Acting Beyond Learning: Imagination-Assisted Decision-Making in the Visual-based Multi-Agent Cooperative Scenarios

Huanhuan Yang[1], Dianxi Shi[2*], Songchang Jin[2], Guojun Xie[3], Yang Chen[4,5], Chunping Qiu[2], Shaowu Yang[1]

[1] College of Computer, National University of Defense Technology, Changsha, China
[2] Academy of Military Sciences, Beijing, China
[3] Nanjing University of Aeronautics and Astronautics, Nanjing, China
[4] School of Computer Science, Peking University, Beijing, China
[5] Tianjin Artificial Intelligence Innovation Center, Tianjin, China
yanghh94@126.com, dxshi@nudt.edu.cn

Abstract

Learning optimal policies in multi-agent cooperative settings with visual observations is significant and challenging. Agents must first perform state representation learning for their image observations and then learn policies in the abstracted state space. Aiming at this problem, we propose a novel model-based MARL method named Contrastive Latent World for Policy Optimization (CLWPO). In CLWPO, we first design a state representation model to facilitate learning in the latent state space. With the support of this model, we construct the latent world and introduce a contrastive variational bound (CVB) to optimize it. Subsequently, we develop a heuristic policy optimization (HPO) scheme, incorporating model-free learning with model-based planning to obtain robust policies that predict future behaviors. In particular, in the planning, we maintain a queue of teammate models and calculate an adaptive rollout length for each agent to support their self-imagination and reduce the model-based return discrepancy. Finally, we conducted extensive experiments in the PettingZoo benchmark, and results show that CLWPO significantly enhances learning efficiency and improves agent performance compared to state-of-the-art MARL methods.

Introduction

Multi-agent Reinforcement Learning (MARL), where multiple agents interact with the environment through trial and error to learn efficient policies, recently has witnessed significant advancements in tackling complex multi-agent tasks, spanning domains like real-time strategy games (OpenAI 2018), sports games (Kurach et al. 2020), autonomous driving (Zhou et al. 2020), etc. There are mainly three learning paradigms in MARL: centralized learning (Ye et al. 2020), independent learning (de Witt et al. 2020), and centralized training with decentralized execution (CTDE) (Lowe et al. 2017). Among them, CTDE is a prevalent paradigm commonly used by researchers, utilizing global information for agents' training while local observations for decision-making. With CTDE, numerous model-free learning approaches have been developed (Sunehag et al. 2017; Rashid et al. 2018; Lowe et al. 2017). While they perform well in simulated environments (Lowe et al. 2017; Vinyals et al. 2017; Berner et al. 2019), one significant weakness is the lower sample efficiency. This implies that agents must collect substantial data to learn optimal policies. For instance, AlphaStar requires 44 days of cumulative training (200 years of gameplay) to defeat professional StarCraft players (Arulkumaran, Cully, and Togelius 2019), and OpenAI Five undergoes ten months of training (over 11,000 years of gameplay) to best Dota 2 world champions (OpenAI 2018).

Furthermore, the lower sample efficiency intensifies when agents face high-dimensional observation spaces, which not only slows down policy convergence speed but also significantly restricts the practical application of MARL. In real-world multi-agent systems, agents utilize multiple sensors to perceive the environment information described by complex features. To learn policies and make decisions more efficiently, they should first perform state representation learning (SRL) to map high-dimensional observations into the low-dimensional latent space. Then, these abstract representations can help agents better understand the inner structure of the environment and improve learning efficiency.

However, there has been limited exploration of multi-agent SRL, primarily due to some inherent challenges of MARL. The major is the non-stationarity (Hernandez-Leal et al. 2017). In MARL, multiple agents concurrently optimize their policies, resulting in an unstable environment from the perspective of each agent. The next is the curse of dimensionality (Hernandez-Leal, Kartal, and Taylor 2019), where the joint state-action space exponentially grows with increased agents. In addition, agents in MARL also face partial observability, coordination, and other challenges. The presence of these challenges exacerbates the optimal policies' learning difficulty. Therefore, how to characterize agents' high-dimensional observations and optimize their policies in visual-based multi-agent settings is a crucial issue that needs to be resolved and well-researched. Aiming at this problem, inspired by the superior performance of model-based reinforcement learning (MBRL) in single-agent domain (Wang et al. 2019; Sun et al. 2019), we focus on multi-agent cooperative tasks and propose the CLWPO method. CLWPO aims to learn optimal policies via a heuristic policy optimization (HPO) scheme in the latent world constructed

*Corresponding author.

by representation, transition, and reward models and optimized by the contrastive variational bound (CVB). Specifically, our main contributions are summarized below:

- To extract task-relevant information from visual observations during agents' interaction and policy learning processes, following characteristics of the CTDE paradigm, we design a representation model consisting of multiple agent representation modules and one state inference module to obtain an abstracted latent state space. Further, in this space, we construct the latent world model and optimize it with the CVB obtained by maximizing the data log-likelihood of joint observations and rewards.
- To make agents learn robust policies capable of predicting long-horizon behaviors, we develop an HPO scheme that integrates model-free learning with model-based planning. In the former, we optimize policies via real interaction data to reduce the negative impact of the inaccurate world model caused by learning bias. For planning, we maintain a queue of teammate models and calculate an adaptive rollout length for each agent to support their self-imagination in the learned latent world while lowering the upper bound of model-based return discrepancy.
- We conduct extensive experiments in the PettingZoo benchmark (Terry et al. 2021). Results show that CLWPO can efficiently represent observations, learns superior cooperative policies, performs better, and is applicable in visual-based settings with diverse agents.

Related Work

Single-agent State Representation

In the single-agent RL, the state representation learning methods have been well-studied. Current approaches primarily revolve around developing various self-supervised auxiliary tasks to train encoders capable of extracting low-dimensional and informative features from the agent's high-dimensional or pixel observations. These approaches include reconstructing the agent's original observations (Lange and Riedmiller 2010; Lange, Riedmiller, and Voigtländer 2012; Yarats et al. 2021), employing data augmentation with contrastive or multi-view learning to assess the similarity between states (Kim et al. 2019; Laskin, Srinivas, and Abbeel 2020; Mazoure et al. 2020), or capturing common and critical task-related information from disparate view data (Chen et al. 2017; Li et al. 2019; Fan and Li 2022; Yang et al. 2022), etc. In particular, the latent world (Oh, Singh, and Lee 2017; Hafner et al. 2019, 2020, 2021, 2023; Ma et al. 2021), a special case of model-based RL in the latent state space, represents a prominent approach in this research domain, where the agent initially performs state representation learning to derive an abstracted latent state space, followed by the construction and training of environment models within this space. Among these works, PlaNet (Hafner et al. 2019) enables the agent to learn environment dynamics from pixels and choose actions via online planning in the latent space. Dreamer (Hafner et al. 2020) and its variants (Hafner et al. 2021, 2023) learn policies that can solve long-horizon tasks with image observations via latent imagination in the compact state space.

