Filtered Observations for Model-Based Multi-agent Reinforcement Learning

Linghui Meng[1,2], Xuantang Xiong[1,2], Yifan Zang[1,2], Xi Zhang[1], Guoqi Li[1,2],
Dengpeng Xing[1,2(✉)], and Bo Xu[1,2(✉)]

[1] Institute of Automation, Chinese Academy of Sciences, Beijing, China
{menglinghui2019,xiongxuantang2021,dengpeng.xing,xubo}@ia.ac.cn
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences,
Beijing, China

Abstract. Reinforcement learning (RL) pursues high sample efficiency in practical environments to avoid costly interactions. Learning to plan with a world model in a compact latent space for policy optimization significantly improves sample efficiency in single-agent RL. Although world model construction methods for single-agent can be naturally extended, existing multi-agent schemes fail to acquire world models effectively as redundant information increases rapidly with the number of agents. To address this issue, we in this paper leverage guided diffusion to filter this noisy information, which harms teamwork. Obtained purified global states are then used to build a unified world model. Based on the learned world model, we denoise each agent observation and plan for multi-agent policy optimization, facilitating efficient cooperation. We name our method UTOPIA, a model-based method for cooperative multi-agent reinforcement learning (MARL). Compared to strong model-free and model-based baselines, our method shows enhanced sample efficiency in various testbeds, including the challenging StarCraft Multi-Agent Challenge tasks.

Keywords: Model-based planning · Multi-agent reinforcement learning · Generative models

1 Introduction

Cooperative multi-agent reinforcement learning (MARL) has recently attracted much attention due to the success of its applications in many practical tasks, such as real-time strategy games [19], return-based card games [1,8], and unmanned aerial vehicles [32]. Despite the empirical success [18,20,30], the most extensively studied methods are built on the model-free reinforcement learning paradigm. They always face low sample efficiency since the policy optimization relies on a large number of costly interactions with the environment [28]. In addition, the policy exploration spaces which explode with the agent number in multi-agent systems further deteriorate this problem [29].

To rectify model-free RL's sample efficiency issue, a common practice of the alternative model-based RL in single-agent settings is to learn a world model [4,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
D. Koutra et al. (Eds.): ECML PKDD 2023, LNAI 14172, pp. 540–555, 2023.
https://doi.org/10.1007/978-3-031-43421-1_32

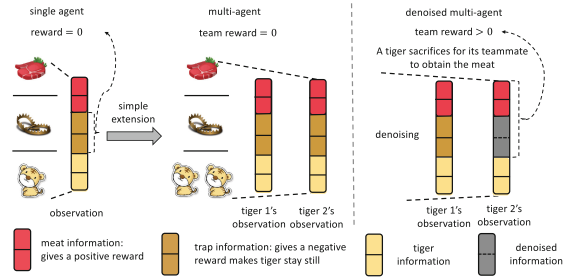


Fig. 1. Illustrative example: each tiger can see four kinds of information and aim to get the meat behind the trap. Each trapped tiger will die and receive negative feedback, which makes the tiger, in the left two cases, stay still to receive a zero reward. Once one tiger gets the meat, the team will receive maximum positive feedback.

6,13,26,27]. It builds transition and reward models to predict trajectories based on global states embedded in a compact latent space. By planning with the learned model rather than interacting with the environment as those model-free methods, these types of approaches efficiently optimize policies and achieve state-of-the-art performance in single-agent RL tasks.

When it comes to the multi-agent field, the embedded global state to construct the world model becomes the research concentration. Although the observation encoder in single-agent RL can be directly extended to embed multiple agents' observations by sharing and concatenating them, it ignores the cooperative relations in the team and leads to the fusion of noise, which hinders the system from getting higher rewards. An illustrative example in Fig. 1 describes an animal trap as noise. A single-agent tiger needs to avoid the trap and turn around for a living resulting in zero rewards. Once extending this problem by sharing and concatenating observations to the multi-agent case, two tigers unite as a multi-agent team to optimize the team reward, but the reward is still zero. A reasonable solution is to mask the noise (trap) in the observation of at least one agent to sacrifice it by covering the trap and finally cooperating for the meat to achieve a positive team reward. Accordingly, obtaining embedded global states and constructing a world model in a multi-agent scenario requires filtering out the above noise in the combination of each agent's local observations.

