Reidentify: Context-Aware Identity Generation for Contextual Multi-Agent Reinforcement Learning

Zhiwei Xu [1] **Kun Hu** [2] **Xin Xin** [1] **Weiliang Meng** [3] **Yiwei Shi** [4] **Hangyu Mao** [5] **Bin Zhang** [3] **Dapeng Li** [3] **Jiangjin Yin** [6]

Abstract

Generalizing multi-agent reinforcement learning (MARL) to accommodate variations in problem configurations remains a critical challenge in real-world applications, where even subtle differences in task setups can cause pre-trained policies to fail. To address this, we propose Context-Aware Identity Generation (CAID), a novel framework to enhance MARL performance under the Contextual MARL (CMARL) setting. CAID dynamically generates unique agent identities through the agent identity decoder built on a causal Transformer architecture. These identities provide contextualized representations that align corresponding agents across similar problem variants, facilitating policy reuse and improving sample efficiency. Furthermore, the action regulator in CAID incorporates these agent identities into the action-value space, enabling seamless adaptation to varying contexts. Extensive experiments on CMARL benchmarks demonstrate that CAID significantly outperforms existing approaches by enhancing both sample efficiency and generalization across diverse context variants.

1. Introduction

In recent years, the field of multi-agent reinforcement learning (MARL) has advanced rapidly, shifting its focus from theoretical exploration to practical applications. For instance, MARL algorithms have been successfully applied to energy scheduling in power systems (Yan & Xu, 2020), coordinated control in drone formations (Ge et al., 2018), and node optimization in communication networks (Mao et al., 2020), yielding promising results. Nevertheless, a significant challenge in reinforcement learning, which is equally prominent in MARL, lies in generalizing across task variants (Kirk et al., 2023). Many existing approaches struggle to maintain comparable performance when faced with even minor task modifications, as the policies trained on original tasks often fail to adapt effectively. Since dynamic problems are common in real-world scenarios, training distinct policies for every variation is infeasible, particularly in multi-agent systems.

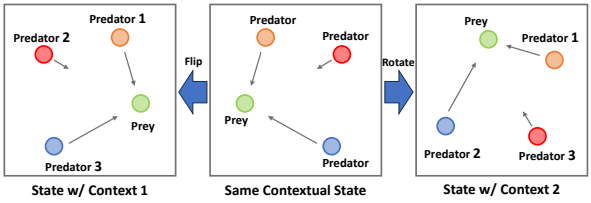


Figure 1. An example of CMARL. The central part depicts a fundamental state from the classic predator-prey task. By introducing different contexts to this state, the resulting environmental states on the left and right sides exhibit substantial differences.

Several single-agent studies have focused on generalization across distinct problem variants within the same domain. One approach integrates meta-learning with reinforcement learning to optimize gradient directions that balance multiple objectives (Finn et al., 2017a; Yu et al., 2019; Nagabandi et al., 2019), thereby improving the robustness of reinforcement learning algorithms in multi-task settings. Another line of research leverages transfer learning to fine-tune pre-trained policies (Taylor & Stone, 2009), enabling adaptation to the distributional shifts between original and target tasks. This approach is particularly prominent in robotics, where sim-to-real methods (Zhao et al., 2020) address the discrepancies between simulated environments and the real world. Additionally, inspired by the success of large models (OpenAI, 2023) in natural language processing (NLP), the intelligent decision-making field has begun scaling up policy networks to improve model performance (Reed et al., 2022; Lee et al., 2022). However,

[1]Shandong University [2]National University of Defense Technology [3]Institute of Automation, Chinese Academy of Sciences [4]University of Bristol [5]Kuaishou Technology [6]Huazhong Agricultural University. Correspondence to: Jiangjin Yin <jiangjinyin@mail.hzau.edu.cn>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

these efforts often face limitations: they either achieve generalization only in simple tasks, narrowly adapt to specific problem variants through fine-tuning, or use model scaling to combine unrelated task strategies. As a result, achieving robust generalization across problem variants remains a significant and unresolved challenge in the field.

Recently, Contextual Reinforcement Learning (CRL) (Benjamins et al., 2023; Hallak et al., 2015) is introduced to formalize the generalization problem across related problem variants. CRL seeks to find a policy capable of solving a group of similar Markov Decision Processes (MDPs), where each MDP presents a variation of the problem, defined by a context variable that reflects the specific characteristics of each variation (Cobbe et al., 2020; Rajan et al., 2023). Similarly, we introduce **Contextual Multi-Agent Reinforcement Learning** (CMARL), which focuses on learning policies for multiple agents across various tasks. These tasks may vary in aspects such as initial states and agent types. For example, while the two initial scenarios may seem distinct, they can be transformed into identical configurations by applying a specific way, as illustrated in Figure 1. Consequently, in CMARL tasks, some related environment states can often be unified into a single contextual state, thereby simplifying the representation of the task. Furthermore, this transformation disrupts the alignment of agent numbering schemes in the two contexts, which underscores the inherent difficulty in ensuring consistent agent identification across different contexts. Current MARL methods cannot automatically assign the appropriate identity to each agent based on the context of the task. As a result, agents must learn separate policies for each situation, significantly reducing sample efficiency in CMARL tasks.

To improve the performance of existing algorithms in the CMARL setting, we propose a novel framework called **C**ontext-**A**ware **ID**entity Generation (**CAID**). The main idea behind CAID is to dynamically assign each agent a unique identity. This is achieved through a context-aware approach integrating global state information with individual agent attributes. By assigning each agent a distinctive identification within the system, the framework effectively reduces the complexity of the contextual decision space. These identities not only serve as unique labels for agents but also provide a foundation for interaction and cooperation among them. Furthermore, the context-aware design ensures that external information about agents is required only during the training phase, allowing for compatibility with most algorithms following the centralized training decentralized execution (CTDE) paradigm. Extensive experiments on benchmarks such as the Vectorized Multi-Agent Simulator (VMAS) (Bettini et al., 2022), Traffic Signal Control (PyTSC) (Bokade & Jin, 2024) and StarCraft Multi-Agent Challenge (SMACv2) (Ellis et al., 2023) demonstrate the effectiveness of CAID.

2. Related Work

The generalization of RL algorithms to task variations has garnered significant attention, particularly in the single-agent RL domain. The MAML framework (Finn et al., 2017b), for instance, optimizes initialization parameters to enable models to achieve strong performance on new tasks with minimal gradient updates. First-order optimization algorithms (Nichol et al., 2018), such as Reptile (Nichol & Schulman, 2018), further reduce the computational complexity of MAML, making it more feasible in real-world scenarios. Many meta RL approaches also focus on training history-dependent policies, often implemented as recurrent neural networks (RNNs) (Chung et al., 2014) like RL^2 (Duan et al., 2016), which dynamically adapt to interaction histories. MetaGenRL (Kirsch et al., 2020) introduces a meta-learned, low-complexity neural objective function based on diverse agent experiences, leveraging off-policy second-order gradients for improved sample efficiency. Beyond meta RL, transfer RL represents another critical avenue for addressing generalization. For example, a component-based transfer learning framework (Sodhani et al., 2021) leveraging abstract representations has been proposed to facilitate the mastery of complex tasks using prior models. REPAINT (Tao et al., 2021) further enhances deep RL efficiency by combining representation learning and instance transfer techniques. Recent research on in-context learning (Laskin et al., 2023) has explored training models across episodes with task-agnostic processes, enabling generalization to diverse tasks without explicit task-specific adjustments. However, many of these approaches still require fine-tuning to adapt to new tasks, and their applicability is sometimes limited to very simple situations.

Some MARL algorithms enable agents to adaptively define roles or form dynamic groups, allowing them to learn optimal strategies under varying task conditions. REFIL (Iqbal et al., 2021), for example, improves agent robustness to variations in initial states by randomly partitioning environmental entities. ROMA (Wang et al., 2020) and RODE (Wang et al., 2021b) dynamically assign roles to each agent, where each role corresponds to a specific sub-action space, thereby effectively reducing the complexity of the action space. COPA (Liu et al., 2021) employs a global "coach" agent to coordinate partially observable "player" agents through limited information exchange. This design enables COPA to demonstrate zero-shot policy generalization even in teams with dynamic compositions. COLA (Xu et al., 2023) applies the principle of viewpoint invariance from computer vision to map the state space into a discrete representation via contrastive learning. E2GN2 (McClellan et al., 2024) is a framework that improves sample efficiency and generalization through equivariant graph neural networks (GNNs) (Wu et al., 2021). By integrating equivariant and invariant features, E2GN2 overcomes scalability challenges common

in traditional methods. However, these approaches are not specifically tailored for Contextual MARL tasks. In contrast, our proposed CAID assigns an identity to each agent, focusing on maintaining consistent identifiers for corresponding agents across different states. This design minimizes the need for retraining when tasks undergo minor changes.

