In-Context Fully Decentralized Cooperative Multi-Agent Reinforcement Learning

Chao Li[1 2] **Bingkun Bao**[1 3 *] **Yang Gao**[4 5]
[1] School of Computer Science, Nanjing University of Posts and Telecommunications
[2] State Key Laboratory of Tibetan Intelligence [3] Peng Cheng Laboratory
[4] School of Intelligent Science and Technology, Nanjing University
[5] State Key Laboratory for Novel Software Technology, Nanjing University
`chaoli@njupt.edu.cn, bingkunbao@njupt.edu.cn, gaoy@nju.edu.cn`

Abstract

In this paper, we consider fully decentralized cooperative multi-agent reinforcement learning, where each agent has access only to the states, its local actions, and the shared rewards. The absence of information about other agents' actions typically leads to the non-stationarity problem during per-agent value function updates, and the relative overgeneralization issue during value function estimation. However, existing works fail to address both issues simultaneously, as they lack the capability to model the agents' joint policy in a fully decentralized setting. To overcome this limitation, we propose a simple yet effective method named Return-Aware Context (RAC). RAC formalizes the dynamically changing task, as locally perceived by each agent, as a contextual Markov Decision Process (MDP), and addresses both non-stationarity and relative overgeneralization through return-aware context modeling. Specifically, the contextual MDP attributes the non-stationary local dynamics of each agent to switches between contexts, each corresponding to a distinct joint policy. Then, based on the assumption that the joint policy changes only between episodes, RAC distinguishes different joint policies by the training episodic return and constructs contexts using discretized episodic return values. Accordingly, RAC learns a context-based value function for each agent to address the non-stationarity issue during value function updates. For value function estimation, an individual optimistic marginal value is constructed to encourage the selection of optimal joint actions, thereby mitigating the relative overgeneralization problem. Experimentally, we evaluate RAC on various cooperative tasks (including matrix game, predator and prey, and SMAC), and its significant performance validates its effectiveness.

1 Introduction

In recent years, cooperative multi-agent reinforcement learning (MARL) has witnessed great advances in both algorithms (*e.g.,* value decomposition [1–4] and multi-agent policy gradient [5–8] methods) and applications (*e.g.,* autonomous vehicles [9], urban traffic management [10], and vaccine allocation [11]). Most of these advances heavily rely on the centralized training (often with decentralized execution) paradigm, where global information, particularly the joint actions of all agents, is available during training. However, direct access to other agents' actions is often unattainable in real-world settings, such as industrial scenarios where multiple robots from different companies may withhold action information due to privacy concerns or limited communication capabilities. In such cases, the *fully decentralized learning* becomes essential, where each agent learns solely from its local experiences, without access to the actions of other agents, during both training and execution.

[*]Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-4943

However, developing efficient coordinated policies under the fully decentralized learning paradigm is challenging, as the lack of access to other agents' actions typically leads to non-stationarity during per-agent value function updates and relative overgeneralization during value function estimation. In detail, since treating other agents as part of the environment, the local task dynamics perceived by each decentralized agent becomes non-stationary due to the evolving policies of other agents. This non-stationarity undermines the convergence of value function updates and is further exacerbated by experience replay [12]. In addition, the value estimations of each agent's local actions may be biased by other agents' exploratory or sub-optimal action selections, hindering agents from selecting optimal joint actions, a problem known as relative overgeneralization [13]. Consequently, fully decentralized learning often suffers from low efficiency and sub-optimal solutions, limiting efficient coordination.

Accordingly, we divide existing value-based MARL approaches into two major categories. The first category of works [1, 2, 14–16] primarily focuses on the non-stationarity issue. These methods either directly access other agents' actions, design multi-agent importance sampling weights, fingerprints, or ideal transition probabilities to ensure stationary transitions, or adopt alternating policy updates among agents to maintain stationary policy updates. The second category [3, 4, 17–22] mainly targets the relative overgeneralization issue, typically by rectifying the factored global action value function or employing optimistic or lenient value updates to facilitate the selection of optimal joint actions. Although both issues stem from the inaccessibility of other agents' actions, current fully decentralized MARL methods typically address either the non-stationarity issue or the relative overgeneralization problem in isolation, due to their inability to model the agents' joint policy in a fully decentralized setting. As a result, they fall short of simultaneously resolving both challenges.

To overcome this limitation, we propose a simple yet effective method named Return-Aware Context (RAC). RAC formalizes the dynamically changing task locally perceived by each agent as a contextual Markov Decision Process (MDP) [23], and tackles non-stationarity and relative overgeneralization from a context modeling perspective. Specifically, under the contextual MDP, RAC attributes each agent's non-stationary local task dynamics to switches between contexts, each corresponding to a distinct joint policy. Subsequently, to enable fully decentralized modeling of agents' joint policies, RAC leverages training episodic returns to differentiate among different joint policies, constructing context representations using discretized episodic return values. This return-aware modeling operates under the assumption that joint policy changes occur only between episodes, an assumption commonly adopted in existing MARL methods, where all agents update their policies at the end of each episode. Accordingly, RAC learns a context-based value function for each agent to address the non-stationarity problem during value function updates. For value function estimation, RAC derives an individual optimistic marginal value regarding all possible contexts for each agent. This guides agents toward selecting optimal joint actions and effectively mitigates the relative overgeneralization problem.

Empirically, we evaluate RAC against multiple baselines across various cooperative tasks, including matrix game, predator and prey, and StarCraft Multi-Agent Challenge (SMAC) [24]. The experimental results show that RAC achieves superior performance in both learning efficiency and final outcomes, demonstrating its effectiveness in enhancing fully decentralized cooperative policy learning.

2 Related Work

In this section, we review current value-based MARL methods, dividing them into two categories: those addressing the non-stationarity problem and those tackling the relative overgeneralization issue.

The first category of works addresses the non-stationarity problem by promoting stationary transitions or stabilizing policy updates. Specifically, canonical value decomposition methods such as VDN [1] and QMIX [2] assume direct access to agents' joint actions to achieve stationary transitions. However, these methods often suffer from relative overgeneralization due to the representational limitation of the factored global action value function [25]. For independent Q-learning (IQL) [26] agents, the multi-agent importance sampling technique [14] assumes access to other agents' policies and constructs importance weights to decay obsolete data during experience replay. Multi-agent Fingerprints [14] employ other agents' training iteration numbers or exploration rates to estimate their policies, and augment per-agent local transitions with these estimates. However, such direct access to other agents' information is typically impractical. I2Q [16] avoids this limitation by constructing stationary ideal transitions by selecting the next state with the highest QSS value [27], enabling IQL agents to learn optimal joint policies in a fully decentralized manner. In comparison to I2Q, which addresses both non-stationarity and relative overgeneralization by shaping ideal state transitions, this work aims for a novel return-aware perspective to tackle both issues. In addition, to ensure stationary policy updates, MA2QL [15] enforces sequential policy updates among IQL agents, where each agent updates its policy while keeping the others fixed. Although this strategy alleviates non-stationarity, it often leads to low learning efficiency due to limited parallelism in policy updates.

The second category of works tackles the relative overgeneralization issue by rectifying the factored global action value function, or updating per-agent local value functions in an optimistic or lenient manner. Specifically, for value decomposition methods, weighted QMIX [17] places more weights on potentially optimal joint actions to exclusively recover correct value functions for them. QTRAN [3] and QPLEX [4] introduce additional terms to correct the discrepancy between the learned factored global action value function and the true joint ones. For fully decentralized IQL agents, distributed Q-learning [18] employs an optimistic decentralized value function, where each agent assumes that other agents always select their cooperative actions. This optimism encourages agents to identify and select local cooperative actions, thus addressing the relative overgeneralization issue. However, being highly optimistic makes distributed Q-learning vulnerable to stochasticity. Hysteretic Q-learning [19, 20] avoids high optimism by utilizing two learning rates to update per-agent value functions with positive and negative temporal difference errors respectively. Lenient learning [21, 22] shifts from optimistic value estimations to normal value estimations using decreasing lenience. Although promising, the neglect of non-stationarity often limits these approaches from efficient policy learning in practice.

In summary, existing fully decentralized MARL methods face challenges in simultaneously addressing both non-stationarity and relative overgeneralization issues, due to their inability to model the agents' joint action or policy in a fully decentralized setting. To overcome this limitation, this work proposes to formalize per-agent local task dynamics by the contextual MDP, and then introduces return-aware context modeling to tackle both issues, thereby effectively enhancing fully decentralized learning.

