FULLY DECENTRALIZED COOPERATIVE MULTI-AGENT REINFORCEMENT LEARNING IS A CONTEXT MODELING PROBLEM

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we consider fully decentralized cooperative multi-agent reinforcement learning, wherein each agent solely observes the states, its local actions, and the shared team rewards. For each agent, the lack of access to other agents' actions typically induces non-stationarity during value function updates and relative overgeneralization during value function estimation, which together impede effective cooperative policy learning. However, existing works fail to address both issues simultaneously, due to their inability to model the joint action of other agents in a fully decentralized setting. To overcome this limitation, we propose Dynamics-Aware Context (DAC), which formalizes each agent's locally perceived task as a Contextual Markov Decision Process, and addresses both non-stationarity and relative overgeneralization through dynamics-aware context modeling. Specifically, DAC attributes variations in per-agent local task dynamics to switches among unobserved contexts, and instantiate them as latent variables that model the step-wise dynamics distribution. Theoretical analysis shows that each context uniquely represents an equivalence class of other agents' joint actions, thus allowing a context-based value function learned for each agent to address both non-stationarity and relative overgeneralization. Empirically, we evaluate DAC across various cooperative tasks, and its superior performance highlights its effectiveness.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has emerged as a powerful technique for addressing cooperative tasks, driving substantial advancements in both algorithms (*e.g.,* value decomposition methods Sunehag et al. (2017); Rashid et al. (2020b); Son et al. (2019); Wang et al. (2020a) and multi-agent policy gradient methods Lowe et al. (2017); Foerster et al. (2018); Yu et al. (2022); Zhong et al. (2024)) and applications (*e.g.,* traffic signal control Wang et al. (2020b), autonomous vehicles Zhou et al. (2021), and vaccine allocation Hao et al. (2023)). Most of these advances depend on the centralized training (often with decentralized execution) paradigm, where global information, particularly the joint actions of all agents, is accessible during training. However, such direct access to other agents' actions is often unattainable in many real-world domains. For example, in industrial automation scenarios, robots from different companies may be unable to share action information due to privacy concerns or limited communication capabilities. In such cases, the *fully decentralized learning* is required, where each agent learns based on only its own experiences, without access to the actions of other agents, during both training and execution periods.

However, developing effective cooperative policies under the fully decentralized learning paradigm is challenging due to two fundamental issues arising from the lack of access to other agents' actions. The first one is *non-stationarity* (NS) during per-agent value function updates. Since treating other agents as part of the environment, the local task dynamics perceived by each agent becomes non-stationary due to the evolving policies of other agents, which undermines the convergence of value function updates Jiang et al. (2024). The second one is *relative overgeneralization* (RO), wherein the value estimations of each agent's local cooperative actions may be biased by other agents' exploratory or sub-optimal actions, hindering agents from selecting the optimal joint actions Matignon et al. (2012). As a result, fully decentralized learning typically suffers from low learning efficiency and sub-optimal solutions, substantially limiting its effectiveness in multi-agent cooperation.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Accordingly, we classify existing value-based MARL methods into two categories. The first category aims to address the NS issue. These methods either ensure stationary transition data by directly accessing other agents' actions Sunehag et al. (2017); Rashid et al. (2020b), designing multi-agent importance sampling weights, fingerprints Foerster et al. (2017), and constructing ideal transition probabilities Jiang & Lu (2022), or enable stationary policy updates based on the alternative policy update paradigm Su et al. (2024). The second category primarily targets the RO issue, typically by rectifying the learned factored global action value function Son et al. (2019); Wang et al. (2020a); Rashid et al. (2020a) or employing optimistic or lenient value function updates Lauer & Riedmiller (2000); Matignon et al. (2007); Omidshafiei et al. (2017); Panait et al. (2006); Wei & Luke (2016) to encourage the selection of optimal joint actions. Although both issues stem from the lack of access to other agents' actions, existing fully decentralized MARL methods address either the NS issue or the RO problem in isolation, due to their inability to model the joint actions of other agents in a fully decentralized setting. Consequently, a unified framework to resolve both challenges remains lacking.

To overcome this limitation, this paper proposes Dynamics-Aware Context (DAC), a novel method that formalizes the task, as locally perceived by each agent, as a Contextual Markov Decision Process (CMDP) Hallak et al. (2015), and further addresses both NS and RO from a dynamics-aware context modeling perspective. Specifically, DAC attributes variations in per-agent non-stationary local task dynamics to switches among unobserved contexts, and instantiates these contexts as latent variables that model the step-wise dynamics distribution underlying the agent's local trajectory. Based on this representation, DAC learns a context-based value function for each agent to address the NS issue during value function updates. Beyond addressing NS, we give theoretical analysis that each context uniquely represents an equivalence class of other agents' joint actions. This property enables DAC to derive an optimistic marginal value by marginalizing over contexts during value estimation, thereby discarding the effects of other agents' sub-optimal actions and addressing the RO problem. Together, these components enable effective cooperative policy learning in a fully decentralized manner.

Empirically, we evaluate DAC across various cooperative tasks, *i.e.,* matrix game, differential game, predator-prey, and the StarCraft Multi-Agent Challenge (SMAC) Samvelyan et al. (2019), and its consistent performance gains validate its effectiveness in fully decentralized learning.

2 Related Work

We broadly classify current value-based MARL methods into two categories. The first category aims to address the NS problem by constructing stationary transition data or policy updates. Specifically, canonical value decomposition methods, such as VDN Sunehag et al. (2017) and QMIX Rashid et al. (2020b), assume direct access to agents' joint actions to ensure stationary transitions during training. However, these methods often suffer from the RO issue due to the representational limitation of their learned factored global action value functions Gupta et al. (2021). For independent Q-learning (IQL) Tan (1993) agents, the multi-agent importance sampling Foerster et al. (2017) technique assumes direct access to other agents' policies and calculates an importance weight to decay obsolete data during experience replay. Multi-agent fingerprints method Foerster et al. (2017) uses the training iteration numbers and exploration rates of other agents to estimate their policies, and augments per-agent local transitions with these estimates. However, such direct access to other agents' information assumed in above methods is unattainable in practice. I2Q Jiang & Lu (2022) addresses this by shaping ideal transition probabilities for each IQL agent in a fully decentralized manner, and guarantees convergence to the optimal joint policy. In comparison to I2Q's approach of addressing NS and RO by shaping ideal transition probabilities, this work aims for a novel context-aware framework to tackle both issues. In addition, to ensure stationary policy updates, MA2QL Su et al. (2024) enforces sequential policy updates among IQL agents. When an agent updates its policy, all others' policies remain fixed. Despite its promise, the sequential policy update typically leads to sample inefficiency, as it lacks the capacity for parallel policy updates.

The second category addresses the RO issue by rectifying the learned factored global action value function or updating per-agent local value function in optimistic or lenient manners. Specifically, for value decomposition methods with representational limitations, weighted QMIX Rashid et al. (2020a) places more weights on potentially optimal joint actions during value function updates to exclusively recover correct value estimations for these critical actions. QTRAN Son et al. (2019) and QPLEX Wang et al. (2020a) incorporate complementary terms to correct the discrepancy between

the learned factored global action value functions and the true joint ones. For IQL, distributed Q-learning Lauer & Riedmiller (2000) adopts optimistic value function updates for individual agents, enabling them to identify their local cooperative actions and thus address the RO problem. However, such high optimism makes distributed Q-learning vulnerable to stochasticity. To mitigate this issue, Hysteretic Q-learning Matignon et al. (2007); Omidshafiei et al. (2017) updates each agent's value function using two learning rates for positive and negative temporal difference errors, respectively. Lenient learning Panait et al. (2006); Wei & Luke (2016) instead gradually shifts from optimistic to standard value function updates through a decaying lenience mechanism. Nevertheless, two critical issues remain unresolved. First, the lack of access to other agents' actions renders NS inherent in per-agent value function updates, potentially destabilizing learning. Second, the reliance on optimistic value function updates can lead to value overestimation, particularly under function approximation.

In summary, existing fully decentralized MARL methods fail to address both NS and RO in a unified manner. To bridge this gap, we formalize the task locally perceived by each agent as a CMDP, and accordingly develop a principled dynamics-aware context modeling framework that tackles both issues simultaneously, thereby enabling effective fully decentralized cooperative policy learning.

3 PRELIMINARY

3.1 MULTI-AGENT MARKOV DECISION PROCESS

We consider a cooperative multi-agent task that can be modeled as a Multi-Agent Markov Decision Process (MMDP) $\langle N, S, \boldsymbol{A}, P, R, \gamma \rangle$, where $N = \{1, 2, \ldots, n\}$ represents the agent set and S is the state space. $\boldsymbol{A} = A^1 \times A^2 \times \ldots A^n$ is all agents' 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 $a_t^i \in A^i$ according to its decentralized policy $\pi^i(a_t^i|s_t)$. Given the joint action $\boldsymbol{a}_t = (a_t^1, a_t^2, \ldots, a_t^n)$, 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 a team reward r_t based on the reward function $R(s_t, \boldsymbol{a}_t)$. The goal is to learn the optimal joint policy $\boldsymbol{\pi}^* = (\pi^{1,*}, \pi^{2,*}, \ldots, \pi^{n,*})$, which maximizes the expected discounted cumulative return $\mathbb{E}_{\boldsymbol{\pi},P}[\sum_{t=0}^{\infty} \gamma^t r_t]$, where γ denotes a discount factor.

We consider the fully decentralized learning, wherein each agent i solely observes the state s_t, its local action a_t^i, and the team reward r_t. For each agent i, the perceived task can be modeled as a Markov Decision Process (MDP) $\langle S, A^i, P^i, R^i, \gamma \rangle$ with dynamics defined as follows:

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

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

NS. As illustrated in Eq. (1), the local task dynamics perceived by each agent i, characterized by P^i and R^i, depend on other agents $-i$' joint policy π^{-i}. Since other agents' actions are unobserved, their effects must be marginalized over π^{-i} when constructing the local dynamics. Consequently, as π^{-i} continually evolves during learning, the resulting local task dynamics also change, giving rise to NS and undermining the convergence of per-agent value function updates. In contrast, if a_t^{-i} is accessible, the value function could be conditioned directly on (a_t^i, a_t^{-i}), under which the underlying transition and reward functions are independent of π^{-i} and remain stationary.

RO. Due to the lack of access to a_t^{-i}, the value estimation of per-agent local cooperative actions may be biased by exploratory or sub-optimal actions taken by other agents. As a result, the sub-optimal joint actions are preferred over the optimal ones, a problem known as RO.

3.2 CONTEXTUAL MARKOV DECISION PROCESS

A CMDP is often defined as a tuple $\langle \mathcal{C}, S, A, M(c) \rangle$, where $\mathcal{C}$ represents the context space, S and A denote the state and action spaces, respectively, and $M(c)$ maps each context $c \in \mathcal{C}$ to a MDP $\langle S, A, P^c, R^c, \gamma \rangle$. A CMDP therefore defines a family of MDPs that share the same state and action spaces but differ in their state transition and reward functions.