Following the above noise filtering intuition, we in this paper propose UTOPIA, a model-based method that denoises each agent observation to acquire the purified global states and learns a world model for the multi-agent system. Instead of building each agent a specific encoder that is impractical and unscalable, we designed a single conditioned noise filter for all agents. Concretely, we leverage a variational lower bound to train a diffusion model to represent the global states in the latent space. Moreover, we define the noise for a guided model to discriminate the noise given latent global states. It is learned from the perspectives of transition and reward for each agent. We conduct the noise filter using the gradient from the guided model to a diffusion model during sampling. The learned noise filter enables tailored denoising for each agent to purify global states. Our approach thus utilizes the purified global state to construct a multi-agent world model, which plans in latent space to improve sample efficiency.

In order to study the effectiveness of our method, we first examined its necessity on a toy multi-agent MDP (MMDP) example and then validated it on the challenging cooperative multi-agent task, Starcraft multi-agent challenges (SMAC). We show that UTOPIA can improve the sample efficiency compared to strong baselines in the toy example and challenging tasks. Our contributions are summarized as follows: 1) **Addressing model-based cooperative multi-agent reinforcement learning.** We propose a multi-agent algorithm based on a learnable world model and emphasize the model-based MARL to solve the low sample efficiency problem. 2) **Learning to denoise each agent observation for purified global states.** We explicitly propose the definition of task-relevant information. To utilize the explicit objective, we regard the definition as a condition and employ a guided model to enforce the diffusion model sampling. 3) **Empirical performance on cooperative MARL** Our experimental results show superior performance in sample efficiency compared to model-free baselines and a model-based approach extended from single-agent methods.

2 Related Work

The problem of learning to collaborate among multiple agents has been commonly addressed with the framework of the partially observed Markov decision process (POMDP) that partly masks the global state for each agent observation. Recent work has combined deep neural networks with multi-agent policy within this framework as centralized-training decentralized execution (CTDE) [15,18] or advantage decomposition approaches [10]. These approaches are based on value iteration or actor-critic even combined with sequential models, such as VDN [20], QMIX [18], MAPPO [30], MAT [25], which achieve remarkable performance in challenging tasks. However, these approaches are model-free methods limited by the low sample efficiency without accessing the environment [29]. An intuitive approach is to build model-based methods to augment the buffer generated for the policy learning [17,24]. In addition, it remains unclear whether these methods can incorporate the environment to construct a model-based method. In the single-agent field, a few works have attempted to abstract the input as global states via reconstruction loss to build a world model and plan it in the latent space, such as PlaNet, Dreamer, DreamerV2 [4–6]. [3] even leverages the goal as a condition to derive the imagined trajectories autoregressively. Although these methods have shown supreme performance on various single-agent tasks, the implicit relations and properties of partially observable limit their development to effectively represent the global states in the multi-agent domain [31]. [12] proposes that neighboring agents could have similar representations of observations in multi-agent. [21,22] claim the role-based method to represent the multi-agent observations via clustering them in latent space to discriminate each agent by sharing a role selector network automatically. These methods mainly focus on how to represent multi-agent observations well to be the policy input. We focus on abstracting the global states to construct the world model for planning. In contrast with directly abstracting the multi-agent observations for the global state extended from single-agent methods [6], we claim the noise fusion problem in the simple aggregation methods, which harms teamwork. The closest work emerges in the single-agent field, Denoised MDP, which splits latent states based on whether they are controllable or reward-relevant via reconstruction. Accordingly, we define noise based on teammate information from transition and reward perspectives and employ guided diffusion [2,7] to purify multi-agent observations conditionally. To combine with model-free MARL approaches, we also integrate prior works with multi-agent behavior learning. Our novelty lies in the definition of the noise in multi-agent observations and the method of conditionally purifying them to build a world model to improve the sample efficiency.