3. Preliminaries

3.1. Decentralized Partially Observable Markov Decision Process

In this paper, we investigate a fully cooperative multi-agent task, which can be modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek & Amato, 2016). A Dec-POMDP is a framework commonly used to formalize cooperative decision-making problems in partially observable and decentralized environments. Formally, it is defined by the tuple $G = \langle S, U, A, P, r, Z, O, n, \gamma \rangle$, where S represents the set of possible states of the environment, and $A = \{1, \ldots, n\}$ denotes the set of n agents involved. At each time step, each agent $a \in A$ selects an action $u^a \in U$ based on its local observation $z^a \in Z$, obtained through the observation function $O(s, a) : S \times A \to Z$. Here, $s \in S$ denotes the true state of the environment. The joint action of all agents is denoted as $\boldsymbol{u} \in \boldsymbol{U}$. The state transition dynamics are determined by the function $P(s_{t+1}|s_t, \boldsymbol{u}_t) : S \times \boldsymbol{U} \times S \to [0, 1]$. All agents share a common reward function $r(s, \boldsymbol{u}) : S \times \boldsymbol{U} \to \mathbb{R}$, and $\gamma \in [0, 1)$ is the discount factor. The primary objective in the Dec-POMDP framework is to maximize the discounted return $\sum_{j=0}^{\infty} \gamma^j r_{t+j}$.

3.2. Value Decomposition Methods

Value decomposition methods (Sunehag et al., 2018; Rashid et al., 2018; Son et al., 2019; Yang et al., 2020) are among the most widely adopted techniques in MARL, particularly for addressing challenges related to coordination. A fundamental concept in value decomposition methods is decomposability, which ensures alignment between global and individual agent objectives. This is formalized through the Individual-Global-Max (IGM) assumption (Son et al., 2019), which stipulates that the optimal action for each agent $\arg\max_{u^a} Q_a^*(\tau^a, u^a)$, must be consistent with the optimal joint action of all agents $\arg\max_{\boldsymbol{u}} Q_{tot}^*(\boldsymbol{\tau}, \boldsymbol{u})$. Mathematically, this condition is expressed as:

$$\arg\max_{\boldsymbol{u}} Q_{tot}^*(\boldsymbol{\tau}, \boldsymbol{u}) = \arg\max_{u^a} Q_a^*(\tau^a, u^a), \quad \forall a \in A, \tag{1}$$

where $\boldsymbol{\tau} \in T^n$ represents the joint action-observation histories of all agents, Q_{tot} is the global action-value function, and Q_a denotes the individual action-value function for agent a. Several value decomposition methods have been developed based on this principle, and our proposed CAID framework can be integrated into these methods to enhance their overall performance.

3.3. Contextual Multi-Agent Reinforcement Learning

Contextual Reinforcement Learning (CRL) (Benjamins et al., 2023) extends traditional reinforcement learning by addressing scenarios where a collection of related tasks needs to be solved, with each task influenced by a specific context. The context provides additional information that may not be directly observable but can significantly affect decision-making. CRL formalizes such problems using Contextual Markov Decision Processes (CMDPs) (Hallak et al., 2015), where the state space is augmented to include the environment state and a context variable that captures the conditions under which the agent operates. Formally, the state is represented as $\bar{s} = (s, c)$, where s is the environment state and c denotes the contextual information relevant to the task.

In CMDPs, the context is assumed to remain fixed throughout an episode, ensuring consistency during the agent's interaction with the environment. However, the context can vary across episodes, leading to a distribution of possible contexts $\rho(c)$. Consequently, the initial state distribution in a CMDP is given by $\rho(\bar{s}) = \rho(c)\rho(s|c)$, where $\rho(s|c)$ represents the distribution of the environment state s conditioned on the context c. For a fixed context c, the CMDP behaves as a traditional Markov Decision Process (MDP). CRL aims to determine an optimal policy π^* that maximizes the expected return across all CMDPs. The expected return for a policy π within a specific CMDP $\mathcal{M}_c$ is denoted as $\mathcal{R}(\pi, \mathcal{M}_c)$. The overall objective is to maximize this return across the distribution of contexts:

$$\pi^* = \max_{\pi} \left[\mathbb{E}_{c \sim \rho(c)} \left[\mathcal{R}\left(\pi, \mathcal{M}_c\right) \right] \right].$$

In multi-agent settings, CRL introduces additional complexity, as each CMDP now represents a multi-agent control problem where multiple agents interact both with the environment and with one another. In this paper, we extend CRL to Contextual Multi-Agent Reinforcement Learning (CMARL) by modeling each CMDP as a Dec-POMDP, a framework well-suited for real-world scenarios. The objective of CMARL is to find an optimal joint policy:

$$\boldsymbol{\pi}^* = \max_{\boldsymbol{\pi}} \left[\mathbb{E}_{c \sim \rho(c)} \left[\mathcal{R}\left(\boldsymbol{\pi}, \mathcal{M}_c\right) \right] \right], \quad \mathcal{M}_c \sim G,$$

where $\mathcal{M}_c$ represents the Dec-POMDP induced by the given context c. Our motivation stems from real-world CMARL tasks where the semantics of an environment can shift significantly even within a single episode. For instance, in a traffic control task, the agent behavior required during morning rush hours may differ substantially from that during the evening, despite both being within one episode. We allow the context vector to evolve over time, effectively treating the episode as a sequence of sub-episodes, each governed by a different latent context.

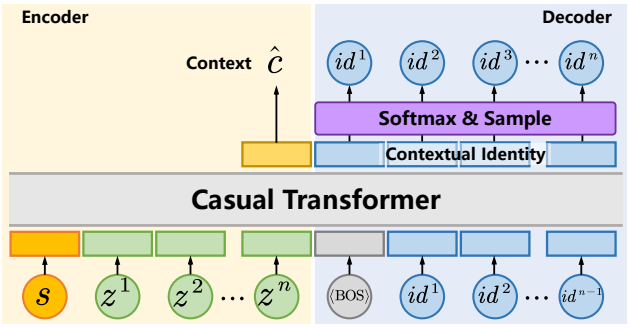


Figure 2. The overall architecture of the contextual state encoder and the agent identity decoder. The input includes the global state and observations from all agents, while the output comprises the inferred context and the identity assigned to each agent.

4. Methodology

This section presents a detailed overview of the CAID framework, which comprises three core modules. The first two modules are the **Contextual State Encoder** and the **Agent Identity Decoder**, both built upon a Transformer network architecture (Vaswani et al., 2017). These modules work collaboratively to infer contextual information from the environment and generate unique identities for each agent. The third module, the **Action Regulator**, transforms the actions selected by each agent based on its assigned identity, ensuring context-aware decision-making. A key feature of CAID is its ability to map the environment state space into a compact, low-dimensional space. This mapping facilitates the identification of similar contextual scenarios, allowing agents to infer analogous contexts and assign identical identities even when operating in significantly different environment states. By leveraging these identities, agents can effectively reuse previously learned policies, improving adaptability and efficiency in complicated multi-agent environments.

4.1. Contextual State Encoder

In practical CMARL tasks, the context c and its distribution $\rho(c)$ are frequently inaccessible, rendering the context-augmented state $\bar{s}$ unavailable during training. As a result, agents must learn their policies based solely on the environment state s, which lacks explicit contextual information. When changes in c cause variations in s, agents must relearn their policies, which dramatically increases training difficulty.

Inspired by the capabilities of large language models (LLMs) (Touvron et al., 2023), we observe that these models often produce consistent responses even when the order of input words changes. Research indicates that this robustness is attributed to the underlying Transformer architecture, particularly its multi-head attention (MHA) mechanism (Vaswani et al., 2017). This mechanism captures the relationships between input tokens while modeling the data comprehensively from multiple perspectives.

Building on this insight, we propose the **Contextual State Encoder**. It produces a context variable $\hat{c}$ by integrating the environment state s with the local observations z of all agents. This context serves as a substitute for the unavailable ground truth c and concurrently assigns unique identities to each agent.