Multi-agent State Representation

Although sufficient single-agent RL research provides valuable insights into multi-agent representation, there is still relatively little research. Both the attention mechanism (Iqbal and Sha 2019; Liu et al. 2020; Shi et al. 2022) and object-centric representation (Liu et al. 2021; Shang et al. 2021) can reduce dimensions of agents' (joint) observations by retaining essential information and inferring environmental internal structure. However, they perform poorly in partially observable or complex visual-based settings. Regarding the latent world-based methods, MBVD (Xu et al. 2022) integrates imaged latent states with current states when evaluating state values. DLC (Schwarting et al. 2021) learns visual control policies in competitive two-player racing games, while MAMBA (Egorov and Shpilman 2022) learns efficient policies in multi-agent cooperative scenarios. Although DLC and MAMBA performed well, there was still room for improvement. Firstly, they decomposed latent states into sub-states, utilizing all sub-states for dynamics learning but only individual sub-states for reward and observation functions learning, partially addressing the non-stationarity issue. Secondly, they relied on all sub-states for representation, making agents infer sub-states of other agents or communicate with others in the interaction, potentially introducing compounding errors or increasing communication complexity. Thirdly, they leave the multi-agent cooperative settings with image observations for future work.

Based on the above analysis, we present CLWPO, which follows the CTDE paradigm. We carefully design the representation model, using individual sub-states for decentralized execution to avoid extra errors or communication while inferring global states from all sub-states to predict transition dynamics and rewards. To learn robust policies that can predict the future, we design the HPO scheme, integrating model-free MARL learning with adaptive model-based planning. Importantly, CLWPO is versatile in our tested visual-based cooperative environments with varying agents.

Preliminaries

Partially Observable Stochastic Game

We consider the multi-agent cooperative scenarios, which can be defined as a partially observable stochastic game (POSG) (Hansen, Bernstein, and Zilberstein 2004) $\langle \mathbb{D}, S, \mathbf{A}, T, \mathbf{O}, \mathbb{O}, \mathbf{r}, n, \gamma, b_0, h \rangle$, where $\mathbb{D}$ is the set of n agents, S is the finite set of environment states. $\mathbf{O} = \times_{i \in \mathbb{D}} \mathbf{O}^i$ and $\mathbf{A} = \times_{i \in \mathbb{D}} \mathbf{A}^i$ are the sets of joint observations and actions. $T : S \times \mathbf{A} \to S$ is the state transition function, $\mathbf{r} : S \times \mathbf{A} \to \mathbb{R}$ is the reward function and $\mathbb{O} : S \times \mathbf{A} \to \mathbf{O}$ is the observation function. $\gamma \in [0, 1]$ is the discount factor, b_0 is the initial environment state distribution, and h is the finite task's horizon. In POSG, each agent's policy $\pi_i : \tau^i \to \mathbf{A}^i$, is conditioned on their Action-Observation History (AOH) $\tau^i = \{a_0^i, o_1^i, \ldots, a_{t-1}^i, o_t^i\}$. At each timestep t, each agent i observes o_t^i, executes action a_t^i, forming joint observation $\mathbf{o}_t = \langle o_t^1, o_t^2, \ldots, o_t^n \rangle$ and joint action $\mathbf{a}_t = \langle a_t^1, a_t^2, \ldots, a_t^n \rangle$. Given $\mathbf{r}, \gamma$ and h, agents aim to learn cooperative policies that maximize the expected cumulative discounted reward $R = \sum_{t=0}^{h} \sum_{i=1}^{n} \gamma^t r_i(s_t, \mathbf{a}_t)$.

It is important to emphasize that we assume an unknown environment in the paper, indicating that we have no prior knowledge regarding the transition function T, reward function $\mathbf{r}$, and observation function $\mathbb{O}$.

Variational Auto-Encoders

As one of the most influential techniques in unsupervised learning, variational auto-encoders (VAE) (Kingma and Welling 2013) have been widely used for image generation (Razavi, Van den Oord, and Vinyals 2019), representation learning (Ha and Schmidhuber 2018; Huang et al. 2020), etc. In VAE, given the dataset $\mathbf{X} = \{x_1, \ldots, x_N\}$ sampled from an unknown distribution $p(x)$, we want to learn a latent-variable model $p_\theta(x) = \int p_\theta(x, z)d_z = \int p_\theta(x|z)p(z)d_z$ to approximate $p(x)$. Typically, θ is optimized by maximizing the average marginal log-likelihood $\frac{1}{N}\log p(\mathbf{X})$. However, once θ is parameterized by a neural network, computing the log-likelihood $\log p(x_i)$ becomes intractable, introducing optimization challenges. Thus, VAE instead maximizes the following evidence lower bound (ELBO):

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}\Big[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}\Big] = \text{ELBO}$$
$$= \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x)||p(z)\right) \quad (1)$$

Where $D_{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence between two distributions. $p(z)$ is the prior. $q_\phi(z|x)$ is the variational posterior (encoder of VAE), responsible for generating continuous latent representations, $p_\theta(x|z)$ is the generative model (decoder of VAE), responsible for the reconstruction of the observed data, and ϕ, θ are their parameters.

Multi-Agent Soft Actor-Critic (MASAC)

The multi-agent soft actor-critic (MASAC) is an off-policy MARL algorithm that extends soft actor-critic (SAC) (Haarnoja et al. 2018) to multi-agent settings. In MASAC, each agent learns a decentralized stochastic policy π_{φ_i} to maximize a γ-discounted and maximum entropy-based return (Haarnoja et al. 2017):

$$\bigtriangledown_{\varphi_i} J = \mathbb{E}_{\mathbf{o}_t\sim\mathcal{B}, \mathbf{a}_t\sim\boldsymbol{\pi_\varphi}}\big[\bigtriangledown_{\varphi_i}\log(\pi_{\varphi_i}(a_t^i|o_t^i))\big(-\alpha_i$$
$$\log\pi_{\varphi_i}(a_t^i|o_t^i) + \min_{m=1,2} Q_{w_i}^m(\mathbf{o}_t, \mathbf{a}_t)\big)\big] \quad (2)$$

