JOINT-PREDICTIVE REPRESENTATIONS FOR MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The recent advances in reinforcement learning have demonstrated the effectiveness of vision-based self-supervised learning (SSL). However, the main efforts on this direction have been paid on single-agent setting, making multi-agent reinforcement learning (MARL) lags thus far. There are two significant obstacles that prevent applying off-the-shelf SSL approaches with MARL on a partially observable multi-agent system : (a) each agent only gets a partial observation, and (b) previous SSL approaches only take consistent temporal representations into account, while ignoring the characterization that captures the interaction and fusion among agents. In this paper, we propose **M**ulti-**A**gent **Jo**int-Predictive **R**epresentations (MAJOR), a novel framework to explore self-supervised learning on cooperative MARL. Specifically, we treat the latent representations of local observations of all agents as the sequence of masked contexts of the global state, and we then learn effective representations by predicting the future latent representations for each agent with the help of the agent-level information interactions in a joint transition model. We have conducted extensive experiments on wide-range MARL environments, including both vision-based and state-based scenarios, and show that our proposed MAJOR achieves superior asymptotic performance and sample efficiency against other state-of-the-art methods.

1 INTRODUCTION

Representation learning has played an important role in recent developments of reinforcement learning (RL) algorithms. Especially self-supervised representation learning (SSL) has attracted more and more attention due to its success in other research fields (He et al., 2019; Devlin et al., 2018; Liu et al., 2019; Lan et al., 2019). Recently, numerous works(Srinivas et al., 2020; Zhu et al., 2022; Yarats et al., 2021; Schwarzer et al., 2021a; Yu et al., 2022) have borrowed insights from different areas and attempted to use SSL-based auxiliary tasks to learn more effective representations of RL and thus improve the empirical performance. Through augmentation for inputs, it can conduct multiple views for building SSL learning objectives, allowing the agent to improve data efficiency and generalization to obtain better task-related representations. Moreover, many proper auxiliary self-supervision priors proposed predictive representations with the help of an additional learned dynamic model, which is utilized to encourage the representations to be temporally predictive and consistent.

However, when meeting partially observable multi-agent systems, it is challenging to apply such self-supervision priors to learn compact and informative feature representations in multi-agent reinforcement learning (MARL). The critical obstacle to learning effective representations in MARL is that agents in partially observable multi-agent systems only have access to their observations, which means that other agents' behavior influences each agent's observations. As a result, independently building representation priors for each agent may be failed due to imperfect information. Furthermore, in the MARL context, it is more important to focus on the representations that embody the interaction and fusion between agents in the environment but not temporal representations for each agent. In other words, it is necessary to learn the representations that can take the other agents into account.

In this work, we propose a novel representation learning framework for MARL, named **M**ulti-**A**gent **Jo**int-Predictive **R**epresentations (MAJOR), which trains better representations for MARL by forc-

ing representations to be temporally predictive and consistent for all agents at the same timestep. We posit that in one timestep, latent representations of local observations of all agents can be treated as the sequence of masked contexts of the global state so that we can predict the future latent representations for each agent with the representations of corresponding actions. Accordingly, we construct a joint transition model as the bridge to connect all agents and implement the interaction of their individual information. The joint transition model treats the encoded representations of individual observations and actions as a sequence and attempts to predict future representations in latent space. Meanwhile, we can also get another view of the subsequent timestep representations by feeding sampled observations into the encoder. In this way, we build the SSL objective by enforcing consistency across different perspectives of each observation. Besides, our proposed framework is a plug-and-play module for almost common-used MARL methods.

Additionally, to maximize its power, we implement an instantiation of MAJOR on the basis of the recently proposed MARL algorithm, named Multi-Agent Transformer (MAT, (Wen et al., 2022)), which solves MARL issues via a sequential updating mechanism and an encoder-decoder architecture. In MAT, the encoder seeks to extract post-interaction representations from observations; the decoder then uses them to generate actions sequentially by a cross-attention mechanism. Moreover, our proposed MAJOR can employ both representations generated from the encoder and decoder, and the gradient derived from our representation learning objective can be back-propagated to both the encoder and decoder.

To evaluate our proposed algorithm, we construct extensive experiments on several common-used cooperative MARL benchmarks, including vision- and state-based environments in discrete and continuous scenarios against current state-of-the-art baselines such as HAPPO, MAPPO, and MAT. Results demonstrate its state-of-the-art performance across all tested tasks.

2 BACKGROUND

2.1 DEC-POMDP

Cooperative MARL problems are often modeled by decentralized Partially Observable Markov Decision Processes (Dec-POMDPs, (Oliehoek & Amato, 2016)) $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}_i\}, \mathcal{T}, R, \Omega, \mathcal{O}, \gamma)$. Here, $\mathcal{N} = 1, \ldots, n$ is the set of agents, $\mathcal{S}$ is a set of states, $\mathcal{A} = \times_i \mathcal{A}_i$ is the set of joint actions, $\mathcal{T}$ is a set of conditional transition probabilities between states, $\mathcal{T}(s, \boldsymbol{a}, s') = P(s' \mid s, \boldsymbol{a})$, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $\mathcal{O} = \times_i \mathcal{O}_i$ is a set of observations for agent i, Ω is a set of conditional observation probabilities $\Omega(s', \boldsymbol{a}, \boldsymbol{o}) = P(\boldsymbol{o} \mid s', \boldsymbol{a})$, and $\gamma \in [0, 1]$ is the discount factor. At each time step, each agent takes an action a_i, the state updates based on the transition function (using the current state and the joint action), each agent observes an observation based on the observation function $\Omega(s', \boldsymbol{a}, \boldsymbol{o})$ (using the next state and the joint action) and a reward is generated for the whole team based on the reward function $R(s, \boldsymbol{a})$. The goal is to maximize the expected cumulative reward over a finite or infinite number of steps. These time steps repeat until some given horizon (called finite horizon) or forever (called infinite horizon).

