 1.58</td>
<td>6.26 ± 1.52</td>
<td>14.11 ± 2.36*</td>
<td>10.94 ± 2.09</td>
<td>9.66 ± 2.36</td>
<td>0.82 ± 0.08*</td>
<td>0.83 ± 0.02</td>
</tr>
<tr>
<td></td>
<td>15x15-3p-5F</td>
<td>17.78 ± 1.26*</td>
<td>17.81 ± 1.58</td>
<td>14.11 ± 2.36*</td>
<td>11.74 ± 1.87</td>
<td>10.94 ± 2.09</td>
<td>6.26 ± 1.58</td>
<td>0.82 ± 0.08*</td>
<td>0.83 ± 0.02</td>
</tr>
<tr>
<td></td>
<td>15x15-4p-3F</td>
<td>15x15-4p-5F</td>
<td>10x10-3p-3F</td>
<td>15x15-3p-5F</td>
<td>15x15-4p-3F</td>
<td>15x15-4p-5F</td>
<td>2s-vs-1sc</td>
<td>3s-vs-5z</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Episodic Returns and 95% Confidence Interval for All Algorithms in All Tasks except Adversary, with Sparse-Reward Tasks Marked in Bold.

In our experiments, we build all exploration methods on the QMIX backbone [Rashid et al., 2018] with the same network architecture and configurations. For the same reason, we implement the canonical version of all methods where the only different 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 the 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 tasks 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. The maximum average return is highlighted 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 of the other algorithms in every task. The return values are marked 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. In addition to average returns, the table summarizing maximum returns over training is presented in Appendix D.

The results in Table 1 and Figure 3 show that COE significantly outperforms other baselines in sparse-reward tasks that require efficient exploration. Particularly COE has higher sample efficiency in difficult LBF domains. In the very 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 tasks demonstrate the effectiveness of conditional-optimism guided exploration.

In the general MARL coordination tasks, COE has similar performance to 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 in practice.

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** (conditional optimism + conditional Q, without centralized training): 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. 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** (conditional optimism + conditional Q + centralized training): We add a QMIX mixer to COE-Cond-IQ to enable centralized Q-value training. The same mixer computes the centralized Q-value $Q_{\text{idp}}^{\text{dep}}$ for independent networks and $Q_{\text{joint}}^{\text{dep}}$ for dependent networks.

To evaluate the contributions of conditional optimism, we propose the following ablation variants that use non-conditional optimism:

(1) **UCB-Ind** (independent optimism + independent Q): Similar to COE, each agent performs UCB-based exploration except that optimism is not conditioned on other agents’ actions.

(2) **UCB-Cen** (centralized optimism + independent Q): Agents receive UCB optimism only through the intrinsic reward $\frac{c_{\text{in}}}{\sqrt{N(x,a)}}$ during centralized training.

We follow the same evaluation protocol described in Section 5.2 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 description of 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 conditional optimism guides more steady performance improvement.

5.4 DIDACTIC PROBLEM

A rigorous application of tree-based methods 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 a repeated multi-player game problem to re-demonstrate conditional optimism is important whereas conditional value estimation could be unnecessary.

We consider the cooperative multi-player game problem, where the common payoff is based on the joint action of a group of agents. Suppose we have $n$ agents, each has $k$ actions. In our didactic Bernoulli game, only one out of $k^n$ joint actions is optimal with payoff distribution $B(p = 0.9)$, and all other joint actions are sub-optimal with payoff distribution $B(p = p_0)$, where the sub-optimality value $p_0$
We test four UCB-based algorithm variants:
(1) **DepRew-DepOpt**: it performs UCT exploration as in Equation (1), where both payoff estimates and count-based optimism are dependent on prior agents’ joint action;
(2) **IndRew-DepOpt**: each agent maintains its payoff estimates independently, but the optimism is dependent on predecessors;
(3) **IndRew-IndOpt**: both payoff estimates and optimism are independent;
(4) **UCB-Cen**: one UCB learner whose action space is the cartesian product of all agents’ action sets. This variant is merely for performance comparison because it is not factorizable to decentralized agents.

We run experiments on a game 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 4. We evaluate algorithms with two metrics, the expected regret shown in the left column, which is preferably lower, and the percentage of optimal joint action being selected shown in the right column, which is preferably higher. Results show that both DepRew-DepOpt and IndRew-DepOpt quickly converge to the optimal policy across different sub-optimality settings. Their learning curves overlap when sub-optimality is 0.0 or 0.4, and they have only marginal difference when sub-optimality equals 0.8. This suggests that in this game task, conditional optimism robustly drives efficient cooperative exploration, regardless of whether the payoff estimates are learned independently or not. 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 MARL ablation results from 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 costly 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.

Acknowledgements

We acknowledge the computational resources provided by the Digital Research Alliance of Canada. Janarthanan Rajendran acknowledges the support of the IVADO postdoctoral fellowship. Sarath Chandar acknowledges the support of the Canada CIFAR AI Chair program and an NSERC Discovery Grant.

References


Pengyi Li, Hongyao Tang, Tianpei Yang, Xiaotian Hao, Tong Sang, Yun Zheng, Jianye Hao, Matthew E Taylor,


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