3 Preliminary

In this section, we present the multi-agent task addressed by this work, and review the non-stationarity and relative overgeneralization problems in decentralized learning, along with the contextual MDP.

3.1 Multi-Agent Markov Decision Process

We consider a fully cooperative task that can be modeled as an Multi-agent Markov Decision Process (MMDP) $\langle N, S, \boldsymbol{A}, P, R, \gamma \rangle$, where $N = \{1, 2, \dots, n\}$ represents the agent set and S denotes the state space. $\boldsymbol{A} = A^1 \times A^2 \times \dots \times A^n$ is the joint action space and A^i denotes the local action space of agent $i \in N$. At each time step t, each agent i observes the state $s_t \in S$ and selects its local action a_t^i using its policy $\pi^i(a_t^i|s_t)$. Based on all agents' joint action $\boldsymbol{a}_t = (a_t^1, a_t^2, \dots, a_t^n)$, the environment transits to the next state s_{t+1} according to the state transition function $P(s_{t+1}|s_t, \boldsymbol{a}_t)$ and provides all agents with the shared reward r_t based on the reward function $R(s_t, \boldsymbol{a}_t)$. The goal of all agents is to learn the optimal joint policy $\boldsymbol{\pi}^* = (\pi^{1,*}, \pi^{2,*}, \dots, \pi^{n,*})$ that maximizes the expected cumulative return $\mathbb{E}_{\boldsymbol{\pi},P}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where γ is a discount factor. Additionally, we also evaluate our approach under the partially observable setting, as detailed in Sec. 5 and Appendix A.

We consider the fully decentralized learning, where each agent i observes only the state s_t, its local action a_t^i, and the shared reward r_t. From the perspective of each agent i, the perceived task can be modeled as an Markov Decision Process (MDP) $\langle S, A^i, P^i, R^i, \gamma \rangle$ with dynamics defined below:

$$\begin{aligned} P^i(s_{t+1}|s_t, a_t^i) &= \sum\nolimits_{a_t^{-i}} \pi^{-i}(a_t^{-i}|s_t) P(s_{t+1}|s_t, \boldsymbol{a}_t), \\ R^i(s_t, a_t^i) &= \sum\nolimits_{a_t^{-i}} \pi^{-i}(a_t^{-i}|s_t) R(s_t, \boldsymbol{a}_t), \end{aligned} \tag{1}$$

where π^{-i} and a_t^{-i} denote the joint policy and the joint action of all other agents $-i$ except agent i.

Non-Stationarity. Eq. (1) illustrates that each agent i's local task dynamics, represented by P^i and R^i, depends on the joint policy π^{-i} of other agents $-i$. As these agents continuously update their policies, the per-agent local task dynamics becomes non-stationary. Furthermore, when employing off-policy experience replay, the sampled transitions from the replay buffer can be viewed as following P^i and R^i with respect to the average joint policy $\bar{\pi}^{-i}$ of other agents $-i$ over the course of training. This non-stationarity disrupts the convergence guarantee of each agent i's value function updates.

https://doi.org/10.52202/085713-4943

Relative Overgeneralization. Due to the lack of access to other agents' actions, the value estimations of per-agent local cooperative actions may be biased by other agents' exploratory or sub-optimal action selections and thus lead to sub-optimal joint actions, a problem known as relative overgeneralization.

Specifically, the local value function $Q^i(s_t, a_t^i)$ of each decentralized agent i can be regarded as a projection of the joint action value function $Q(s_t, a_t^i, a_t^{-i})$. IQL follows an average-based projection:

$$Q^{i,\boldsymbol{\pi}}(s_t, a_t^i) = \sum\nolimits_{a_t^{-i}} \pi^{-i}(a_t^{-i}|s_t) Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i}), \tag{2}$$

where $Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i})$ is the joint action value function under a given joint policy $\boldsymbol{\pi} = (\pi^i, \pi^{-i})$. It is obvious that the average-based projection is easily affected by other agents' sub-optimal actions and suffers from the relative overgeneralization. In contrast, the optimistic projection is defined below:

$$Q^{i,opt}(s_t, a_t^i) = \max\nolimits_{a_t^{-i}} Q^*(s_t, a_t^i, a_t^{-i}), \tag{3}$$

where $Q^*(s_t, a_t^i, a_t^{-i})$ represents the joint action value function of an optimal joint policy $\boldsymbol{\pi}^*$. The optimistic projection assumes that other agents $-i$ always select their cooperative local actions, thus eliminating the impact of other agents' non-cooperation. Hysteretic Q-learning directly approximates $Q^{i,opt}(s_t, a_t^i)$ by an optimistic value update. In contrast, our approach estimates it using an optimistic marginal value derived from a context-based value function. We detail this distinction in Appendix. B.

3.2 Contextual Markov Decision Process

A contextual MDP is defined as a tuple $\langle \mathcal{C}, S, A, M(c) \rangle$, where $\mathcal{C}$ denotes the context space, S is the state space, and A is the action space. The function $M(c)$ maps each context $c \in \mathcal{C}$ to a specific MDP $\langle S, A, P_c, R_c, \gamma \rangle$, thereby defining a family of MDPs that share the same state and action spaces but differ in their transition dynamics and reward functions. In this work, we use the contextual MDP to model each agent's varying local task dynamics, which is determined by other agents' joint policies.

4 Methodology

In this section, we provide a detailed explanation of our method, Return-Aware Context (RAC). We begin by introducing the task formalization based on contextual MDP, and then delve into the process of modeling contexts using the training episodic return. Subsequently, for each decentralized agent, we propose to learn a context-based value function and derive an individual optimistic marginal value aimed at selecting the optimal joint actions. Finally, we summarize the overall learning procedure.

4.1 Task Formalization

As detailed in Sec. 3.1, from the perspective of each agent i, the perceived task can be modeled as an MDP $\langle S, A^i, P^i, R^i, \gamma \rangle$, where the state transition dynamics P^i and reward function R^i depend on the joint policy π^{-i} of the other agents $-i$. For all possible π^{-i}, the local task experienced by agent i can be further decomposed into a family of MDPs that share the same state and action spaces but differ in their transition and reward functions, each conditioned on a specific π^{-i}. By associating each context $c \in \mathcal{C}$ with a corresponding π^{-i}, we propose to formalize the local task of agent i as a contextual MDP, which is defined as follows:

$$\langle \mathcal{C}, S, A^i, M(c) \rangle, \;\; \text{where}\, M(c) : c \rightarrow \langle S, A^i, P_c^i, R_c^i, \gamma \rangle, \tag{4}$$

where S is the state space and A^i is agent i's local action space. The dynamics $P_c^i(s_{t+1}|s_t, a_t^i, c)$ and $R_c^i(s_t, a_t^i, c)$ are explicitly conditioned on the context $c \in \mathcal{C}$, each corresponding to a unique π^{-i}.

Within the contextual MDP formalization, when each agent operates in a fully decentralized manner, the task dynamics is determined by an underlying context, which is inherently tied to other agents' current joint policy. When an agent encounters different contexts at different time steps, the same state and local action may result in different next states and rewards. Consequently, the lack of context information prevents agent i from fully capturing the task dynamics, leading to the non-stationarity.

This contextual MDP formalization attributes non-stationarity to switches between contexts, and provides a principled framework to address this challenge by explicit context modeling. The context can be instantiated as either an estimate of other agents' current joint policy or a real-time representation

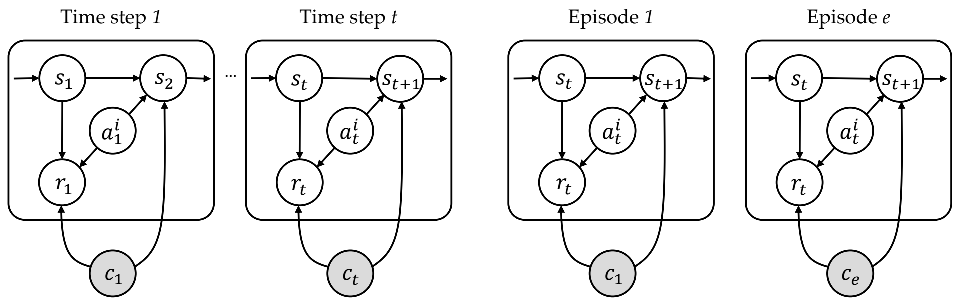

Figure 1: Two cases of context changes. Case 1 refers to the scenario where other agents update their joint policy at every time step or every few time steps. For example, in the left plot, the dynamics is determined by c_1 at time step 1 and by c_t at time step t. Case 2 refers to the scenario where agents update policies only between episodes; for instance, c_1 persists throughout episode 1 and c_e persists until the termination of episode e, as shown in the right plot. Empty and solid circles respectively represent observable and unobservable stochastic variables.

of the current local task dynamics distribution. By augmenting per-agent state-local-action pair with the inferred context, the resulting transitions become stationary, thereby enabling efficient learning.