2.2 MULTI-AGENT TRANSFORMER

Multi-Agent Transformer (MAT, Wen et al. (2022)) effectively casts cooperative MARL into Sequential Modeling (SM) problems wherein the task is to map agents' observation sequence to agents' optimal action sequence. Its sequential update scheme is built on Multi-Agent Advantage Decomposition Lemma and Heterogeneous-Agent Proximal Policy Optimisation (HAPPO, Kuba et al. (2022)). The lemma provides an intuition guiding the choice of incrementally improving actions, and HAPPO fully leverages the lemma to implement multi-agent trust-region learning with a monotonic improvement guarantee. Unfortunately, HAPPO requests the sequential update scheme in the permutation for agents' orders, meaning that HAPPO cannot be run in parallel. To address the drawback of HAPPO, MAT produces Transformer-based implementation for multi-agent trust-region learning.

Concretely, MAT maintains an encoder-decoder structure, where the encoder maps an input sequence of tokens to latent representations and then the decoder generates a sequence of desired outputs in an auto-regressive manner wherein at each step of inference, the Transformer takes all

previously generated tokens as the input. In other words, MAT treats a team of agents as a sequence, thus implementing the sequence modeling paradigm for MARL. The encoder ϕ takes a sequence of observations $\boldsymbol{o} \triangleq (o^{i_1}, \ldots, o^{i_n})$ in arbitrary order [1] and passes them through L computational blocks. Each of these blocks has a self-attention mechanism(Vaswani et al., 2017) and a multi-layer perceptron (MLP), as well as residual connections to prevent gradient vanishing and network degradation as depth increases. Thus we can obtain the encoding of the observations as $\hat{\boldsymbol{o}}$ containing interrelationships among agents. Feeding the representations into the value head (an MLP), denoted as f_ϕ, will get value estimations. The encoder's learning objective is to minimize the individual version of empirical Bellman error by:

$$\mathcal{L}_{\phi, f_\phi}^{\text{MAT}_{encoder}}(\boldsymbol{o}_t) = \frac{1}{Tn} \sum_{m=1}^{n} \sum_{t=0}^{T-1} \left[R\left(s, \boldsymbol{a}_t\right) + \gamma V_{\phi', f'_\phi}\left(\hat{o}_{t+1}^{i_m}\right) - V_{\phi, f_\phi}\left(\hat{o}_t^{i_m}\right) \right]^2 \tag{1}$$

where ϕ' is the target network, which is non-differentiable and updated every few epochs. Meanwhile, the decoder θ passes the embedding joint action to a sequence of decoding blocks. Crucially, the decoding block replaces the encoder's self-attention mechanism with a masked self-attention mechanism, i.e. the attention of the action to be generated in the current step is computed only among previously computed agents' actions. The output to the last decoder block is a sequence of representations of the joint actions. Same as the value head, this is fed into a policy head (also an MLP), denoted as f_θ, that outputs the policy $\pi_\theta^{i_m}\left(\mathrm{a}^{i_m} \mid \hat{\boldsymbol{o}}^{i_{1:n}}, \boldsymbol{a}^{i_{1:m-1}}\right)$. Besides, the decoder's learning objective is to minimize the clipping PPO objective proposed in HAPPO of

$$\mathcal{L}_{\theta, f_\theta}^{\text{MAT}_{Decoder}}(\boldsymbol{o}_t, \boldsymbol{a}_t) = -\frac{1}{Tn} \sum_{m=1}^{n} \sum_{t=0}^{T-1} \min\left(\mathrm{r}_t^{i_m}(\theta)\hat{A}_t, \operatorname{clip}\left(\mathrm{r}_t^{i_m}(\theta), 1 \pm \epsilon\right) \hat{A}_t\right) \tag{2}$$

where $\mathbf{r}_t^{i_m}(\theta) = \frac{\pi_\theta^{i_m}\left(a_t^{i_m} | \hat{\boldsymbol{o}}_t^{i_{1:n}}, \hat{\boldsymbol{a}}_t^{i_{1:m-1}}\right)}{\pi_{\theta_{\text{old}}}^{i_m}\left(a_t^{i_m} | \hat{\boldsymbol{o}}_t^{i_{1:n}}, \hat{\boldsymbol{a}}_t^{i_{1:m-1}}\right)}$ and $\hat{A}_t$ is an estimation of the joint advantage function, e.g. GAE(Schulman et al., 2015).

3 Our Method

3.1 Connection Between Joint Transition Model and SSL in MARL

We start with the intuition that encourages the observations and actions representations integrating other agents' information and embodying agent-level relationships and interactions should reduce the non-stationary during the learning process, thus improving the data efficiency of MARL. Inspired by recent works applying SSL in RL via a transition model, we exploit a multi-agent world model in latent space as the bridge to connect both observation representations and action representations of overall agents. Naturally, as described in the formulation of Dec-POMDP, one can feed the global state with joint actions into the world model $\mathcal{T}$ to predict the next state, i.e. implement $\mathcal{T}\left(s, \boldsymbol{a}, s'\right) = P\left(s' \mid s, \boldsymbol{a}\right)$. In this work, we learn a transformer-based joint transition model $\hat{\mathcal{T}}$ as an approximate version of the multi-agent world model, whose inputs are joint observation representations and joint action representations, and its outputs are the joint observation representations in the next timestep.

On the one hand, to achieve the goal of implementing information fusion among agents, it is better to use individual observations rather than a global state as the input since this way can execute the learned joint transition model to (a) reconstruct the global state from individual observations (b) predict the future state of next timestep (c) implement the observation mapping functions for each agent. In this way, the joint transition model has to infer the influences caused by others and attempt to integrate all imperfect information. As a result, enforcing consistency across different views of individual observations can lead to better representations generated from encoder networks.