3 Background and Notations

3.1 Cooperative Multi-Agent Problem

We consider the formulation in cooperative multi-agent reinforcement learning, which is a partial observable extension of the standard Markov Decision Process (MDP) named partially-observed MDP (POMDP) [14]. $\hat{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \gamma, n, \mathcal{R}, \mathcal{T})$. $\mathcal{S}$ denotes the state space of multiple agents, $\mathcal{O} : \mathcal{S} \rightarrow \Omega$ denotes the observation function, where Ω represents the observation space, $\mathcal{A} = \{\mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n\}$ represents the joint action space of n agents, respectively, $\mathcal{R}$ represents the reward function. To clarify the notation, we denote the time t for variables of all agents by sub-scripts but the time for variables with agent index by super-scripts. We denote $s_t \in \mathcal{S}$ to refer to the global state of the environment. At time step t, let $\boldsymbol{o}_t = \{o_i^t\}_{i=1}^n$ with each $o_i^t \in \mathcal{O}(s_t, i)$ be the partial observation of agent i accessing the global state s_t. Accordingly, let $\boldsymbol{a}_t = \{a_i^t\}_{i=1}^n$ with each $a_i^t \in \mathcal{A}_i$ indicating the action taken by the agent i. $\mathcal{T}(s_{t+1}|s_t, \boldsymbol{a}_t) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition function. $r(s_t, \boldsymbol{a}_t) \in \mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ indicates the team reward function is shared across agents from the environment. $\gamma \in [0, 1)$ is the discount factor. Let $\pi_i(a_i^t|o_i^t)$ be a stochastic policy for agent i and denote $\pi = \{\pi_i\}_{i=1}^n$. Let $J(\pi) = \mathbb{E}_t[R_t]$ with $R_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l}(s_{t+l}, \{a_i^{t+l}\}_{i=1}^n)$ denoting the expected discounted reward, where $a_i^{t+l} \sim \pi_i(a_i^{t+l}|o_i^{t+l})$ for $i \in [1, n]$. The problem aims to find the optimal multi-agent policy $\pi^\star = \{\pi_i^\star\}_{i=1}^n$ that achieve the maximum expected team reward $J(\pi)$.

3.2 Latent Imagination for Learning Policy

Learning to plan via a predictive model named *world model* can facilitate learning complex behaviors [4,5]. The world model aims to estimate the real dynamics and reward conditioned on the input state with deep neural networks $p(s^{'}|s)$ and $p(r|s)$, respectively, where s denotes the global state usually encoded by the generative model VAEs in latent space. The behavior policy $\pi(a|s)$ is built on the latent states to control the agent. Once the world model is learned, it imagines batch trajectories starting from the true current state s^t, $\tau^t = \{s^j, a^j, r^j\}_{j=t}^{t+h}$, where $s^j \sim p(s^j|s^{j-1})$ and $a^j \sim \pi(a^j|s^j)$, h denotes the imagination horizon. The objective is to maximize the expected reward on the imagined trajectories, $\mathbb{E}_\pi[\sum_{j=t}^{\infty}\gamma^{j-t}r(\tau^j)]$ with respect to the behavior policy, where γ denotes the discount factor. To extend this mechanism to multi-agent fields, we follow the above procedure to build a world model and learn a policy for multi-agent from the trajectories planned by the world model. The challenges in the multi-agent case, mainly emerge in how to abstract the global states for the multi-agent world model and the policy learning.

3.3 Guided Diffusion Models

Score-based generative models, such as diffusion models [7], achieve great advancements in image synthesis tasks. The diffusion (forward) process gradually destroys inputs via a fixed Markov chain, such as adding weighted noise conditioned on the previous input, $x_d \sim \mathcal{N}(x_d; \sqrt{\bar{\alpha_d}}x_{d-1}, \sqrt{1-\bar{\alpha_d}}\mathcal{I})$, where $\bar{\alpha_d} = \prod_{d=1}^{D}\alpha_d$, and d denotes the index of the diffusion process, while D denotes the diffusion horizon. To reconstruct the original input, we need to approximate the unknown posterior, $q(x_{d-1}|x_d)$ with a parameterized neural network, $p_\theta(x_{d-1}|x_d) = \mathcal{N}(x_{d-1}; \mu_\theta(x_d, d), \Sigma_d)$. The original inputs can be deblurred from the standard normal distribution. The induced data distribution is given by:

$$p_\theta(x_0) = \int p(x_D)\prod_{d=1}^{D} p_\theta(x_{d-1}|x_d)dx_{1:D} \tag{1}$$

