MA2CL:Masked Attentive Contrastive Learning for Multi-Agent Reinforcement Learning

Haolin Song[1] , **Mingxiao Feng**[1] , **Wengang Zhou**[1,2] and **Houqiang Li**[1,2]

[1]EEIS Department, University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China
{hlsong, fmxustc}@mail.ustc.edu.cn, {zhwg, lihq}@ustc.edu.cn,

Abstract

Recent approaches have utilized self-supervised auxiliary tasks as representation learning to improve the performance and sample efficiency of vision-based reinforcement learning algorithms in single-agent settings. However, in multi-agent reinforcement learning (MARL), these techniques face challenges because each agent only receives partial observation from an environment influenced by others, resulting in correlated observations in the agent dimension. So it is necessary to consider agent-level information in representation learning for MARL. In this paper, we propose an effective framework called **M**ulti-**A**gent **M**asked **A**ttentive **C**ontrastive **L**earning (MA2CL), which encourages learning representation to be both temporal and agent-level predictive by reconstructing the masked agent observation in latent space. Specifically, we use an attention reconstruction model for recovering and the model is trained via contrastive learning. MA2CL allows better utilization of contextual information at the agent level, facilitating the training of MARL agents for cooperation tasks. Extensive experiments demonstrate that our method significantly improves the performance and sample efficiency of different MARL algorithms and outperforms other methods in various vision-based and state-based scenarios.

1 Introduction

Recent advances in reinforcement learning (RL) and multi-agent reinforcement learning (MARL) have led to remarkable progress in developing artificial agents that can cooperate to solve complex tasks. While the performance is encouraging, the agents require extensive training time and millions of interactions with the environment, especially in a vision-based setting (agents learn from visual observation). Therefore, enhancing the sample efficiency has become a challenge in both RL and MARL communities.

Various techniques have been proposed to improve the sample efficiency of RL in single-agent environments through joint learning, which combines the RL loss with auxiliary tasks in the form of self-supervised learning (SSL). Some methods utilize data augmentation to generate multiple views representation for constructing SSL learning objectives. Some other methods employ a dynamic model to predict future states given the current state and future action sequences, then use the prediction results and the ground truth to construct SSL loss. These dynamic models aim to learn temporally aware representations and thus improve the agent's ability to predict the potential outcomes of its actions over time. The SSL auxiliary tasks provide additional representation supervision and facilitate the acquisition of informative representations, which better serve policy learning.

The dynamic model has been demonstrated effective to learn temporally aware representations in single-agent RL, where agents have a global observation of the environment. However, in the multi-agent setting, the situation becomes vastly different, as each agent only receives a partial observation from the environment that is influenced by others agents. This makes building a dynamic model for each agent with incomplete information challenging. The observation of different agents may be interrelated and contain agent-level information. Furthermore, in cooperative tasks, all agents need to collaborate to achieve a common goal, and the agent-level information in their observation should be considered when making decisions. So it is necessary for agents to learn representations with team awareness in MARL. MAJOR [Feng *et al.*, 2023] extends the dynamic model in SPR [Schwarzer *et al.*, 2020] for multi-agent settings, but only focuses on temporal awareness, without specifically taking advantage of the correlation among the agents. At present, it appears that recent works in MARL have not explicitly incorporated representation with agent-level information as a learning objective.

In this paper, we introduce a novel representation learning framework called **M**ulti-**A**gent **M**asked **A**ttentive **C**ontrastive **L**earning (MA2CL). MA2CL aims to encourage representations to be both temporal and agent-level predictive, achieved by reconstructing the "masked agent's" observation in latent space. We sample observations of all agents at the same time step from the replay buffer and treat them as a sequence with contextual relationships in the agent dimension. Next, we randomly mask several agents with information from the previous time step, then map the masked observations into latent space. We utilize an attention reconstruction model truct the masked agent in latent space, generating another view of the masked agent's representation. We construct a contrastive learning objective for training, based on the intuition that the reconstructed representations should be similar to the ground truth while dissimilar to others. In this way, we build a representation learning objective optimized together with the policy learning objectives, as shown in Fig. 1.

It is worth noting that our algorithm can be easily integrated as an auxiliary task module in many multi-agent algorithms. In order to assess the effectiveness of MA2CL, we implement it based on the state-of-the-art MARL baselines and compared their performance with our approach across various multi-agent environments involving both vision-based and state-based scenarios. Our method outperformed the baselines in these evaluations.

The contributions of our work are summarized as follows:

1) We propose MA2CL, an attentive contrastive representation learning framework for MARL algorithms, which encourages agents to learn effective representations.
2) We implement MA2CL on both MAT and MAPPO, demonstrating its flexibility and ability to be incorporated into various off-policy MARL algorithms.
3) Through extensive experiments, we demonstrate that MA2CL outperforms previous methods and achieves state-of-the-art performance in both vision-based and state-based multi-agent environments.

2 Related Works

Sample-Efficient Reinforcement Learning Sample efficiency assesses how well interaction data are utilized for model training [Huang *et al.*, 2022]. Sample-efficient RL tries to maximize the expected return of the policy during training by interacting with the environment as little as feasible [Ma *et al.*, 2022]. To improve the sample efficiency of RL that learns a policy network from high-dimensional inputs in single-agent settings, recent works design auxiliary tasks to explicitly improve the learned representations [Yarats *et al.*, 2021; Srinivas *et al.*, 2020; Zhu *et al.*, 2022; Lee *et al.*, 2020; Schwarzer *et al.*, 2020; Zhao *et al.*, 2022; Ye *et al.*, 2021; Yu *et al.*, 2021b; Yu *et al.*, 2022], or adopt data augmentation techniques, to improve the diversity of data used for training [Laskin *et al.*, 2020; Yarats *et al.*, 2020]. In multi-agent environments, the observations of various agents already provide a diverse range of observations, and the need for additional data augmentation may be less pressing.