On the other hand, it is necessary to emphasize that we **treat the individual observations as a sequence of masked contexts of global state** in the joint transition model. Based on the definition of Dec-POMDP, observations are the outputs of the mapping function, in other orders, individual observations can be seen as multiple views of the global state. This property revealed by Dec-POMDP's

[1] Because the algorithm makes no assumptions about the order of the agents and don't change the original order, we subsequently omit the superscript about the agents' order for the sake of simplicity of presentation.

formulation in cooperative MARL inspires us to propose a self-supervised prior tailored MARL. We utilize the joint transition model as a Masked Language Model (MLM, Devlin et al. (2018); Lewis et al. (2019)) under core components in MARL, which is general in all MARL settings.

3.2 Multi-Agent Joint-Predictive Representations

Multi-Agent Joint-Predictive Representations(MAJOR) is an auxiliary objective to promote the learned representations from the latent space of the sequential observation and action embeddings. The core process in the joint transition model of MAJOR is to **implicitly reconstruct the global state and then infer the future observation representation of each agent**. This mechanism enables the better use of agent-cross information when learning observation and action representations, further enhancing the understanding of MARL agents for individual messages. The overall framework of MAJOR is shown in Figure 1. We will introduce the components shown in the framework in the following subsections.

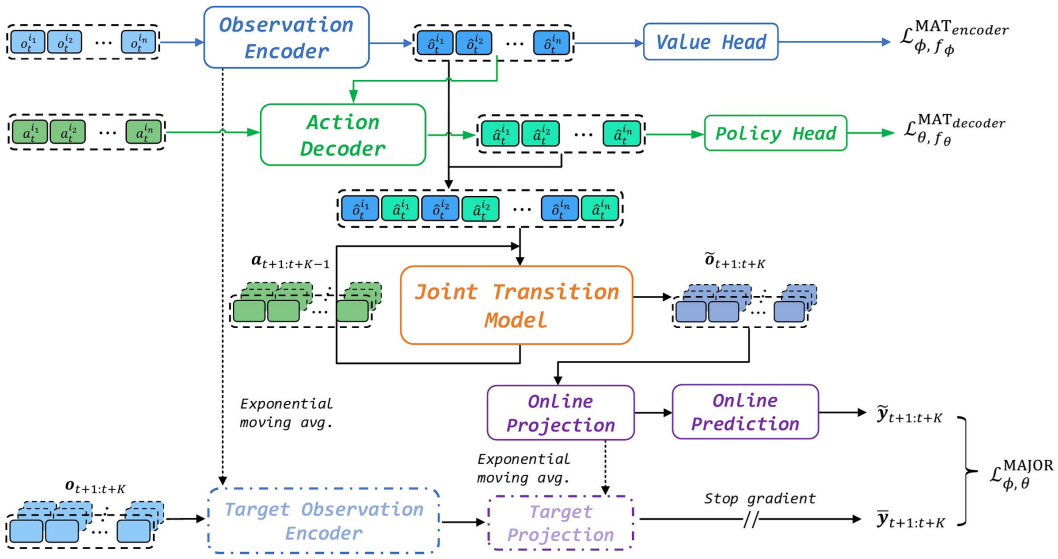


Figure 1: An illustration of the full MAJOR method. representations from the online observation encoder are used in the action decoder, value head, and policy head. They will also be utilized for the prediction of future representations from the target observation encoder via the joint transition model. The target observation encoder and projection head are defined as an exponential moving average of their online version and are not updated via gradient descent. For brevity, we illustrate the iterative K-steps results described in Equation 6 for the joint transition model totally in one loop.

Encoding observations and actions: We use MAT's encoder ϕ and decoder θ, excluding the final MLP individual, as the *online* observation encoder and action encoder, respectively. Taking the observations sequence of an arbitrary order $\boldsymbol{o}_t$ as input, the online observation encoder applies a self-attention mechanism and obtains post-interaction representations for agents, noted as $\hat{\boldsymbol{o}}_t$. Similar to the online observation encoder, the online action encoder accept both origin actions sequence $\boldsymbol{a}_t$ and observation representations $\boldsymbol{o}_t^{i_{1:n}}$ and output action representations $\hat{\boldsymbol{a}}_t$ via cross-attention mechanism. We employ these representations with a goal that motivates them to forecast future observation representations up to a given temporal offset K, iteratively. Following prior work (Schwarzer et al., 2021a; Zhu et al., 2022; Yu et al., 2021b; 2022), we utilize another observation encoder for the encoding of original observations. This *target* encoder has the same architecture as the online observation encoder, and its parameters are an exponential moving average (EMA) of the online observation encoder parameters. Denoting the target observation encoder as $\bar{\phi}$ and the momentum coefficient as $\tau \in [0, 1)$, the update scheme of the target observation encoder is:

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

Joint Transition Model: We construct the predictive version of future observation representations using a transformer-based joint transition model $\hat{\mathcal{T}}$. The architecture of the joint transition model is

similar to the online encoder but (a) contains L' Multi-Head Self-Attention (MHSA) layers without masks and (b) takes the combined sequence of both observations and action representations as the input tokens, and then it only outputs the sequence of observation representations of the subsequent timestep. We obtain the t-th observation and action representations by feeding the origin observation sequence and the action sequence into the online observation encoder and the action encoder. Then the input tokens of the latent joint transition model can be mathematically represented as :

$$\boldsymbol{x} = [\hat{o}_t^{i_1}, \hat{a}_t^{i_1}, \ldots, \hat{o}_t^{i_n}, \hat{a}_t^{i_n}] \tag{4}$$

The process of passing token sequence through the joint transition model's l-th layer can be mathematically described as:

$$\boldsymbol{h}^l = \text{MHSA}\left(\text{LN}\left(\boldsymbol{x}^l\right)\right) + \boldsymbol{x}^l, \quad \boldsymbol{x}^{l+1} = \text{FFN}\left(\text{LN}\left(\boldsymbol{h}^l\right)\right) + \boldsymbol{h}^l \tag{5}$$