The centralized critics of agent i are trained to minimize the following Bellman error:

$$L_{Q_{w_i}^m} = \mathbb{E}_{(\mathbf{o}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{o}_{t+1})\sim\mathcal{B}}\big[\big(Q_{w_i}^m(\mathbf{o}_t, \mathbf{a}_t) - y_i\big)^2\big] \quad (3)$$

where y_i is the target value, defined as:

$$y_i = r_t^i + \gamma\mathbb{E}_{\mathbf{a}_{t+1}\sim\boldsymbol{\pi_\varphi}}\big[\min_{m=1,2}\bar{Q}_{\bar{w}_i}^m(\mathbf{o}_{t+1}, \mathbf{a}_{t+1}) - \alpha_i$$
$$\log\pi_{\varphi_i}(a_{t+1}^i|o_{t+1}^i)\big] \quad (4)$$

In Eq. (2) - (4), $\mathcal{B}$ is the replay buffer, storing transition data $(\mathbf{o}_t, \mathbf{a}_t, \mathbf{r}_t, \mathbf{o}_{t+1})$. $Q_{w_i}^m$ and $\bar{Q}_{\bar{w}_i}^m$ $(m = 1, 2)$ are two critics and target critics of agent i. Parameters w_i, $\bar{w}_i$, and φ_i correspond to the critics, target critics, and actors. $\bar{w}_i$ is softly updated based on w_i, defined as $\bar{w}_i \leftarrow \zeta_w \cdot w_i + (1 - \zeta_w) \cdot \bar{w}_i$, where ζ_w is a hyper-parameter that controls the updating rate. Given the target entropy $\mathcal{H}_i$ of agent i's policy distribution, temperature parameter α_i is updated by:

$$L_{\alpha_i} = \mathbb{E}_{a_t^i\sim\pi_{\varphi_i}}[-\alpha_i\log\pi_{\varphi_i}(a_t^i|o_t^i) - \alpha_i\mathcal{H}_i] \quad (5)$$

Method

In this section, we present our proposed Contrastive Latent World for Policy Optimization (CLWPO), a model-based method that learns efficient policies in multi-agent cooperative scenarios with image observations. In CLWPO, we first design the representation model, construct the latent world, and formulate the CVB objective to optimize the world. Next, we develop the HPO scheme to learn robust policies that can predict the future. In particular, we provide the detailed procedure of CLWPO in Appendix D.

CLWPO Framework

In CLWPO, to enable fast trajectory prediction in the compact latent state space, we fully leverage the CTDE paradigm and carefully design the representation model as multiple agent representation modules and one state inference module. Based on this model, we illustrate the overall framework of CLWPO in Fig. 1, with three parts below. **(1) Environment interaction.** Agents utilize their trained (or randomly initialized) agent representation modules and policies to encode image observations, select actions, interact with the environment, and then collect and store the transition data into the replay buffer to subsequently update relevant models. **(2) Latent World Learning.** On the foundation of the representation model, we construct environment models—including transition and reward models—that, together with the representation model, constitute the latent world and can be optimized through the CVB objective. **(3) Policy Optimization.** In CLWPO, we develop the HPO scheme, which integrates model-free learning with model-based planning to learn robust policies that predict long-horizon behaviors. As part of the planning, we maintain a queue of teammate models and calculate an adaptive rollout length h_i for each agent to effectively reduce the model-based return discrepancy bound when self-imagination (planning) in the latent world.

Latent World Learning

Representation Model. In the multi-agent domain, we can naively maintain a centralized representation model — mapping joint observation $\mathbf{o}_t$ into global latent state s_t (Schwarting et al. 2021; Egorov and Shpilman 2022). However, it is intractable in the CTDE paradigm, as agents can only access their local observations during the decentralized execution. Although information (like observation) transmission and agent modeling are two general practices for this challenge, they both have limitations. The former yields communication complexity, while the latter introduces compounding errors in environment interaction. Alternatively, another way is to maintain multiple decentralized representation models to map individual observation o_t^i into local latent state s_t^i. Unlike the centralized model, models in this way are more flexible. However, they potentially suffer learning instability as the dynamics, observation, and reward models occur in the local latent state space.

Based on these analyses, in this paper, we consider the characteristics of the CTDE paradigm, defining the representation model as multiple (n) agent representation modules and one shared state inference module, as shown in Appendix B and Eq.(6). During decentralized execution, each

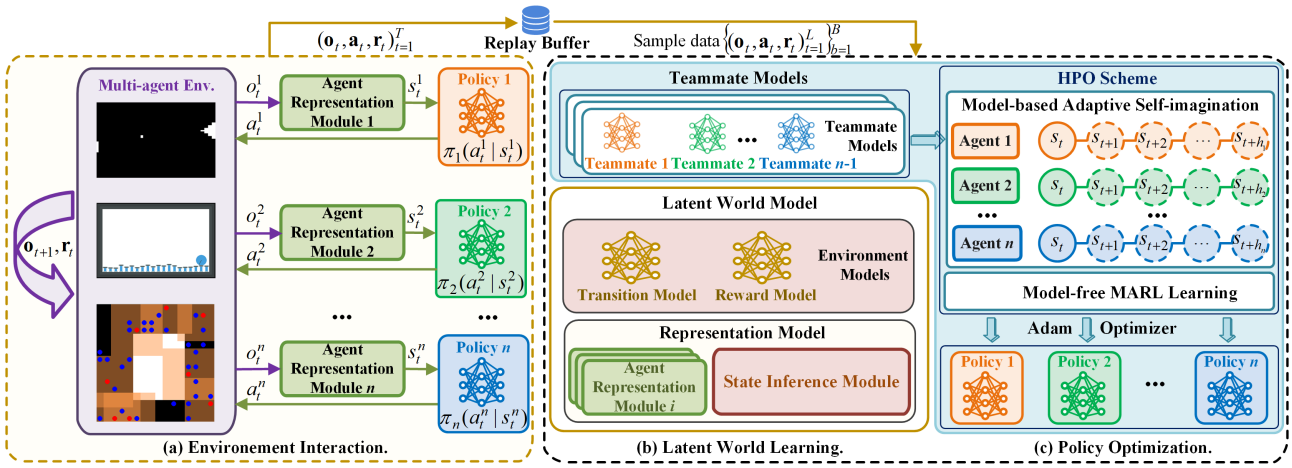


Figure 1: CLWPO framework. It mainly contains three parts: (a) environment interaction, (b) latent world learning, and (c) policy optimization. In (a), agents interact with the environment to store the transition data in the replay buffer. In (b), the built latent world consisting of the representation and environment models is learned via the CVB objective. In (c), agents optimize their policies via the HPO scheme, which integrates model-free MARL learning with model-based adaptive self-imagination.

agent utilizes its agent representation module to transform complex observation o_t^i into local latent state s_t^i. Then, they use the shared state inference module to infer global posterior state s_t from $\mathbf{s}_t^i = \{s_t^1, \cdots, s_t^i, \cdots, s_t^n\}$ for centralized training. Specifically, each agent representation module consists of three key components: an encoder, a GRU network, and an MLP. The encoder maps o_t^i into a low-dimensional embedding e_t^i, and we implement it as CNN in the visual-based tasks. By taking the previously hidden information h_{t-1}^i, local latent state s_{t-1}^i and action a_{t-1}^i as input, the GRU network outputs current hidden information h_t^i, which incorporates historical information into the state representation process. After that, the MLP fuses h_t^i and e_t^i as a local latent state s_t^i. The state inference module, shared among all agents, is defined as a standard Variational Auto-encoder (VAE) (Kingma and Welling 2013), consisting of a variational encoder and decoder. The former infers s_t from $\mathbf{s}_t^i$, while the latter reconstructs $\mathbf{s}_t^i$ based on s_t.