The frequency of context changes determines the degree of non-stationarity. As illustrated in Fig. 1, we define two cases: (1) **Context changes between time steps**, where other agents update their joint policy every (several) time steps, leading to significant non-stationarity; and (2) **Context changes between episodes**, where agents keep their policies fixed within each episode and update them only between episodes, resulting in moderate non-stationarity. The latter case is widely adopted by existing MARL methods. In this work, we take a step toward explicitly modeling contexts within it.

4.2 Context Modeling and Usage

Based on case (2) that agents update their policies only between episodes, we propose to represent all agents' fixed joint policy within each episode by the training episodic return. Given a local episodic trajectory $\tau^i = (s_0, a_0^i, r_0, s_1, a_1^i, r_1, s_2, \ldots, s_H)$ of agent i, its generation probability is defined as:

$$p(\tau^i) = p(s_0) \prod_{t=0}^{H-1} \underbrace{\pi^i(a_t^i|s_t)\pi^{-i}(a_t^{-i}|s_t)}_{①} \underbrace{P(s_{t+1}|s_t, \boldsymbol{a}_t)R(s_t, \boldsymbol{a}_t)}_{②}, \tag{5}$$

where term ① represents all agents' joint policy and term ② that contains both the state transition probability $P(s_{t+1}|s_t, \boldsymbol{a}_t)$ and reward emission probability $R(s_t, \boldsymbol{a}_t)$ is entirely determined by the environment. For fixed joint policy during each episode, the generated local trajectory can be viewed as a Monte Carlo sampling of all possible trajectories induced by this joint policy under the stochastic or deterministic environment. The associated episodic return implicitly represents the agents' joint policy in the return space. For joint policies that generate similar local trajectories for each agent, the episodic returns will provide them with nearby representations, and vice versa.

Accordingly, we use the episodic return $R(\tau^i)$ of per-agent local trajectory τ^i to estimate the agents' current joint policy $\boldsymbol{\pi} = (\pi^i, \pi^{-i})$, and refer to it as the context. Although each agent's local task dynamics solely depends on other agents' joint policy π^{-i}, achieving a fully decentralized estimation of π^{-i} is intractable in practice and we instead turn to directly estimate $\boldsymbol{\pi}$ using the episodic return $R(\tau^i)$. This enables fully decentralization and greater scalability, eliminating the need for each agent i to separately estimate other agents $-i$' policies. We empirically demonstrate that estimating $\boldsymbol{\pi}$ effectively mitigates the non-stationarity issue, as discussed in Sec. 5.

Specifically, during training, each agent respectively calculates the episodic returns for sampled local trajectories. To avoid too many continuous values, we discretize the training episodic return into m intervals. Each interval κ covers a range of $[\frac{R_{\max}-R_{\min}}{m} \times \kappa, \frac{R_{\max}-R_{\min}}{m} \times (\kappa+1))$, where $R_{\max}$ and $R_{\min}$ respectively denote the maximum and minimum episodic returns. For a training episodic

 https://doi.org/10.52202/085713-4943

return $R(\tau^i)$, the corresponding interval index κ can be calculated as follows:

$$\kappa = \lfloor \frac{R(\tau^i)}{\frac{R_{\max}-R_{\min}}{m}} \rfloor, \tag{6}$$

where $\lfloor \cdot \rfloor$ is the floor function and κ is enforced to be an integer between 0 and $m-1$.

Context-Based Value Function. We use the one-hot encoding c_κ of κ as the context of trajectory τ^i, and learn a value function $Q^i(s_t, c_\kappa, a_t^i)$ for each agent i, which is conditioned on the state, the local action and the context. Based on the augmented transition $(s_t, a_t^i, c_\kappa, r_t, s_{t+1})$, the Q^i is updated by:

$$\mathcal{L}_{\mathrm{C}}(\theta^i) = \mathbb{E}_{(s_t, a_t^i, c_\kappa, r_t, s_{t+1}) \sim \mathcal{D}^i} \left[(r_t + \gamma \max_{a_{t+1}^i} Q^i(s_{t+1}, c_\kappa, a_{t+1}^i) - Q^i(s_t, c_\kappa, a_t^i))^2 \right], \tag{7}$$

where θ^i denotes the parameters of Q^i and we sample batches of trajectories from agent i's replay buffer $\mathcal{D}^i$ to conduct the update. c_κ enables both stationary transitions and value function updates.

Individual Optimistic Marginal Value. There are still two issues during value estimation: (1) The relative overgeneralization limits agents from identifying and selecting their local cooperative actions; and (2) Each agent can not select actions using the context-based value function within the episode, where the context based on episodic return is available only when the episode terminates. To address these two issues, we propose the individual optimistic marginal value as defined below:

$$\phi^i(s_t, a_t^i) = \max_{c_\kappa \in \mathcal{C}} Q^i(s_t, c_\kappa, a_t^i), \tag{8}$$

where we shape marginal value functions of per-agent local actions using the corresponding maximum context-based value estimations across all possible contexts. Eq. (8) adheres to an optimistic belief that other agents always select their optimal cooperative actions, and thus the individual optimistic marginal value regarding per-agent local action a_t^i can reach the optimal value $\max_{c_\kappa \in \mathcal{C}} Q^i(s_t, c_\kappa, a_t^i)$. The individual optimistic marginal value discards the effect caused by exploratory or sub-optimal action selections of other agents, and accordingly enables each agent to select its local cooperative action, addressing the relative overgeneralization issue and leading to the optimal joint policy.

4.3 Overall Learning Procedure

Based on the learned individual optimistic marginal value, each agent i can select its local cooperative action by $\arg\max_{a_t^i} \phi^i(s_t, a_t^i) = \arg\max_{a_t^i} \max_{c_\kappa \in \mathcal{C}} Q^i(s_t, c_\kappa, a_t^i)$. However, accurately learning the context-based value function necessitates a comprehensive coverage across the entire return space, which is unavailable during the early training process and leads to sub-optimal outcomes. To address this issue, we separately learn a value function $Q_{\mathrm{S}}^i(s_t, a_t^i)$ and update it by the following TD loss:

$$\mathcal{L}_{\mathrm{TD}}(\sigma^i) = \mathbb{E}_{(s_t, a_t^i, r_t, s_{t+1}) \sim \mathcal{D}^i} \left[(r_t + \gamma \max_{a_{t+1}^i} Q_{\mathrm{S}}^i(s_{t+1}, a_{t+1}^i) - Q_{\mathrm{S}}^i(s_t, a_t^i))^2 \right], \tag{9}$$

where σ^i denotes the parameters of Q_{S}^i. We make each agent select actions based on its $Q_{\mathrm{S}}^i(s_t, a_t^i)$ to generate informative transitions during the early training process, and accordingly enable an efficient update of $Q^i(s_t, c_\kappa, a_t^i)$ following Eq. (7). Furthermore, we propose an auxiliary supervision loss:

$$\mathcal{L}_{\mathrm{sup}}(\sigma^i) = \mathbb{E}_{(s_t, a_t^i, r_t, s_{t+1}) \sim \mathcal{D}^i} [D_{\mathrm{KL}}[\pi^i(\cdot|s_t) || \pi_{\mathrm{S}}^i(\cdot|s_t))]], \tag{10}$$

where $\pi^i(\cdot|s_t)$ and $\pi_{\mathrm{S}}^i(\cdot|s_t)$ respectively denote the Boltzmann policy with respect to $\phi^i(s_t, a_t^i)$ and $Q_{\mathrm{S}}^i(s_t, a_t^i)$, which are defined as follows:

$$\pi^i(a_t^i|s_t) = \frac{\exp(\phi^i(s_t, a_t^i))}{\sum_{a^i \in A^i} \exp(\phi^i(s_t, a^i))}, \quad \pi_{\mathrm{S}}^i(a_t^i|s_t) = \frac{\exp(Q_{\mathrm{S}}^i(s_t, a_t^i))}{\sum_{a^i \in A^i} \exp(Q_{\mathrm{S}}^i(s_t, a^i))}. \tag{11}$$

The intuition behind Eq. (10) is that each agent's π^i is capable of addressing both non-stationarity and relative overgeneralization, and we make the decision policy π_{S}^i to imitate it. As a result, we update $Q_{\mathrm{S}}^i(s_t, a_t^i)$ using the loss $\mathcal{L}_{\mathrm{S}}(\sigma^i) = \mathcal{L}_{\mathrm{TD}}(\sigma^i) + \beta \mathcal{L}_{\mathrm{sup}}(\sigma^i)$, where β is a scaling factor.