The structure of the contextual state encoder is illustrated in Figure 2. Inspired by the Seq2Seq architecture (Sutskever et al., 2014), which is widely used in NLP, we employ the encoder component as the contextual state encoder. In CMARL scenarios, the environment state s is highly dynamic, making it challenging to capture contextual information or assign consistent identities based merely on s. To address this, we represent the input as a sequence of length $n+1$, comprising the environment state s and the local observations of all agents $\boldsymbol{z} = \{z^1, z^2, \ldots, z^n\}$. Formally, the encoder $f^{Encoder}$ generates the contextual information as follows:

$$\hat{c} = f^{Encoder}(s, \boldsymbol{z}). \tag{2}$$

While the contextual state encoder does not directly reconstruct the ground truth c, it achieves two key objectives: (1) mapping the complex state space into a compact, low-dimensional representation, and (2) extracting relationships between the environment state and agents' local observations. These relationships capture a critical subset of the contextual information, allowing for more efficient policy learning and identity assignment in multi-agent systems.

4.2. Agent Identity Decoder

After generating the contextual information, the next step focuses on assigning unique identities to agents. In most existing MARL frameworks (Lowe et al., 2017; Kurach et al., 2020), agent identities are determined using **heuristic rules**. However, in scenarios with similar contextual settings, such methods often fail to ensure consistent alignment of agent identities across different environments. To overcome this limitation, we propose a context-aware identity assignment module capable of automatically assigning each agent a a unique identity id^a based on the generated contextual information.

As shown in Figure 2, the **Agent Identity Decoder** is designed using a causal Transformer decoder. Inspired by natural language generation tasks and the Pointer Network (Vinyals et al., 2015), the identities are determined in an autoregressive manner. The process begins with a special token ⟨BOS⟩, commonly employed in NLP tasks to indicate the start of decoding. Representing the agent identity decoder as $f^{Decoder}$, the probability of identity assignment

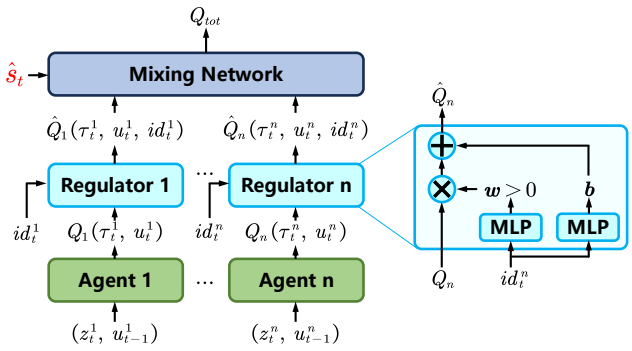


Figure 3. Illustration of value decomposition methods with the action regulators.

for each agent a can be expressed as:

$$\mathcal{P}(a) = \text{Softmax}\left(f^{Decoder}(s, \boldsymbol{z}, id^1, \ldots, id^{a-1})\right). \quad (3)$$

Each identity $id^a \in \{1, 2, \ldots, n\}$ is discrete and must be unique to distinguish agents. Enforcing identity uniqueness is a critical constraint, formally expressed as follows:

$$\forall i \neq j, \quad id^i \neq id^j.$$

To satisfy this constraint, the agent identity decoder employs a dynamic masking mechanism M, which ensures that newly generated identities do not duplicate previously assigned ones. The mask M is defined as:

$$M[id] = \begin{cases} 0 & \text{if identity } id \text{ has already been assigned,} \\ 1 & \text{otherwise.} \end{cases}$$

During each decoding step, the probability distribution is dynamically adjusted based on Equation (3) through the application of the specified mask, resulting in:

$$\mathcal{P}(a) = \text{Softmax}(f^{Decoder}(s, \boldsymbol{z}, id^1, \ldots, id^{a-1}) \cdot M). \quad (4)$$

To prevent premature convergence to suboptimal identity assignments, identities are sampled rather than greedily selected. Then the identity assignment for each agent a can be formulated as:

$$id^a = \text{Draw}\left(\mathcal{P}(a)\right), \quad \forall a \in \mathcal{A}. \quad (5)$$

This stochastic approach reduces the likelihood of a particular agent consistently dominating a specific identity, promoting more balanced identity assignments among agents.

4.3. Action Regulator

After assigning unique identities to agents, the next step is to incorporate this identity information into decision-making processes. In multi-agent systems, agents with the same identity may have identical absolute semantics in their action spaces, but the relative semantics of their actions can vary significantly across different states. It is necessary to transform their action spaces accordingly.

During the centralized training phase, the contextual state encoder and agent identity decoder generate $\hat{c}$ and id^a. However, these identities cannot be used as inputs to the agent networks during the decentralized execution phase. We draw inspiration from hypernetworks (Ha et al., 2017) to overcome this limitation and propose an **Action Regulator** to align agents' action spaces based on their identities. The action regulator operates on the Q-values output by each agent and consists of two linear layers. Both layers take the agent's identity id^a as input, generating the weights $\boldsymbol{w}$ and bias $\boldsymbol{b}$ for a hypernetwork. The transformed Q-value output of the agent network is then expressed as:

$$\hat{Q}_a(\tau_t^a, u_t^a, id_t^a) = \boldsymbol{w}(id_t^a) Q_a(\tau_t^a, u_t^a) + \boldsymbol{b}(id_t^a). \quad (6)$$

By introducing an additional hypernetwork layer to perform an affine transformation on the raw Q-values, the action regulator effectively aligns the action spaces of agents, ensuring that their decisions are consistent with their assigned identities.

To enable seamless integration of the action regulator into existing value decomposition methods for MARL, it must satisfy the Individual-Global-Max (IGM) paradigm, as described in Equation (1). Precisely, the global optimal action should align with the individual optimal actions of the agents. We adopt a simple yet effective approach to meet this requirement: ensuring that $\boldsymbol{w}$ remains positive by taking its absolute value. This guarantees the transformation preserves the consistency between global and individual optimal actions.

In summary, the action regulator serves as a critical component of the CAID framework, enabling identity-based decision-making and ensuring compatibility with existing multi-agent reinforcement learning methods while maintaining theoretical guarantees of the IGM condition.

4.4. End-to-End Training

The three main components of CAID, along with their inputs and outputs, have been detailed earlier. The causal Transformer architecture includes a contextual state encoder and an agent identity decoder. The encoder generates a contextual state $\hat{c}$ based on the environment state s and the local observations of all agents z. This contextual state $\hat{s} = (s, \hat{c})$ replaces the ground truth $\bar{s}$ and serves as input to the mixing network in value decomposition methods, providing a more comprehensive representation of CMARL tasks compared to the original environment state s. The agent identity decoder sequentially predicts each agent's identity in an autoregressive manner. These identities are

subsequently used as inputs to the action regulator, aligning agents' action spaces. The entire CAID framework is trained end-to-end, where the three modules are optimized jointly with the reinforcement learning module. The loss function for this process is as follows:

$$\mathcal{L} = \left(y_{tot} - Q_{tot}(\boldsymbol{\tau}, \boldsymbol{u}, \hat{s})\right)^2, \tag{7}$$

where y_{tot} is the target joint value function, computed as $y_{tot} = r + \gamma \max_{\boldsymbol{u}'} Q_{tot}(\boldsymbol{\tau}', \boldsymbol{u}', \hat{s}')$. All parameters within CAID are optimized by minimizing the temporal-difference (TD) error. A significant challenge arises because each agent's identity is sampled, preventing direct gradient backpropagation to the agent identity decoder. To address this, we utilize **Straight-Through Gradients** (Bengio et al., 2013), a technique easily implemented using automatic differentiation. The agent identity in Equation (5) is modified as follows:

$$id^a = \mathrm{Draw}\left(\mathcal{P}(a)\right) + \mathcal{P}(a) - \mathrm{StopGrad}\left(\mathcal{P}(a)\right), \forall a \in \mathcal{A}, \tag{8}$$

where the $\mathrm{StopGrad}$ function prevents gradient flow through its input during backpropagation. This approach ensures that agent identities are sampled to avoid premature convergence to suboptimal solutions while still allowing gradients to propagate back to optimize the agent identity decoder.

The overall CAID framework is illustrated in Figure 3. By integrating the contextual state $\hat{s}$ and agent identities into value decomposition methods, CAID provides a more expressive representation of CMARL tasks. This design enables CAID to be incorporated into existing value decomposition methods, improving its capability to handle complex multi-agent scenarios.