Then the evidence lower bound can be derived based on the maximized likelihood estimation to optimize this generative model. The simplified loss is shown as:

$$L_{simple} = \mathbb{E}_{x_0,z_d}[||z_d - z_\theta(\sqrt{\bar{\alpha_d}}x_0 + \sqrt{1-\bar{\alpha_d}}z_d, d)||] \tag{2}$$

where z_d is the Gaussian noise. Recent works propose classifier-guided diffusion to generate a specified sample, such as generating an image with a label. The modified posterior $q(x_{d-1}|x_d, y)$ can be estimated with an integration between the diffusion model pre-trained and a classifier $p(y|x_d)$. Therefore, the reverse process can be approximated as the following prediction problem [2]:

$$p_\theta(x_{d-1}|x_d, y) \approx \mathcal{N}(x_{d-1}; \mu + \Sigma g, \Sigma) \tag{3}$$

where μ and Σ are the output of the pre-trained diffusion model, $g = \nabla \log p(y|x)|_{x=x_d}$ denotes the gradient of the classifier. Therefore, we can guide the diffusion model with predefined objectives to generate samples of specific attributes without additionally retraining the diffusion model.

In this work, we define the task-relevant objective for the purified states in the multi-agent world model. UTOPIA guides the diffusion model to denoise each agent observation conditioned on the objective to filter out the noise that harms teamwork. The global purified states are composed of denoised observations to construct a multi-agent world model. By planning batch trajectories with the world model, the multi-agent policy is optimized based on the planned trajectories in a cooperative Markov Game.

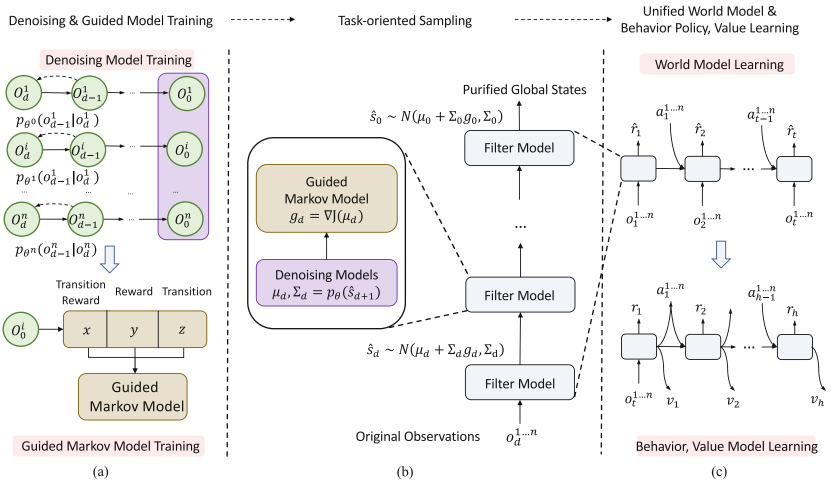


Fig. 2. Overview of UTOPIA learning process: (a) Denoising (up) and guided (down) model learning, where the original observations are fed into the diffusion model with the sub-scripts denoising index d gradually decreasing, and the guided model learns to discriminate the noise; (b): Guide diffusion model purifies each agent original observation and concatenate them as purified global states; (c): Unified World (up) and behavior, value model (down) learning. The filter model in (b) is used to purify observations to predict the true reward and next state with a supervised style. Given observations in a time, the world model plans to roll out based on the purified global state, which fed for Behavior policy and Value model learning.

4 Methodology

In this section, we first highlight the critical motivation to learn a purified world model for model-based MARL. We then elaborate on each module of our method for learning to control multi-agent cooperation. In addition, we show the learning objective of the multi-agent world and behavior models. To illustrate the use of input-conditional guided sampling for each agent purified observation, we describe the training and sampling process of the task-oriented filter model.

4.1 Motivation

1) There exists much noise having no effect on the transition and reward functions, then noise increases the redundant exploration of the expanded policy space. To improve efficient cooperation, we need to filter it out to obtain the purified global states. 2) The noise in each agent observation should be eliminated conditioned on their local observations and global information rather than be denoised individually. Therefore, based on the above two points, we need to develop the conditioned denoising network for purifying the global states to construct the multi-agent world model.