Representation Learning in MARL To the best of our knowledge, there are few works that have investigated the promotion of representation in the context of multi-agent reinforcement learning (MARL). In [Shang *et al.*, 2021], it utilizes an agent-centric predictive objective, where each agent is tasked to predict its future location and incorporates this objective into an agent-centric attention module. However, although an auxiliary task was first introduced to MARL by [Shang *et al.*, 2021], it is only designed to predict the position of agent in a 2D environment and is not adaptable. Another work focus representation learning in MARL is MAJOR [Feng *et al.*, 2023], which employs a joint transition model to predict the future latent representations of all agents. However, constructing the auxiliary task using predictions from all agents results in focusing on the entire team's information in the temporal sequence rather than correlation in the agent level. In this work, our auxiliary task encourages agents to take full advantage of agent-level information which is important in various cooperative tasks.

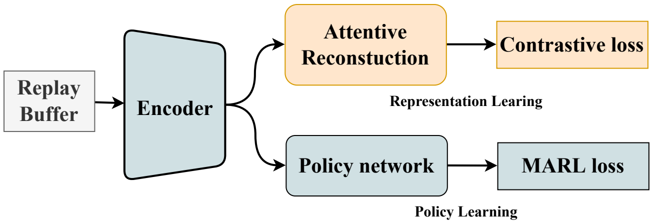


Figure 1: The learning process of MA2CL. In policy learning, the inputs are processed by the encoder and policy network to construct MARL loss for training. In representation learning, the masked inputs are processed by the same encoder and reconstructed by the attentive model and training with contrastive loss.

3 Backgroud

Dec-POMDP Cooperative MARL problems are often modeled using decentralized partially observable Markov decision processes (Dec-POMDPs) [Oliehoek and Amato, 2016], $\langle \mathcal{N}, \boldsymbol{\mathcal{O}}, \boldsymbol{\mathcal{A}}, R, P, \gamma \rangle$. $\mathcal{N} = \{1, \ldots, n\}$ is the set of agents, $\boldsymbol{\mathcal{O}} = \prod_{i=1}^{n} \mathcal{O}^i$ is the product of local observation spaces of the agents, namely the joint observation space, $\boldsymbol{\mathcal{A}} = \prod_{i=1}^{n} \mathcal{A}^i$ is the joint action space, $R : \boldsymbol{\mathcal{O}} \times \boldsymbol{\mathcal{A}} \to \mathbb{R}$ is the joint reward function, $P : \boldsymbol{\mathcal{O}} \times \boldsymbol{\mathcal{A}} \times \boldsymbol{\mathcal{O}} \to \mathbb{R}$ is the transition probability function, and $\gamma \in [0, 1)$ is the discount factor. At each time step $t \in \mathbb{N}$, each agent observes a local observation $o_t^i \in \mathcal{O}^i$ and takes an action a_t^i according to its policy π^i. The next set of observations $\mathbf{o}_{t+1}$ is updated based on a transition probability function, and the entire team receives a joint reward $R(\mathbf{o}_t)$. The goal is to maximize the expected cumulative joint reward over a finite or infinite number of steps. In our framework, we use the base MARL algorithm as a policy learning part.

Multi-Agent Proximal Policy Optimization (MAPPO) MAPPO [Yu *et al.*, 2021a] is a method for applying the Proximal Policy Optimization (PPO) [Schulman *et al.*, 2017] algorithm to multi-agent reinforcement learning (MARL). Each agent in MAPPO has a representation encoder and a policy network. The representation encoder process the observation of one agent, and the policy network generates an action based on the representation. The parameters of the representation encoder and policy network are shared by all agents for training, MAPPO updates the parameters using the aggregated trajectories of all agents.

Multi-Agent Transformer (MAT) MAT [Wen *et al.*, 2022] is a transformer encoder-decoder architecture that changes the joint policy search problem into a sequential decision-making process. The encoder maps an input sequence of observations to latent representations and the decoder generates a sequence of actions in an auto-regressive manner. MAT simplifies the sequence modeling paradigm for multi-agent reinforcement learning by treating the team of agents as a