5. Experiments

In this section, we evaluate CAID in three well-known CMARL environments: StarCraft Multi-Agent Challenge (SMACv2) (Ellis et al., 2023), Vectorized Multi-Agent Simulator (VMAS) (Bettini et al., 2022) and Traffic Signal Control (PyTSC) (Bokade & Jin, 2024). The tasks in these domains exhibit considerable variability across episodes, primarily in the agents' positions, agent types, and target locations. First, the performance of CAID is evaluated through a comparison with several classical algorithms, including Weighted QMIX (Rashid et al., 2020), QPLEX (Wang et al., 2021a), and the baseline QMIX (Rashid et al., 2018), along with recently proposed methods such as RIIT (Hu et al., 2021), COLA (Xu et al., 2023), and VMIX (Su et al., 2021). Then the contributions of individual modules within the framework are discussed. To ensure the reliability of the results, each experiment is repeated five times with different random seeds. For a fair comparison, all hyperparameters, except those introduced specifically by CAID, are kept consistent with the original methods. *Unless explicitly stated, CAID refers to the variant implemented on QMIX.* Details on algorithm hyperparameters are provided in Appendix A.

5.1. StarCraft Multi-Agent Challenge

Based on the renowned real-time strategy game StarCraft II, SMAC (Samvelyan et al., 2019) is among the most widely used platforms for multi-agent micro-management experiments. It contains a variety of mini-scenarios, including homogeneous, heterogeneous, symmetric, and asymmetric problem types. Each task requires controlling a team of agents to defeat an AI team controlled by heuristic strategies. Agents can only access information within their local observation range and share a global reward function. However, recent studies (Ellis et al., 2023) have shown that agents in SMAC can achieve near-optimal policies by relying solely on the current timestep and fixed agent IDs, completely disregarding their local observations. This finding highlights a critical limitation of SMAC: a lack of stochasticity, which is essential in real-world problems. To address this limitation, SMACv2 was introduced, where agents' positions and types can change dynamically in each episode, posing new challenges for existing MARL algorithms. We conducted extensive experiments in this classic CMARL environment to evaluate various algorithms, and scenario details are provided in Appendix B.1.

The experimental results in SMACv2 are presented in Figure 4. Across 12 tested scenarios, the inherent stochasticity of the environment occasionally led to scenarios where achieving victory was entirely infeasible, preventing convergence to near-perfect win rates. Our primary observation is that CAID outperforms the baseline QMIX algorithm across all tasks. Furthermore, CAID outperforms other algorithms in most scenarios, particularly in the *Terrain* and *Protoss* series scenarios. In the Zerg series scenarios, the results derived from integrating CAID into QMIX were constrained by QMIX's suboptimal credit assignment capabilities. Nonetheless, CAID consistently demonstrates the fastest initial convergence rate among all tested algorithms.

5.2. Vectorized Multi-Agent Simulator

In addition to the StarCraft II game environment, we seek additional effective CMARL environments. The Vectorized Multi-Agent Simulator (VMAS) is a vectorized and fully differentiable simulator designed for efficient MARL benchmarking. It features a high-performance, vectorized 2D physics engine and a diverse suite of challenging multi-robot scenarios. The modular and user-friendly design of the simulator facilitates the creation of custom scenarios, encouraging contributions from the research community. Some scenarios in VMAS represent tasks such as intelligent

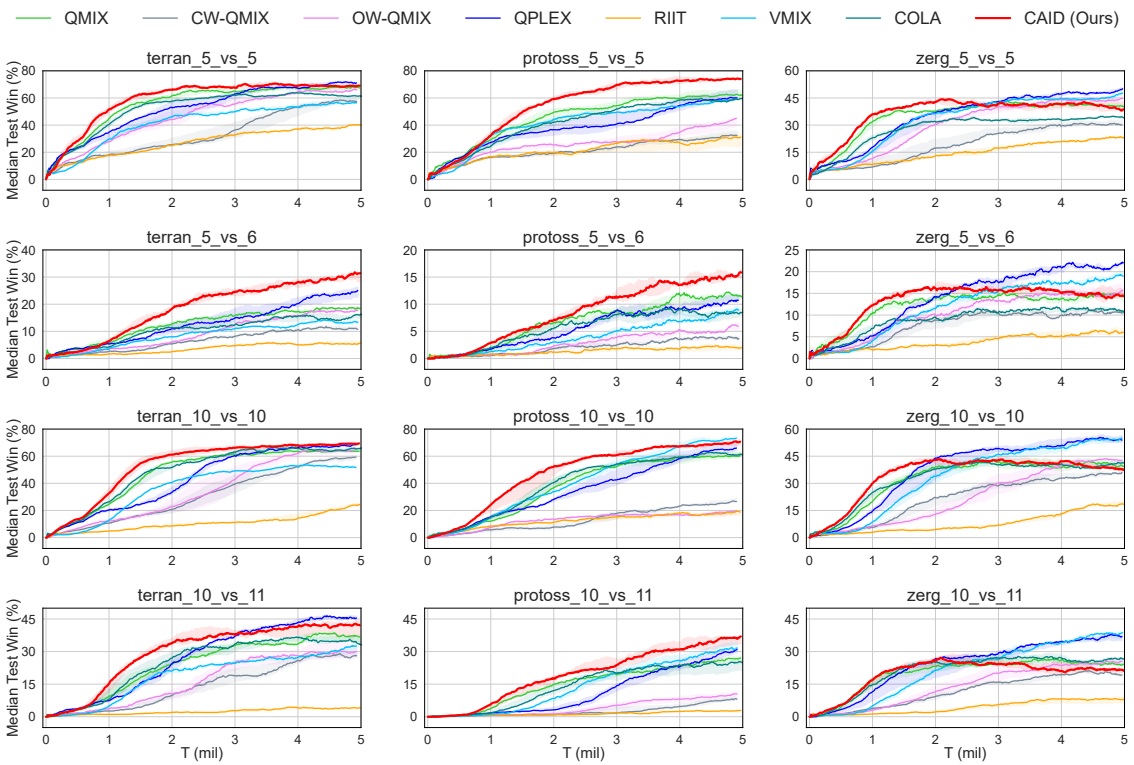


Figure 4. Performance comparison with baselines in different SMACv2 scenarios.

formations and obstacle avoidance for drones or uncrewed vehicles, which are of significant practical relevance. Most notably, nearly all tasks can be configured to follow the CMARL setting, where both the initial and goal positions of agents are randomly assigned in each episode. Detailed descriptions of the scenarios employed in VMAS can be found in Appendix B.2.

We used VMAS to compare the performance of three classic value decomposition MARL algorithms, VDN (Sunehag et al., 2018), QMIX, and QPLEX, along with their CAID variants. The learning curves for the CAID variants and their original counterparts are illustrated in Figure 5. It is evident that, in most cases, the CAID variants outperform the original baseline algorithms. This comparison among the three distinct value decomposition methods and their respective variants demonstrates that CAID is applicable to existing value decomposition approaches and can improve their performance. Furthermore, as a distinct value decomposition method, VDN lacks a mixing network module, implying that the contextual state $\hat{s}$ inferred within CAID-VDN is not used. Consequently, the performance improvements observed in CAID-VDN are entirely attributable to the reidentification of agents. This finding underscores the contribution of the agent identity decoder. In the following sections, we systematically evaluate the role and importance of each component in the CAID framework.

5.3. Ablation Study

An ablation study was carried out on CAID to evaluate its performance in Traffic Signal Control (TSC) scenarios. PyTSC (Bokade & Jin, 2024) is a development environment tailored for research in traffic signal control, designed to facilitate the rapid development of reinforcement learning-based solutions. In PyTSC, the tasks emulate urban traffic management by controlling traffic lights to optimize vehicle movement through intersections. The project currently integrates with SUMO (López et al., 2018) and CityFlow (Zhang et al., 2019) as simulation backends, offering researchers practical tools to utilize open-source traffic signal control datasets. Notably, it supports the CMARL setting, where the contexts (e.g., traffic flow) in each episode are dynamically randomized. Details regarding the environment are provided in Appendix B.3.