Latent World Model. Upon the representation model, we define two environment models: transition $p_\theta(s_t|s_{t-1}, \mathbf{a}_{t-1})$ and reward $p_\theta(\mathbf{r}_t|s_t)$, as in Eq. (6). In the equation, $p_\theta(\cdot)$ and $q_\theta(\cdot)$ are distributions in the latent state space, with θ denoting the combined parameter vector. The representation model $q_\theta(s_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})$, contains four components: f_{1_θ}, f_{2_θ}, f_{3_θ}, and f_{4_θ}, corresponding to the encoder, GRU, MLP, and VAE illustrated in the above and Appendix B.

$$
\begin{cases}
\text{Representation Model: } s_t \sim q_\theta(s_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t}) \\
\quad \hookrightarrow \begin{cases}
\text{Encoder: } e_t^i \sim f_{1_\theta}(e_t^i|o_t^i) \\
\text{GRU: } \quad h_t^i \sim f_{2_\theta}(h_t^i|h_{t-1}^i, s_{t-1}^i, a_{t-1}^i) \\
\text{MLP: } \quad s_t^i \sim f_{3_\theta}(s_t^i|h_t^i, e_t^i) \\
\text{VAE: } \quad s_t \sim f_{4_\theta}(s_t|s_t^1, s_t^2, \cdots, s_t^n)
\end{cases} \\
\text{Environment Models:} \\
\quad \hookrightarrow \begin{cases}
\text{Transition: } s_t \sim p_\theta(s_t|s_{t-1}, \mathbf{a}_{t-1}) \\
\text{Reward: } \quad \mathbf{r}_t \sim p_\theta(\mathbf{r}_t|s_t)
\end{cases}
\end{cases}
\tag{6}
$$

Learning of the latent world model. In MBRL, incorporating the observation model into the learned world (Schwarting et al. 2021; Egorov and Shpilman 2022) results in the reconstruction of the observation space and inevitably encodes task-irrelevant information into the latent states. This, in turn, hinders agents from obtaining an accurate latent world and further increases learning instability. Aiming for this challenge, we introduce an optimization objective called the contrastive variational bound (CVB), benefiting from the potential of contrastive learning in representation learning (Laskin, Srinivas, and Abbeel 2020). To derive CVB, we maximize the data log-likelihood of joint observations and rewards for the sequential trajectory data $\{\mathbf{o}_{1:T}, \mathbf{a}_{1:T}, \mathbf{r}_{1:T}\}$. Then, by leveraging the importance weighting, Jensen's inequality, and contrastive learning techniques, we obtain:

$$
\begin{aligned}
&\ln p(\mathbf{o}_{1:T}, \mathbf{r}_{1:T}|\mathbf{a}_{1:T}) \geq \sum_{t=1}^{T} \Big(\\
&\mathbb{E}_{q(s_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t})} \Big(\underbrace{\ln p(s_t|\mathbf{o}_t) - \textstyle\sum_{\mathbf{o}_t'} p(s_t|\mathbf{o}_t')}_{\text{contrastive}} + \underbrace{\ln p(\mathbf{r}_t|s_t)}_{\text{reconstruction}} \Big) - \\
&\underbrace{\mathbb{E}_{q(s_{t-1}|\mathbf{o}_{\leq t-1}, \mathbf{a}_{<t-1})} D_{KL}\big(q(s_t|\mathbf{o}_{\leq t}, \mathbf{a}_{<t}) || p(s_t|s_{t-1}, \mathbf{a}_{t-1})\big)}_{\text{transition}} \\
&\underbrace{- D_{KL}\left(q\left(s_t|\mathbf{s}_t^{\mathbf{i}}\right) || p(s_t)\right) + \mathbb{E}_{q(s_t|\mathbf{s}_t^{\mathbf{i}})} \ln p\left(\mathbf{s}_t^{\mathbf{i}}|s_t\right)}_{\text{VAE}} \Big)
\end{aligned}
\tag{7}
$$

Where $p(s_t|\mathbf{o}_t)$ is the state model[1]. Note that we use the InfoNCE contrastive learning loss (Poole et al. 2019) to avoid the reconstruction of complex observations. $q(s_t|\mathbf{s}_t^{\mathbf{i}})$ and $p(\mathbf{s}_t^{\mathbf{i}}|s_t)$ are the variational encoder and decoder of the state inference module in the representation model, respectively. $p(s_t) \sim \mathcal{N}(0, I)$ is the variational distribution. Detailed derivations of Eq. (7) are given in Appendix A.

[1]As the state model is merely used in the latent world learning process, we thus omit it in the latent world.

Policy Optimization

Adaptive Self-imagination. In the planning, to allow agents to optimize their policies via self-imagination in the latent world, we first maintain a queue of $n-1$ teammate models $\hat{\boldsymbol{\pi}}_{\phi_{-i}}(\hat{\mathbf{a}}_t^{-i}|\mathbf{s}_t^{-i})$ for each agent to infer behaviors of other agents. We consider both discrete and continuous action cases. For discrete actions, we utilize the cross-entropy loss:

$$L_{\hat{\boldsymbol{\pi}}_{\phi_{-i}}} = -\mathbb{E}_{\mathbf{s}_t^{-i}\sim\mathcal{B}}\left[\log \hat{\boldsymbol{\pi}}_{\phi_{-i}}(\hat{\mathbf{a}}_t^{-i}|\mathbf{s}_t^{-i})\right] \tag{8}$$

For tasks with continuous action spaces, we adopt the following smooth-L1 (Huber) loss:

$$L_{\hat{\boldsymbol{\pi}}_{\phi_{-i}}} = \begin{cases} 0.5(\mathbf{a}_t^{-i} - \hat{\mathbf{a}}_t^{-i})^2, & |\mathbf{a}_t^{-i} - \hat{\mathbf{a}}_t^{-i}| < 1 \\ |\mathbf{a}_t^{-i} - \hat{\mathbf{a}}_t^{-i}| - 0.5, & otherwise \end{cases} \tag{9}$$

Where $\mathbf{a}_t^{-i}$, $\hat{\mathbf{a}}_t^{-i}$, and $\mathbf{s}_t^{-i}$ are the actual actions, predicted actions, and local latent states of teammate agents.