Here LN and FFN denote LayerNorm and Feed-Forward Network mentioned in (Vaswani et al., 2017). Note that if the permutation order is known, one can also add agent ids' embedding and positional embedding on $\boldsymbol{x}$. And we only select the odd elements of the output tokens of the joint transition model as the corresponding predictive results for the latent future representations inferred from previous observation and action representations. Furthermore, in the k-th step of generating future representations where $k = 2, \ldots, K$, we use the joint transition model output representations instead of the online observation encoders as the input latent observation tokens. The process mentioned above can be denoted as:

$$\begin{aligned} &\tilde{\boldsymbol{o}}_{t+1} \triangleq \hat{\mathcal{T}}\left(\hat{\boldsymbol{o}}_t, \hat{\boldsymbol{a}}_t\right), \text{ where } \hat{\boldsymbol{o}}_t \triangleq \phi\left(\boldsymbol{o}_t\right), \text{ and } \hat{\boldsymbol{a}}_t \triangleq \theta\left(\hat{\boldsymbol{o}}_t, \boldsymbol{a}_t\right) \\ &\tilde{\boldsymbol{o}}_{t+k} \triangleq \hat{\mathcal{T}}\left(\tilde{\boldsymbol{o}}_{t+k-1}, \tilde{\boldsymbol{a}}_{t+k-1}\right), \text{ where } \tilde{\boldsymbol{a}}_{t+k-1} \triangleq \theta\left(\tilde{\boldsymbol{o}}_{t+k-1}, \boldsymbol{a}_{t+k-1}\right), \forall k = 2, \ldots, K \end{aligned} \tag{6}$$

The joint transition model and prediction loss operate in the latent space, thus, avoiding pixel-based reconstruction objectives and making MAJOR robust for vision-based and state-based MARL settings. In the following, we elaborate on the predictive loss between the prediction results and corresponding targets.

Prediction Loss: Motivated by the success of BYOL (Grill et al., 2020) in SSL and sample-efficient RL(Schwarzer et al., 2021a; Yu et al., 2021b; 2022), we compute the future prediction loss for MAJOR by summing over cosine similarities between the predicted and observed representations. Concretely, from the outputs of the joint transition model, i.e. the sequence of observation representations set $\tilde{\boldsymbol{o}}_{t+1:t+K}$, we use a projection head g and a prediction head q to obtain the final sequence of predictions result in $\tilde{\boldsymbol{y}}_{t+1:t+K} \triangleq q(g(\tilde{\boldsymbol{o}}_{t+1:t+K}))$. Then we utilize a target projection head $\bar{g}$ (i.e. follows the same EMA update strategy in the target observation encoder) to process the encoded results of original observations, which is denoted as $\bar{\boldsymbol{y}}_{t+1:t+K} \triangleq \bar{g}(\bar{\boldsymbol{o}}_{t+1:t+K})$ where $\bar{\boldsymbol{o}}_{t+1:t+K} \triangleq \bar{\phi}(\boldsymbol{o}_{t+1:t+K})$. Here, we apply a stop-gradient operation as illustrated in Figure 1 to avoid model collapse, following BYOL. Finally, the objective of MAJOR is to enforce the final prediction result $\tilde{\boldsymbol{y}}_{t+1:t+K}$ to be as close as possible to its corresponding target $\bar{\boldsymbol{y}}_{t+1:t+K}$. And we define the following mean squared error between the normalized predictions and target projections overall agents and offset timesteps as:

$$\mathcal{L}_{\phi,\theta}^{\text{MAJOR}}(\boldsymbol{o}_{t:t+K}, \boldsymbol{a}_{t:t+K-1}) = -\frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}\left(\frac{\tilde{y}_{t+k}^i}{\|\tilde{y}_{t+k}^i\|_2}\right)^\top\left(\frac{\bar{y}_{t+k}^i}{\|\bar{y}_{t+k}^i\|_2}\right) \tag{7}$$

Total learning objective: The proposed MAJOR is an auxiliary task, which is optimized together with MAT. So that the overall loss function for MARL agents training is:

$$\mathcal{L}_{total} = \mathcal{L}_{\phi,f_\phi}^{\text{MAT}_{encoder}} + \mathcal{L}_{\theta,f_\theta}^{\text{MAT}_{decoder}} + \lambda\mathcal{L}_{\phi,\theta}^{\text{MAJOR}} \tag{8}$$

where $\mathcal{L}_{\phi,f_\phi}^{\text{MAT}_{encoder}}$, $\mathcal{L}_{\theta,f_\theta}^{\text{MAT}_{decoder}}$ and $\mathcal{L}_{\phi,\theta}^{\text{MAJOR}}$ are the MARL loss functions of the based MAT algorithm mentioned before and our proposed joint-predictive representation learning objective, respectively. λ is a hyperparameter for balancing the items. It is worth noting that, unlike other suggested SSL algorithms in CV and RL, MAJOR can be employed with or without data augmentation, especially in situations when data augmentation is unavailable or counterproductive. Moreover, MAJOR mainly focuses on capturing the relationships among agents via the joint transition model. The proposed framework can also be transferred into other MARL algorithms that follow the Centralize Training Decentralized Execution(CTDE) paradigm, such as MAPPO(Yu et al., 2021a)/HAPPO(Kuba et al., 2022) and QMIX(Rashid et al., 2018)/QPLEX(Wang et al., 2020), etc. Here we provide the pseudo-code of MAJOR in Algorithm 1.