Three variants of CAID were designed, each modifying a single component of the original algorithm. To investigate the importance of the contextual state encoder, we proposed CAID w/o CS, where the contextual state $\hat{s}$ is excluded from the mixing network in the base algorithm. For the agent identity decoder, we introduced CAID w/o AI, in which the agent network does not utilize the newly assigned identity information. Lastly, in CAID w/o ST, new identities for agents are generated using a greedy algorithm instead of

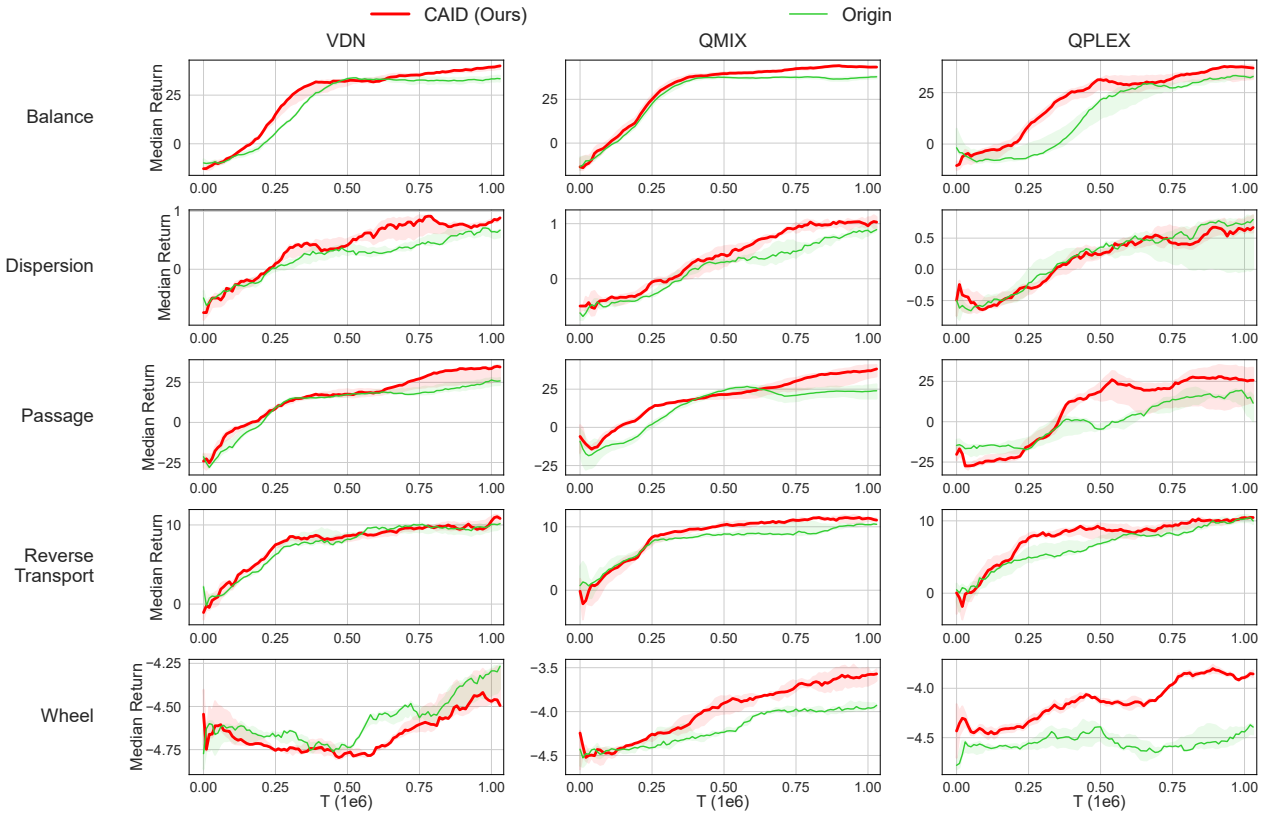


Figure 5. Comparison of our approach against baseline algorithms on Vectorized Multi-Agent Simulator.

Table 1. Mean of metrics for various algorithms across scenarios in PyTSC.

Algos	Jinan			Hangzhou			New York		
	Queue	Delay	Travel Time	Queue	Delay	Travel Time	Queue	Delay	Travel Time
CAID	**463.39**	**0.471**	329.28	**146.50**	**0.651**	336.52	**479.48**	**0.901**	369.05
w/o CS	476.52	0.473	**326.86**	155.79	0.653	340.00	488.39	0.908	372.13
w/o AI	663.59	0.498	364.05	171.37	0.659	346.77	480.41	0.902	**368.38**
w/o ST	490.70	0.482	332.34	150.11	0.662	**330.65**	487.28	0.903	371.38
QMIX	563.13	0.507	338.15	170.58	0.660	349.25	496.08	0.902	381.57

sampling, potentially leading to suboptimal solutions. The ablation experiment results are illustrated in Table 1. The best performances of the above algorithms are bold.

We selected three key performance metrics that provide a more direct assessment of performance: queue length, delay, and travel time. The results demonstrate that all three ablated variants degrade the performance of CAID. Among them, CAID w/o CS shows the least performance drop, followed by CAID w/o ST. These findings highlight the importance of the identities produced by the agent identity decoder in enabling agents to adapt effectively to dynamic environments. In summary, the designs of the contextual state encoder and agent identity decoder are critical to the effectiveness of CAID.

5.4. Visualization

We use a dataset consisting of 20,000 timesteps. The left side of the figure shows the 2D t-SNE embedding of the contextual states produced by the Contextual State Encoder, while the right side presents the 2D t-SNE embedding of the raw states. Each point corresponds to a state, and the color of each point encodes a specific permutation of agent identities in that state. Specifically, we enumerate all permutations of the agent indices $\{0, 1, 2, 3, 4\}$, resulting in $5! = 120$ possible arrangements. Each permutation is mapped to a unique integer in $[0, 119]$ using a bijective function based on Cantor expansion (Lehmer code), enabling us to assign a distinct color to each identity configuration. For comparison, traditional methods typically assign agent identities using heuristic rules, leading to only a single

fixed permutation—represented by a single color in the figure. Notably, the middle of the figure highlights a pair of mirror-symmetric and semantically similar states. In the raw state embedding (right), these states appear distant from each other, whereas in the contextual state embedding (left), they are close together and share similar identity patterns across agents.

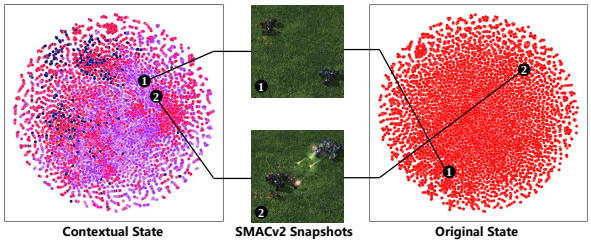


Figure 6. The analysis and visualization of the agent identities in the *terrain_5_vs_5* scenario.

6. Conclusion

In this paper, we introduce the Context-Aware Identity Generation (CAID) framework, which leverages the global state and local observations from all agents to construct contextual states and dynamically assign agent identities. This design enables agents to adapt effectively to different task variations. The CAID framework can be fully trained in an end-to-end manner using the reinforcement learning paradigm. It is compatible with existing MARL value decomposition algorithms, significantly improving sample efficiency. The extensive experiments in diverse contextual MARL scenarios demonstrate compelling performance, providing strong evidence for the practicality of the proposed framework. We believe this work represents a meaningful step toward bridging the gap between multi-agent reinforcement learning and real-world applications. Promising directions for future research include enhancing the robustness of CAID in contextual MARL tasks with dynamic agent populations and extending its applicability to zero-shot generalization scenarios. Furthermore, identifying a more informative representation of contextual MARL, similar to the agent identities generated by CAID, would be an intriguing avenue for exploration.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 62302189 and in part by the Fundamental Research Funds for the Central Universities under Grant 2662022XXQD002.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bengio, Y., Léonard, N., and Courville, A. C. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

Benjamins, C., Eimer, T., Schubert, F., Mohan, A., Döhler, S., Biedenkapp, A., Rosenhahn, B., Hutter, F., and Lindauer, M. Contextualize me - the case for context in reinforcement learning. *Trans. Mach. Learn. Res.*, 2023, 2023.

Bettini, M., Kortvelesy, R., Blumenkamp, J., and Prorok, A. VMAS: A vectorized multi-agent simulator for collective robot learning. In Bourgeois, J., Paik, J., Piranda, B., Werfel, J., Hauert, S., Pierson, A., Hamann, H., Lam, T. L., Matsuno, F., Mehr, N., and Makhoul, A. (eds.), *Distributed Autonomous Robotic Systems - 16th International Symposium, DARS 2022, Montbéliard, France, 28-30 November 2022*, volume 28 of *Springer Proceedings in Advanced Robotics*, pp. 42–56. Springer, 2022. doi: 10.1007/978-3-031-51497-5_4.