As presented in Algorithm 1, the learning procedure of RAC is as follows. We begin by initializing Q^i, Q_{S}^i for each agent $i \in N$. During each episode, each agent i selects its local action based on Q_{S}^i and the local episode data is stored into its replay buffer $\mathcal{D}^i$. For agents with ϵ-greedy policy, we keep ϵ fixed during each episode and decrease it between episodes, which is adopted by PyMARL [24] by default. This approach enables fixed policy within each episode, and is consistent with the proposed case (2) (context changes between episodes). During training, for each agent i, we sample batches of episodes from $\mathcal{D}^i$, and construct the context c_κ for each episodic trajectory τ^i based on the episodic return $R(\tau^i)$. Finally, we calculate the loss functions $\mathcal{L}_{\mathrm{C}}(\theta^i), \mathcal{L}_{\mathrm{S}}(\sigma^i)$, and update all components by gradient descent. The entire procedure continues until the maximum training episode is reached.

Algorithm 1: Return-Aware Context (RAC)

```
Initialize necessary hyper-parameters
foreach agent i ∈ N do
    Initialize parameters θ^i, σ^i of Q^i, Q^i_S
if the maximum training episode is not reached then
    for Each episode do
        foreach time step t do
            foreach agent i ∈ N do
                Select its local action a^i_t based on Q^i_S(s_t, a^i_t)
    Store each agent i's episode into its replay buffer D^i
    Decrease ϵ for each agent if using ϵ-greedy policy
    if train then
        foreach agent i ∈ N do
            Sample batches of episodes from D^i
            Calcualte R(τ^i) for each episodic trajectory τ^i
            Construct κ, c_κ based on Eq. (6)
            Calculate L_C(θ^i), L_S(σ^i) based on Eq. (7), (9), (10)
            Update all components with gradient descent
```

5 Experiment

In this section, we design experiments to answer the following two questions: (1) Can RAC benefit fully decentralized learning by addressing both non-stationarity and relative overgeneralization? (See Sec. 5.1) (2) If so, which component contributes the most to its performance gain? (See Sec. 5.2)

For question (1), we compare RAC against fully decentralized baselines, IQL [26], Hysteretic Q-learning [19], and I2Q [16], on Matrix Game, Predator and Prey, and StarCraft Multi-Agent Challenge (SMAC). For question (2), we conduct ablation studies to assess the effect of each component of RAC. All results are illustrated with the median performance and the standard error across five random seeds. More details about experimental setups and hyper-parameters can be found in Appendix C.

A^1 \ A^2	a^1	a^2	a^3
a^1	8	-12	-12
a^2	-12	6	0
a^3	-12	0	6

(a) Payoff matrix.

A^1 \ κ	0	1	2	3
$a^1(7.99)$	-12	$\cancel{-5.8}$	$\cancel{2.42}$	7.99
$a^2(6.02)$	-11	0.04	6.02	$\cancel{-5.2}$
$a^3(5.94)$	-11	0.03	5.94	$\cancel{-4.1}$

(b) $Q^1(s, c_\kappa, a)$ and $\phi^1(s, a)$.

A^2 \ κ	0	1	2	3
$a^1(7.99)$	-11	$\cancel{-6.0}$	$\cancel{2.27}$	7.99
$a^2(6.01)$	-11	0.02	6.01	$\cancel{-5.2}$
$a^3(5.94)$	-12	0.04	5.94	$\cancel{-3.8}$

(c) $Q^2(s, c_\kappa, a)$ and $\phi^2(s, a)$.

Table 1: The payoff matrix and value functions learned by RAC. We define 4 episodic return intervals: $[-12, 0), [0, 6), [6, 8), [8, +\infty)$. The context-based value functions $Q^1(s, c_\kappa, a)$ and $Q^2(s, c_\kappa, a)$ for all contexts κ are presented in (b) and (c). The individual optimistic marginal values, $\phi^1(s, a)$ and $\phi^2(s, a)$, appear in the first column. $Q^i(s, c_\kappa, a)$ with unattainable κ is marked by $\cancel{Q^i(s, c_\kappa, a)}$.

5.1 Comparison Result

Matrix Game. We begin by evaluating all algorithms on a didactic matrix game shown in Tab. 1a, where two agents must select the optimal joint action (a^1, a^1) to achieve the best reward. However, each agent maintains higher value estimations of its local actions a^2 and a^3 when others uniformly select actions, thereby leading to relative overgeneralization where (a^2, a^2) and (a^3, a^3) are preferred.

Fig. 2 shows the comparison results in the matrix game. One can observe that IQL struggle in the sub-optimal joint actions with 6 rewards, which demonstrates that the average-based value projection is susceptible to the relative overgeneralization issue. In contrast, RAC maintains an optimistic value estimation and accordingly excludes the effects caused by other agents' sub-optimal actions. This enables an efficient selection of the optimal joint action with the best reward 8. This also applies to

 https://doi.org/10.52202/085713-4943

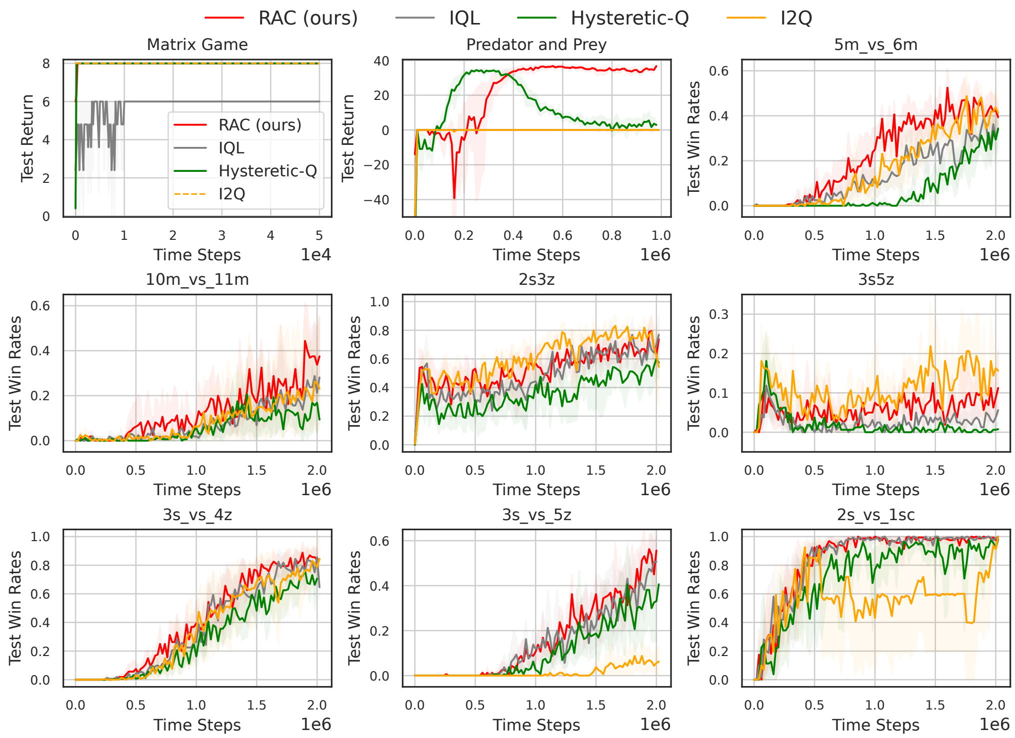

Figure 2: Comparison results in the matrix game, predator and prey, and seven SMAC maps.

the Hysteretic Q-learning, which follows an optimistic value update and demonstrates efficiency in simple tasks. When facing with complex tasks, such optimistic value update heavily suffers from the overestimation and non-stationarity issues, leading to poor performance. We validate this insight in subsequent comparisons. Similarly, I2Q shapes ideal transitions, which implicitly assume that other agents select cooperative policies, and learning policies on them leads to the optimal joint actions.