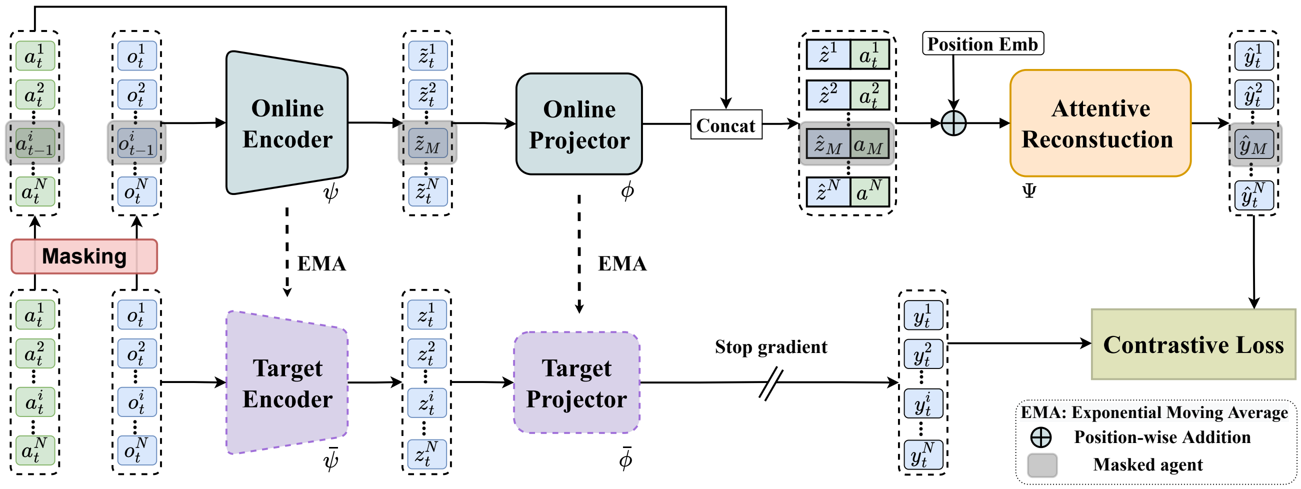


Figure 2: The framework of the MA2CL. $\{o_i\}, \{a_i\}$ are observations and actions of all agents at timestep t from the replay buffer. We randomly select a subset of agents and mask them with observation in time step $t-1$. The masked observation sequence will be mapped into latent space using an encoder and a projector. An attentive model will reconstruct the latent feature of masked agents given the masked action sequence and identity position embedding. Our method trains the attentive model truct accurately, using a contrastive loss between the predicted features of the masked agent and the target features inferred from the original observation sequence. Notice that the encoder and the processing of the observation sequence are the **same** as that in the basic MARL algorithm.

single sequence. The learning objectives for MAT are represented by $L_{Encoder}$ and $L_{Decoder}$.

The learning objectives of MAPPO and MAT serve as the MARL loss and are detailed in the supplementary materials.

Contrastive Learning Contrastive learning is an optimization objective for self-supervised algorithms and is used in many auxiliary tasks for representation learning in RL. Contrastive learning can be regarded as an instance classification problem, where one instance should be distinguished from others. Given a query q and keys $\mathbb{K} = \{k_0, k_1, \ldots\}$ and an explicitly known partition of $\mathbb{K}$ (different view of q), $P(\mathbb{K}) = (\{k_+\}, \mathbb{K}\setminus\{k_+\})$. the goal of contrastive learning is to ensure that q matches with k_+ relatively more than any of the keys in $\mathbb{K}\setminus\{k_+\}$. We utilize the InfoNCE loss in [Oord *et al.*, 2018] as the contrastive loss function. It can be mathematically defined as follows:

$$\mathcal{L}_q = -\log \frac{\exp\left(\omega(q^T, k_+)\right)}{\exp\left(\omega(q^T, k_+)\right) + \sum_{i=0}^{K-1} \exp\left(\omega(q^T, k_i)\right)}, \quad (1)$$

where $\omega(q, k) = q^T W k$ is a bi-linear inner-product similarity function in [Srinivas *et al.*, 2020], and W is a learned matrix.

4 Method

Multi-Agent Masked Attentive Contrastive Learning (MA2CL) is an auxiliary task that aims to enhance representation learning in MARL by reconstructing masked agent with information from other agents and historical trajectory. This allows for better utilization of temporal and agent-level information in observation, further improving the sample efficiency in MARL tasks. Besides, MA2CL can be integrated easily into various off-policy MARL algorithms. The framework of MA2CL is shown in Fig. 2, and each component will be introduced in the following subsections.

4.1 Observation and Action Masking

We sample the observations and actions of all agents at the same time step t from the replay buffer $\mathcal{B}$. Denote the observation and action as a sequence:

$$\mathbf{o}_t = \{o_t^1, o_t^2 \ldots o_t^i \ldots o_t^N\}, \quad \mathbf{a}_t = \{a_t^1, a_t^2 \ldots a_t^i \ldots o_t^N\},$$

where N represents the total number of agents. Then, we will randomly select N_m agents for masking, as indicated by:

$$\mathbf{M} = \{M_1, M_2, \ldots, M_i, \ldots, M_N\}, M_i = 1 \text{ or } 0. \quad (2)$$

If $M_i = 1$ then the agent will be masked. In our masking strategy, if agent i_m is selected, its observation and action $(o_t^{i_m}, a_t^{i_m})$ will be modified as follows: $o_t^{i_m}$ is replaced by $o_{t-1}^{i_m}$ and $a_t^{i_m}$ is replaced by $a_{t-1}^{i_m}$, where $(o_{t-1}^{i_m}, a_{t-1}^{i_m})$ is the observation and action of agent i_m at time step $t-1$, then we get the masked observation and action sequence as :

$$\tilde{\mathbf{o}}_t = \{o_t^1, \ldots o_{t-1}^{i_m} \ldots o_t^N\}, \quad \tilde{\mathbf{a}}_t = \{a_t^1, \ldots a_{t-1}^{i_m} \ldots a_t^N\}.$$

We will utilize the masked observation sequence during the encoding and reconstruction stages. The masked actions are only employed in the reconstruction stage. Our aim is truct the masked observation by leveraging information from both agent and time domains. The number of masked agents, denoted as $N_m (1 \leq N_m \leq N)$ is a hyperparameter that would be set prior to training.