Bokade, R. and Jin, X. Pytsc: A unified platform for multi-agent reinforcement learning in traffic signal control. *CoRR*, abs/2410.18202, 2024. doi: 10.48550/ARXIV.2410.18202.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

Cobbe, K., Hesse, C., Hilton, J., and Schulman, J. Leveraging procedural generation to benchmark reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2048–2056. PMLR, 2020.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl$ˆ2$: Fast reinforcement learning via slow reinforcement learning. *CoRR*, abs/1611.02779, 2016.

Ellis, B., Cook, J., Moalla, S., Samvelyan, M., Sun, M., Mahajan, A., Foerster, J. N., and Whiteson, S. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017a.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 2017b.

Ge, X., Han, Q., and Zhang, X. Achieving cluster formation of multi-agent systems under aperiodic sampling and communication delays. *IEEE Trans. Ind. Electron.*, 65 (4):3417–3426, 2018. doi: 10.1109/TIE.2017.2752148.

Ha, D., Dai, A. M., and Le, Q. V. Hypernetworks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Hallak, A., Castro, D. D., and Mannor, S. Contextual markov decision processes. *CoRR*, abs/1502.02259, 2015.

Hu, J., Jiang, S., Harding, S. A., Wu, H., and Liao, S.-w. Rethinking the implementation tricks and monotonicity constraint in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2102.03479*, 2021.

Iqbal, S., de Witt, C. A. S., Peng, B., Boehmer, W., Whiteson, S., and Sha, F. Randomized entity-wise factorization for multi-agent reinforcement learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4596–4606. PMLR, 2021.

Kirk, R., Zhang, A., Grefenstette, E., and Rocktäschel, T. A survey of zero-shot generalisation in deep reinforcement learning. *J. Artif. Intell. Res.*, 76:201–264, 2023. doi: 10.1613/JAIR.1.14174.

Kirsch, L., van Steenkiste, S., and Schmidhuber, J. Improving generalization in meta reinforcement learning using learned objectives. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Kurach, K., Raichuk, A., Stanczyk, P., Zajac, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., and Gelly, S. Google research football: A novel reinforcement learning environment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 4501–4510. AAAI Press, 2020. doi: 10.1609/AAAI.V34I04.5878.

Laskin, M., Wang, L., Oh, J., Parisotto, E., Spencer, S., Steigerwald, R., Strouse, D., Hansen, S. S., Filos, A., Brooks, E. A., Gazeau, M., Sahni, H., Singh, S., and Mnih, V. In-context reinforcement learning with algorithm distillation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Lee, K., Nachum, O., Yang, M., Lee, L., Freeman, D., Guadarrama, S., Fischer, I., Xu, W., Jang, E., Michalewski, H., and Mordatch, I. Multi-game decision transformers. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Liu, B., Liu, Q., Stone, P., Garg, A., Zhu, Y., and Anandkumar, A. Coach-player multi-agent reinforcement learning for dynamic team composition. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6860–6870. PMLR, 2021.

López, P. Á., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and WieBner, E. Microscopic traffic simulation using SUMO. In Zhang, W., Bayen, A. M., Medina, J. J. S., and Barth, M. J. (eds.), *21st International Conference on Intelligent Transportation Systems, ITSC 2018, Maui, HI, USA, November 4-7, 2018*, pp. 2575–2582. IEEE, 2018. doi: 10.1109/ITSC.2018.8569938.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6379–6390, 2017.

Mao, H., Liu, W., Hao, J., Luo, J., Li, D., Zhang, Z., Wang, J., and Xiao, Z. Neighborhood cognition con-

sistent multi-agent reinforcement learning. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7219–7226. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6212.

McClellan, J., Haghani, N., Winder, J., Huang, F., and Tokekar, P. Boosting sample efficiency and generalization in multi-agent reinforcement learning via equivariance. *CoRR*, abs/2410.02581, 2024. doi: 10.48550/ARXIV.2410.02581.

Nagabandi, A., Clavera, I., Liu, S., Fearing, R. S., Abbeel, P., Levine, S., and Finn, C. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Nichol, A. and Schulman, J. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3):4, 2018.

Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018.

Oliehoek, F. and Amato, C. A concise introduction to decentralized pomdps. In *SpringerBriefs in Intelligent Systems*, 2016.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774.

Rajan, R., Diaz, J. L. B., Guttikonda, S., Ferreira, F., Biedenkapp, A., von Hartz, J. O., and Hutter, F. MDP playground: An analysis and debug testbed for reinforcement learning. *J. Artif. Intell. Res.*, 77:821–890, 2023. doi: 10.1613/JAIR.1.14314.

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J. N., and Whiteson, S. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4292–4301. PMLR, 2018.

Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. Weighted QMIX: expanding monotonic value function factorisation for deep multi-agent reinforcement learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Reed, S. E., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. A generalist agent. *Trans. Mach. Learn. Res.*, 2022, 2022.

Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C., Torr, P. H. S., Foerster, J. N., and Whiteson, S. The starcraft multi-agent challenge. In Elkind, E., Veloso, M., Agmon, N., and Taylor, M. E. (eds.), *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pp. 2186–2188. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

Sodhani, S., Zhang, A., and Pineau, J. Multi-task reinforcement learning with context-based representations. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9767–9779. PMLR, 2021.

Son, K., Kim, D., Kang, W. J., Hostallero, D., and Yi, Y. QTRAN: learning to factorize with transformation for cooperative multi-agent reinforcement learning. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5887–5896. PMLR, 2019.

Su, J., Adams, S. C., and Beling, P. A. Value-decomposition multi-agent actor-critics. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 11352–11360. AAAI Press, 2021. doi: 10.1609/AAAI.V35I13.17353.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V. F., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., and Graepel, T. Value-decomposition networks for cooperative multi-agent learning based on team reward. In André, E., Koenig, S., Dastani, M., and Sukthankar, G. (eds.), *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pp. 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 3104–3112, 2014.

Tao, Y., Genc, S., Chung, J., Sun, T., and Mallya, S. REPAINT: knowledge transfer in deep reinforcement learning. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10141–10152. PMLR, 2021.

Taylor, M. E. and Stone, P. Transfer learning for reinforcement learning domains: A survey. *J. Mach. Learn. Res.*, 10:1633–1685, 2009. doi: 10.5555/1577069.1755839.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. doi: 10.48550/ARXIV.2302.13971.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017.

Vinyals, O., Fortunato, M., and Jaitly, N. Pointer networks. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 2692–2700, 2015.

Wang, J., Ren, Z., Liu, T., Yu, Y., and Zhang, C. QPLEX: duplex dueling multi-agent q-learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a.

Wang, T., Dong, H., Lesser, V. R., and Zhang, C. ROMA: multi-agent reinforcement learning with emergent roles. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9876–9886. PMLR, 2020.

Wang, T., Gupta, T., Mahajan, A., Peng, B., Whiteson, S., and Zhang, C. RODE: learning roles to decompose multi-agent tasks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24, 2021. doi: 10.1109/TNNLS.2020.2978386.

Xu, Z., Zhang, B., Li, D., Zhang, Z., Zhou, G., Chen, H., and Fan, G. Consensus learning for cooperative multi-agent reinforcement learning. In Williams, B., Chen, Y., and Neville, J. (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 11726–11734. AAAI Press, 2023. doi: 10.1609/AAAI.V37I10.26385.

Yan, Z. and Xu, Y. A multi-agent deep reinforcement learning method for cooperative load frequency control of a multi-area power system. *IEEE Transactions on Power Systems*, 35(6):4599–4608, 2020.

Yang, Y., Hao, J., Liao, B., Shao, K., Chen, G., Liu, W., and Tang, H. Qatten: A general framework for cooperative multiagent reinforcement learning. *CoRR*, abs/2002.03939, 2020.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Kaelbling, L. P., Kragic, D., and Sugiura, K. (eds.), *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2019.

Zhang, H., Feng, S., Liu, C., Ding, Y., Zhu, Y., Zhou, Z., Zhang, W., Yu, Y., Jin, H., and Li, Z. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In Liu, L., White, R. W., Mantrach, A., Silvestri, F., McAuley, J. J., Baeza-Yates, R., and Zia, L. (eds.), *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pp. 3620–3624. ACM, 2019. doi: 10.1145/3308558.3314139.