4 METHODOLOGY

In this section, we begin by proposing the task formalization based on CMDP, and then delve into the process of modeling dynamics-aware contexts. Subsequently, we analyze the theoretical properties of the learned contexts and their relation to other agents' joint actions. Building on this, we learn a context-based value function for each agent and derive an optimistic marginal value to address NS and RO, respectively. Finally, we summarize the overall learning procedure of DAC.

4.1 TASK FORMALIZATION

As detailed in Sec. 3.1, the local task of each agent i can be modeled as a MDP $\langle S, A^i, P^i, R^i, \gamma \rangle$, where both the state transition function P^i and the reward function R^i depend on the joint policy π^{-i} of other agents $-i$. As π^{-i} evolves during decentralized learning, it induces different local transition and reward functions for agent i. Accordingly, the local task perceived by agent i can be decomposed into a family of MDPs that share the same state and action spaces but differ in their task dynamics. Therefore, we formalize this family of local tasks as a CMDP:

$$\langle \mathcal{C}, S, A^i, M(c) \rangle, \qquad M(c) : c \to \langle S, A^i, P^i_c, R^i_c, \gamma \rangle, \tag{2}$$

where S represents the state space, A^i is the local action space of agent i, and $c \in \mathcal{C}$ denotes the latent context induced by other agents' π^{-i}. Joint policies that induce identical local state transition P^i_c and reward functions R^i_c are regarded as belonging to the same context. Therefore, changes in π^{-i} induce switches between different contexts and, consequently, changes in the local task dynamics.

Under fully decentralized learning, agent i has access to neither π^{-i} nor the corresponding latent context. As a result, transitions generated under different task dynamics are mixed together, causing the same local state-action pair (s_t, a^i_t) to exhibit time-varying transition and reward distributions. From the perspective of agent i, such unobserved context shifts manifest as the NS problem.

This CMDP formalization attributes NS to switches between unobserved contexts, and provides a principled framework for addressing it through explicit context modeling, without requiring explicit estimation of the joint policy π^{-i}. Since the latent context is ultimately manifested through its induced state transition and reward dynamics, a more direct approach is to model the underlying dynamics distribution currently experienced by the agent. Motivated by this observation, we instantiate the latent CMDP context as a *dynamics-aware context* inferred solely from locally observable transitions. Its concrete modeling is introduced in the following subsection.

4.2 DYNAMICS-AWARE CONTEXT MODELING

Following the above CMDP formalization, we instantiate the latent context by modeling the local task dynamics currently experienced by each agent. Specifically, we introduce a latent variable, termed the *dynamics-aware context*, to represent the underlying local transition and reward dynamics. Since other agents' joint policy evolves during decentralized learning, the corresponding local dynamics distribution also changes over time. This parallels the setting of concept drift Lu et al. (2018), where the underlying data distribution evolves along a data stream. A common strategy for tracking such distributional changes is to maintain a sliding window containing the latest data, which has proven effective in capturing the real-time data distribution, ensuring model adaptability in dynamic settings.

Motivated by this insight, DAC infers the current dynamics-aware context from a sliding window of recent local transitions. At time step t, the sliding window of agent i is represented by the trajectory segment $\tau^i_{t-k+1:t} = (s_{t-k+1}, a^i_{t-k+1}, r_{t-k+1}, s_{t-k+2}, \ldots, s_t, a^i_t, r_t, s_{t+1})$, which contains the latest k local transitions. These recent transitions provide empirical evidence of the current dynamics.

We assume that the dynamics distribution of $\tau^i_{t-k+1:t}$ can be represented by a latent context c^i_t, whose posterior distribution is given by $p^i(c^i_t|\tau^i_{t-k+1:t})$. Since this posterior is intractable, we introduce a variational distribution $q^i(c^i_t|\tau^i_{t-k+1:t})$ to approximate it by minimizing:

$$\begin{aligned} &D_{\mathrm{KL}}(q^i(c^i_t|\tau^i_{t-k+1:t})||p^i(c^i_t|\tau^i_{t-k+1:t})) \\ &= \log p^i(\tau^i_{t-k+1:t}) + D_{\mathrm{KL}}(q^i(c^i_t|\tau^i_{t-k+1:t})||p^i(c^i_t)) - \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log p^i(\tau^i_{t-k+1:t}|c^i_t), \end{aligned} \tag{3}$$

where $p^i(c^i_t)$ denotes the prior distribution of the latent context and $\log p^i(\tau^i_{t-k+1:t})$ is the evidence term that can be regarded as a constant. The derivation is detailed in Appendix A. Accordingly,

minimizing Eq. (3) is equivalent to maximizing the following evidence lower bound:

$$\max \mathbb{E}_{q^i(c_t^i|\tau_{t-k+1:t}^i)} \log p^i(\tau_{t-k+1:t}^i|c_t^i) - D_{\mathrm{KL}}(q^i(c_t^i|\tau_{t-k+1:t}^i)||p^i(c_t^i)). \tag{4}$$

In Eq. (4), the first term is the reconstruction likelihood that ensures c_t^i retains sufficient information of the trajectory segment, whereas the second term regularizes its posterior toward the prior distribution. To explicitly focus c_t^i on transition and reward dynamics, we factorize the reconstruction likelihood:

$$p^i(\tau_{t-k+1:t}^i|c_t^i) = p(s_{t-k+1}) \prod_{h=t-k+1}^{t} p^i(a_h^i|s_h) p^i(s_{h+1}, r_h|s_h, a_h^i, c_t^i), \tag{5}$$

where the initial state distribution $p(s_{t-k+1})$ is determined by the environment and $p^i(a_h^i|s_h)$ corresponds to agent i's decentralized policy conditioned on the state. Since neither term characterizes the latent task dynamics, they can be omitted from the context-learning objective. We therefore obtain:

$$\max \mathbb{E}_{q^i(c_t^i|\tau_{t-k+1:t}^i)} \sum_{h=t-k+1}^{t} \log p^i(s_{h+1}, r_h|s_h, a_h^i, c_t^i) - D_{\mathrm{KL}}(q^i(c_t^i|\tau_{t-k+1:t}^i)||p^i(c_t^i)). \tag{6}$$

Through this objective, the inferred context $c_t^i \sim q^i(c_t^i|\tau_{t-k+1:t}^i)$ provides a compact representation of the real-time task dynamics distribution. **At this stage, c_t^i is defined solely by its ability to represent the local task dynamics, thereby providing the basis for mitigating NS.** Its relationship with the unobserved joint actions of the other agents is established in the following theoretical analysis.

4.3 Theoretical Validation of the Dynamics-Aware Context

The dynamics-aware context c_t^i is introduced above to characterize the real-time local task dynamics perceived by agent i. We now show that, under suitable conditions, it also identifies an equivalence class of the unobserved joint action a_t^{-i} of other agents $-i$, making it sufficient for joint action-value representation. Full assumptions and proofs are provided in Appendix B.

We consider a deterministic MMDP, where $s_{t+1} = F(s_t, a_t^i, a_t^{-i}), r_t = R(s_t, a_t^i, a_t^{-i})$. For a fixed (s_t, a_t^i), we define the local dynamics outcome induced by a_t^{-i} as:

$$\Psi^i_{s_t,a_t^i}(a_t^{-i}) := (F(s_t, a_t^i, a_t^{-i}), R(s_t, a_t^i, a_t^{-i}).$$

Definition 1 (Dynamics-equivalent joint actions)**.** For a fixed (s_t, a_t^i), two actions a_t^{-i} and $\tilde{a}_t^{-i}$ are *dynamics-equivalent* with respect to agent i, denoted by $a_t^{-i} \sim^i_{s_t,a_t^i} \tilde{a}_t^{-i}$, if and only if $\Psi^i_{s_t,a_t^i}(a_t^{-i}) = \Psi^i_{s_t,a_t^i}(\tilde{a}_t^{-i})$. The corresponding equivalence class is denoted by $[a_t^{-i}]^i_{s_t,a_t^i} := \{\tilde{a}_t^{-i} | \tilde{a}_t^{-i} \sim^i_{s_t,a_t^i} a_t^{-i}\}$.

Hence, exact recovery of a_t^{-i} is unnecessary: joint actions that induce identical local dynamics are indistinguishable from the perspective of agent i. Recall that $c_t^i \sim q^i(c_t^i \mid \tau_{t-k+1:t}^i)$ is learned to reconstruct the current transition and reward. We consider the ideal representation in which c_t^i is sufficient for recovering the current local dynamics and introduces no redundant distinctions between identical dynamics outcomes; the formal conditions are given in Appendix B.1.

Theorem 1 (Context-joint-action equivalence)**.** *Under the conditions above, for any fixed (s_t, a_t^i), two transitions are represented by the same dynamics-aware context if and only if their corresponding joint actions of the other agents are dynamics-equivalent: $c_t^i = \tilde{c}_t^i \iff a_t^{-i} \sim^i_{s_t,a_t^i} \tilde{a}_t^{-i}$. Consequently, there exists a one-to-one correspondence between the set of reachable dynamics-aware contexts and the set of reachable dynamics-equivalence classes: $c_t^i \longleftrightarrow [a_t^{-i}]^i_{s_t,a_t^i}$.*

The proof is provided in Appendix B.2. Theorem 1 shows that, although DAC never observes a_t^{-i}, each dynamics-aware context uniquely represents the equivalence class of the other agents' joint actions that could have generated the currently experienced local dynamics.

Theorem 2 (Joint action-value representation sufficiency)**.** *Under the conditions of Theorem 1, for any joint policy $\boldsymbol{\pi}$, there exists a context-based value function $\bar{Q}^{i,\boldsymbol{\pi}}$ such that, for every reachable transition generated by (s_t, a_t^i, a_t^{-i}), $\bar{Q}^{i,\boldsymbol{\pi}}(s_t, a_t^i, c_t^i) = Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i})$.*

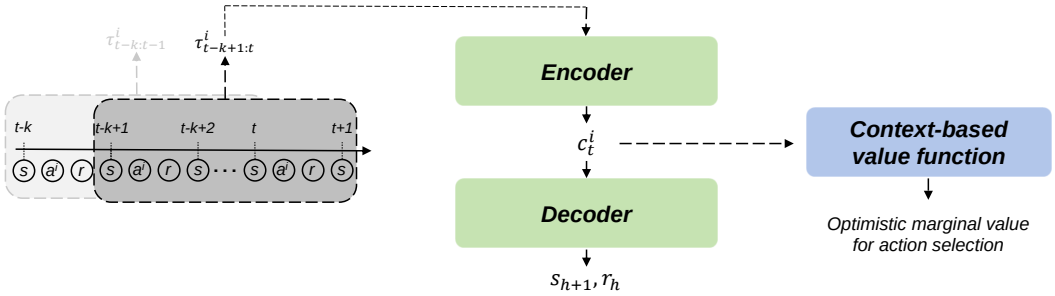


Figure 1: The workflow of DAC.