4.2 Encoding Observation Sequence

We use two encoders to obtain representations from masked and original observation sequences respectively. The *online* encoder, denoted as ψ, is the same as the representation(encoder) network in the MARL agent. It can be a centralized encoder that processes all agent's observations simultaneously or a decentralized encoder set: $\psi =$

$\{\psi^1, \psi^2, \dots, \psi^N\}$, where each agent's observation is processed by its own encoder. For simplicity, we use ψ to denote the *online* encoder. This encoder maps the masked observation $\widetilde{\mathbf{o}}_t$ into latent space, represented as $\tilde{\mathbf{z}}_t$. This process is formulated as $\tilde{\mathbf{z}}_t = \psi(\widetilde{\mathbf{o}}_t)$. Similar to [Srinivas *et al.*, 2020], we employ a *target* encoder, denoted as $\bar{\psi}$, to encode the original observation sequence, formulated as $\mathbf{z}_t = \bar{\psi}(\mathbf{o}_t)$. The *target* encoder has the same architecture as the *online* encoder, and its parameters are updated by an exponential moving average (EMA) of the online encoder parameters. Given *online* encoder ψ parameterized by θ , *target* encoder $\bar{\psi}$ parameterized by $\bar{\theta}$ and the momentum coefficient $\tau \in [0, 1)$, the target encoder will be updated as follows:

$$\bar{\theta} \leftarrow \tau\bar{\theta} + (1-\tau)\theta. \tag{3}$$

Previous works [Chen *et al.*, 2020b; Chen *et al.*, 2020a] have experimentally demonstrated that introducing a nonlinear layer before the contrastive loss can significantly improve performance. Therefore, we also include a non-linear projection layer ϕ before reconstruction in our model. This layer is implemented as an MLP network with GELU activation.

We use the *online* projector ϕ to process the latent state feature sequence $\tilde{\mathbf{z}}_t$, and we get obtain the following result: $\hat{\mathbf{z}}_t = \{\hat{z}_t^1, \hat{z}_t^2, \dots, \hat{z}_t^N\}$, where $\hat{z}_t^i = \phi(\tilde{z}_t^i)$. For the encoded results obtained from original observations, denoted as $\mathbf{z}$, we use a *target* projector $\bar{\phi}$ for further processing. *target* projector has the same structure as the online projector and its parameters are updated with *online* projector following the same EMA update strategy in the encoder. We get the **target** latent feature sequence as follows:

$$\mathbf{y}_t = \{y_t^1, y_t^2, \dots y_t^i, \dots y_t^N\}, \quad \text{where} \quad y_t^i = \bar{\phi}(z_t^i). \tag{4}$$

As shown in Fig. 2, we apply a *stop-gradient* operation on *target* projector to avoid model collapse, following [Grill *et al.*, 2020; Srinivas *et al.*, 2020].

4.3 Attentive Reconstruction

We propose an attentive reconstruction model Ψ to recover the representation of the masked observations. The model Ψ consists of L identical blocks, each of which is an attention layer. To add the identity information of each agent in the team, we adopt relative positional embedding, which are commonly used in the standard Transformer [Vaswani *et al.*, 2017]. truct the masked information in latent space, we leverage three types of information: (a) the representation sequence $\hat{\mathbf{z}}_t$, which is obtained from the masked observation sequence $\tilde{\mathbf{o}}_t$ through processed by *online* encoder and *online* projector sequentially. This sequence contains information from other agents and historical trajectory; (b) the action information $\widetilde{\mathbf{a}}_t$, which corresponds to $\hat{\mathbf{z}}_t$ provides decision information from each agent, including the masked ones; (c) the identity information $\mathbf{p} = (p^1, p^2, \dots, p^N)$, which is obtained from the positional embedding and reflects the identity of each agent in the sequence. To integrate these three types of information, we concatenate $\hat{z}_t^i$ with the masked action $\tilde{a}_t^i$, and add identity information p^i to the concatenating vector, as the identity information is related to both $\hat{z}_t^i$ and $\tilde{a}_t^i$. Thus, the inputs tokens of the predictive model can be expressed as:

$$\mathbf{x} = \{[\hat{z}_t^1 : \tilde{a}_t^1] + p^1, [\hat{z}_t^2 : \tilde{a}_t^2] + p^2, \dots, [\hat{z}_t^N : \tilde{a}_t^N] + p^N\}. \tag{5}$$

Algorithm 1 Training Process for MA2CL

```
Input: number of agents $N$, number of masking agents $N_M$
Parameter: parameters in online encoder $\psi$, online projector $\phi$, similarity function $W$,policy network, target encoder $\bar{\psi}$ , target projector $\bar{\phi}$.
Determine EMA coefficient $\tau$
while Training do
    Interact with the environment and collect the transition: $\mathcal{B} \leftarrow \mathcal{B} \cup (\mathbf{o_t}, \mathbf{a_t}, r_t, \mathbf{o_{t+1}})$
    Sample a minibatch $(\mathbf{o_t}, \mathbf{a_t}, r_t, \mathbf{o_{t+1}})$ from $\mathcal{B}$.
    Calculate RL loss $\mathcal{L}_{\text{rl}}$ based on a given basic MARL algorithm ($e.g.$ MAT, MAPPO)
    Sample another minibatch $(\mathbf{o_t}, \mathbf{a_t})$ from $\mathcal{B}$
    Randomly masks $N_M$ agent: $\tilde{\mathbf{o}}_\mathbf{t}, \tilde{\mathbf{a}}_\mathbf{t} \leftarrow \text{Mask}(\mathbf{o}_t, \mathbf{a}_t)$
    Process $\tilde{\mathbf{o}}_\mathbf{t}$ and $\mathbf{o_t}$ based on Eq. (4),Eq. (5) and Eq. (7), obtain $\hat{\mathbf{y_t}}$ and $\mathbf{y_t}$.
    Calculate contrastive loss $\mathcal{L}_{\text{cl}}$ based on Eq. (8)
    Calculate total loss: $\mathcal{L}_{\text{toatal}} = \mathcal{L}_{\text{rl}} + \mathcal{L}_{\text{cl}}$
    Update online encoder $\psi$, online projector, similarity function $W$, policy network with $\mathcal{L}_{\text{total}}$
    Update target encoder and projector based on Eq. (3)
end while
```

The input token sequence is processed by L attention layers in the attentive reconstruction model. The l-th layer processes the input token sequence according to the following steps:

$$\begin{aligned} \mathbf{h}^l &= \text{MSHA}\left(\text{LN}\left(\mathbf{x}^l\right)\right) + \mathbf{x}^l, \\ \mathbf{x}^{l+1} &= \text{FFN}\left(\text{LN}\left(\mathbf{h}^l\right)\right) + \mathbf{h}^l. \end{aligned} \tag{6}$$

The Layer Normalization (LN), Multi-Headed Self-Attention (MSHA) layer, and Feed-Forward Network(FFN) are the same as those used in the Transformer [Vaswani *et al.*, 2017].