Algorithm 1 Multi-Agent Joint-Predictive Representations

Input: Stepsize α, number of agents n, episodes K, steps per episode T.
Initialize: Observation encoder ϕ, Action decoder θ, Value head f_ϕ, Policy head f_θ, Replay buffer $\mathcal{B}$, Joint transition model $\hat{\mathcal{T}}$, Online projection head g, Online prediction head q, Target observation Encoder $\bar{\phi}$, Target projection head $\bar{g}$.
while Training **do**
  Sample a minibatch $(\boldsymbol{o}_t, \boldsymbol{a}_t, r_t, \boldsymbol{o}_{t+1}) \sim \mathcal{B}$.
  Calculate $\mathcal{L}^{\text{MAT}_{encoder}}_{\phi, f_\phi}(\boldsymbol{o}_t)$ with Equation (1).
  Calculate $\mathcal{L}^{\text{MAT}_{Decoder}}_{\theta, f_\theta}(\hat{\boldsymbol{o}}_t, \boldsymbol{a}_t)$ with Equation (2).
  Sample another sequential minibatch $(\boldsymbol{o}_{t:t+K}, \boldsymbol{a}_{t:t+K-1}) \sim \mathcal{B}$.
  Get projection/prediction representations $\tilde{\boldsymbol{y}}_{t+1:t+K}$ and $\hat{\boldsymbol{y}}_{t+1:t+K}$ with Equation (6).
  Calculate $\mathcal{L}^{\text{MAJOR}}_{\phi,\theta}(\boldsymbol{o}_{t:t+K}, \boldsymbol{a}_{t:t+K-1})$ with Equation (7)
  Update the encoder/decoder, value/policy head, joint transition model, and online projection/prediction head by minimising $\mathcal{L}_{total}$ with gradient descent according to Equation (8).
  Update target observation encoder and target projection head with Equation (3)
end while

3.3 Implement details for MAJOR

During several training iterations of MAT, defined as a hyperparameter to control the update frequency with respect to MAT, we sample a unique batch of B' samples from the trajectories collected using the latest policy. Each sample consists of a temporal set of observations and corresponding actions, i.e. $(\boldsymbol{o}_{t:t+K}, \boldsymbol{a}_{t:t+K})$. For the projection and prediction head, we do not use BatchNorm Layer and replace ReLU with GELU activation units, which is different from what BYOL did. In addition, our joint transition model's architecture is the same as the encoder used in MAT. In vision-based settings, we use three convolutional layers with a ReLU layer after each convolutional layer, which is the same as DQN's, as the feature extractor in all algorithms. The full hyperparameter of MAJOR can be found in Appendix A.

4 Related Work

4.1 Overview of Representation Learning and Self-Supervised Learning

Self-supervised learning empowers us to exploit a variety of labels that come with the data for free. With self-supervised learning, we can utilize inexpensive unlabeled data and establish the learning objectives properly from designed pretexts to gain supervision from the data itself. SSL has been developed in CV and NLP areas and can be divided into the various self-supervised pretexts in the literature into four broad families(Ericsson et al., 2022): *Masked Prediction*, *Transformation Prediction*, *Instance Discrimination*, and *Clustering*. (1) Masked Prediction methods (Mikolov et al., 2013; Baevski et al., 2020; Pathak et al., 2016; Hu et al., 2020) mask a portion of word tokens or image pixels from the input sentence or image and train the model to predict the masked components to obtain effective representations. (2) Transformation Prediction methods (Gidaris et al., 2018; Sarkar & Etemad, 2020; Xu et al., 2019) apply a transformation that maps from canonical views to alternative views and trains the model to predict what transformation has been applied. (3)Instance Discrimination methods (Velickovic et al., 2019; Chen et al., 2020; He et al., 2019; Tian et al., 2020) apply some transformation process in one instance to obtain multiple views of it and attempt to formalize the contrastive instance discrimination. (4) And Clustering methods (Caron et al., 2018; 2020; Zhan et al., 2020; Alwassel et al., 2020) focus on dividing the training data into several groups with high intra-group similarity and low inter-group similarity. We recommend readers read (Ericsson et al., 2022) to get more information.

4.2 Self-Supervised Learning and Reinforcement Learning

There exist substantial works taking advantage of SSL techniques to promote representation learning in RL. A popular approach is to jointly learn policy learning objectives and auxiliary objectives. As for constructing auxiliary SSL objectives, the primary way is to build multiple views of the same

input through masked-latent reconstruction or dynamic models with augmentations. For instance, (Srinivas et al., 2020; Zhu et al., 2022) tried to extract high-level features from raw pixels using contrastive learning and performed off-policy control on top of the extracted features. (Schwarzer et al., 2021a; Yu et al., 2021b; 2022; Zhang et al., 2021) leverage a dynamic model to obtain a predicted version of the subsequent observation and then use contrastive learning to enforce consistency between the raw future observation and the prediction version of it in latent space. Another alternative way of obtaining good representations is to pre-train the observation encoder to learn effective representations before policy learning(Yarats et al., 2021; Stooke et al., 2021; Schwarzer et al., 2021b; Yang & Nachum, 2021; Campos et al., 2021).

4.3 Multi-Agent Reinforcement Learning and Representation Learning

Recent works (Yeh et al., 2019; Sun et al., 2019; Zhan et al., 2018; Yuan et al., 2021) study the representation learning for developing models that predict trajectories over all agents at once but not under the MARL context. And as far as we know, only a few works consider promoting representation in the MARL context. The most similar works are (Shang et al., 2021) and (Zhang et al., 2022). (Shang et al., 2021) task each agent to predict its future location, arriving at an agent-centric predictive objective to be combined in their proposed agent-centric attention module in the football game. (Zhang et al., 2022) is a model-based MARL method that proposed a graph-assisted predictive state representation learning framework that leverages the agent connectivity graphs to aggregate local representations computed by each agent. Note that the SSL prior proposed in (Shang et al., 2021) only be used in football-like environments and is not flexible. Additionally, our method aims to build a general plugin for model-free MARL approaches so that model-based MARL methods are not directly comparable to our method. We focus on the auxiliary task line in this work.

5 Experiments

We consider wide-rage MARL benchmarks for evaluating MAJOR and compared MARL algorithms, including vision-based and state-based MARL benchmarks. All hyperparameters settings can be found in Appendix C.