Furthermore, we analyze the learned context-based value functions and per-agent individual optimistic marginal values. As shown in Tab. 1b and Tab. 1c, for agents 1 and 2, the context-based value functions can accurately approximate the rewards of all joint actions, thus validating the efficiency of episodic return-aware context in representing the agents' joint policy and action. Additionally, the individual optimistic marginal value of each agent's local action adheres to the optimistic property, and equals the highest rewards when other agents select their cooperative actions. By enforcing per-agent value function to imitate it using Eq. (10), the selection of agents' optimal joint actions is achieved.

Predator and Prey. We further test all methods in the predator and prey, a partially observable task featured by relative overgeneralization. We control 8 predators to capture 8 preys, and a +10 reward is issued when two predators simultaneously capture a prey otherwise -2 is emitted for solitary hunting.

As depicted in Fig. 2, IQL suffers from the relative overgeneralization issue, and fails to learn effective policies, leading to 0 reward. While Hysteretic Q-learning succeeds in learning cooperative policies to achieve +40 rewards, its poor performance suggests the overestimation and non-stationarity issues caused by the optimistic value update. In contrast, RAC maintains standard Bellman update of the context-based value function, and solely derives optimistic value estimations regarding it to calculate the individual optimistic marginal value. Therefore, RAC exhibits notable strengths in terms of both learning efficiency and asymptotic performance. I2Q fails to learn effective policies and struggles in 0 reward. We hypothesis that the accurate approximation to QSS value is hardly achieved in partially observable setting, hindering I2Q from shaping ideal transitions and learning cooperative policies.

SMAC. To validate the scalability of RAC in more complex tasks, we compare it against baselines on seven SMAC maps (*i.e.,* 2s3z, 3s5z, 3s_vs_4z, 3s_vs_5z, 5m_vs_6m, 10m_vs_11m, and 2s_vs_1sc). The results, illustrated in Fig. 2, show that Hysteretic Q-learning performs the worst across all maps due to the overestimation and non-stationarity caused by its optimistic value update. These two issues

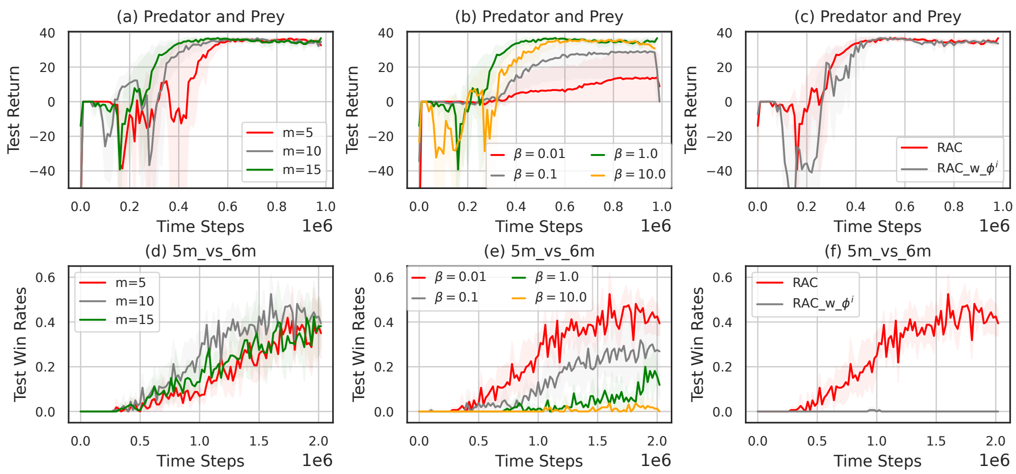

Figure 3: Ablation studies regarding m and β of RAC in the predator and prey and 5m_vs_6m.

are exacerbated in complex tasks and hinder learning efficiency. In contrast, while IQL achieves competitive performance on most maps, RAC significantly outperforms it on 5m_vs_6m, 10m_vs_11m, 3s5z, and 3s_vs_4z, demonstrating its effectiveness in addressing both non-stationarity and relative overgeneralization in fully decentralized learning. On the remaining maps (2s3z, 3s_vs_5z, and 2s_vs_1sc), RAC performs comparably to IQL. We hypothesis that these maps pose fewer challenges from non-stationarity and relative overgeneralization, and thus the benefit brought by RAC is not obvious. I2Q surpasses RAC on 2s3z and 3s5z, indicating that shaping ideal transitions enhances fully decentralized cooperative policy learning. However, I2Q shows inconsistent performance, performing comparably or worse on other maps. This discrepancy may be attributed to its inability of accurately estimating QSS values under partial observability, limiting its adaptability to certain maps.

5.2 Ablation Study

To examine the impact of m and β on RAC's performance, we set m to 5, 10, 15 and β to 0.01, 0.1, 1.0, 10.0, which serve as multiple baselines. Additionally, we introduce RAC_w_ϕ^i, a variant where the use of Q^i_{S} is excluded, and each agent i selects actions solely based on ϕ^i. This baseline allows us to evaluate the effect of separately learning Q^i_{S}. As depicted in Fig. 3 (c), in the predator and prey task, RAC_w_ϕ^i performs similarly to RAC, demonstrating that Q^i and ϕ^i can be efficiently learned across the entire return space in simple tasks. As a result, as shown in Fig. 3 (a) and (b), increasing m leads to finer partitioning of the episodic return, enabling ϕ^i to be learned more accurately. Moreover, RAC's approach of making Q^i_{S} imitate ϕ^i enhances its ability to learn cooperative policies, particularly as β increases. However, in more complex tasks like 5m_vs_6m, a comprehensive coverage over the entire return space is hardly achieved during early training, making it difficult to efficiently learn both Q^i and ϕ^i. Consequently, RAC_w_ϕ^i, which relies only on ϕ^i for action selection, fails to learn a cooperative policy, as shown by its poor performance in Fig. 3 (f). In such scenario, an appropriate selection of m and β is necessitated for balancing the TD loss (Eq. (9)) and the supervision loss (Eq. (10)) of Q^i_{S}. As depicted in Fig. 3 (d) and (e), RAC with m=10 and β=0.01 yields the best performance, and these settings are adopted as the default for most SMAC maps.

6 Discussion

In this section, we provide a complementary discussion on three aspects of RAC: (1) its performance compared with MARL methods under the centralized training with decentralized execution paradigm; (2) its flexibility with respect to major components; and (3) its computational complexity analysis.

Empirical Assessment. We compare RAC with two representative baselines, QMIX [2] and VDN [1]. As shown in Fig. 4 in Appendix C.4, RAC significantly outperforms both QMIX and VDN on the matrix game and the predator and prey tasks, where both QMIX and VDN suffer from sub-optimal policies due to their representational limitations regarding the learned factored global action value

 https://doi.org/10.52202/085713-4943

function [25]. On multiple SMAC maps, RAC exhibits superior performance in comparison to fully decentralized baselines but underperforms relative to QMIX and VDN. We attribute this to RAC's limited ability to handle partial observability and to adequately cover the whole return space.

Flexibility Analysis. The context-based value function Q^i necessitates comprehensive coverage of the return space for efficient updates. When only Q^i is learned, during the early training process, coordinated behaviors are limited, and the achieved returns tend to fall within a low-value region (*i.e.,* small values of c_κ). Consequently, when computing the individual optimistic marginal value $\phi^i(s_t, a_t^i) = \max_{c_\kappa \in \mathcal{C}} Q^i(s_t, c_\kappa, a_t^i)$, $\phi^i(s_t, a_t^i)$ tends to approximate $Q^i(s_t, c_\kappa, a_t^i)$ associated with small c_κ, which in turn corresponds to sub-optimal actions yielding low episodic returns.

For complex tasks such as SMAC maps, we empirically find that separately learning $Q^i_{\mathrm{S}}(s_t, a_t^i)$ and optimizing it with the TD loss and the supervision loss lead to satisfactory performance. Importantly, we clarify that the inclusion of Q^i_{S} is not a unique choice, and other alternatives are viable. In the context of SMAC maps, the ability of Q^i_{S} to aid in return-space coverage can be attributed to two main factors: (1) the competitive performance of vanilla IQL, which makes Q^i_{S} a reasonable candidate for generating informative transitions; and (2) the incorporation of additional supervision signals, such as the proposed supervision loss, through which Q^i helps guide Q^i_{S} in identifying and selecting agents' local cooperative actions, improving its effectiveness in facilitating exploration and learning.

From perspective (1), Q^i_{S} could be replaced by other MARL methods that demonstrate competitive performance. From perspective (2), its exploration capability could be further enhanced by incorporating intrinsic objectives (*e.g.,* curiosity-based rewards) or alternative supervision signals derived from Q^i to promote broader or more targeted exploration and facilitate cooperative action selection.