Given the input $\mathbf{x}$, we try to recover a sequence of all agent observations in the latent space using the attentive reconstruction model, then we get:

$$\hat{\mathbf{y}} = (\hat{y}_t^1, \hat{y}_t^2, \dots, \hat{y}_t^N) = \Psi(\mathbf{x}). \tag{7}$$

Using the reconstructed feature sequence $\hat{\mathbf{y}}_t$ and the target feature sequence $\mathbf{y}$ from Eq. (4), we construct a contrastive loss to improve the accuracy of the reconstruction model and the ability of the encoder, which will be described in detail in the following section.

4.4 Contrastive Loss

Inspired by the success of [Oord *et al.*, 2018; Srinivas *et al.*, 2020], we use contrastive learning to optimize the parameters of *online* encoder, projector, and reconstruction model. The query set $\mathcal{Q} = \{q_i | q_i = \hat{y}^i\}$ is the reconstructed feature of the masked agent according to Eq. (7). The key set $\mathcal{K} = \{k_i | k_i = y^i\}$ is encoded from the non-masked observation. As defined in Eq. (4), (q_i, k_i) is a query-key pair from the same agent. We use the function $\omega(q^T, k)$ from Eq. (1) to measure the similarity between query and key. We can formulate the contrastive loss as follows:

$$\mathcal{L}_{\text{cl}} = \sum_{i=1}^{N} -M_i \log \frac{\exp\left(\omega\left(q_i, k_i\right)\right)}{\sum_{j=1}^{N} \exp\left(\omega\left(q_i, k_j\right)\right)}. \tag{8}$$

The contrastive loss $\mathcal{L}_{\text{cl}}$ in Eq. (8) is designed based on the following intuition: the reconstructed feature $\hat{y}^i$ ($i.e., q_i$) should be similar to its corresponding original feature y^i ($i.e., k_i$) while being distinct from the others. M_i (as specified in Eq. (2)) emphasizes the focus on the reconstructed features of the masked agents rather than those of the unmasked ones. By minimizing the contrastive loss $\mathcal{L}_{\text{cl}}$, we aim to train an accurate attentive reconstruction model and a powerful encoder. The powerful encoder is able to extract an informative representation from observation input, allowing the reconstruction model to recover the sequence in the latent space even when some agents are masked.

4.5 Overall Training Objective

The agent in MARL algorithms consists of two parts: the representation encoder and the policy network. The *online* encoder of MA2CL is the same as the representation encoder in the MARL algorithm for policy learning. MA2CL provides a powerful encoder for basic MARL algorithm, which helps agents learn cooperative policy more quickly. In MA2CL, representation learning is optimized together with policy learning. Thus, the overall training objective of our method is as below:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{rl} + \lambda \mathcal{L}_{\text{cl}}, \tag{9}$$

where $\mathcal{L}_{rl}$ is the loss functions of the base MARL algorithm ($e.g.$ MAT [Wen *et al.*, 2022], MAPPO [Yu *et al.*, 2021a]), and $\mathcal{L}_{\text{cl}}$ is the loss functions of the MA2CL we introduced. λ is a hyperparameter for balancing the two terms. Following [Srinivas *et al.*, 2020], we fix λ as 1. A detailed example of MA2CL is provided in Algorithm 1.

5 Experiments

To evaluate the performance of MA2CL, we use different multi-agent reinforcement learning benchmarks, including both vision-based and state-based scenarios. We set mask agent number $N_m = 1$, attention layer $L = 1$. Other hyperparameters settings can be found in supplementary materials.

5.1 Setup

Vision-based MARL Environments In single-agent settings, representation learning techniques are commonly used in vision-based reinforcement learning (RL) to improve sample efficiency. We demonstrate the effectiveness of our proposed MA2CL model in vision-based MARL settings on the Multi-Agent Quadcopter Control benchmark (MAQC) [Panerati *et al.*, 2021].

In the MAQC environment, each agent receives an RGB video frame $\in \mathbb{R}^{64\times48\times4}$ as an observation, which is captured from a camera fixed on the drone. MAQC allows for continuous action settings of varying difficulty, including RPM and PID. When operating under the RPM setting, the agent's actions directly control the motor speeds. In the PID setting, the agent's actions directly control the PID controller, which calculates the corresponding motor speeds. In this work, we evaluate MA2CL on two cooperative tasks $Flock$ and $LeaderFollower$ in both RPM and PID settings. Some detail about the task can be found in supplementary materials.

State-based MARL Environments In order to demonstrate the applicability of our method in both vision-based and state-based scenarios within MARL environments, and due to the lack of vision-based benchmarks, we conduct an addtional evaluation on commonly utilized state-based MARL benchmarks. We use a variety of state-based MARL scenarios, such as the StarCraft Multi-Agent Challenge (SMAC) [Samvelyan *et al.*, 2019] and Multi-Agent MuJoCo [de Witt *et al.*, 2020], where the performance of the baselines represents the current state-of-the-art performance in MARL.