The proof is provided in Appendix B.4. Theorem 2 establishes that the dynamics-aware context retains sufficient information about the unobserved a_t^{-i} for joint action-value representation. Therefore, with sufficiently expressive and accurately learned value functions, $Q^i(s_t, a_t^i, c_t^i) \approx Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i})$.

Thus, the dynamics-aware context not only characterizes the real-time task dynamics to mitigate NS, but also provides the information required to distinguish otherwise aliased joint action values caused by the inaccessible actions of the other agents, thereby addressing the representational source of RO. This motivates the context-based value learning mechanism introduced next.

4.4 Context-based Value Learning

Context-based Value Function. Following Theorem 2, we learn a context-based value function $Q^i(s_t, a_t^i, c_t^i)$ for each agent i. Since the context c_t^i is inferred solely from agent i's local trajectory segment through $q^i(c_t^i|\tau_{t-k+1:t}^i)$, the value function can be learned in a fully decentralized manner.

Given the context-augmented transition $(s_t, a_t^i, c_t^i, r_t, s_{t+1}, c_{t+1}^i)$, we optimize Q^i using:

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

where Q^i is parameterized by θ^i and $\mathcal{D}^i$ denotes agent i's local replay buffer. By conditioning on the inferred dynamics-aware context, Q^i distinguishes otherwise aliased joint action values while accounting for variations in the local task dynamics, thereby mitigating NS.

Optimistic Marginal Value. Although $Q^i(s_t, a_t^i, c_t^i)$ provides a context-based value estimate, the current context c_t^i is available only after the current transition is completed and thus cannot be directly used for action selection. We therefore eliminate the context through the *optimistic marginal value*:

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

According to Theorem 1, different contexts represent different dynamics-equivalence classes of the other agents' joint actions. Hence, Eq. (8) evaluates each local action under its most favorable dynamics context, reducing the effects of other agents' sub-optimal actions and thereby alleviating RO. Moreover, eliminating the context yields a decentralized action value $\phi^i(s_t, a_t^i)$, from which $\pi^i(a_t^i \mid s_t)$ can be directly derived, consistent with the policy assumption in Eqs. (5) and (6).

4.5 Overall Learning Procedure

To make the maximization over contexts in Eq. (8) tractable, we instantiate the dynamics-aware context as a discrete categorical variable using a VAE-style network Kingma & Welling (2014). As shown in Fig. 1, the encoder takes the most recent k local transitions as input and outputs the logits of a categorical distribution over m contexts, while the decoder reconstructs the local task dynamics.

To obtain discrete context samples while preserving differentiability, we apply Gumbel-Softmax Jang et al. (2016) to the encoder outputs. We further adopt a uniform categorical distribution over the m context categories as the prior $p^i(c_t^i)$. Each agent infers its dynamics-aware context, updates the context-based value function, and selects actions according to the optimistic marginal value in a fully decentralized manner. Additional algorithmic details are provided in Appendix C.

5 EXPERIMENT

We design experiments to answer four questions. (1) Can DAC capture non-stationary task dynamics through its learned discrete contexts? (See Sec. 5.1) (2) Can DAC improve fully decentralized cooperative policy learning by alleviating both NS and RO? (See Sec. 5.2) (3) Which components contribute to its performance gains? The complete ablation study is presented in Appendix D.4. It examines the effects of k and m, compares context maximization with random context selection when computing ϕ^i, and evaluates DAC against teammate-modeling baselines. (4) How large is the performance gap between DAC and representative value decomposition methods? (See Appendix D.6)

5.1 DIDACTIC EXAMPLE

For question (1), we introduce a didactic single-agent task, named Switching Wind, where a single agent moves along a line and an external wind (modeled as periodically changing Gaussian noise) changes the transition dynamics. This abstracts multi-agent settings, where changes in other agents' policies induce NS in each agent's local dynamics. Formally, the state $s_t \in \mathbb{R}$ is the agent's position and the action $a_t \in [-1, 1]$ is a movement scalar. The state transition follows $s_{t+1} = s_t + a_t + \epsilon_t$, where ϵ_t depends on the current task mode: Calm ($\epsilon \sim \mathcal{N}(0, 0.1)$), Headwind ($\epsilon \sim \mathcal{N}(-1.5, 0.3)$), or Tailwind ($\epsilon \sim \mathcal{N}(1.5, 0.3)$). An episode lasts 60 steps and the task mode switches every 20 steps.

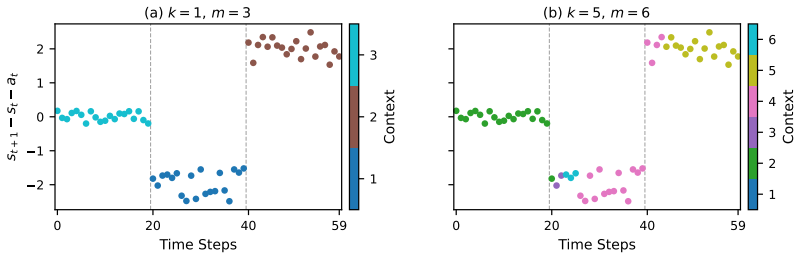


Figure 2: Visualization of the learned dynamics-aware contexts in Switching Wind.

We train the VAE-style network using one episode of randomly sampled transitions, and visualize the learned discrete contexts. As shown in Fig. 2, with a single-transition window ($k = 1$) and 3 contexts ($m = 3$), the learned contexts align uniquely with the underlying task modes, demonstrating the model's ability to capture non-stationary dynamics. With longer sliding windows ($k > 1$), transitions around mode boundaries contain dynamics from multiple modes and therefore cannot be represented by the three mode-specific contexts alone. Accordingly, with $k = 5$ and $m = 6$, additional contexts emerge around the switching points to characterize the mixed dynamics within the sliding window.

Compared with single-transition estimation ($k = 1$), sliding-window-based estimation introduces a short delay in detecting dynamics changes but provides more robust context inference. Aggregating multiple transitions reduces sensitivity to stochastic noise and produces more stable context assignments, whereas single-transition estimation responds immediately to dynamics changes but is more susceptible to noisy transitions, resulting in frequent context oscillations. We further evaluate the effectiveness of sliding-window-based context estimation in the subsequent experiments.

5.2 COMPARISON RESULTS

For question (2), we compare DAC against multiple fully decentralized baselines, including value-based methods (IQL, Hysteretic Q-learning, and I2Q) and policy-based ones (DPO Su & Lu (2024) and TVPO Su & Lu (2025)), across the matrix game, differential game, predator-prey, and SMAC.[1] Detailed descriptions of the baselines, tasks, and experimental settings are provided in Appendix D.

Matrix Game. As shown in Tab. 1a, two agents within this game need to select the optimal joint action (a^1, a^1) to achieve the best reward $+8$. However, under the fully decentralized learning paradigm, each agent maintains higher value estimations regarding its local actions a^2 and a^3, when the other agent selects actions uniformly at random. This gives rise to the RO problem.

[1]Our primary focus is the comparison among value-based fully decentralized methods. DPO and TVPO are included only as empirical references and are therefore not discussed in depth.

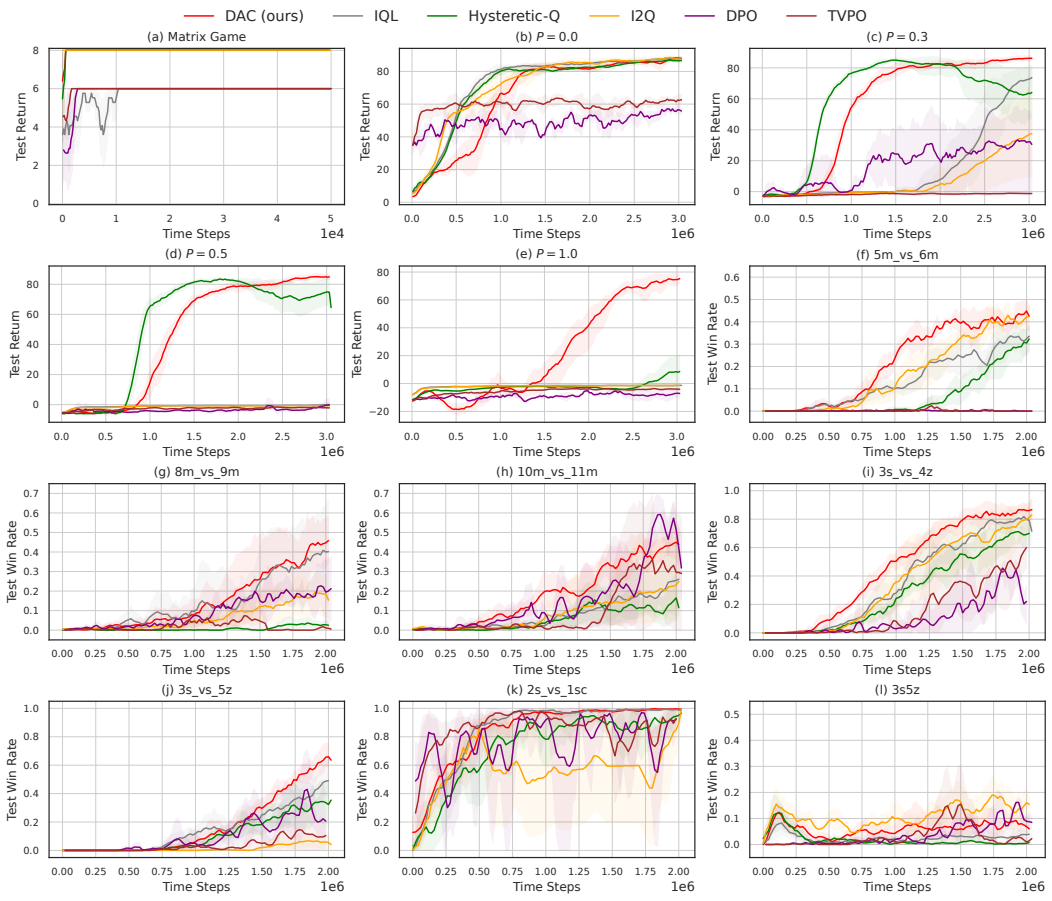


Figure 3: Comparison results on the matrix game, predator-prey, and several SMAC maps.

As shown in Fig. 3 (a), IQL converges to a sub-optimal joint action with reward $+6$, demonstrating that the average-based projection followed by IQL is susceptible to RO. In contrast, DAC learns the optimal joint action with reward $+8$ by selecting actions according to optimistic marginal values. Hysteretic Q-learning and I2Q also solve this simple task through optimistic value function updates or idealized transitions, whereas DPO and TVPO remain at suboptimal returns.

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 \ c	0	1	2	3
$a^1(8.03)$	~~-0.7~~	-11	8.03	~~-5.0~~
$a^2(5.96)$	0.01	-12	5.96	0.00
$a^3(5.95)$	0.06	-12	5.95	-0.0

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

A^2 \ c	0	1	2	3
$a^1(8.03)$	~~-0.7~~	-11	8.03	~~-5.0~~
$a^2(5.96)$	0.01	-12	5.96	0.01
$a^3(5.94)$	0.07	-12	5.94	-0.0

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

Table 1: The payoff matrix and value functions learned by DAC. We set the number of discrete contexts to 4. The context-based value functions $Q^1(s, a, c)$ and $Q^2(s, a, c)$ over all contexts are shown in (b) and (c), respectively. The optimistic marginal values $\phi^1(s, a)$ and $\phi^2(s, a)$ are reported in parentheses after each local action. $Q^i(s, a, c)$ with unattainable value is marked by ~~$Q^i(s, a, c)$~~.