Complexity Analysis. The computational complexity of RAC comprises two main components:

(a) In terms of the context-based value function Q^i, its inputs include both states and contexts, and the outputs correspond to the value estimates of all possible local actions of agent i. Its input space scales linearly with the dimension (or number) of contexts, and deep neural networks can generalize well across such space. The number of its output is only $|A^i|$, where A^i is agent i's local action space.

(b) In terms of the individual optimistic marginal value ϕ^i, when we set the number of contexts m too large, the enumeration of all possible contexts typically leads to high computational complexity. A promising approach is to use sampling-based derivation-free heuristic search methods, such as the Cross-Entropy Method (CEM) [28], to approximate the maxima. Specifically, we iteratively draw a batch of N_c random context samples from a candidate distribution $\mathcal{D}_k$, *e.g.,* a Gaussian, at each iteration k. The best $M < N_c$ samples (with the highest context-based value estimates) are then used to fit a new Gaussian distribution $\mathcal{D}_{k+1}$, and this process repeats K times. Although sampling-based approximation can efficiently reduce the complexity caused by exhaustive search over an enormous context space and make RAC more tractable when m is too large, maintaining multiple sampling processes and performing iterative updates of the Gaussian distribution add additional complexity.

7 Conclusion

For fully decentralized cooperative MARL, this paper presents RAC to address both non-stationarity and relative overgeneralization issues in a unified framework. RAC formalizes the local task dynamics of each agent as a contextual MDP, and constructs contexts using discretized episodic return values. Consequently, RAC learns a context-based value function for each agent to enable stationary policy updates, and derives an individual optimistic marginal value to facilitate the selection of optimal joint action. Extensive experiments across various cooperative tasks demonstrate its effectiveness.

Limitation and Future Work. Despite its strengths, there are three critical limitations that warrant further exploration. First, RAC necessitates comprehensive coverage of the entire return space, and learning a separate decentralized value function performs poorly in complex tasks. This limitation could be addressed by combining RAC with efficient coordinated exploration techniques. Second, RAC currently relies on manual discretization of prior episodic return bounds to construct contexts, which limits adaptability and introduces high variance. To address this issue, we intend to explore adaptive context modeling techniques and alternative context representations. Third, RAC currently focuses only on the scenario where context changes between episodes (case (2)). To address case (1), where context changes between time steps, we plan to construct contexts by modeling the local task dynamics distribution over different time intervals. We leave these directions as our future work.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China (No.62506172, No.62325206, No.62192783, No.62532003), the Key Research and Development Program of Jiangsu Province (No.BE2023016-4), the Jiangsu Science and Technology Major Project (No.BG2024031), the Natural Science Foundation of Jiangsu Province (No.BK20250658), and the Natural Science Research Start-up Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications (No.NY225026). What's more, the authors greatly thank all anonymous reviewers for their valuable comments, which significantly contribute to improving the quality and clarity of this work.

References

[1] 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.

[2] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 21(1):7234–7284, 2020.

[3] 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.

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

[5] Ryan Lowe, Yi 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.

[6] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[7] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

[8] Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

[9] Ming Zhou, Jun Luo, Julian Villella, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, Montgomery Alban, Iman Fadakar, Zheng Chen, et al. Smarts: An open-source scalable multi-agent rl training school for autonomous driving. In *Conference on robot learning*, pages 264–285. PMLR, 2021.

[10] Xiaoqiang Wang, Liangjun Ke, Zhimin Qiao, and Xinghua Chai. Large-scale traffic signal control using a novel multiagent reinforcement learning. *IEEE transactions on cybernetics*, 51 (1):174–187, 2020.

[11] Qianyue Hao, Wenzhen Huang, Tao Feng, Jian Yuan, and Yong Li. Gat-mf: Graph attention mean field for very large scale multi-agent reinforcement learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 685–697, 2023.

[12] Jiechuan Jiang, Kefan Su, and Zongqing Lu. Fully decentralized cooperative multi-agent reinforcement learning: A survey. *arXiv preprint arXiv:2401.04934*, 2024.

[13] Laetitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems. *The Knowledge Engineering Review*, 27(1):1–31, 2012.

 https://doi.org/10.52202/085713-4943

[14] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning*, pages 1146–1155. PMLR, 2017.

[15] Kefan Su, Siyuan Zhou, Jiechuan Jiang, Chuang Gan, Xiangjun Wang, and Zongqing Lu. Multi-agent alternate q-learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 1791–1799, 2024.

[16] Jiechuan Jiang and Zongqing Lu. I2q: A fully decentralized q-learning algorithm. *Advances in Neural Information Processing Systems*, 35:20469–20481, 2022.

[17] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10199–10210, 2020.

[18] Martin Lauer and Martin A Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the seventeenth international conference on machine learning*, pages 535–542, 2000.

[19] Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. 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*, pages 64–69. IEEE, 2007.

[20] Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P How, and John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*, pages 2681–2690. PMLR, 2017.

[21] Liviu Panait, Keith Sullivan, and Sean Luke. Lenient learners in cooperative multiagent systems. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 801–803, 2006.

[22] Ermo Wei and Sean Luke. Lenient learning in independent-learner stochastic cooperative games. *Journal of Machine Learning Research*, 17(84):1–42, 2016.

[23] Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

[24] 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.

[25] Tarun Gupta, Anuj Mahajan, Bei Peng, Wendelin Böhmer, and Shimon Whiteson. Uneven: Universal value exploration for multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 3930–3941. PMLR, 2021.

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

[27] Ashley Edwards, Himanshu Sahni, Rosanne Liu, Jane Hung, Ankit Jain, Rui Wang, Adrien Ecoffet, Thomas Miconi, Charles Isbell, and Jason Yosinski. Estimating q (s, s') with deep deterministic dynamics gradients. In *International Conference on Machine Learning*, pages 2825–2835. PMLR, 2020.

[28] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss existing limitations and future works in Sec. 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/085713-4943

Justification: This work does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide both algorithmic and experimental details in Appendix C.3. What's more, the source code is available in the supplementary materials. Together, these resources ensure the reproducibility of the experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the source code of our algorithm in the supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All training details can be found in Appendix C.3 and the source code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All results are illustrated with the median performance and standard errors over five random seeds, as depicted in all figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

https://doi.org/10.52202/085713-4943

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: See Sec. C.5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work fully complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work has no societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This work does not present such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We build our algorithm and baselines upon the PyMARL [24] framework, and have appropriately cited the corresponding paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/085713-4943

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We include the codebase of RAC in the supplemental material, and provide a README regarding its usage.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: We use LLM exclusively for polishing the writing.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-4943

A Partially Observable Setting

Task Formalization. A partially observable cooperative multi-agent task where agents make decision in a fully decentralized manner is usually formalized as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), which is denoted by a tuple $\langle N, S, \boldsymbol{A}, P, R, \boldsymbol{O}, \boldsymbol{Z}, \gamma \rangle$. Dec-POMDP differs from MMDP in that each agent solely has access to its local observation rather than the state. At each time step t, each agent i receives its local observation $o_t^i \in Z^i \in \boldsymbol{Z}$ based on its local observation function $O^i(o_t^i|s_t) \in \boldsymbol{O}$, where Z^i denotes agent i's observation space. $\boldsymbol{Z}$ and $\boldsymbol{O}$ represent all agents' joint observation space and joint observation function, respectively. The partial observability limits agents from fully perceiving the states, thereby exacerbating the non-stationarity. To deal with the partial observability challenge, each agent i conditions its local policy $\pi^i(a_t^i|\tau_t^i)$ and local value function $Q^i(\tau_t^i, a_t^i)$ on its local action-observation history $\tau_t^i = (o_0^i, a_0^i, o_1^i, a_1^i, \dots, a_{t-1}^i, o_t^i)$.