Then, corresponding to $\hat{\boldsymbol{\pi}}_{\phi_{-i}}(\hat{\mathbf{a}}_t^{-i}|\mathbf{s}_t^{-i})$, it is crucial to determine a suitable rollout length associated with the theoretical discrepancy between expected returns in the real and learned environment that assesses how well the learned world impacts an agent's performance compared to the actual environment. Note that agents in CLWPO share environment models but own separate teammate models. Thus, from the perspective of agent i, its return discrepancy is:

Proposition 1. *Assume that the expected total variation distance between the learned transition model and real transition model at each timestep t is bounded by* $\max_t \mathbb{E}_{(s_t,a_t^i,\mathbf{a}_t^{-i})\sim\pi_{D_i},\boldsymbol{\pi}_{D_{-i}}}[D_{TV}(T(\cdot|s_t,a_t^i,\mathbf{a}_t^{-i})\|p_\theta(\cdot|s_t,a_t^i,\mathbf{a}_t^{-i}))] \leq \varepsilon'_m$*, prediction errors of teammate models are bounded as* $\max_{s_t^j} D_{TV}(\pi_j(\cdot|s_t^j)\|\hat{\pi}_j(\cdot|s_t^j)) \leq \epsilon_{\hat{\pi}}^j$*, and policies' divergence are bounded as* $\max_{s_t^i} D_{TV}(\pi_i(\cdot|s_t^i)\|\pi_{D_i}(\cdot|s_t^i)) \leq \epsilon_\pi^i, \max_{s_t^j} D_{TV}(\pi_j(\cdot|s_t^j)\|\pi_{D_j}(\cdot|s_t^j)) \leq \epsilon_\pi^j$*, where $j \in \{-i\}$, subscript D identify data collecting policies, and $r_{\max}^i = \max r_i(s_t,\mathbf{a}_t)$. Then the discrepancy bound of return in the real environment $\eta_1^i = \eta_i[\pi_i,\boldsymbol{\pi}_{-i}]$ and in the learned world (using the learned transition model and teammate models) with k-branched rollout $\eta_2^i = \eta_i^{branch}[(\pi_{D_1},\hat{\pi}_1),\ldots,(\pi_{D_i},\pi_i),\ldots,(\pi_{D_n},\hat{\pi}_n)]$ is expressed as:*

$$|\,\eta_1^i - \eta_2^i\,| \leq 2r_{\max}^i \Bigg[\underbrace{k\epsilon'_m + (k+1)\sum_{j\in\{-i\}}\epsilon_{\hat{\pi}}^j}_{\textit{model prediction error}} + \underbrace{\gamma^{k+1}\left(\epsilon_\pi^i + \textstyle\sum_{j\in\{-i\}}\epsilon_\pi^j\right) + \frac{\gamma^{k+1}\left(\epsilon_\pi^i + \sum_{j\in\{-i\}}\epsilon_\pi^j\right)}{1-\gamma}}_{\textit{policy distribution shift}} \Bigg]$$

Proof. See *Theorem 2* in Zhang et al. (2021). □

In Proposition 1, the return discrepancy is highly related to the prediction errors of teammate models. To reduce the bound, in Eq. (10), we calculate an adaptive branched (rollout) length h to replace the fixed k. Specifically, we multiply k by an adaptive weight proportional to the minimum and maximum prediction errors of teammate models to obtain h. Our intention is straightforward: to select an appropriate length that fully leverages teammate models and minimizes compounding errors negatively impacting performance.

$$h = k * \left\lfloor \frac{\min_{j\in\{-i\}}\epsilon_{\hat{\pi}}^j}{\max_{j\in\{-i\}}\epsilon_{\hat{\pi}}^j} \right\rfloor \tag{10}$$

With h, each agent i can adaptively perform h times self-imagination in the latent world, with the specific process in Appendix C. Next, we backpropagate the analytic gradient of estimated values along imagined trajectories to update the actor and critic of agent i, as shown in Eq.(11) and Eq.(12).

$$L_{\varphi_i} = \max{}_{\varphi_i}\mathbb{E}_{p_\theta,\pi_{\varphi_i}}\Big(\sum_{\tau=t}^{t+h} V_\lambda(s_\tau)\Big) \tag{11}$$

$$L_{\psi_i} = \min{}_{\psi_i}\mathbb{E}_{p_\theta,\pi_{\varphi_i}}\Big(\sum_{\tau=t}^{t+h}\|v_{\psi_i}(s_\tau) - V_\lambda(s_\tau)\|^2\Big) \tag{12}$$

Where $V_\lambda(s_\tau)$ is an exponentially-weighted average of the l-step value estimates $V_M^l(s_\tau)$, $v_{\psi_i}(\cdot)$ is the state value function. Given $h_0 = \min(\tau+l, t+h)$, we have:

$$V_\lambda(s_\tau) = (1-\lambda)\sum_{m=1}^{h-1}\lambda^{m-1}V_M^m(s_\tau) + \lambda^{h-1}V_M^h(s_\tau)$$

$$V_M^l(s_\tau) = \mathbb{E}_{p_\theta,\pi_{\varphi_i}}\Big(\sum_{m=\tau}^{h_0-1}\gamma^{m-\tau}r_m^i + \gamma^{h_0-\tau}v_{\psi_i}(s_{h_0})\Big)$$

HPO scheme. Although agents can learn to predict the future through planning (self-imagining), the effectiveness of the learned policies heavily relies on the accuracy of the latent world model. When the model is inaccurate, planning within this world will significantly slow down the learning process. Notably, even with our well-designed representation model and CVB objective, this issue is still troubling in MARL settings with pixel observations. We thus develop an HPO scheme, incorporating model-free MARL learning into policy optimization and combining it with model-based planning. In this way, we can obtain an auxiliary training signal that corrects deviations caused by planning via the real interaction trajectory data. Specifically, in HPO, agents first weight the MASAC loss in Eq. (2) by a hyperparameter α_{HPO} and use this weighted loss to optimize their policies. Then, they perform model-based self-imagination based on these optimized policies via the loss in Eq. (11) to further update policies, resulting in the final HPO objective: $L_{final} = \alpha_{HPO} * L_{MASAC} + L_{self-imagination} = \alpha_{HPO} *$ Eq. (2) + Eq. (11), which allows us to learn robust, optimal policies that can predict long-horizon behaviors.