Furthermore, to analyze the representational capabilities of the contexts, we present the per-agent context-based value function and optimistic marginal value learned by DAC. As depicted in Tab. 1b and Tab. 1c, for agent 1 or 2, the context-based value function accurately approximates the rewards of all possible joint actions, which validates the effectiveness of dynamics-aware contexts in representing other agents' joint actions. Moreover, the optimistic marginal values of per-agent local actions satisfy the optimistic property, *i.e.,* which equal to the highest rewards only achieved when other agents select their cooperative actions. Accordingly, each agent can accurately select its local cooperative actions and the efficient selection of optimal joint actions is achieved.

Predator-Prey. To further evaluate the effectiveness of DAC in mitigating RO, we adopt the modified predator-prey task Son et al. (2019). Two predators receive a team reward of $+1$ for simultaneously capturing the prey, whereas a penalty of $-P$ is incurred if only one predator attempts the capture.

Figure 3 (b)-(e) shows that the performance gap widens as the penalty P increases, corresponding to more severe RO. When $P = 0$, the value-based methods learn effective cooperative policies, whereas DPO and TVPO converge more slowly and plateau at lower returns. As P increases, IQL and I2Q degrade markedly. Hysteretic Q-learning remains competitive under moderate penalties ($P = 0.3$ and 0.5) but suffers from degraded asymptotic performance, consistent with overestimation under function approximation. In the most challenging setting, $P = 1.0$, only DAC learns a high-return policy. This robustness arises from DAC's decoupled estimation of the stationary context-based value function and its optimistic marginalization. In contrast, Hysteretic Q-learning injects optimism directly into a single value estimator, making it more susceptible to overestimation.

SMAC. To evaluate the scalability of DAC to more complex tasks, we test all methods on a range of SMAC maps. As depicted in Fig. 3 (f)-(l), three key observations emerge. (1) On moderately difficult maps, including 5m_vs_6m, 8m_vs_9m, 10m_vs_11m, 3s_vs_4z, and 3s_vs_5z, DAC consistently outperforms the baselines. This advantage stems from its effective selection of cooperative joint actions based on the learned optimistic marginal values. Hysteretic Q-learning performs less reliably as overestimation and NS become more pronounced, while IQL also exhibits degraded performance on most maps due to NS. I2Q generally achieves low win rates, suggesting that learning ideal transitions becomes difficult under partial observability. DPO and TVPO exhibit mixed performance: both remain near zero on 5m_vs_6m and substantially underperform DAC on 8m_vs_9m, 3s_vs_4z, and 3s_vs_5z. Both methods improve on 10m_vs_11m, although DPO exhibits considerable instability, while TVPO achieves comparatively better performance on 3s_vs_4z. Nevertheless, their final performance remains below DAC on most maps. (2) On the simpler 2s_vs_1sc map, DAC, IQL, DPO, and TVPO all attain high win rates, although DPO and TVPO exhibit greater variability. Successful cooperative behaviors occur more frequently in this task, enabling most methods to learn effective policies and making the advantage of DAC less pronounced. (3) On the challenging 3s5z map, all methods fail to learn effective policies, with DPO and TVPO also achieving only low win rates.

Discussion. DAC currently relies on full observability and deterministic dynamics to learn informative dynamics-aware contexts. Although SMAC is partially observable, its rich local observations Wang et al. (2019) and largely deterministic dynamics enable effective context learning. To further examine the applicability of DAC, we consider both continuous-control tasks and environments with stronger partial observability and stochasticity. As shown in Appendix D.5, DAC outperforms the baselines in the differential game but loses its advantage in SMACv2 Ellis et al. (2023). This contrast reveals a limitation of DAC's current formulation under severe partial observability and stochastic dynamics.

6 Conclusion

This paper presents DAC as a unified dynamics-aware context modeling framework to address both NS and RO in fully decentralized cooperative MARL. DAC formalizes the local task perceived by each agent as a CMDP, and learns dynamics-aware contexts from step-wise transition distributions. Our theoretical analysis establishes that each context uniquely corresponds to an equivalence class of the other agents' joint actions. Based on these contexts, DAC learns a context-based value function for each agent to enable stationary policy updates, and derives an optimistic marginal value to encourage the selection of cooperative joint actions. Extensive experiments empirically validate its effectiveness.

Limitations and Future Work. We identify four limitations that warrant further investigation. (1) DAC employs a VAE-style architecture to learn dynamics-aware contexts, whose instability during early training may interfere with value function learning. This issue could be mitigated by pretraining the context model during a warm-up phase and subsequently freezing it, or by adopting non-learning-based context construction methods, such as hash tables. (2) Discrete contexts may have limited representational capacity, particularly in complex environments. (3) Unconstrained maximization over all contexts may select out-of-distribution contexts and produce unreliable marginal values. Addressing (2) and (3) requires richer context representations and scalable maximization restricted to the valid context support conditioned on each state–local-action pair. (4) Severe partial observability and stochastic dynamics impair context inference. Developing belief-based, uncertainty-aware context models may improve the robustness of DAC in such tasks. We leave these directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist in developing and refining certain steps of the theoretical proofs, polishing the manuscript for grammar, clarity, and academic style, and modifying code used to visualize experimental results. All AI-assisted proof arguments were independently examined and verified by the authors. Suggested textual revisions were manually reviewed for technical accuracy and consistency with our intended claims, while AI-assisted plotting code was inspected and tested to ensure that the resulting figures accurately reflected the underlying experimental data. Generative AI was not used to generate experimental data or make final scientific conclusions, and no other disclosure-relevant uses apply. The authors take full responsibility for the final content of this work, including all text, claims, proofs, code, and artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work focuses on a methodological study of fully decentralized cooperative MARL, with all experiments conducted in simulated environments. It does not involve human participants, personal or sensitive data, or real-world deployment. We therefore do not identify any specific ethical concerns beyond standard requirements for research integrity, transparency, and reproducibility.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the source code of DAC as part of the supplementary materials. Detailed descriptions of the algorithm and experimental settings are provided in Appendix C and Appendix D, respectively. The complete theoretical analysis, including the assumptions and proofs supporting our claims, is presented in Appendix B. All benchmarks used in our experiments, including the matrix game, predator-prey, SMAC, and SMACv2, have publicly available implementations, and their corresponding original papers are cited in the manuscript.

REFERENCES

Jiahan Cao, Lei Yuan, Jianhao Wang, Shaowei Zhang, Chongjie Zhang, Yang Yu, and De-Chuan Zhan. Linda: Multi-agent local information decomposition for awareness of teammates. *Science China Information Sciences*, 66(8):182101, 2023.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in neural information processing systems*, 36:37567–37593, 2023.

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*, pp. 1146–1155. PMLR, 2017.

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.

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*, pp. 3930–3941. PMLR, 2021.

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

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*, pp. 685–697, 2023.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

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

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. URL http://arxiv.org/abs/1312.6114.

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*, pp. 535–542, 2000.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

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.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE transactions on knowledge and data engineering*, 31(12):2346–2363, 2018.

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*, pp. 64–69. IEEE, 2007.

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.

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*, pp. 2681–2690. PMLR, 2017.

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*, pp. 801–803, 2006.

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, 2020a.

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, 2020b.

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.

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*, pp. 5887–5896. PMLR, 2019.

Kefan Su and Zongqing Lu. A fully decentralized surrogate for multi-agent policy optimization. *Transactions on Machine Learning Research*, 2024.

Kefan Su and Zongqing Lu. f-divergence policy optimization in fully decentralized cooperative MARL. *Transactions on Machine Learning Research*, 2025.

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*, pp. 1791–1799, 2024.

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.

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

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

Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decomposable value functions via communication minimization. *arXiv preprint arXiv:1910.05366*, 2019.

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, 2020b.

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

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.

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.

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*, pp. 264–285. PMLR, 2021.

A VARIATIONAL INFERENCE

For modeling the task dynamics distribution of $\tau^i_{t-k+1:t}$, we assume that this distribution can be represented by a latent context c^i_t, and the mapping from the trajectory segment to the context adheres to an unknown distribution $p^i(c^i_t|\tau^i_{t-k+1:t})$. We introduce a variational distribution $q^i(c^i_t|\tau^i_{t-k+1:t})$ to approximate it, and optimize this approximation by minimizing the KL-divergence between them:

$$
\begin{aligned}
&D_{\mathrm{KL}}(q^i(c^i_t|\tau^i_{t-k+1:t})||p^i(c^i_t|\tau^i_{t-k+1:t})) \\
&= \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log \frac{q^i(c^i_t|\tau^i_{t-k+1:t})}{p^i(c^i_t|\tau^i_{t-k+1:t})} \\
&= \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log q^i(c^i_t|\tau^i_{t-k+1:t}) - \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log p^i(c^i_t|\tau^i_{t-k+1:t}) \\
&= \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log q^i(c^i_t|\tau^i_{t-k+1:t}) - \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log \frac{p^i(c^i_t, \tau^i_{t-k+1:t})}{p^i(\tau^i_{t-k+1:t})} \\
&= \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log q^i(c^i_t|\tau^i_{t-k+1:t}) - \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log \frac{p^i(\tau^i_{t-k+1:t}|c^i_t)p^i(c^i_t)}{p^i(\tau^i_{t-k+1:t})} \\
&= \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log \frac{q^i(c^i_t|\tau^i_{t-k+1:t})}{p^i(c^i_t)} - \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log p^i(\tau^i_{t-k+1:t}|c^i_t) + \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log p^i(\tau^i_{t-k+1:t}) \\
&= \log p^i(\tau^i_{t-k+1:t}) + D_{\mathrm{KL}}(q^i(c^i_t|\tau^i_{t-k+1:t})||p^i(c^i_t)) - \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log p^i(\tau^i_{t-k+1:t}|c^i_t),
\end{aligned}
\tag{9}
$$

where $p^i(c^i_t)$ denotes the prior distribution of the latent context, and $\log p^i(\tau^i_{t-k+1:t})$ is the evidence term that can be regarded as a constant. Thereby, minimizing $D_{\mathrm{KL}}(q^i(c^i_t|\tau^i_{t-k+1:t})||p^i(c^i_t|\tau^i_{t-k+1:t}))$ can be achieved by the equation below:

$$
\max \mathbb{E}_{q^i(c^i_t|\tau^i_{t-k+1:t})} \log p^i(\tau^i_{t-k+1:t}|c^i_t) - D_{\mathrm{KL}}(q^i(c^i_t|\tau^i_{t-k+1:t})||p^i(c^i_t)). \tag{10}
$$

B Complete Theoretical Analysis

This appendix provides the complete theoretical analysis supporting Sec. 4.3. In particular, we first formalize the assumptions underlying the dynamics-aware context, then establish the identifiability of the dynamics-equivalence class from local dynamics, and finally provide the complete proofs of Theorems 1 and 2.