Extension to Dec-POMDP. To extend RAC to Dec-POMDP, we learn a context-based value function $Q^i(\tau_t^i, c_\kappa, a_t^i)$ for each agent i, and update it using the loss function defined as follows:

$$\mathcal{L}_{\mathrm{C}}(\theta^i) = \mathbb{E}_{(o_t^i, a_t^i, c_\kappa, r_t, o_{t+1}^i) \sim \mathcal{D}^i} \left[(r_t + \gamma \max_{a_{t+1}^i} Q^i(\tau_{t+1}^i, c_\kappa, a_{t+1}^i) - Q^i(\tau_t^i, c_\kappa, a_t^i))^2 \right], \quad (12)$$

and the individual optimistic marginal value is derived as follows:

$$\phi^i(\tau_t^i, a_t^i) = \max_{c_\kappa \in \mathcal{C}} Q^i(\tau_t^i, c_\kappa, a_t^i). \quad (13)$$

Furthermore, we condition the separately learned value function $Q_{\mathrm{S}}^i(\tau_t^i, a_t^i)$ on each agent i's τ_t^i. The corresponding TD loss is defined as follows:

$$\mathcal{L}_{\mathrm{TD}}(\sigma^i) = \mathbb{E}_{(o_t^i, a_t^i, r_t, o_{t+1}^i) \sim \mathcal{D}^i} \left[(r_t + \gamma \max_{a_{t+1}^i} Q_{\mathrm{S}}^i(\tau_{t+1}^i, a_{t+1}^i) - Q_{\mathrm{S}}^i(\tau_t^i, a_t^i))^2 \right], \quad (14)$$

and the auxiliary supervision loss is defined below:

$$\mathcal{L}_{\mathrm{sup}}(\sigma^i) = \mathbb{E}_{(o_t^i, a_t^i, r_t, o_{t+1}^i) \sim \mathcal{D}^i} [D_{\mathrm{KL}}[\pi^i(\cdot|\tau_t^i) || \pi_{\mathrm{S}}^i(\cdot|\tau_t^i))]], \quad (15)$$

where $\pi^i(\cdot|\tau_t^i)$ and $\pi_{\mathrm{S}}^i(\cdot|\tau_t^i)$ respectively denote the Boltzmann policy with respect to $\phi^i(\tau_t^i, a_t^i)$ and $Q_{\mathrm{S}}^i(\tau_t^i, a_t^i)$, which are defined as follows:

$$\pi^i(a_t^i|\tau_t^i) = \frac{\exp(\phi^i(\tau_t^i, a_t^i))}{\sum_{a^i \in A^i} \exp(\phi^i(\tau_t^i, a^i))}, \quad \pi_{\mathrm{S}}^i(a_t^i|\tau_t^i) = \frac{\exp(Q_{\mathrm{S}}^i(\tau_t^i, a_t^i))}{\sum_{a^i \in A^i} \exp(Q_{\mathrm{S}}^i(\tau_t^i, a^i))}. \quad (16)$$

Empirical Validation. In Sec. 5, we evaluate RAC on the predator and prey and seven SMAC maps, which are featured by the partially observable challenge. The results demonstrate that RAC yields superior performance on these tasks, validating its effectiveness in solving partially observable tasks.

B The Distinction Clarification

For each agent i, the decentralized value function $Q^i(s_t, a_t^i)$ can be regarded as a projection of the true joint action value function $Q(s_t, a_t^i, a_t^{-i})$. IQL adheres to an average-based projection as below:

$$Q^{i,\boldsymbol{\pi}}(s_t, a_t^i) = \sum_{a_t^{-i}} \pi^{-i}(a_t^{-i}|s_t) Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i}), \quad (17)$$

where $Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i})$ is the joint action value function under a given joint policy $\boldsymbol{\pi} = (\pi^i, \pi^{-i})$. It is obvious that the average-based projection is easily affected by other agents' sub-optimal actions and suffers from the relative overgeneralization. In contrast, the optimistic projection is defined below:

$$Q^{i,opt}(s_t, a_t^i) = \max_{a_t^{-i}} Q^*(s_t, a_t^i, a_t^{-i}), \quad (18)$$

where $Q^*(s_t, a_t^i, a_t^{-i})$ represents the joint action value function of an optimal joint policy $\boldsymbol{\pi}^*$. The optimistic projection assumes that other agents $-i$ always select their cooperative local actions, thus eliminating the impact of other agents' non-cooperation. As a result, each agent i can identify and select its local cooperative action based on $Q^{i,opt}(s_t, a_t^i)$, leading to the optimal joint policy.

Hysteretic Q-learning. Hysteretic Q-learning directly approximates $Q^{i,opt}(s_t, a_t^i)$ by $Q^i(s_t, a_t^i)$, which is updated based on an optimistic value update below:

$$Q^i(s_t, a_t^i) \leftarrow \begin{cases} Q^i(s_t, a_t^i) + \delta_t^i & \text{if } \delta_t^i \geq 0 \\ Q^i(s_t, a_t^i) + \beta \delta_t^i & \text{else} \end{cases}, \quad (19)$$

Table 2: Descriptions of maps used in this work.

Name	Ally Units	Enemy Units	Type
2s_vs_1sc	2 Stalkers	1 Spine Crawler	Asymmetric & Homogeneous
2s3z	2 Stalkers, 3 Zealots	2 Stalkers, 3 Zealots	Symmetric & Heterogeneous
3s5z	3 Stalkers, 5 Zealots	3 Stalkers, 5 Zealots	Symmetric & Heterogeneous
3s_vs_4z	3 Stalkers	4 Zealots	Asymmetric & Homogeneous
3s_vs_5z	3 Stalkers	5 Zealots	Asymmetric & Homogeneous
5m_vs_6m	5 Marines	6 Marines	Asymmetric & Homogeneous
10m_vs_11m	10 Marines	11 Marines	Asymmetric & Homogeneous

where $\beta < 1$ is a complement factor and δ_t^i denotes the temporal difference error (TD-error). It is defined as follows:

$$\delta_t^i = \underbrace{r_t + \gamma \max_{a_{t+1}^i} Q^i(s_{t+1}, a_{t+1}^i)}_{\text{learning target}} - Q^i(s_t, a_t^i). \tag{20}$$

Although Distributed Q-learning theoretically demonstrates that $Q^i(s_t, a_t^i)$ with the optimistic value update converges to the optimal joint policy, the entire update process (Eq. (19)) solely relies on the decentralized value function $Q^i(s_t, a_t^i)$. When facing with complex tasks where multiple agents strongly influence each other, a fully decentralized value update often leads to instability and poor convergence. Furthermore, Hysteretic Q-learning with function approximators (particularly the deep neural networks) is susceptible to the overestimation issue, resulting in sub-optimal solutions. The poor performance of Hysteretic Q-learning empirically validates this insight.

Return-Aware Context (RAC). In contrast, RAC decomposes the learning of $Q^{i,opt}(s_t, a_t^i)$ into two distinct sub-processes. The first is that we learn a context-based value function $Q^i(s_t, c_\kappa, a_t^i)$. The context c_κ represents the agents' joint policy and thus $Q^i(s_t, c_\kappa, a_t^i)$ approximates the true joint action value function $Q(s_t, a_t^i, a_t^{-i})$, as empirically demonstrated by experiments in the matrix game. Furthermore, the contexts enable stationary value updates of $Q^i(s_t, c_\kappa, a_t^i)$. Based on the Bellman update, $Q^i(s_t, c_\kappa, a_t^i)$ finally converges to $Q^*(s_t, a_t^i, a_t^{-i})$.

The second is that we derive the individual optimistic marginal value $\phi^i(s_t, a_t^i)$ based on $\phi^i(s_t, a_t^i) = \max_{c_\kappa \in \mathcal{C}} Q^i(s_t, c_\kappa, a_t^i)$. Consequently, as $Q^i(s_t, c_\kappa, a_t^i)$ converges to $Q^*(s_t, a_t^i, a_t^{-i})$ following a stationary value update, $\phi^i(s_t, a_t^i)$ approaches $Q^{i,opt}(s_t, a_t^i)$.

By implementing these two sub-processes, we ensure the stationary approximation of $Q^{i,opt}(s_t, a_t^i)$ by $\phi^i(s_t, a_t^i) = \max_{c_\kappa \in \mathcal{C}} Q^i(s_t, c_\kappa, a_t^i)$. In comparison to Hysteretic Q-learning, the context-based value function of RAC is free from the non-stationarity and overestimation issues. The superior performance of RAC across various cooperative tasks further validates its effectiveness in enhancing fully decentralized cooperative policy learning.

C Experimental Details

C.1 Environments

Matrix Game. Matrix game is a simple two-agent cooperative stage game where two agents must select the optimal joint action (a^1, a^1) to receive the best reward 8. In addition to the optimal joint action (a^1, a^1), there are two sub-optimal joint actions (a^2, a^2) and (a^3, a^3) that lead to 6 rewards. For each decentralized agent, when the other agent selects the action uniformly, the sub-optimal local actions a^2 and a^3 may be preferred over the optimal ones a^1, leading to the relative overgeneralization.