5.1 Setup

Vision-based MARL Environments. Representation Learning and SSL techniques show their strength in vision-based RL environments, such as DMControl and Atari. To evaluate whether our proposed MAJOR also is powerful in such vision-based MARL settings, we run it on three physics-based cooperative tasks in Multi-Agent Quadcopter Control(dubbed MAQC) (Panerati et al., 2021). MAQC is an open-source, OpenAI Gym-like multi-quadcopter simulator that provides vision-based observations and multi-agent controlling interfaces. The observations of drones include video frames in each drone's perspective (towards the positive direction of the local x-axis) for the RGB ($\in \mathbb{R}^{64\times48\times4}$), depth, and segmentation ($\in \mathbb{R}^{64\times48\times1}$) views. The action of drones is continuous velocity and the magnitude of the velocity.

Here we briefly introduce the three scenarios, named *Meetup*, *Flock*, and *LeaderFollower* in MAQC. Denote i-th agent's xyz coordinates as $\mathbf{x} = (x, y, z)$, individual reward as r_i, team reward is $R = \sum_{i=1}^{n} r_i$, and then: (i) The goal of *Meetup* is to make the distance between (i)-th and $(n-i)$-th drone as close as possible(here we assume that there are even number of drones in the simulator), i.e. $r_i = r_{n-i} = -||\mathbf{x}_i - \mathbf{x}_{n-i}||_2^2$, $\forall i = 1, \dots, n$. (ii) In *Flock* , the first agent should keep its position with a predefined location (e.g. $\mathbf{p}$) as close as possible and i-th agents($i > 1$) need to track $(i-1)$-th agent's latitude, i.e $r_1 = -||\mathbf{p} - \mathbf{x}_1||_2^2, \quad r_i = -(y_i - y_{i-1})^2 \ \forall i = 2, \dots, n$ (iii) The goal of *LeaderFollower* in MAQC is to train the *follower* drones to track the *leader* drone, and the leader drone needs to keep its position with a predefined position as close as possible, i.e. $r_1 = -||\mathbf{p} - \mathbf{x}_1||_2^2, \quad r_i = -\frac{1}{n}(z_i - z_1)^2 \ \forall i = 2, \dots, n$. An overview of MAQC is shown in Figure 2. Note that in our experiments, we only use the RGB information provided by the simulator.

State-based MARL Environments. Since our method can be used not only in vision-based but also in state-based scenarios in MARL environments, and we also test our method in state-based cooperative MARL scenarios. Our evaluation include StarCraftII Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019), Multi-Agent MuJoCo (de Witt et al., 2020), and Google Research

Figure 2: Overview of Multi-Agent Quadcopter Control environment and the observation for the quadrotor. A quadrotor is (i) an easy-to-understand mobile robot platform whose (ii) control can be framed as a continuous states and actions problem but, beyond 1-dimension, (iii) it adds the complexity that many candidate policies lead to unrecoverable states, violating the assumption of the existence of a stationary state distribution on the entailed Markov chain.

Football (Kurach et al., 2020), in where the compared baselines show the current state-of-the-art performance.

Baselines and evaluations. We select MAPPO, HAPPO, and MAT as the compared baselines because they are general to both continuous and discrete MARL environments. Moreover, they have reached state-of-the-art performance in common-used MARL benchmarks. We keep the hyperparameters proposed in their papers for a fair comparison of all algorithms. We run five random seeds to get the mean and standard deviation of the evaluation metrics. We calculate the winning rate in SMAC and the playing scores in Google Football. For other environments, we use episode rewards as the evaluation metric. Additionally, we only select three super-hard difficulty scenarios in the SMAC environment. And we play four drones in MAQC simulator to test the performance of candidates to be compared. As for Multi-Agent MuJoCo, we enforce agents can only observe the state of their own joints and bodies but not their immediate neighbor's joints and bodies.

5.2 RESULTS AND ANALYSIS

Multi-Agent Quadcopter Control. Figure 3 shows the performance of MAJOR, compared with strong baselines in the MAQC environment. The three tasks need drones(agents) to learn to fly to a specific goal and keep a lower distance from others by controlling the velocity vector. The results in Flock and LeaderFollwer demonstrates the strength of MAJOR since it conducts the joint transition model and implicitly captures the relationships of euclidean distance via the learned joint transition model so that for the goal of these two scenarios, being closed with other drones, MAJOR can inject more helpful information into networks thus bring improvement w.r.t other algorithms. On the other hand, for the failure case shown in Meetup, we speculate that the reason is that agents need to get others' IDs to track the specific drone as close as possible. Still, they cannot gain the corresponding messages from the RGB observation, which leads to no algorithm that can work on this task. In our opinion, there are still many challenges in applying MARL algorithms in vision-based environments, and our proposed representation learning framework shows its strength if the underlying approaches don't work at all.

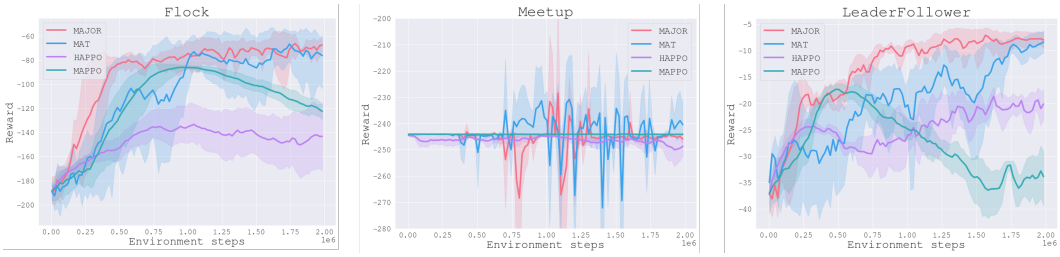
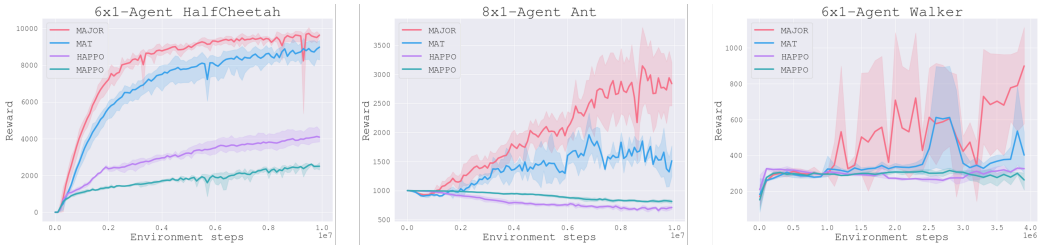