B.1 Preliminaries and Representation Assumptions

As in Sec. 4.3, we consider a deterministic MMDP.

Assumption 1 (Deterministic environment)**.** *For any state s_t and joint action $\boldsymbol{a}_t = (a_t^i, a_t^{-i})$, the next state and team reward are uniquely determined by:*

$$s_{t+1} = F(s_t, a_t^i, a_t^{-i}), \qquad r_t = R(s_t, a_t^i, a_t^{-i}), \tag{11}$$

where F and R are deterministic.

For a fixed (s_t, a_t^i), we define the local dynamics outcome induced by a_t^{-i} as:

$$\Psi^i_{s_t, a_t^i}(a_t^{-i}) := \Big(F(s_t, a_t^i, a_t^{-i}), R(s_t, a_t^i, a_t^{-i})\Big). \tag{12}$$

Under Assumption 1, the influence of the unobserved joint action a_t^{-i} on the current local transition is completely manifested through $\Psi^i_{s_t, a_t^i}(a_t^{-i})$. Distinct joint actions of the other agents therefore need not be distinguished if they induce exactly the same local dynamics.

For completeness, we recall the dynamics-equivalence relation introduced in Sec. 4.3.

Definition 2 (Dynamics-equivalent joint actions)**.** For a fixed state-local-action pair (s_t, a_t^i), two joint actions a_t^{-i} and $\tilde{a}_t^{-i}$ of the other agents are *dynamics-equivalent* with respect to agent i, denoted by $a_t^{-i} \sim^i_{s_t, a_t^i} \tilde{a}_t^{-i}$, if and only if $\Psi^i_{s_t, a_t^i}(a_t^{-i}) = \Psi^i_{s_t, a_t^i}(\tilde{a}_t^{-i})$. Equivalently,

$$F(s_t, a_t^i, a_t^{-i}) = F(s_t, a_t^i, \tilde{a}_t^{-i}), \quad R(s_t, a_t^i, a_t^{-i}) = R(s_t, a_t^i, \tilde{a}_t^{-i}). \tag{13}$$

The corresponding equivalence class is

$$[a_t^{-i}]^i_{s_t, a_t^i} := \left\{ \tilde{a}_t^{-i} \in \boldsymbol{A}^{-i} | \tilde{a}_t^{-i} \sim^i_{s_t, a_t^i} a_t^{-i} \right\}. \tag{14}$$

Definition 2 partitions the other agents' joint action space $\boldsymbol{A}^{-i}$ according to their effects on the local dynamics experienced by agent i. Therefore, recovering the exact identity of a_t^{-i} is unnecessary: agent i only needs to distinguish joint actions that induce different transition or reward outcomes.

We first establish that the realized local dynamics are sufficient to identify such an equivalence class.

Lemma 1 (Identifiability from local dynamics)**.** *Under Assumption 1, for a fixed (s_t, a_t^i), the observed outcome (s_{t+1}, r_t) uniquely determines the dynamics-equivalence class $[a_t^{-i}]^i_{s_t, a_t^i}$.*

Proof. By Assumption 1,

$$(s_{t+1}, r_t) = \Psi^i_{s_t, a_t^i}(a_t^{-i}). \tag{15}$$

Consider any $\tilde{a}_t^{-i} \in \boldsymbol{A}^{-i}$. According to Definition 2,

$$\tilde{a}_t^{-i} \in [a_t^{-i}]^i_{s_t, a_t^i} \tag{16}$$

if and only if:

$$\Psi^i_{s_t, a_t^i}(\tilde{a}_t^{-i}) = \Psi^i_{s_t, a_t^i}(a_t^{-i}) = (s_{t+1}, r_t). \tag{17}$$

Hence, all and only the joint actions belonging to the same equivalence class produce the same local dynamics outcome under the fixed (s_t, a_t^i). Therefore, (s_{t+1}, r_t) uniquely identifies $[a_t^{-i}]^i_{s_t, a_t^i}$. □

Lemma 1 establishes an important intermediate result: although a_t^{-i} is inaccessible, the equivalence class containing a_t^{-i} is identifiable from the local dynamics that it induces. We next show that the learned dynamics-aware context inherits this identifiability.

Recall that the context is inferred as $c_t^i \sim q^i\left(c_t^i \mid \tau_{t-k+1:t}^i\right)$,where $\tau_{t-k+1:t}^i$ denotes a recent window of local transitions. The temporal window provides additional evidence for robust context inference, whereas the inferred c_t^i is intended to characterize the local dynamics currently governing the transition at time t. We characterize the ideal representation learned by the dynamics-modeling objective through the following two properties.

Assumption 2 (Dynamics sufficiency)**.** *The dynamics-aware context preserves sufficient information about the current local dynamics outcome. Specifically, there exists a decoder D^i such that, for every reachable transition, $D^i(s_t, a_t^i, c_t^i) = (s_{t+1}, r_t)$.*

Assumption 2 corresponds to the ideal optimum of the dynamics reconstruction objective in Eq. (6). It requires that, conditioned on (s_t, a_t^i), no information necessary for recovering the realized transition and reward is lost in c_t^i.

Dynamics sufficiency alone does not exclude redundant latent distinctions: two different context values may still decode to exactly the same dynamics outcome. We therefore additionally characterize an ideal compressed dynamics representation.

Assumption 3 (Dynamics minimality)**.** *For a fixed (s_t, a_t^i), local transitions with identical dynamics outcomes are assigned the same dynamics-aware context: $(s_{t+1}, r_t) = (\tilde{s}_{t+1}, \tilde{r}_t) \Longrightarrow c_t^i = \tilde{c}_t^i$.*

Assumption 3 requires the context to introduce no distinctions that are irrelevant to the local dynamics outcome. Together, Assumptions 2 and 3 characterize c_t^i as an ideal minimal sufficient representation of the currently experienced local dynamics. In our implementation, the reconstruction term directly encourages dynamics sufficiency, while the KL regularization encourages compression of information unnecessary for dynamics prediction. Assumptions 2 and 3 describe the idealized representation toward which the learning objective is directed.

Remark 1 (Role of the sliding transition window). Although the encoder conditions on the recent transition window $\tau_{t-k+1:t}^i$, the theoretical object represented by c_t^i is the dynamics currently governing the transition at time t. The preceding assumptions are therefore imposed on the resulting context representation rather than directly on the encoder input. Intuitively, the additional transitions in the window provide evidence for inferring the current dynamics, while histories corresponding to the same current local dynamics are ideally compressed into the same dynamics-aware context.

B.2 Proof of the Context-Joint-Action Equivalence Correspondence

We now provide the complete proof of Theorem 1 in Sec. 4.3 (*i.e.,* establish the correspondence between the learned dynamics-aware context and the unobserved joint actions of the other agents).

Theorem 3 (Context-joint-action equivalence)**.** *Under Assumptions 1 2 3, for any fixed (s_t, a_t^i), two reachable transitions are represented by the same dynamics-aware context if and only if their corresponding joint actions of the other agents are dynamics-equivalent: $c_t^i = \tilde{c}_t^i \iff a_t^{-i} \sim^i_{s_t, a_t^i} \tilde{a}_t^{-i}$. Consequently, for a fixed (s_t, a_t^i), there exists a one-to-one correspondence between the set of reachable dynamics-aware contexts and the set of reachable dynamics-equivalence classes: $c_t^i \longleftrightarrow [a_t^{-i}]^i_{s_t, a_t^i}$.*

Proof. We prove the two directions separately.

First, suppose

$$a_t^{-i} \sim^i_{s_t, a_t^i} \tilde{a}_t^{-i}. \tag{18}$$

By Definition 2,

$$\Psi^i_{s_t, a_t^i}(a_t^{-i}) = \Psi^i_{s_t, a_t^i}(\tilde{a}_t^{-i}), \tag{19}$$

which implies

$$(s_{t+1}, r_t) = (\tilde{s}_{t+1}, \tilde{r}_t). \tag{20}$$

By Assumption 3, identical local dynamics outcomes are represented by the same dynamics-aware context:

$$c_t^i = \tilde{c}_t^i. \tag{21}$$

Conversely, suppose

$$c_t^i = \tilde{c}_t^i. \tag{22}$$

By Assumption 2,

$$(s_{t+1}, r_t) = D^i(s_t, a_t^i, c_t^i), \quad (\tilde{s}_{t+1}, \tilde{r}_t) = D^i(s_t, a_t^i, \tilde{c}_t^i). \tag{23}$$

Since $c_t^i = \tilde{c}_t^i$,

$$(s_{t+1}, r_t) = (\tilde{s}_{t+1}, \tilde{r}_t). \tag{24}$$

Under Assumption 1, this is equivalent to

$$\Psi^i_{s_t, a_t^i}(a_t^{-i}) = \Psi^i_{s_t, a_t^i}(\tilde{a}_t^{-i}). \tag{25}$$

Definition 2 therefore gives

$$a_t^{-i} \sim^i_{s_t, a_t^i} \tilde{a}_t^{-i}. \tag{26}$$

Hence,

$$c_t^i = \tilde{c}_t^i \iff a_t^{-i} \sim^i_{s_t, a_t^i} \tilde{a}_t^{-i}. \tag{27}$$

□

Theorem 3 reveals an additional interpretation of the dynamics-aware context beyond its original role in characterizing non-stationary local task dynamics. Although DAC never observes a_t^{-i}, an informative and compact dynamics-aware context implicitly identifies the equivalence class of the other agents' joint actions that could have generated the currently experienced local dynamics. Thus, c_t^i should not be interpreted as recovering the exact identity of a_t^{-i}; instead, it represents the aspect of a_t^{-i} that is identifiable through its effect on agent i's local dynamics.

B.3 Joint Action-Value Consistency

We next establish that joint actions belonging to the same dynamics-equivalence class are also indistinguishable with respect to their joint action values.

Corollary 1 (Value consistency within a dynamics-equivalence class)**.** *Under Assumption 1, for any joint policy $\boldsymbol{\pi}$, if $a_t^{-i} \sim^i_{s_t, a_t^i} \tilde{a}_t^{-i}$, then $Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i}) = Q^{\boldsymbol{\pi}}(s_t, a_t^i, \tilde{a}_t^{-i})$.*

Proof. Under deterministic dynamics,

$$Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i}) = R(s_t, a_t^i, a_t^{-i}) + \gamma V^{\boldsymbol{\pi}}\left(F(s_t, a_t^i, a_t^{-i})\right). \tag{28}$$

If $a_t^{-i} \sim^i_{s_t, a_t^i} \tilde{a}_t^{-i}$, Definition 2 guarantees:

$$R(s_t, a_t^i, a_t^{-i}) = R(s_t, a_t^i, \tilde{a}_t^{-i}), \quad F(s_t, a_t^i, a_t^{-i}) = F(s_t, a_t^i, \tilde{a}_t^{-i}). \tag{29}$$

Therefore, the two joint actions have identical immediate rewards and next states, which gives:

$$Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i}) = Q^{\boldsymbol{\pi}}(s_t, a_t^i, \tilde{a}_t^{-i}). \tag{30}$$

□

Corollary 1 shows that the distinctions discarded by the dynamics-aware context are also irrelevant to value estimation. Conversely, whenever two joint actions of the other agents induce different joint action values, they cannot belong to the same dynamics-equivalence class and, under Theorem 3, cannot share the same dynamics-aware context. This establishes that the context retains all distinctions in a^{-i} required for representing the joint action value.