Zhao, W., Queralta, J. P., and Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020, Canberra, Australia, December 1-4, 2020*, pp. 737–744. IEEE, 2020. doi: 10.1109/SSCI47803.2020.9308468.

A. Implementation Details

A.1. Algorithmic Description

The pseudo-code of CAID is shown in Algorithm 1.

Algorithm 1 Context-Aware Identity Generation (CAID)

for each episode **do**
  Get the global state s_1 and the local observations $\boldsymbol{z}_1 = \{z_1^1, z_1^2, \ldots, z_1^n\}$ of all agents
  for $t \leftarrow 1$ to $T-1$ **do**
    for $a \leftarrow 1$ to n **do**
      Select action u_t^a according to the agent network
    end for
    Carry out the joint action $\boldsymbol{u}_t = \{u_t^1, \ldots, u_t^n\}$
    Get the global reward r_{t+1}, the next local observations $\boldsymbol{z}_{t+1}$, and the next state s_{t+1}
  end for
  Store the trajectory in the replay buffer $\mathcal{D}$.
  Sample a batch of episodes $\mathcal{B} \sim \text{Uniform}(\mathcal{D})$.
  Compute the context variable $\hat{c}$ using Equation (2).
  Generate agent identities based on Equation (8).
  Evaluate the transformed Q-values for all agents using Equation (6).
  Update the parameters of the CAID model using Equation (7).
  Periodically update the parameters of the target network.
end for

A.2. Hyperparameters

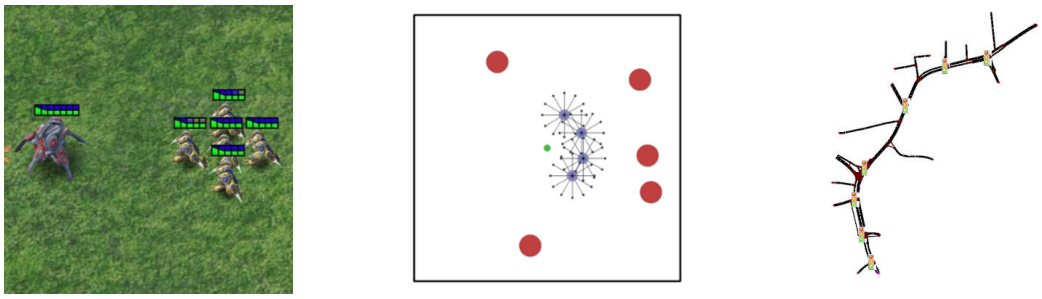
Unless specified otherwise, the hyperparameter configurations across different environments are presented in Table 2. These settings are identical to those provided in PyMARL2 (Hu et al., 2021). All experiments in this study were conducted using NVIDIA GeForce RTX 2080 Ti graphics cards and Intel(R) Xeon(R) Silver 4114 CPUs. For all methods, exploration during training is achieved via independent ϵ-greedy action selection, with ϵ linearly annealed from 1.0 to 0.05 over 50,000 steps. In SMACv2, training ends after 5 million timesteps, whereas it concludes after 1 million timesteps in VMAS and 2 million timesteps in PyTSC.

Table 2. Hyperparameter settings.

Description	**Value**
Learning rate	0.001
Type of optimizer	Adam
How many episodes to update target networks	200
Reduce global norm of gradients	10
Batch size	128
Capacity of replay buffer	5000
Batch size for parallel execution	8
Discount factor	0.99

B. Introduction for Environments

In this paper, we employed three experimental environments: SMACv2, VMAS, and PyTSC. Given the limited research on Contextual MARL and the recent introduction of some new environments, it is essential to provide a detailed overview of these environments tailored to the CMARL setting. The following sections aim to familiarize readers with the objectives of these tasks and the corresponding evaluation metrics, offering a foundation for understanding this emerging field.



(a) SMACv2 (b) VMAS (c) PyTSC

Figure 7. Screenshots of the two experimental platforms used in this paper.

B.1. SMACv2

SMACv2 (StarCraft Multi-Agent Challenge version 2) is an enhanced benchmark environment designed to assess the effectiveness of cooperative MARL algorithms. Compared to its predecessor, SMACv2 incorporates procedurally generated scenarios, requiring agents to adapt their collaborative strategies to dynamically changing environments, including diverse enemy configurations and map layouts. This design intensifies the challenges posed by partial observability, compelling agents to base their decisions on real-time sensory input rather than relying on scripted behaviors. In each episode, the positions and types of both ally and enemy agents are subject to change.

The primary focus of SMACv2 is on evaluating the agents' generalization capabilities in previously unseen scenarios. This is achieved through metrics such as win rates, collaboration efficiency, and adaptability in novel environments. By introducing more demanding scenarios and imposing stricter evaluation standards, SMACv2 establishes a rigorous platform for advancing research on multi-agent reinforcement learning algorithms. It fosters algorithmic development in complex, dynamic contexts and ensures robustness and reliability in real-world applications. Descriptions of the scenarios are provided in Table 3.

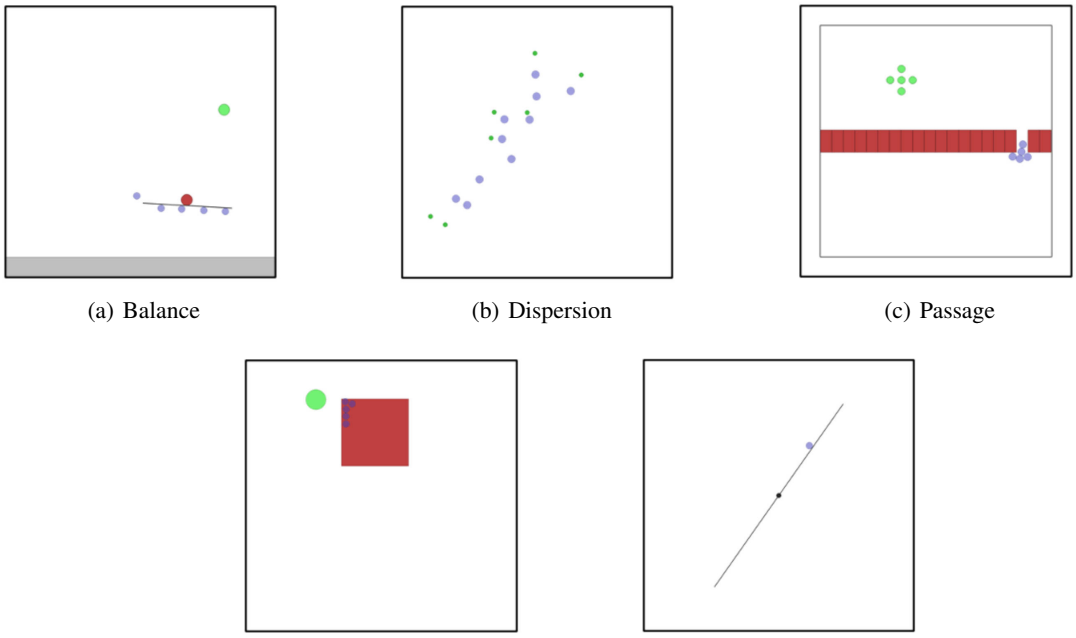
Table 3. Maps in different scenarios.

Race	Unit Types	Probability of Generation	Scenarios
Terran	Marine Marauder Medivac	0.45 0.45 0.1	terran_5_vs_5 terran_5_vs_6 terran_10_vs_10 terran_10_vs_11
Protoss	Stalker Zealot Colossus	0.45 0.45 0.1	protoss_5_vs_5 protoss_5_vs_6 protoss_10_vs_10 protoss_10_vs_11
Zerg	Zergling Baneling Hydralisk	0.45 0.1 0.45	zerg_5_vs_5 zerg_5_vs_6 zerg_10_vs_10 zerg_10_vs_11

B.2. VMAS

VMAS (Vectorized Multi-Agent Simulator) is an efficient and scalable simulation framework for MARL, featuring a differentiable and vectorized 2D physics engine. VMAS includes a comprehensive suite of complex scenarios that encompass both cooperative and competitive tasks. It supports highly customizable features, including sensors, elastic collisions, rotations, joints, and communication mechanisms. Each scenario integrates tailored reward mechanisms and termination conditions to rigorously evaluate the agents' collaborative strategies and division of labor under diverse task objectives.