Baseline Different current state-of-the-art baselines are selected for comparison, including MAT [Wen *et al.*, 2022] and MAJOR [Feng *et al.*, 2023], as they have been shown to perform well in both continuous and discrete MARL environments. The hyperparameters are kept consistent with those used in the original papers for a fair comparison. A total of five random seeds were used to obtain the mean and standard deviation of various evaluation metrics such as episode rewards in Multi-Agent MuJoCo, and the winning rate measured in the SMAC.

Additionally, we play four drone agents in the MAQC environment, incorporating different action settings such as RPM and PID. In the Multi-Agent MuJoCo environment, agents were able to observe the state of joints and bodies from themselves and their immediate neighbors.

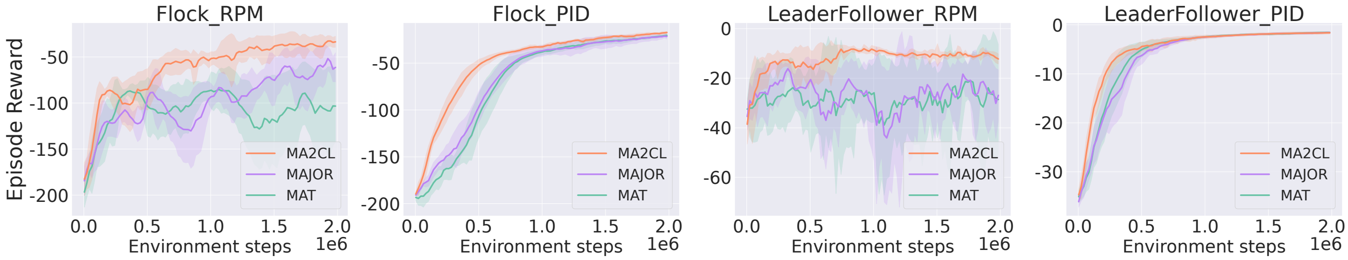
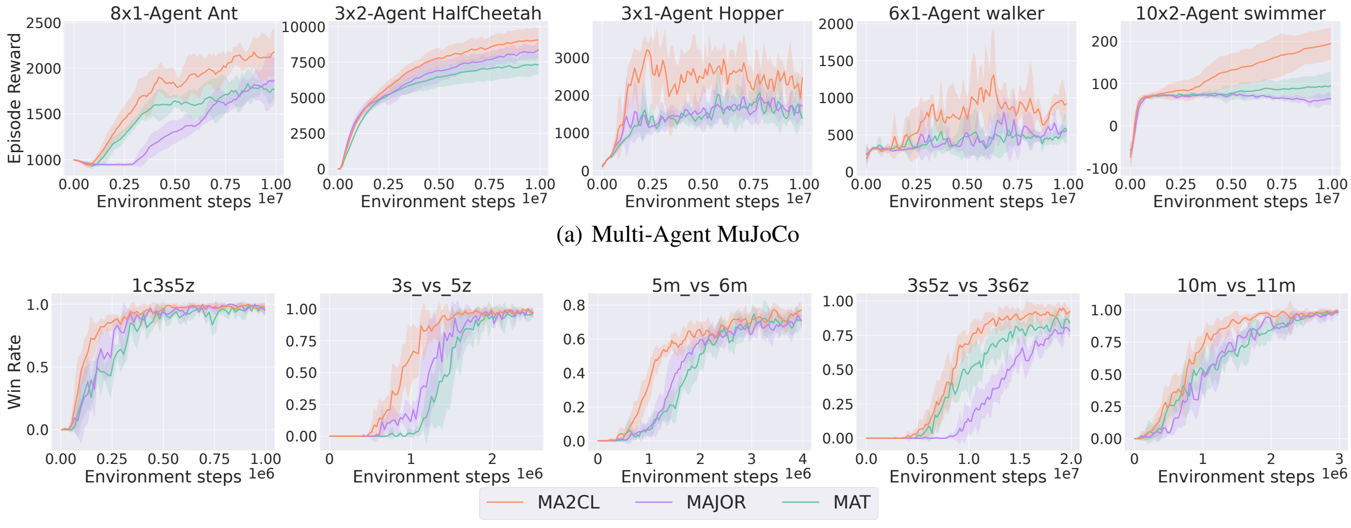


Figure 3: Results on Multi-Agent Quadcopter Control. MA2CL outperforms other baseline methods on three out of four tasks as measured by episode reward, and exhibits improved sample efficiency in all tasks. The orange line is MA2CL, the purple line is MAJOR, and the green line is MAT. The shaded region represents the standard deviation of the average return over 5 seeds. It is worth noting that both MA2CL and MAJOR utilize the MARL loss from MAT for training the policy.



(a) Multi-Agent MuJoCo

(b) StarCraftII Multi-Agent Challenge

Figure 4: Results on state-based MARL environment: (a) Multi-Agent MuJoCo, (b) StarCraftII Multi-Agent Challenge, The results indicate that MA2CL outperforms MAJOR and MAT in both performance and sample efficiency across different state-based scenarios. Due to the page limit, additional results on other scenarios can be found in the supplementary materials.

5.2 Result and Discussion

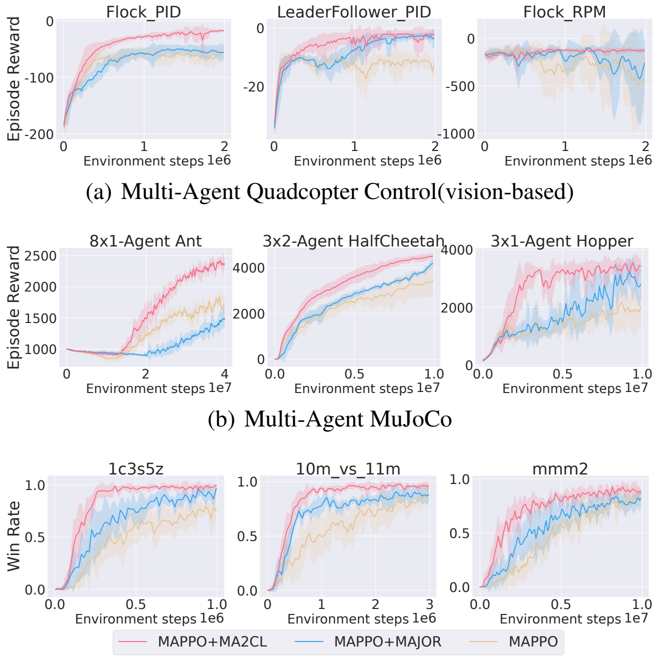
To ensure a fair comparison, we implement MA2CL on MAT, as the MAJOR algorithm is also proposed based on MAT.