B.4 Proof of Sufficiency for Joint Action-Value Representation

We finally prove Theorem 2 in Sec. 4.3.

Theorem 4 (Joint action-value representation sufficiency)**.** *Under Assumptions 1 2 3, for any joint policy $\boldsymbol{\pi}$, there exists a context-based value function $\bar{Q}^{i,\boldsymbol{\pi}} : S \times A^i \times \mathcal{C} \to \mathbb{R}$ such that, for every reachable transition generated by (s_t, a_t^i, a_t^{-i}), $\bar{Q}^{i,\boldsymbol{\pi}}(s_t, a_t^i, c_t^i) = Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i})$.*

Proof. Fix (s_t, a_t^i) and a reachable context $c_t^i \in \mathcal{C}$. Define the set of joint actions of the other agents represented by c_t^i as $A^{-i}_{s_t,a_t^i}(c_t^i) = \{a_t^{-i} | a_t^{-i} \text{ induces } c_t^i \text{ under } (s_t, a_t^i)\}$. Choose an arbitrary representative $a_c^{-i} \in A^{-i}_{s_t,a_t^i}(c_t^i)$ and define:

$$\bar{Q}^{i,\boldsymbol{\pi}}(s_t, a_t^i, c_t^i) := Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_c^{-i}). \tag{31}$$

It remains to verify that this definition is independent of the selected representative. Consider any $a_t^{-i}, \tilde{a}_t^{-i} \in A^{-i}_{s_t,a_t^i}(c_t^i)$. Both joint actions induce the same context. Therefore, by Theorem 3,

$$a_t^{-i} \sim^i_{s_t,a_t^i} \tilde{a}_t^{-i}. \tag{32}$$

Corollary 1 then gives:

$$Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i}) = Q^{\boldsymbol{\pi}}(s_t, a_t^i, \tilde{a}_t^{-i}). \tag{33}$$

Hence, every joint action represented by the same context has the same joint action value, and Eq. (31) is independent of the selected representative.

In particular, for the context c_t^i induced by any reachable a_t^{-i},

$$\bar{Q}^{i,\boldsymbol{\pi}}(s_t, a_t^i, c_t^i) = Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i}). \tag{34}$$

□

Theorem 4 establishes that explicit access to a_t^{-i} is unnecessary for representing its effect on the joint action value. Instead, the dynamics-aware context serves as a sufficient surrogate:

$$Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i}) = \bar{Q}^{i,\boldsymbol{\pi}}(s_t, a_t^i, c_t^i). \tag{35}$$

Accordingly, if the learned local value function $Q^i(s_t, a_t^i, c_t^i)$ has sufficient representational capacity and accurately approximates the context-based value function $Q^i(s_t, a_t^i, c_t^i) \approx \bar{Q}^{i,\boldsymbol{\pi}}(s_t, a_t^i, c_t^i)$, then

$$Q^i(s_t, a_t^i, c_t^i) \approx Q^{\boldsymbol{\pi}}(s_t, a_t^i, a_t^{-i}). \tag{36}$$

This result explains the role of the dynamics-aware context in alleviating the representational ambiguity underlying RO. A context-free local value function $q^i(s_t, a_t^i)$ cannot distinguish different joint action values associated with the same (s_t, a_t^i). In contrast, conditioning on c_t^i retains sufficient information about the dynamics-relevant effect of the unobserved joint action a_t^{-i} to represent the corresponding joint action value.

C Algorithmic Details

C.1 Optimization of the VAE-Style Network

For the VAE-style network, we parameterize the variational distribution $q^i(c_t^i | \tau^i_{t-k+1:t})$ using an encoder that takes the most recent k local transitions as input and outputs the logits of a categorical context distribution. Given the inferred context, the decoder reconstructs the corresponding next states and rewards within the sliding window.

To learn discrete contexts, we consider two objectives. First, the posterior $q^i(c_t^i | \tau^i_{t-k+1:t})$ is regularized toward a discrete prior $p^i(c_t^i)$, which is defined as a uniform categorical distribution. Accordingly, the KL regularization term in Eq. (4) is implemented as:

$$\mathcal{L}_{\mathrm{KL}}(\omega^i_{\mathrm{enc}}) = D_{\mathrm{KL}}(q^i(c_t^i | \tau^i_{t-k+1:t}) || p^i(c_t^i)) = \sum_{j=1}^{m} q_j^i(c_t^i | \tau^i_{t-k+1:t}) \log \frac{q_j^i(c_t^i | \tau^i_{t-k+1:t})}{1/m}, \tag{37}$$

where the encoder is parameterized by ω^i_{enc}, m denotes the number of discrete context categories (*i.e.,* the dimension of context), and q^i_j represents the probability that c^i_t belongs to the j-th category. This probability is obtained by applying a softmax function to the encoder outputs.

Second, to ensure discrete context sampling while retaining differentiable, we employ the Gumbel-Softmax Jang et al. (2016) to the encoder outputs. Given a sampled context c^i_t, the decoder reconstructs the local task dynamics by maximizing $\sum_{h=t-k+1}^{t} \log p^i(s_{h+1}, r_h | s_h, a^i_h, c^i_t)$, which corresponds to the following reconstruction loss:

$$\begin{aligned} &\mathcal{L}_{\text{REC}}(\omega^i_{\text{enc}}, \omega^i_{\text{dec}}) \\ &= \mathbb{E}_{\tau^i_{t-k+1:t} \sim \mathcal{D}^i, c^i_t \sim q^i(c^i_t | \tau^i_{t-k+1:t})} \sum_{h=t-k+1}^{t} [(s_{h+1} - f_{\text{s}}(s_h, a^i_h, c^i_t))^2 + (r_h - f_{\text{r}}(s_h, a^i_h, c^i_t))^2], \end{aligned} \tag{38}$$

where the decoder consists of two separate networks, f_{s} and f_{r}, used to respectively predict the next states and rewards. ω^i_{dec} denotes the decoder parameters. The Gumbel-Softmax enables differentiable training of the encoder. As a result, the encoder and decoder are jointly optimized using the loss function $\mathcal{L}_{\text{V}} = \mathcal{L}_{\text{REC}} + \beta \mathcal{L}_{\text{KL}}$, where β is a scaling factor. We empirically find that $\beta = 0.001$ works well across the evaluated tasks.

C.2 Implementation

We consider two architectures for the context-based value function $Q^i(s_t, a^i_t, c^i_t)$.

Normal-Net. The network follows a standard FC-GRU backbone, consisting of a fully connected layer with 64 units and ReLU activation followed by a GRU module. The resulting hidden state is concatenated with the inferred context and fed into the final linear layer to produce Q-values for all local actions.

Hyper-Net. The network uses the same FC-GRU backbone, while the weights and biases of the final linear layer are generated by two separate hyper-networks conditioned on the inferred context. The remaining implementation follows standard IQL in PyMARL Samvelyan et al. (2019).

We use Hyper-Net by default because it allows the context to directly modulate the value function parameters. For tasks with large action spaces, however, generating the output-layer parameters can introduce substantial computational overhead, in which case we adopt Normal-Net instead. In Normal-Net, the context is injected after the recurrent representation rather than concatenated with the raw input. This prevents its influence from being diluted by high-dimensional observations and avoids unnecessarily propagating historical context information through the GRU, while maintaining a simple and computationally efficient architecture.

C.3 Pseudocode

Algorithm 1 summarizes the learning procedure of DAC. We initialize a context-based value function, an encoder, and a decoder for each agent $i \in N$. During data collection, each agent independently selects its local action according to the optimistic marginal value $\phi^i(s_t, a^i_t)$. After each episode, the locally available trajectory of each agent is stored in its individual replay buffer $\mathcal{D}^i$.

During training, each agent samples batches of episodes from $\mathcal{D}^i$ and constructs the sliding-window trajectory segments $\tau^i_{t-k+1:t}$. The corresponding contexts are inferred using $q^i(c^i_t | \tau^i_{t-k+1:t})$. We then compute the context-based value function loss $\mathcal{L}_{\text{C}}$, reconstruction loss $\mathcal{L}_{\text{REC}}$, and KL regularization loss $\mathcal{L}_{\text{KL}}$, and jointly update the associated network parameters. This procedure is repeated until the maximum number of training episodes is reached.

D Experimental Details

D.1 Benchmarks

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

Algorithm 1: Dynamics-Aware Context (DAC)

Initialize hyperparameters and individual replay buffers $\{\mathcal{D}^i\}_{i \in N}$
foreach *agent* $i \in N$ **do**
  Initialize parameters $\theta^i, \omega^i_{\text{enc}}, \omega^i_{\text{dec}}$ of the context-based value function, the encoder, and the decoder
while *the maximum number of training episodes is not reached* **do**
  foreach *time step* t **do**
    foreach *agent* $i \in N$ **do**
      Select a^i_t based on $\phi^i(s_t, a^i_t)$ (Eq. 8)
    Execute the joint action and observe the resulting local transitions
  Store each agent's local trajectory in its replay buffer $\mathcal{D}^i$
  if *the training condition is satisfied* **then**
    foreach *agent* $i \in N$ **do**
      Sample a batch of episodes from $\mathcal{D}^i$
      Construct sliding-window trajectory segments $\tau^i_{t-k+1:t}$ for each sampled trajectory τ^i
      Infer contexts c^i_t using $q^i(c^i_t \mid \tau^i_{t-k+1:t})$
      Compute $\mathcal{L}_{\text{C}}(\theta^i)$, $\mathcal{L}_{\text{REC}}(\omega^i_{\text{enc}}, \omega^i_{\text{dec}})$, and $\mathcal{L}_{\text{KL}}(\omega^i_{\text{enc}})$ according to Eqs. (7), (37), and (38)
      Update θ^i, ω^i_{enc}, and ω^i_{dec} via gradient descent

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 agent, when the other agent uniformly selects its local actions, the sub-optimal local actions a^2 and a^3 may be preferred over the optimal ones a^1, leading to the RO issue.

Predator-Prey. We adopt the modified predator-prey task Son et al. (2019) to evaluate the ability of DAC to alleviate RO. The task is played on a 5×5 grid by two predators and one moving prey. Each predator observes its own position and identity, together with the relative position of the prey, and can move left, right, up, or down, or remain stationary. The predators receive a team reward of $+1$ for simultaneously capturing the prey, a penalty of $-P$ for a unilateral capture attempt, and zero reward for avoiding the prey. After being captured, the prey respawns at a random position. Each episode lasts 100 time steps, and we consider $P \in \{0.0, 0.3, 0.5, 1.0\}$.