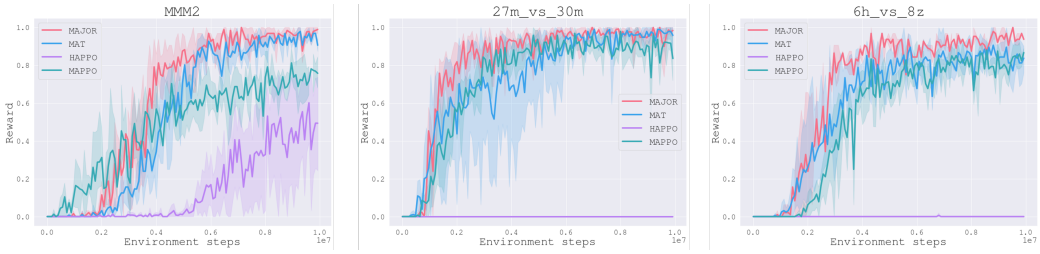
Figure 3: Results on Multi-Agent Quadcopter Control. MAJOR outperforms all prior methods on two of three tasks and all methods cannot work on the Meetup.

SMAC, MA MuJoCo and Google Football. We further compare MAJOR with the SOTA MARL methods for typical state-based cooperative MARL benchmarks. According to Figure 4b, Figure 4a, and Figure 4c, MAJOR also shows superior performance, meaning that even though the observations

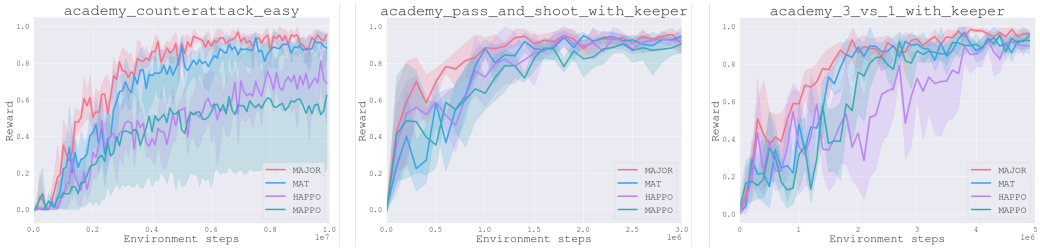
are non-visual signals, MAJOR still embodies the interaction and fusion between agents through the joint transition model in the latent space. This can be significantly shown in Multi-Agent MuJoCo. Agents in the robot have to attempt to infer the information of other joints and bodies to cooperate. Other compared methods only considered the feature encoding process from the perspective of policy optimization, but MAJOR can take into account other agents' position and velocity by the extra self-supervised representation learning task, which makes it natural to achieve better performance. Besides, our algorithm can also improve MAT's sample efficiency in Google Football and SMAC benchmarks.



(a) Multi-Agent MuJoCo



(b) StarCraftII Multi-Agent Challenge



(c) Google Research Football

Figure 4: Performance for MAJOR, MAT, HAPPO, and MAPPO on (a) Multi-Agent MuJoCo, (b) StarCraftII Multi-Agent Challenge, and (c) Google Research Football. MAT outperforms compared current state-of-the-art baselines in common cooperative MARL benchmarks.

6 Conclusion

In this paper, we introduce Multi-Agent Joint-Predictive Representations(MAJOR), a self-supervised representation learning algorithm designed to improve the data efficiency of MARL algorithms. MAJOR treats the individual observations as a masked sequence and learns the representations that are jointly temporally predictive and consistent across different views overall agents, by implicitly reconstructing the global state and directly predicting representations of observations produced by a joint transition model and a target encoder. Experimental results on both vision-based and state-based cooperative MARL benchmarks(i.e. Multi-agent Quadcopter Control, SMAC, Multi-agent MuJoCo, and Google Football) show MAJOR's state-of-the-art performance. Based on the established connection between MARL and SSL, in the future, we will consider incorporating cycle-consistent predictive or other powerful SSL priors into MAJOR and combining offline representation pretraining with task-specific finetuning in the paradigm.

REFERENCES

Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *Advances in Neural Information Processing Systems*, 33:9758–9770, 2020.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

Víctor Campos, Pablo Sprechmann, Steven Hansen, Andre Barreto, Steven Kapturowski, Alex Vitvitskyi, Adria Puigdomenech Badia, and Charles Blundell. Beyond fine-tuning: Transferring behavior in reinforcement learning. *arXiv preprint arXiv:2102.13515*, 2021.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Christian Schröder de Witt, Bei Peng, Pierre-Alexandre Kamienny, Philip H. S. Torr, Wendelin Böhmer, and Shimon Whiteson. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. *CoRR*, abs/2003.06709, 2020. URL https://arxiv.org/abs/2003.06709.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M Hospedales. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine*, 39(3):42–62, 2022.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

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.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=EcGGFkNTxdJ.

Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Google research football: A novel reinforcement learning environment. *ArXiv*, abs/1907.11180, 2020.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, pp. 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc.