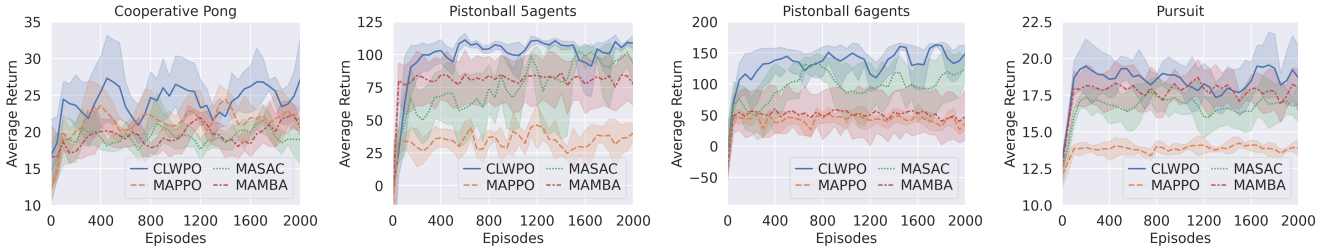
Experiments

Experimental Setup

In this paper, we use the PettingZoo (Terry et al. 2021) to evaluate CLWPO. While it consists of multiple environment classes, we focus on visual-based tasks in the Butterfly and SISL, i.e., Cooperative Pong, Pistonball 5agents & 6agents, and Pursuit, with further details given in Appendix

2K Episode Steps	CLWPO	MAPPO	MASAC	MAMBA
Cooperative Pong (↑)	**68.32 ± 12.96**	61.03 ± 10.87	60.44 ± 8.81	61.30 ± 7.15
Pistonball 5agents (↓)	**31.00 ± 4.52**	83.61 ± 8.37	49.33 ± 20.98	65.87 ± 28.15
Pistonball 6agents (↓)	**50.16 ± 18.46**	101.78 ± 4.01	68.59 ± 13.93	91.90 ± 16.82
Pursuit (↓)	500	500	500	500
1K Episode Steps	CLWPO	MAPPO	MASAC	MAMBA
Cooperative Pong (↑)	**68.66 ± 10.73**	59.05 ± 4.82	59.24 ± 7.01	58.15 ± 10.34
Pistonball 5agents (↓)	**42.47 ± 17.37**	79.57 ± 6.78	54.16 ± 9.59	62.78 ± 30.90
Pistonball 6agents (↓)	**55.93 ± 22.75**	93.45 ± 5.37	82.96 ± 29.90	85.61 ± 15.07
Pursuit (↓)	500	500	500	500

Table 1: Average episode steps (mean and standard deviation) on 1K and 2K episodes of 4 tasks.



(a) Cooperative Pong (b) Pistonball 5agents (c) Pistonball 6agents (d) Pursuit

Figure 2: Average Returns of CLWPO over five seeds with mean and standard error in 4 tasks of the PettingZoo benchmark.

E. We base our CLWPO on the implementation of MAMBA (Egorov and Shpilman 2022) and run 2000 episodes for each experiment on a desktop with an 8-core CPU and two Nvidia GeForce RTX 3090. Throughout the paper, unless otherwise stated, we present the result as the mean and standard error across five different random seeds in the table and figures. Other implementation details are in Appendix F.

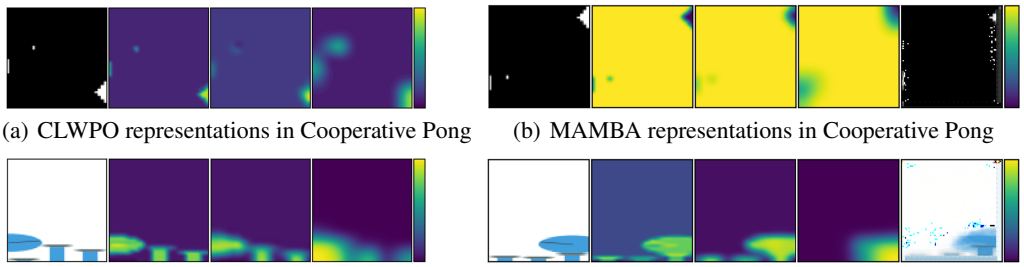
We compare CLWPO with the following state-of-the-art methods and our ablations. MAMBA (Egorov and Shpilman 2022) updated the latent world model based on a self-defined communication protocol. MASAC follows the idea of SAC to learn policies. MAPPO (Yu et al. 2022) extends PPO (Schulman et al. 2017) to multi-agent settings. CLWPO-rec_obs, CLWPO-fixed_roll, and CLWPO-image are variants of CLWPO, which replace the contrastive learning with the reconstruction of agents' original observations in the CVB objective, the adaptive rollout length h with a fixed k, and solely retains model-based adaptive planning (self-imagination) for policy learning, respectively.

Main Results

We first evaluate CLWPO in the four above tasks, with learning curves in Fig. 2(a) - 2(d). Overall, CLWPO can scale to tasks with varying agents and significantly outperform baselines in Cooperative Pong and Pistonball tasks while performing comparably in the Pursuit task. This result is consistent with the data listed in Table 1, where the CLWPO agents stay longer in the screen and use fewer timesteps to move the ball to the left wall in the Cooperative Pong and Pistonball tasks. We attribute this to the CVB and HPO, where the former urges agents to learn an accurate latent world via efficient state representation of image observations, which, in turn, empowers the latter to optimize policies better. Notably, in the Pursuit task, we configured the number of pursuers to 6, evaders to 20, and pursuers that catch an evader to 3, significantly increasing the task difficulty. Furthermore, agents only receive sparse rewards when they encounter or catch the prey, which slows down policy learning efficiency, thus making all methods fail to capture all evaders before the 500-step limit and the improvement of CLWPO not obvious.

Then, we visualize the representations learned from pixel observations by CLWPO and MAMBA in Fig. 3, along with the reconstructed images of MAMBA agents. The figure shows that the CLWPO encoder, trained with the CVB objective, can effectively preserve task-relevant features, filter out irrelevant details, and tend to focus on the paddle agents, neighboring piston agents, and the balls. In contrast, the MAMBA encoder fails to capture crucial information about the paddle and adjacent piston agents, simultaneously introducing additional noise into the reconstructed images. These figures further highlight the limitations of reconstruction-based state representation learning methods, which prompt agents to retain task-irrelevant information, especially some scene details, in the representations of image observations.

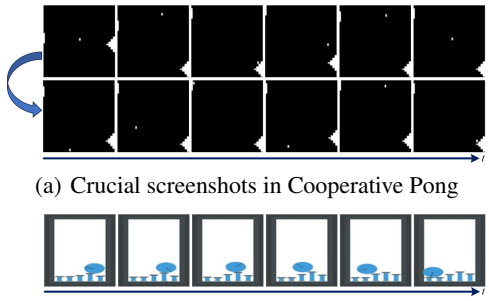
Next, we provide a detailed view of the critical actions executed by CLWPO agents in the Cooperative Pong and Pistonball 5agents tasks to show the learned policies' effectiveness in tackling tasks collaboratively in Fig. 4. In Fig. 4(a), both the paddle agents learn to predict the future movements of the ball and thus take appropriate actions to place themselves in locations where the ball is likely to fall, further increasing hitting chances. Likewise, in Fig. 4(b), the piston agents learn to coordinate their movements to sequentially roll the ball from the right edge of the screen to the left.