Result in vision-based Environment As shown in Fig. 3, the performance of the MA2CL is evaluated in the MAQC environment by comparing it to the MAJOR and MAT algorithms. We can observe that MA2CL demonstrates superior performance and sample efficiency in both RPM and PID action settings. Specifically, MA2CL achieves the highest episode reward in three out of four tasks. Additionally, MA2CL requires fewer steps to obtain high episode rewards compared to other baseline algorithms. The superior performance of MA2CL can be attributed to its ability to effectively capture task-relevant information from high-dimensional inputs (RGB image in MAQC), which enables it to generate more informative representations that incorporate both temporal and team awareness. This enhances the ability of the drones to learn good policy and effectively collaborate and complete cooperative tasks.

Result in state-based Environment In the state-based MARL environment, MA2CL also exhibits strong performance when compared to MAJOR and MAT. Fig. 4(a) and Fig. 4(b) show the performance of MA2CL and baselines in different tasks of Multi-Agent MuJoCo and SMAC respectively. The results indicate that MA2CL demonstrates robust performance, improving the performance of MAT in most tasks and surpassing that of MAJOR. These findings suggest that the MA2CL algorithm not only possesses the ability to extract task-relevant information from high-dimensional (image) input but also to effectively integrate information from agent-level in state-based observations, which provide more direct information. For instance, in the Multi-Agent MuJoCo environment, the decision-making process of the agents, in terms of the joints of the body, must take into account the velocities and positions of other joints to produce human-like joint actions. Our MA2CL algorithm enables agents to fully leverage the information of neighboring agents present in their observations, allowing for decisions to be made based on a representation of team awareness and resulting in superior performance.

Effect on other MARL Algorithms To exhibit the plug-and-play nature of our proposed MA2CL framework across various MARL algorithms, we implement MA2CL on the MAPPO algorithm, denated as MAPPO+MA2CL. We select MAPPO as a representative example of the actor-critic algorithm in MARL due to its utilization of a CNN encoder (for vision-based tasks) or an MLP encoder (for state-based tasks) to encoder observation from every single agent, in contrast to the transformer-based encoder processing observations from all agents in MAT. To ensure fairness in comparison, we also choose MAPPO+MAJOR, which is the implementation of MAJOR on MAPPO, and the standard MAPPO algorithm as baselines for evaluation. Our experiments are conducted in both visual and state-based MARL environments.

As shown in Fig. 5(a), Fig. 5(b), and Fig. 5(c), MA2CL is found to significantly enhance the performance of MAPPO in both visual and state-based tasks and outperform the MAPPO+MAJOR in these tasks. The representation encoder in MAPPO only utilizes observations from a single agent for decision-making, as opposed to the sequential decision model used in MAT, where agents have explicit agent-level information from previous agents' observations in decision-making. We can observe that, with the introduction of MA2CL, we obtain a more notable improvement effect on the MAPPO than on the MAT. This suggests that the MA2CL framework enhances the representation encoder in MAPPO to extract more task-relevant information and exploit the agent-level informa-



(a) Multi-Agent Quadcopter Control(vision-based)

(b) Multi-Agent MuJoCo

(c) StarCraftII Multi-Agent Challenge

Figure 5: Result for MAPPO+MA2CL, MAPPO+MAJOR, and MAPPO in both vision and state-based MARL environments. Due to the page limit, additional results on other scenarios can be found in supplementary materials.

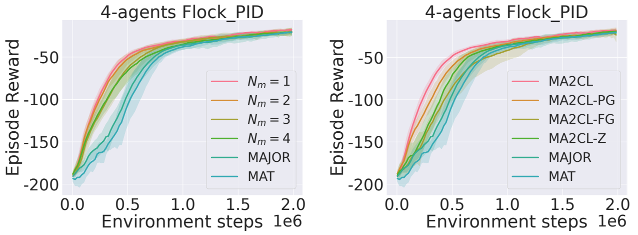
tion from observation. This enhanced informative representation enables agents to effectively learn cooperative strategies, resulting in higher performance and increased sample efficiency in MAPPO+MA2CL. This highlights the potential of our MA2CL to promote various MARL algorithms.

5.3 Ablation Studies

To study the impact of masked agent numbers and masking strategy on MA2CL, we conduct ablation studies in which we varied the number of masked agents or masking strategy while holding other parameters constant. Additional ablation results can be found in the supplementary materials.

Variation in masked agent numbers. In MA2CL, the number of masked agents, N_m, determines the ratio of accurate agent-level information from the same time step. As N_m increases, more information comes from the historical trajectory. Therefore, the agent-level information utilized by the reconstruction model decreases and becomes outdated. As shown in Fig. 6(a), when we perform masking on three or four agents in a task involving four drones, the performance decreases slightly but still surpasses the baseline. This is because the agent-level information from the history is not accurate but also contains information in the time dimension, which is also beneficial for agent decision-making.