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

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)*, 2021.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International conference on machine learning*, pp. 4295–4304. PMLR, 2018.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Pritam Sarkar and Ali Etemad. Self-supervised learning for ecg-based emotion recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3217–3221. IEEE, 2020.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=uCQfPZwRaUu.

Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R Devon Hjelm, Philip Bachman, and Aaron C Courville. Pretraining representations for data-efficient reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12686–12699, 2021b.

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

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *international conference on machine learning*, 2020.

Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International Conference on Machine Learning*, pp. 9870–9879. PMLR, 2021.

Chen Sun, Per Karlsson, Jiajun Wu, Joshua B Tenenbaum, and Kevin Murphy. Stochastic prediction of multi-agent interactions from partial observations. *arXiv preprint arXiv:1902.09641*, 2019.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6827–6839. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *ICLR (Poster)*, 2(3):4, 2019.

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

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.

Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10334–10343, 2019.

Mengjiao Yang and Ofir Nachum. Representation matters: offline pretraining for sequential decision making. In *International Conference on Machine Learning*, pp. 11784–11794. PMLR, 2021.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Reinforcement learning with prototypical representations. 2021.

Raymond A Yeh, Alexander G Schwing, Jonathan Huang, and Kevin Murphy. Diverse generation for multi-agent sports games. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4610–4619, 2019.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games, 2021a.

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, 2021b.

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

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generative multi-agent behavioral cloning. *arXiv*, 2018.

Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6688–6697, 2020.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

Zhi Zhang, Zhuoran Yang, Han Liu, Pratap Tokekar, and Furong Huang. Reinforcement learning under a multi-agent predictive state representation model: Method and theory. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PLDOnFoVm4.

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*, pp. 1–1, 2022. doi: 10.1109/TPAMI.2022.3176413.

A ADDITIONAL HYPER-PARAMETER IN MAJOR

Table 1: Hyperparameters used for MAJOR.

Hyperparameter	**Value**
Number of prediction steps K	1
Auxiliary batch size for MAJOR B'	128
Weight for MAJOR loss λ	1
Hidden units in projection/prediction head	512
Encoder MEA τ	0.01 SMAC and Google Football 0.05 MA MoJoCo and MAQC
EMA update frequency	1
Number of blocks for joint transition model	1
Number of heads for joint transition model	1

B DYNAMIC PROCESS AND SOURCE CODE OF MAJOR

Please refer to https://anonymous.4open.science/r/MAJOR

C HYPER-PARAMETER SETTINGS FOR EXPERIMENTS

During experiments, the implementations of baseline methods are consistent with their official repositories, all hyper-parameters left unchanged at the origin best-performing status. The hyper-parameters adopted for different algorithms and tasks are listed in Table 2-8. In particular, the *ppo epochs* and *ppo clip* across different SMAC scenarios are unified to 10 and 0.05 respectively.

Table 2: Common hyper-parameters used for MAJOR, MAT, MAPPO and HAPPO in the SMAC domain.

hyper-parameters	value	hyper-parameters	value	hyper-parameters	value
critic lr	5e-4	actor lr	5e-4	use gae	True
gain	0.01	optim eps	1e-5	batch size	3200
training threads	16	num mini-batch	1	rollout threads	32
entropy coef	0.01	max grad norm	10	episode length	100
optimizer	Adam	hidden layer dim	64	use huber loss	True

Table 3: Different hyper-parameters used for MAJOR and MAT in the SMAC domain.

Maps	ppo epochs	ppo clip	num blocks	num heads	stacked frames	steps	γ
27m vs 30m	5	0.2	1	1	1	1e7	0.99
MMM2	5	0.05	1	1	1	1e7	0.99
6h vs 8z	15	0.05	1	1	1	1e7	0.99

Table 4: Different hyper-parameters used for MAPPO and HAPPO in the SMAC domain.

Maps	ppo epochs	ppo clip	hidden leyer	stacked frames	network	steps	γ_{MAPPO}	γ_{HAPPO}
27m vs 30m	5	0.2	2	1	rnn	1e7	0.99	0.95
MMM2	5	0.2	2	1	rnn	1e7	0.99	0.95
6h vs 8z	5	0.2	2	1	mlp	1e7	0.99	0.95

Table 5: Common hyper-parameters used for all methods in the Multi-Agent MuJoCo and Multi-Agent Quadcopter Control domain.

hyper-parameters	value	hyper-parameters	value	hyper-parameters	value
gamma	0.99	steps	1e7	stacked frames	1
gain	0.01	optim eps	1e-5	batch size	4000
training threads	16	num mini-batch	40	rollout threads	40
entropy coef	0.001	max grad norm	0.5	episode length	100
optimizer	Adam	hidden layer dim	64	use huber loss	True

Table 6: Different hyper-parameter used in the Multi-Agent MuJoCo and Multi-Agent Quadcopter Control domain.

Maps	MAJOR	MAT	MAPPO	HAPPO
critic lr	5e-5	5e-5	5e-3	5e-3
actor lr	5e-5	5e-5	5e-6	5e-6
ppo epochs	10	10	5	5
ppo clip	0.05	0.05	0.2	0.2
num hidden layer	/	/	2	2
num blocks	1	1	/	/
num head	1	1	/	/

Table 7: Common hyper-parameters used for all methods in the Google Research Football domain.

hyper-parameters	value	hyper-parameters	value	hyper-parameters	value
critic lr	5e-4	actor lr	5e-4	gamma	0.99
gain	0.01	optim eps	1e-5	batch size	4000
training threads	16	num mini-batch	1	rollout threads	20
entropy coef	0.01	max grad norm	0.5	episode length	200
optimizer	Adam	hidden layer dim	64	stacked frames	1

Table 8: Different hyper-parameter used in the Google Research Football domain.

Maps	MAJOR	MAT	MAPPO	HAPPO
ppo epochs	10	10	5	5
ppo clip	0.05	0.05	0.2	0.2
num hidden layer	/	/	2	2
num blocks	1	1	/	/
num head	1	1	/	/