(a) CLWPO representations in Cooperative Pong (b) MAMBA representations in Cooperative Pong

(c) CLWPO representations in Pistonball 5agents (d) MAMBA representations in Pistonball 5agents

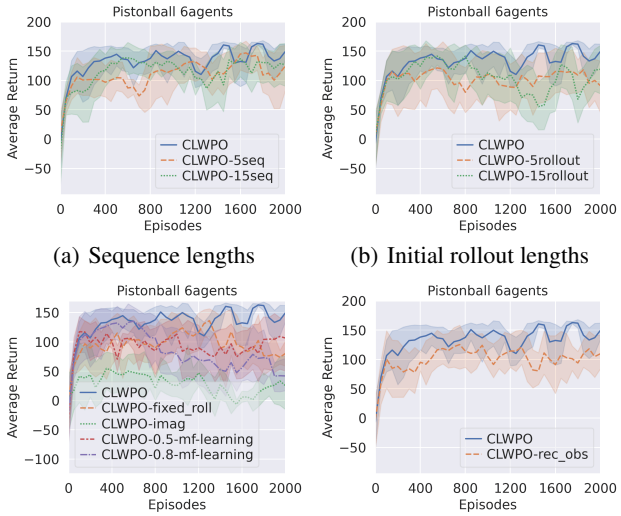
Figure 3: The spatial attention maps for each convolutional layer of the CLWPO (left) and MAMBA (right) agents' encoders at a random intermediate episode around 1500 in the Cooperative Pong and Pistonball 5agents tasks.



(a) Crucial screenshots in Cooperative Pong

(b) Crucial screenshots in Pistonball 5agents

Figure 4: Critical action sequences of CLWPO agents in (a) Cooperative Pong and (b) Pistonball 5agents tasks.



(a) Sequence lengths (b) Initial rollout lengths

(c) Policy learning (d) Representation learning

Figure 5: Results for ablation studies. (a)-(d) compares sequence lengths, initial rollout lengths k, policy learning and representation learning methods, respectively.

Ablation Studies

Finally, in Fig. 5, we investigate how CLWPO is affected by sequence lengths, initial rollout lengths k, policy learning methods, and representation learning methods in the Pistonball 6agents task, with additional results and analysis in Appendix G. Specifically, Fig. 5(a) indicates that the CLWPO agents trained with an appropriate sequence length can learn predictive representations that capture the RL temporal structure well. Fig. 5(b) and Fig. 5(c) demonstrate that the adaptive rollout length h derived from a well-chosen initial length k efficiently balances the utilization of learned models and maintaining of policy performance. Furthermore, introducing auxiliary training signals of ground-truth interaction data generated through model-free MARL learning and setting α_{HPO} to one to ensure that both the model-free learning and model-based planning impact the training equally can help CLWPO agents acquire robust policies that can predict long-horizon behavior. Fig. 5(d) reveals that compared with reconstructing agents' observation space, the contrastive learning-based CVB objective in CLWPO learns more efficient representation and thus performs better.

Conclusions

In this paper, we introduce CLWPO, a multi-agent model-based method for cooperative tasks with visual observations. CLWPO first designs a representation model to represent complex observations and facilitate model learning in the latent state space. Upon this model, it constructs a latent world and derives the CVB objective to optimize the world. Then, it develops the HPO scheme, combining model-free learning with model-based planning to acquire robust policies that predict future behaviors. In line with planning, it maintains a queue of teammate models and calculates an adaptive rollout length for each agent, enabling them to reduce the model-based return discrepancy bound during self-imagination in the world. To evaluate CLWPO, we conducted extensive experiments, and the results show that CLWPO outperforms state-of-the-art MARL baselines, enhancing learning efficiency and improving asymptotic performance. In the future, we aim to combine CLWPO with advanced exploration methods and validate it across a broader range of cooperative, competitive, mixed, and realistic environments.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 91948303).

References

Arulkumaran, K.; Cully, A.; and Togelius, J. 2019. Alphastar: An evolutionary computation perspective. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 314–315.

Berner, C.; Brockman, G.; Chan, B.; Cheung, V.; Dkebiak, P.; Dennison, C.; Farhi, D.; Fischer, Q.; Hashme, S.; Hesse, C.; et al. 2019. Dota 2 with large scale deep reinforcement learning. arXiv:1912.06680.

Chen, J.; Bai, T.; Huang, X.; Guo, X.; Yang, J.; and Yao, Y. 2017. Double-task deep q-learning with multiple views. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 1050–1058.

de Witt, C. S.; Gupta, T.; Makoviichuk, D.; Makoviychuk, V.; Torr, P. H.; Sun, M.; and Whiteson, S. 2020. Is independent learning all you need in the starcraft multi-agent challenge? arXiv:2011.09533.

Egorov, V.; and Shpilman, A. 2022. Scalable Multi-Agent Model-Based Reinforcement Learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, 381–390.

Fan, J.; and Li, W. 2022. Dribo: Robust deep reinforcement learning via multi-view information bottleneck. In *International Conference on Machine Learning*, volume 162, 6074–6102. PMLR.

Ha, D.; and Schmidhuber, J. 2018. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2455–2467.

Haarnoja, T.; Tang, H.; Abbeel, P.; and Levine, S. 2017. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, volume 70, 1352–1361. PMLR.

Haarnoja, T.; Zhou, A.; Hartikainen, K.; Tucker, G.; Ha, S.; Tan, J.; Kumar, V.; Zhu, H.; Gupta, A.; Abbeel, P.; et al. 2018. Soft actor-critic algorithms and applications. arXiv:1812.05905.

Hafner, D.; Lillicrap, T.; Ba, J.; and Norouzi, M. 2020. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations*.

Hafner, D.; Lillicrap, T.; Fischer, I.; Villegas, R.; Ha, D.; Lee, H.; and Davidson, J. 2019. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, volume 97, 2555–2565. PMLR.

Hafner, D.; Lillicrap, T. P.; Norouzi, M.; and Ba, J. 2021. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations*.

Hafner, D.; Pasukonis, J.; Ba, J.; and Lillicrap, T. 2023. Mastering Diverse Domains through World Models. arXiv:2301.04104.

Hansen, E. A.; Bernstein, D. S.; and Zilberstein, S. 2004. Dynamic programming for partially observable stochastic games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 4, 709–715.