Increasing P leaves the optimal joint policy unchanged but raises the cost of failed coordination. Let q denote the probability that the other predator coordinates. The expected immediate reward of attempting a capture is $q - (1 - q)P$, which is positive only if $q > P/(1 + P)$. Accordingly, the required coordination probability increases from 0 to approximately 0.23, 0.33, and 0.5 as P increases from 0 to 0.3, 0.5, and 1.0, respectively. Because successful coordination is rare during early exploration, larger penalties cause unilateral failures to dominate the marginalized action value. Independent learners may therefore undervalue the action associated with the globally optimal joint policy and instead favor the risk-free but suboptimal behavior of avoiding the prey. Thus, increasing P progressively aggravates RO.

Differential Game. Differential game is a fully observable cooperative task with continuous action spaces. The joint state is defined as $s_t = (x^1_t, \ldots, x^N_t) \in [-1, 1]^N$, and each agent selects an action $a^i_t \in [-1, 1]$, which is mapped to an inward effort $e^i_t \in [0, 1]$. The joint transition depends on $C_t = \prod_i e^i_t$: sufficient coordination moves the state toward the global optimum near the origin, whereas insufficient coordination leads to a suboptimal outer region. For agent i, the influence of the other agents is summarized by $z^i_t = \prod_{j \neq i} e^j_t$. Because inward effort is beneficial only when the other agents coordinate, independent learners may undervalue it and converge to the risk-dominant outer region, resulting in RO. We evaluate $N = 3$ and $N = 5$. If each teammate coordinates independently with probability q, successful coordination occurs with probability q^{N-1}, corresponding to q^2 and q^4, respectively. Thus, increasing the number of agents aggravates RO.

SMAC. The StarCraft Multi-Agent Challenge (SMAC) is a widely used benchmark for cooperative MARL. It requires agents to control allied units in a decentralized manner and coordinate against built-

in AI opponents. We evaluate DAC on seven maps with diverse unit compositions and coordination requirements: 5m_vs_6m, 8m_vs_9m, 10m_vs_11m, 3s_vs_4z, 3s_vs_5z, 2s_vs_1sc, and 3s5z.

SMACv2. SMACv2 extends SMAC with procedurally generated combat scenarios that introduce stronger partial observability and stochastic task dynamics. Unit types, team compositions, and initial positions are randomized in each episode, while race-specific sight and attack ranges restrict the information available to individual agents. Agents must therefore infer the underlying task configuration from local observations and learn decentralized closed-loop policies that generalize across diverse, previously unseen scenarios.

D.2 BASELINES

In this paper, we compare DAC with both value-based baselines (IQL, Hysteretic Q-learning, and I2Q), and policy-based baselines (DPO and TVPO). Our primary focus is on value-based fully decentralized MARL; the policy-based methods are included only as empirical references. We briefly introduce each baseline below.

IQL. IQL learns a decentralized value function $Q^i(s_t, a_t^i)$ for each agent i, and updates it following the standard Q-learning paradigm. However, since IQL agent ignores other agents' actions and policies, it suffers from NS during value function updates. What's more, the $Q^i(s_t, a_t^i)$ learned by IQL is an average-based projection of the true joint action value function, which induces RO issue during value estimation.

Hysteretic Q-learning. This algorithm employs an optimistic value function update for each agent i's $Q^i(s_t, a_t^i)$ to directly approximate the optimistic projection. Such optimistic update assumes that other agents always select their cooperative actions, thereby eliminating the negative effects caused by other agents' sub-optimal or exploratory action selections and addressing the RO issue. However, the NS issue remains unsolvable.

I2Q. I2Q addresses both NS and RO by shaping ideal transitions, which are constructed by selecting next states with the highest QSS value. These transitions implicitly assume that other agents follow cooperative local policies. By learning based on such transitions, I2Q guides IQL agents toward optimal cooperative policies.

DPO. DPO derives a fully decentralized surrogate objective that lower-bounds joint-policy improvement and decomposes across agents. Under its theoretical assumptions, independently optimizing the resulting local actor-critic objectives guarantees monotonic improvement and convergence of the joint return. In practice, DPO uses two adaptive coefficients to penalize both the square root of the KL divergence and the KL divergence between the old and updated local policies. Its practical implementation nevertheless relies on several approximations that may weaken the theoretical guarantee.

TVPO. TVPO formulates independent policy optimization under a general f-divergence framework and instantiates it using total variation distance. It introduces decentralized value functions and a policy-iteration procedure for which monotonic joint-policy improvement and convergence can be established theoretically. In practice, an adaptive coefficient regulates local policy changes. Compared with the heuristic clipping used by IPPO and the KL-based surrogate adopted by DPO, TVPO provides a principled total-variation-based formulation while retaining fully decentralized training and execution.

D.3 EXPERIMENTAL SETUPS

We implement all value-based methods within the PyMARL framework. For IQL, we use the original PyMARL implementation. For I2Q, we adopt the official implementation and recommended hyperparameters for the predator-prey and SMAC experiments, while reporting its matrix-game results from the original paper. For Hysteretic Q-learning, we set $\beta = 0.01$ on the matrix game. On predator-prey, SMAC, and SMACv2, we tune β over $0.01, 0.1, 0.3, 0.5, 0.7$ for each task and conduct multi-seed evaluations using the best-performing setting. For the policy-based methods DPO and TVPO, we use their official implementations across all evaluated tasks.

For DAC, we instantiate $Q^i(s_t, a_t^i, c_t^i)$ using Hyper-Net on the matrix game, predator-prey, and differential game, and Normal-Net on SMAC and SMACv2. Unless otherwise specified, common

Table 2: Hyper-parameters of DAC across all tasks.

Tasks	k	m	β	$T_{epsilon}$	T_{max}	N_{max}	N_{batch}	Optimizer	α	γ	ϵ_{max}	ϵ_{min}
matrix game	1	4		50k	50000							
$P = 0.0$	10	10		1500k	3050000							
$P = 0.3$	10	10		1500k	3050000							
$P = 0.5$	10	10		1500k	3050000							
$P = 1.0$	5	10		1500k	3050000							
2s_vs_1sc	10	10		50k	2050000							
3s5z	10	5		50k	2050000							
3s_vs_4z	15	20		50k	2050000							
3s_vs_5z	15	20		50k	2050000							
5m_vs_6m	15	20	0.001	50k	2050000	5000	32	RMSprop	0.0005	0.99	1.0	0.05
8m_vs_9m	5	10		50k	2050000							
10m_vs_11m	15	10		50k	2050000							
protoss_5_vs_5	10	20		100k	10050000							
protoss_10_vs_10	10	10		100k	10050000							
terran_5_vs_5	10	10		100k	10050000							
terran_10_vs_10	10	20		100k	10050000							
zerg_5_vs_5	15	10		100k	10050000							
terran_10_vs_11	5	20		100k	10050000							
differential game	1	5		/	550000	550000	100		0.001		/	/

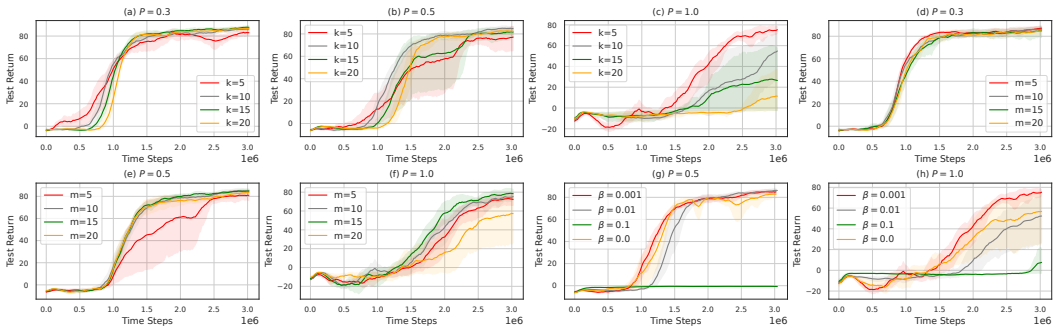


Figure 4: Ablation results for different values of k, m, and β.

training settings, including the learning rate and batch size, are kept consistent across methods. Detailed hyperparameter settings for DAC are provided in Table 2. Here, k is the sliding-window length, and m is the context dimension, corresponding to the number of discrete contexts. The coefficient β balances the KL-divergence and reconstruction losses. For ϵ-greedy exploration, T_ϵ denotes the number of steps over which ϵ decreases from $\epsilon_{\max}$ to $\epsilon_{\min}$. Moreover, $T_{\max}$ is the total number of training steps, $N_{\max}$ is the capacity of each agent's replay buffer, N_{batch} is the sampled batch size, α is the learning rate, and γ is the discount factor.

D.4 Ablation Study

We conduct a series of ablation studies to examine the key design choices and mechanisms of DAC: (1) the sliding-window length k, (2) the number of discrete contexts m, (3) the KL-loss coefficient β, (4) maximization over contexts when computing $\phi^i(s_t, a_t^i) = \max_{c \in \mathcal{C}} Q^i(s_t, a_t^i, c)$, and (5) dynamics-aware context modeling compared with explicit teammate modeling.

For components (1), (2), and (3), we evaluate $k \in \{5, 10, 15, 20\}$, $m \in \{5, 10, 15, 20\}$, and $\beta \in \{0.0, 0.001, 0.01, 0.1\}$. As depicted in Fig. 4, DAC with $m = 20$ exhibits degraded performance in predator-prey with $P = 1.0$. This indicates that an excessively large m results in inefficient updates of the context-based value function due to the expanded context space, particularly in challenging tasks where RO limits effective reward signals. Conversely, a small number of discrete contexts (*e.g.,* $m = 5$) fails to represent diverse local task dynamics and the corresponding other agents' diverse joint actions, as evidenced by its poor performance in predator-prey with $P = 0.5$.

Regarding the sliding window length k, a small value may lead to abrupt fluctuations in the inferred dynamics distribution, while a large value may fail to capture timely distributional changes. The

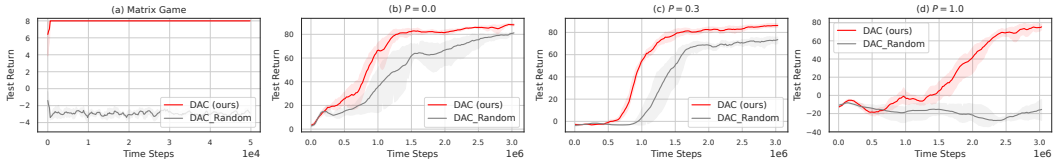


Figure 5: Comparison results between DAC and DAC_Random.