Predator and Prey. Predator and prey is a partially observable multi-agent task involving 8 predators and 8 preys. When a predator solely tries to capture a prey, a -2 punishment is emitted. In contrast, when two predators cooperate to capture a prey simultaneously, they receive $+10$ reward.

SMAC. The StarCraft Multi-Agent Challenge (SMAC) serves as a widely used benchmark in which a set of challenging maps is provided. These maps require cooperative MARL algorithms to make

https://doi.org/10.52202/085713-4943

Table 3: Challenges faced by all algorithms.

Algorithm	Non-stationarity	Relative Overgeneralization
IQL	✓	✓
Hysteretic Q-learning	✓	✗
I2Q	✗	✗

decentralized control for allied units against build-in AI enemies and achieve high win rates. In this paper, we choose seven maps including 5m_vs_6m, 10m_vs_11m, 2s3z, 3s5z, 3s_vs_4z, 3s_vs_5z, and 2s_vs_1sc to evaluate our algorithm. Details regarding these maps can be found in Tab. 2.

C.2 Baselines

We compare RAC with several baselines, namely IQL, Hysteretic Q-learning, and I2Q, as presented in Tab. 3. Below is a brief introduction to each algorithm.

IQL. IQL learns a decentralized $Q^i(s_t, a_t^i)$ for each agent i, and updates it using the standard TD loss. However, since IQL ignores the actions and policies of other agents, it suffers from non-stationarity during value function updates. Moreover, the $Q^i(s_t, a_t^i)$ learned by IQL is an average-based projection of the true joint action value function, leading to the issue of relative overgeneralization.

Hysteretic Q-learning. In this algorithm, each agent i learns an optimistic projection $Q^{i,opt}(s_t, a_t^i)$ based on the optimistic value update paradigm (Eq. (19)). Such optimistic belief assumes that other agents always select their cooperative actions, thus eliminating the negative effects caused by other agents' sub-optimal actions and addressing the relative overgeneralization issue.

I2Q. I2Q addresses both non-stationarity and relative overgeneralization by shaping ideal transitions, which are constructed by selecting next states with the highest QSS value. These ideal transitions implicitly assume that other agents follow cooperative policies. By learning from such transitions, I2Q guides IQL agents toward optimal cooperative policies.

C.3 Experimental Setups

We implement all algorithms based on the PyMARL framework. For I2Q, we adopt its official source code along with the recommended hyper-parameters to conduct experiments on both predator and prey and SMAC maps. For the matrix game, we directly report I2Q's results from the original paper.

For IQL, we use the original implementation provided in PyMARL. For Hysteretic Q-learning, we set β=0.01 for the matrix game. For the predator and prey and SMAC maps, we set β=0.01, 0.1, 0.3, 0.5, and 0.7, and run multiple seeds using β that yields the best performance.

For our proposed algorithm RAC, the architecture of Q_{S}^i is kept consistent with Q^i of IQL. It consists of a hidden layer (64 units, ReLU activation), a GRU module, and a linear layer (64 units) that outputs the Q values of all local actions. For $Q^i(s_t, c_\kappa, a_t^i)$, we provide two kinds of implementations: (1) Normal-Net. This architecture contains a hidden layer (64 units, ReLU activation), a GRU module, and a linear layer (64 units). The final linear layer takes the hidden states and contexts as inputs, and outputs the Q values of all local actions. (2) Hyper-Net. It is comprised by a hidden layer (of 64 units, ReLU activation), a GRU module, and a linear layer (of 64 units). The weights and biases of the final linear layer are generated by two separate hyper-networks, which take the context as input.

For the matrix game and predator and prey tasks, we instantiate $Q^i(s_t, c_\kappa, a_t^i)$ using the Hyper-Net architecture. For the SMAC maps, we use the Normal-Net to implement $Q^i(s_t, c_\kappa, a_t^i)$. Empirically, we find these configurations work well. All other settings (*e.g.,* learning rate, batch size) are kept consistent across all algorithms.

The hyper-parameters of RAC across all tasks are provided in Tab. 4. In particular, m denotes the number of episodic return intervals, while β is a scaling factor balancing the TD loss and the supervision loss of Q_{S}^i. $T_{epsilon}$ denotes the anneal time steps of ϵ when ϵ-greedy policy is used for exploration. T_{max} represents the total number of training time steps, and N_{max} is the size of the replay buffer. N_{batch} represents the size of sampled batches per training. α is the learning rate and γ

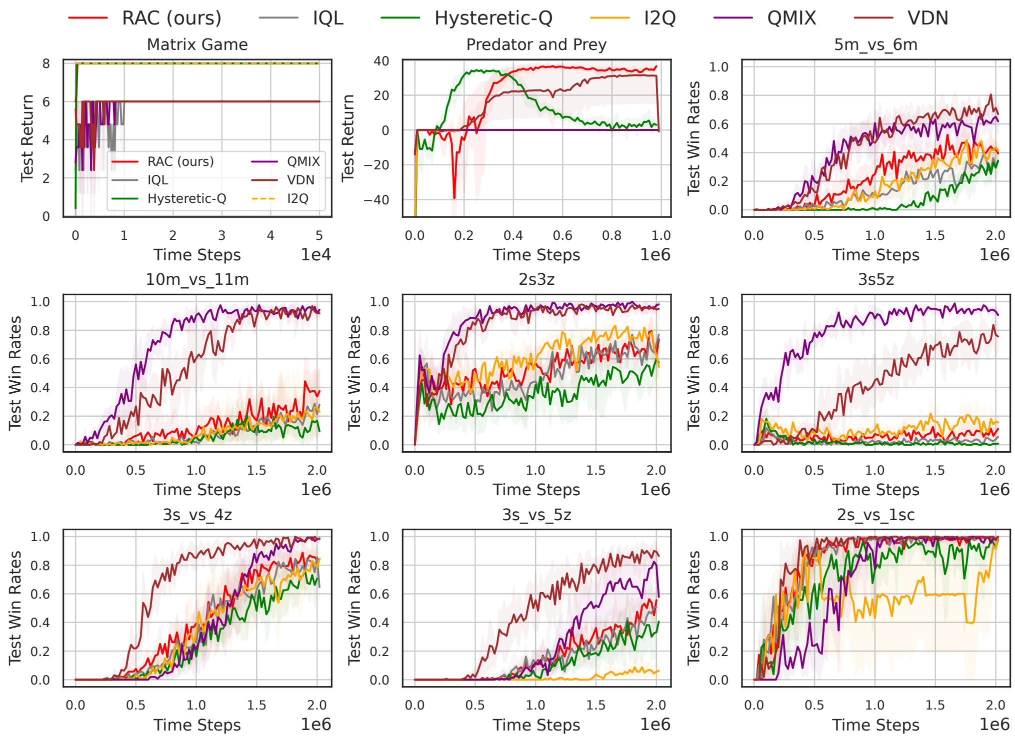

Figure 4: Complementary comparison results of RAC against QMIX and VDN.

Table 4: Hyper-parameters of RAC across all tasks.

Tasks	m	β	$T_{epsilon}$	T_{max}	N_{max}	N_{batch}	Optimizer	α	γ	ϵ_{max}	ϵ_{min}
matrix game	4	10.0	50k	50000	5000	32	RMSprop	0.0005	0.99	1.0	0.05
predator and prey	15	1.0	50k	1000000							
2s_vs_1sc	10	0.01	50k	2050000							
2s3z	10	0.01	50k	2050000							
3s5z	10	0.01	50k	2050000							
3s_vs_4z	5	0.01	50k	2050000							
3s_vs_5z	10	0.001	50k	2050000							
5m_vs_6m	10	0.01	50k	2050000							
10m_vs_11m	10	0.01	50k	2050000							

is the discounted factor. We decrease ϵ from ϵ_{max} to ϵ_{min} within $T_{epsilon}$ time steps. In addition, we utilize RMSprop technique to update all networks of RAC using gradient descent.

C.4 Complementary Comparison

To further evaluate the performance gap between RAC and MARL methods under the centralized training with decentralized execution (CTDE) paradigm, we compare RAC against two representative CTDE-based baselines, QMIX and VDN. The comparison results are presented in Fig. 4.

C.5 Computational Cost

We run all experiments under five different random seeds, and plot the mean/std in all figures. The experiments are carried out on a server, which comprises a AMD EPYC 7542 32-Core Processor CPU, 504GB RAM, and 8 NVIDIA GeForce RTX 4090 D GPUs.

https://doi.org/10.52202/085713-4943