Different masking strategy. On the other hand, the masking strategy in MA2CL is also a crucial factor in the reconstruction process. One consideration is that in MA2CL, we only use $o_{t-1}^{i_m}$ to mask the agents, which may be viewed as "replacement" rather than "masking". Therefore, we explored two other masking strategies:



(a) Different mask agent number N_m in **four** agents scenario.

(b) Study on different masking strategy.

Figure 6: Result for ablation study on different mask agent number N_m and masking strategy

- (1) Use $o_{t-1}^{i_m}$ plus gaussian noise to mask $o_t^{i_m}$, namely $Mask(o_t^{i_m}) = o_{t-1}^{i_m} + \mathcal{N}(0, 1)$, denoted as MA2CL-PG;
- (2) Mask $o_t^{i_m}$ with full gaussian noise, namely $Mask(o_t^{i_m}) = \mathcal{N}(0, 1)$, denoted as MA2CL-FG.
- (3) Mask $o_t^{i_m}$ with zero, namely $Mask(o_t^{i_m}) = \mathbf{0}$, denoted as MA2CL-Z.

As shown in Fig. 6(b), even when using zero to completely mask the agent's observation, in which case only agent-level information can be used for reconstruction, MA2CL-Z still demonstrates superior performance. This indicates that MA2CL indeed learns informative representation with agent-level information and helps the agent learn better cooperation strategies. Additionally, the performance of MA2CL-PG is slightly worse than MA2CL, but better than MA2CL-FG and MA2CL-Z, indicating that using o_{t-1} for masking actually introduces temporal information. Therefore, MA2CL simultaneously considers valuable information in the time dimension and agent level, thus learning more informative representations to promote agent policy learning.

5.4 Conclusion

In this work, we present the Multi-Agent Masked Attentive Contrastive Learning (MA2CL) framework which utilizes an attentive reconstruction model that encourages the learning of temporal and team awareness representations, thereby enhancing the sample efficiency and performance of MARL algorithms. Extensive experiments on both vision-based and state-based cooperative MARL benchmarks show MA2CL set the state-of-the-art performance.

In terms of future work, there are several avenues that warrant further exploration. One potential direction is to investigate the use of more elaborate masking strategies to allow for attention to different types of information in MARL settings. Additionally, given the importance of sample efficiency in reinforcement learning, future efforts could be directed toward methods that are specifically designed to address this issue in vision-based multi-agent environments.What's more, we only consider the encoder of actor, and it is also worth thinking about the approach of treat the encoder of both critic and actor as the target of representation learning.

Acknowledgments

This work was supported in part by NSFC under Contract 61836011, and in part by the Fundamental Research Funds for the Central Universities under contract WK3490000007.

References

[Chen *et al.*, 2020a] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[Chen *et al.*, 2020b] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[de Witt *et al.*, 2020] Christian Schroeder de Witt, Bei Peng, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. *arXiv preprint arXiv:2003.06709*, 2020.

[Feng *et al.*, 2023] Mingxiao Feng, Wengang Zhou, Yaodong Yang, and Houqiang Li. Joint-predictive representations for multi-agent reinforcement learning, 2023. Paper rejected from the International Conference on Learning Representations (ICLR), https://openreview.net/forum?id=S80ioOGLpD9.

[Grill *et al.*, 2020] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[Huang *et al.*, 2022] Jiawei Huang, Jinglin Chen, Li Zhao, Tao Qin, Nan Jiang, and Tie-Yan Liu. Towards deployment-efficient reinforcement learning: Lower bound and optimality. *arXiv preprint arXiv:2202.06450*, 2022.

[Laskin *et al.*, 2020] Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

[Lee *et al.*, 2020] Alex X Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 2020.

[Ma *et al.*, 2022] Guozheng Ma, Zhen Wang, Zhecheng Yuan, Xueqian Wang, Bo Yuan, and Dacheng Tao. A comprehensive survey of data augmentation in visual reinforcement learning. *arXiv preprint arXiv:2210.04561*, 2022.

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

[Oord *et al.*, 2018] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[Panerati *et al.*, 2021] Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P Schoellig. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7512–7519. IEEE, 2021.

[Samvelyan *et al.*, 2019] 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.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Schwarzer *et al.*, 2020] Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

[Shang *et al.*, 2021] Wenling Shang, Lasse Espeholt, Anton Raichuk, and Tim Salimans. Agent-centric representations for multi-agent reinforcement learning. *arXiv preprint arXiv:2104.09402*, 2021.

[Srinivas *et al.*, 2020] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

[Yarats *et al.*, 2020] Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2020.

[Yarats *et al.*, 2021] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10674–10681, 2021.

[Ye *et al.*, 2021] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, 34:25476–25488, 2021.

[Yu *et al.*, 2021a] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

[Yu *et al.*, 2021b] Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, Zhizheng Zhang, and Zhibo Chen. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5276–5289, 2021.

[Yu *et al.*, 2022] Tao Yu, Zhizheng Zhang, Cuiling Lan, Zhibo Chen, and Yan Lu. Mask-based latent reconstruction for reinforcement learning. *arXiv preprint arXiv:2201.12096*, 2022.

[Zhao *et al.*, 2022] Jian Zhao, Youpeng Zhao, Weixun Wang, Mingyu Yang, Xunhan Hu, Wengang Zhou, Jianye Hao, and Houqiang Li. Coach-assisted multi-agent reinforcement learning framework for unexpected crashed agents. *Frontiers of Information Technology & Electronic Engineering*, 23(7):1032–1042, 2022.

[Zhu *et al.*, 2022] Jinhua Zhu, Yingce Xia, Lijun Wu, Jiajun Deng, Wengang Zhou, Tao Qin, Tie-Yan Liu, and Houqiang Li. Masked contrastive representation learning for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.