VMAS offers a range of pre-built multi-agent reinforcement learning scenarios, addressing both cooperative and adversarial



(a) Balance (b) Dispersion (c) Passage

(d) Reverse Transport (e) Wheel

Figure 8. Illustration of benchmark tasks in VMAS.

tasks. Descriptions of representative scenarios are as follows:

Balance. Agents collaborate to balance a spherical object on a linear platform while transporting it to a designated target. Any fall of the object or platform results in a substantial penalty. Successful completion requires coordinated task allocation to maintain equilibrium while steadily advancing toward the goal.

Dispersion. Starting from a common origin, agents must efficiently disperse to distinct target points. This scenario highlights the importance of decentralized coordination, minimizing clustering, and optimizing task completion.

Passage. A group of robots must traverse a wall with a narrow passage to reach their assigned destinations. The challenge involves avoiding collisions and orchestrating the use of the passage in a sequential and cooperative manner, testing path-planning capabilities and localized teamwork.

Reverse Transport. Agents cooperate to transport a high-mass package to a target location. Given the package's significant weight, individual agents cannot accomplish the task alone. Success demands precise synchronization of force direction and magnitude among all agents.

Wheel. Agents encircle a rotating beam anchored at the origin, coordinating their efforts to achieve a target angular velocity. The task requires an optimal distribution of forces to precisely control the beam's speed without overexertion, which could cause deviations from the target.

These scenarios are designed to rigorously evaluate agents' collaborative efficiency, adaptability, and task performance across diverse objectives and constraints. Notably, all scenarios can be configured to the CMARL settings.

B.3. PyTSC

PyTSC is a simulation environment tailored to multi-agent reinforcement learning (MARL) in the domain of traffic signal control (TSC). Its primary goal is to reduce global congestion across the traffic network, quantified by metrics such as total delay, average waiting time, or vehicle queue length, through the coordinated control of multiple traffic signal agents. In a fully cooperative MARL framework, agents adapt to dynamic traffic patterns and develop efficient control strategies by interacting with both the environment and other agents, ultimately achieving global optimization.

The evaluation of PyTSC focuses on two key dimensions: traffic efficiency and algorithmic performance. Traffic efficiency is

assessed using metrics such as average vehicle waiting time, average travel time, and total network delay, while algorithmic performance is measured by factors like convergence speed and policy stability. Additionally, PyTSC allows for testing the adaptability of algorithms under dynamic traffic flow scenarios, such as morning and evening peak periods, providing a comprehensive assessment of MARL approaches in TSC applications.

Table 4. Scenarios in PyTSC environments.

Scenarios	Simulators	Network Types	Agent Types	Total Agents
Jinan	CityFlow	Real-world	Homogeneous	12
Hangzhou	CityFlow	Real-world	Homogeneous	16
New York	CityFlow	Real-world	Homogeneous	16

C. Additional Results

C.1. Results of Dynamic Role Assignment Methods

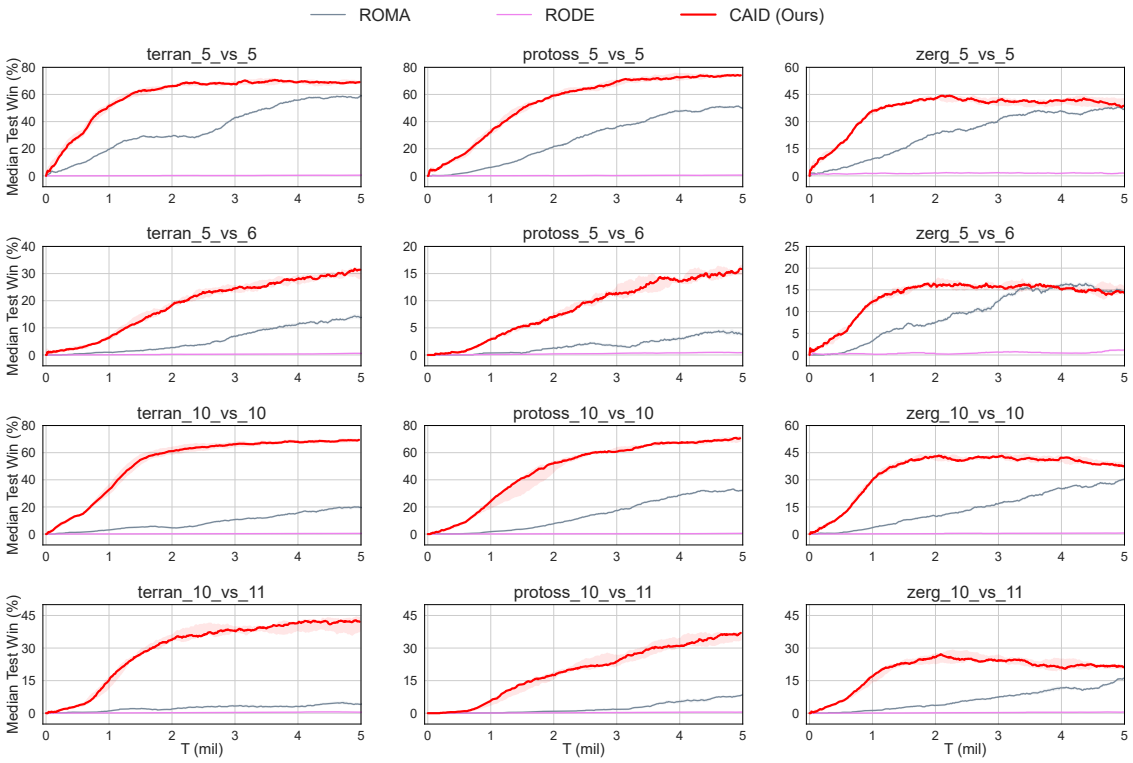


Figure 9. Performance comparison with dynamic role assignment methods in different SMACv2 scenarios. We selected ROMA and RODE as two representative methods for role assignment. All methods were configured with identical hyperparameters, including the optimizer and techniques such as eligibility traces, to ensure consistency with other algorithms implemented in PyMARL2. It is worth noting that RODE fails to perform properly in the SMACv2 environment. This limitation arises because the original RODE implementation modifies the SMAC environment by sorting enemy units in the environment file. Such changes simplify the original SMAC environment, which undermines its complexity and makes the implementation incompatible with the more challenging and standardized SMACv2. As a result, RODE is not applicable to the Contextual MARL setting. CAID achieves superior performance and faster convergence.

C.2. Additional Ablation Study

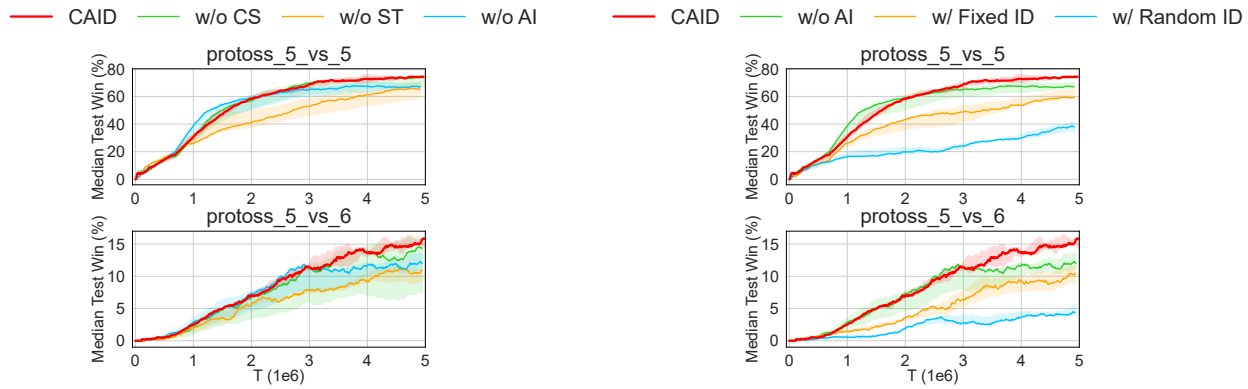


Figure 10. **Left:** The learning curves of the three variants discussed in Section 5.3 on the SMACv2 benchmark. CAID w/o AI refers to the variant where only the Action Regulator component of CAID is removed, while all other components remain unchanged. The results demonstrate that all three variants underperform compared to the full CAID model, highlighting the essential role each module plays in the overall performance of CAID. **Right:** The learning curves of different variants for identity modeling, including ablation and alternative strategies. CAID with Fixed ID denotes a configuration where each agent is assigned a static identity based on its index in the agent list. CAID with Random ID refers to a setting where agents are assigned randomly generated but episode-consistent identities.