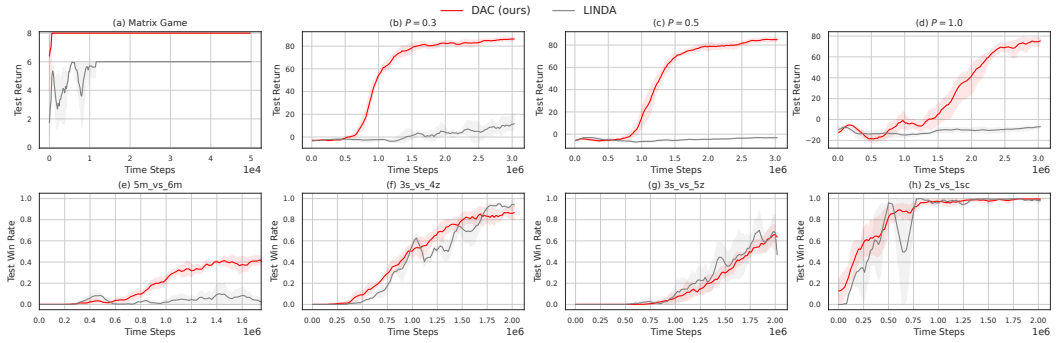


Figure 6: Comparison results between DAC and LINDA.

superior performance of DAC with $k = 10$ in comparison to $k = 5, 15, 20$ in predator-prey with $P = 0.3$ and 0.5 supports this observation. However, DAC with $k = 5$ achieves strong performance in predator-prey with $P = 1.0$, indicating that the optimal choice of k is task-dependent.

For the KL loss coefficient, DAC with $\beta = 0.001$ consistently outperforms $\beta = 0.0$, indicating that the KL loss encourages effective utilization of all discrete contexts and accordingly results in better performance. However, overly large values (*e.g.,* $\beta = 0.01$ and 0.1) degrade performance, highlighting the need to balance reconstruction and KL regularization objectives.

For (4), we construct DAC_Random by replacing context maximization with random context selection: $\phi^i(s_t, a_t^i) = \text{Random}_{c \in \mathcal{C}}\, Q^i(s_t, a_t^i, c)$. This variant isolates the contribution of optimistic marginalization to mitigating RO. As shown in Fig. 5, DAC_Random performs substantially worse when $P = 0.0$ and 0.3 and fails to learn effective policies on the matrix game and predator-prey with $P = 1.0$. These results demonstrate that maximizing over contexts is critical for alleviating RO and promoting cooperative joint-action selection.

For (5), we compare DAC with LINDA Cao et al. (2023), which explicitly infers the joint action of the other agents, $\bar{a}_t^{-i}$, and selects $\arg\max_{a_t^i} Q^i(s_t, a_t^i, \bar{a}_t^{-i})$. Because the original LINDA is built upon QMIX, we integrate its teammate-modeling mechanism into IQL for a fair comparison. As shown in Fig. 6, DAC substantially outperforms LINDA on tasks characterized by RO, including the matrix game and predator-prey. LINDA performs comparably to DAC on 3s_vs_4z, 3s_vs_5z, and 2s_vs_1sc but performs poorly on 5m_vs_6m. This suggests that explicit teammate modeling alone may provide limited benefits when agents fail to learn effective local policies. Overall, the comparison supports the effectiveness of dynamics-aware contexts in representing other agents' joint actions and of optimistic marginalization in mitigating RO.

In summary, the ablations confirm that optimistic marginalization is essential for addressing RO and that dynamics-aware contexts effectively represent other agents' joint actions. The choices of k and m remain task-dependent, motivating adaptive context modeling that reduces manual tuning. Across the evaluated tasks, $\beta = 0.001$ works well and provides a robust balance between context utilization and dynamics reconstruction.

D.5 Complementary Comparison Results

We further examine the applicability of DAC beyond the main benchmarks along two dimensions: continuous action spaces and environments with stronger partial observability and stochasticity. For

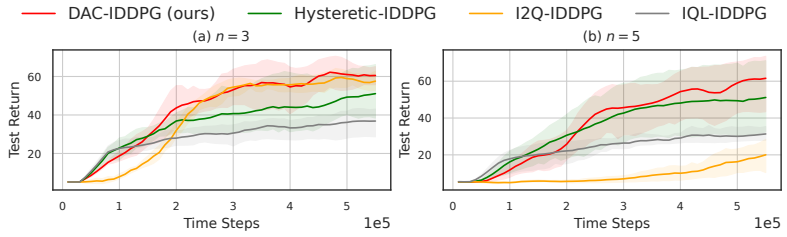


Figure 7: Comparison results on the differential game.

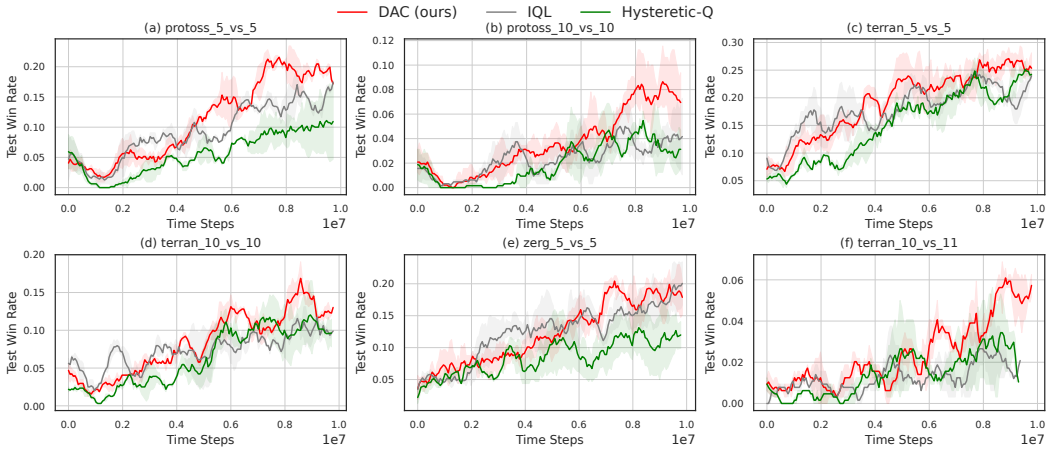


Figure 8: Comparison results on several SMACv2 maps.

continuous control, we implement DAC, Hysteretic Q-learning, I2Q, and IQL within an independent DDPG framework Lillicrap et al. (2015), yielding DAC-IDDPG, Hysteretic-IDDPG, I2Q-IDDPG, and IQL-IDDPG, respectively. As shown in Fig. 7, DAC-IDDPG outperforms all baselines for both $N = 3$ and $N = 5$, demonstrating its ability to mitigate RO in continuous-action tasks.

DAC is currently designed for fully observable environments with deterministic dynamics, which limits its ability to infer reliable contexts under severe partial observability and stochasticity. As shown in Fig. 8, DAC performs comparably to IQL and Hysteretic Q-learning on SMACv2, rather than retaining its advantage observed on SMAC. These results reveal a limitation of the current context-learning formulation in environments with partially observable and stochastic task dynamics.

D.6 Comparison against QMIX and VDN

To analyze the performance gap between DAC and representative value decomposition methods, we evaluate QMIX and VDN on the matrix game, predator-prey, and seven SMAC maps. As shown in Fig. 9, DAC significantly outperforms QMIX and VDN on the matrix game and predator-prey under $P = 0.3, 0.5, 1.0$, where both QMIX and VDN fail to address RO due to their representational limitations regarding the factored global action value function and suffer from sub-optimal policies.

On the seven SMAC maps, DAC outperforms the fully decentralized baselines but trails QMIX and VDN. This gap reflects the advantage of centralized training and joint value factorization, whereas DAC must infer task dynamics from decentralized experience. Its performance is further constrained by partial observability and insufficient coverage of the local task-dynamics space. Incorporating belief-state representations and more effective exploration strategies may help narrow this gap.

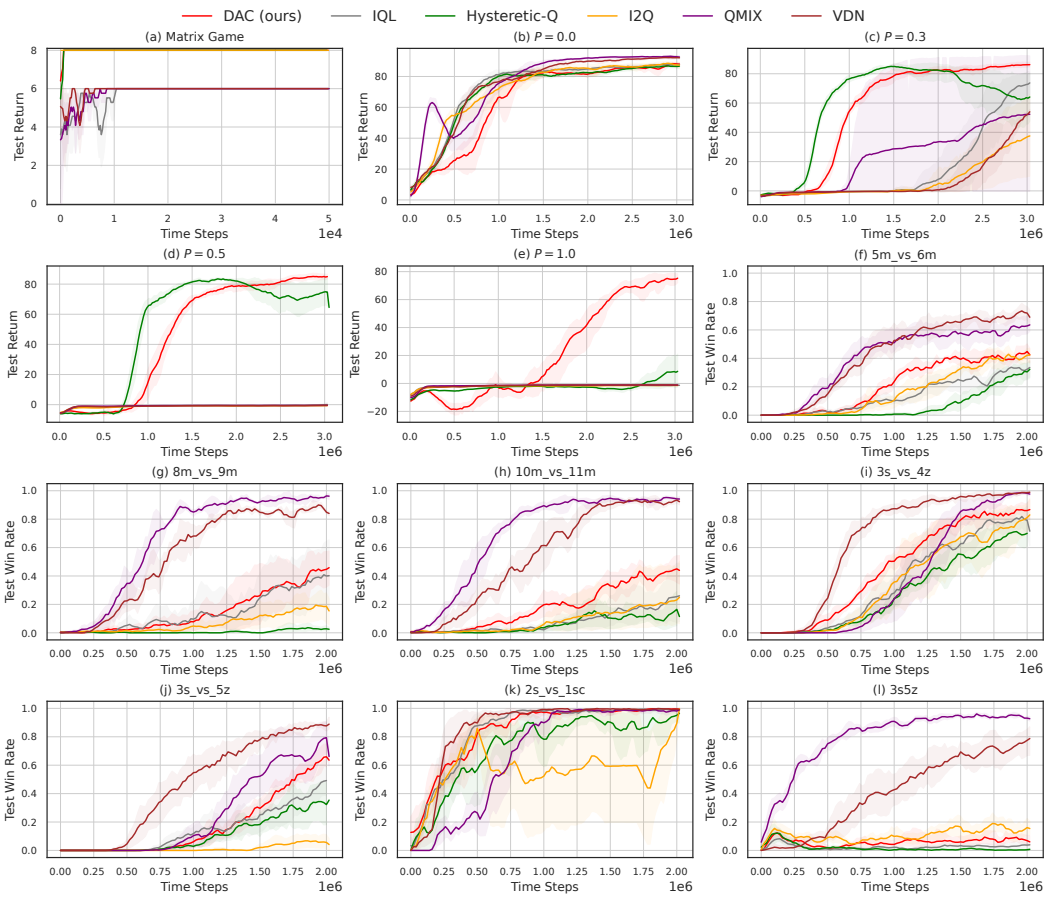


Figure 9: Performance comparison among DAC, QMIX, and VDN.

D.7 Computational Cost

All experiments are conducted on two compute servers: one equipped with an AMD EPYC 7542 32-core CPU, 504GB of RAM, and 8 NVIDIA GeForce RTX 4090D GPUs, and the other equipped with an AMD EPYC 9554 64-core CPU, 128 GB of RAM, and 2 NVIDIA GeForce RTX 5090 GPUs.