Hernandez-Leal, P.; Kaisers, M.; Baarslag, T.; and De Cote, E. M. 2017. A survey of learning in multiagent environments: Dealing with non-stationarity. arXiv:1707.09183.

Hernandez-Leal, P.; Kartal, B.; and Taylor, M. E. 2019. A survey and critique of multiagent deep reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 33(6): 750–797.

Huang, S.; Su, H.; Zhu, J.; and Chen, T. 2020. SVQN: Sequential Variational Soft Q-Learning Networks. In *International Conference on Learning Representations*.

Iqbal, S.; and Sha, F. 2019. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, volume 97, 2961–2970. PMLR.

Kim, H.; Kim, J.; Jeong, Y.; Levine, S.; and Song, H. O. 2019. Emi: Exploration with mutual information. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 3360–3369. PMLR.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. arXiv:1312.6114.

Kurach, K.; Raichuk, A.; Stanczyk, P.; Zajkac, M.; Bachem, O.; Espeholt, L.; Riquelme, C.; Vincent, D.; Michalski, M.; Bousquet, O.; et al. 2020. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 4501–4510.

Lange, S.; and Riedmiller, M. 2010. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Lange, S.; Riedmiller, M.; and Voigtländer, A. 2012. Autonomous reinforcement learning on raw visual input data in a real world application. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Laskin, M.; Srinivas, A.; and Abbeel, P. 2020. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, volume 119, 5639–5650.

Li, M.; Wu, L.; Ammar, H. B.; and Wang, J. 2019. Multi-view reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 32, 1418–1429.

Liu, I.-J.; Ren, Z.; Yeh, R. A.; and Schwing, A. G. 2021. Semantic tracklets: An object-centric representation for visual multi-agent reinforcement learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5603–5610. IEEE.

Liu, Y.; Wang, W.; Hu, Y.; Hao, J.; Chen, X.; and Gao, Y. 2020. Multi-agent game abstraction via graph attention neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 7211–7218.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, volume 30, 6379–6390.

Ma, X.; Chen, S.; Hsu, D.; and Lee, W. S. 2021. Contrastive variational reinforcement learning for complex observations. In *Conference on Robot Learning*, volume 155, 959–972.

Mazoure, B.; Combes, R. T. d.; Doan, T.; Bachman, P.; and Hjelm, R. D. 2020. Deep reinforcement and infomax learning. In *Advances in Neural Information Processing Systems*, volume 33, 3686–3698.

Oh, J.; Singh, S.; and Lee, H. 2017. Value prediction network. In *Advances in Neural Information Processing Systems*, volume 30, 6118–6128.

OpenAI. 2018. Openai five. https://blog.openai.com/openai-five/. Accessed: 2018.

Poole, B.; Ozair, S.; Van Den Oord, A.; Alemi, A.; and Tucker, G. 2019. On variational bounds of mutual information. In *International Conference on Machine Learning*, volume 97, 5171–5180. PMLR.

Rashid, T.; De Witt, C.; Farquhar, G.; Foerster, J.; Whiteson, S.; and Samvelyan, M. 2018. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, volume 80, 4292–4301.

Razavi, A.; Van den Oord, A.; and Vinyals, O. 2019. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, volume 32, 14837–14847.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. arXiv:1707.06347.

Schwarting, W.; Seyde, T.; Gilitschenski, I.; Liebenwein, L.; Sander, R.; Karaman, S.; and Rus, D. 2021. Deep Latent Competition: Learning to Race Using Visual Control Policies in Latent Space. In *Conference on Robot Learning*, volume 155, 1855–1870. PMLR.

Shang, W.; Espeholt, L.; Raichuk, A.; and Salimans, T. 2021. Agent-centric representations for multi-agent reinforcement learning. arXiv:2104.09402.

Shi, D.; Zhao, C.; Wang, Y.; Yang, H.; Wang, G.; Jiang, H.; Xue, C.; Yang, S.; and Zhang, Y. 2022. Multi actor hierarchical attention critic with RNN-based feature extraction. *Neurocomputing*, 471: 79–93.

Sun, W.; Jiang, N.; Krishnamurthy, A.; Agarwal, A.; and Langford, J. 2019. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, volume 99, 2898–2933. PMLR.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; et al. 2017. Value-decomposition networks for cooperative multi-agent learning. arXiv:1706.05296.

Terry, J.; Black, B.; Grammel, N.; Jayakumar, M.; Hari, A.; Sullivan, R.; Santos, L. S.; Dieffendahl, C.; Horsch, C.; Perez-Vicente, R.; et al. 2021. Pettingzoo: Gym for multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, 15032–15043.

Vinyals, O.; Ewalds, T.; Bartunov, S.; Georgiev, P.; Vezhnevets, A. S.; Yeo, M.; Makhzani, A.; Küttler, H.; Agapiou, J.; Schrittwieser, J.; et al. 2017. Starcraft ii: A new challenge for reinforcement learning. arXiv:1708.04782.

Wang, T.; Bao, X.; Clavera, I.; Hoang, J.; Wen, Y.; Langlois, E.; Zhang, S.; Zhang, G.; Abbeel, P.; and Ba, J. 2019. Benchmarking model-based reinforcement learning. arXiv:1907.02057.

Xu, Z.; Zhang, B.; Zhan, Y.; Baiia, Y.; Fan, G.; et al. 2022. Mingling Foresight with Imagination: Model-Based Cooperative Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 35, 11327–11340.

Yang, H.; Shi, D.; Xie, G.; Peng, Y.; Zhang, Y.; Yang, Y.; and Yang, S. 2022. Self-supervised representations for multi-view reinforcement learning. In *The 38th Conference on Uncertainty in Artificial Intelligence*, volume 180, 2203–2213. PMLR.

Yarats, D.; Zhang, A.; Kostrikov, I.; Amos, B.; Pineau, J.; and Fergus, R. 2021. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 10674–10681.

Ye, D.; Liu, Z.; Sun, M.; Shi, B.; Zhao, P.; Wu, H.; Yu, H.; Yang, S.; Wu, X.; Guo, Q.; et al. 2020. Mastering complex control in moba games with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6672–6679.

Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; and Wu, Y. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, volume 35, 24611–24624.

Zhang, W.; Wang, X.; Shen, J.; and Zhou, M. 2021. Model-based Multi-agent Policy Optimization with Adaptive Opponent-wise Rollouts. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, 3384–3391.

Zhou, M.; Luo, J.; Villella, J.; Yang, Y.; Rusu, D.; Miao, J.; Zhang, W.; Alban, M.; Fadakar, I.; Chen, Z.; et al. 2020. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. In *4th Conference on Robot Learning, CoRL 2020*, volume 155, 264–285.