4.2 Control with Denoised World Models

To resolve the problems that emerged in our motivation, we formulate the multi-agent behavior policy learning that takes input as the global states in a unified world model with several modules, mainly including the *unified world model* and *agent components* parts. Figure 2 shows the whole framework.

Unified World Model. We introduce six components for the unified world model learning and leave the demonstration of how to define and utilize the objective in the next section. Learning the joint world model for cooperation across multiple agents is difficult for precisely predicting task-relevant future trajectories. There are three main reasons: 1) Learning the global state from observations of multiple agents requires a purified representation without additional noise, as described in motivation. 2) The relationship among agents is implicit due to their identity, so how to define noisy information is difficult. 3) How to leverage the observation filtering process differently for each agent conditioned on the shared observation encoder and noise definition. Therefore, we introduce six components to construct the presented world model. The six components contain the sharing denoising model facilitated by the diffusion model, the guided model, the transition model, the reward prediction model, the available prediction model, and the mask prediction model. We formulate them as follows:

$$\begin{array}{ll} \text{Denoising model:} & p(\hat{o}_i^t|o_i^t, a_i^{t-1}, \hat{s}_{t-1}) \\ \text{Guided model:} & p(\mathcal{O}_t = 1|o_i^t, a_i^{t-1}) \\ \text{Transition model:} & q(\hat{s}_t|\hat{s}_{t-1}, \boldsymbol{a}_{t-1}) \\ \text{Available model:} & q(ava_i^t|\hat{o}_i^t, a_i^{t-1}) \\ \text{Reward model:} & q(r_t|\hat{s}_t, \boldsymbol{a}_t) \\ \text{Masking model:} & q(m_t|\hat{s}_t) \end{array} \tag{4}$$

1) The denoising model $p(\hat{o}_i^t|o_i^t, a_i^{t-1}, \hat{s}_{t-1})$ is shared across agents and aims to denoise each agent observation to the latent space. 2) The guided model $p(\mathcal{O}_t = 1|o_i^t, a_i^{t-1})$ is input conditional on splitting the noisy information. In this model, we leverage the control-as-inference used in [9,11] to guide the denoising process by defining a binary variable $\mathcal{O}_t$ denoting the optimality of our defined objective which is described in the following sections. Conditioned on the learned guided model, the modified mean and variance for a permuted Gaussian distribution can be computed. The purified state is concatenated by all agents denoised observations $\hat{s} = \hat{o}_{i=1...n}$. 3) To predict the Markov transition, UTOPIA leverages the transition model $q(\hat{s}_t|\hat{s}_{t-1}, \boldsymbol{a}_{t-1})$ used in Dreamer [4] to predict the next global state based on the current ones. 4) The available models $q(ava_i^t|\hat{o}_i^t, a_i^{t-1})$ estimate the currently available actions given the global state, where $ava_i^t \in \mathtt{R}^{[\mathcal{A}_i]}$ represents the available action of agent i at t step, where each dimension denotes the corresponding action is actionable or not. 5) The reward model $q(r_t|\hat{s}_t, \boldsymbol{a}_t)$ is used to predict the current true reward given state and action. 6) The masking model $q(m_t|\hat{s}_t)$ predicts the end step for the whole episode, where m^t is a binary

variable representing whether the game is over or not 0/1 at t step. They could emerge in a complicated environment like SMAC. We represent how to learn a specific model to tackle them for accurate planning, although they may be not necessary for many tasks. The denoised observation embeddings are concatenated together as the purified global state. The learned world model is leveraged to update the behavior and value models to maximize the expected reward in Markov games.

Agent Components We introduce two components used in UTOPIA, multi-agent behavior policy and value model.

$$\pi_i(a_i^t|\hat{o}_i^t) \quad \text{and} \quad V_\pi^t(\hat{s}_t) \tag{5}$$

where $\hat{o}_i^t$ denotes the denoised observation embedding for agent i in timestep t, π_i denotes the behavior policy of agent i mapping the denoised observation $\hat{o}_i^t$ to the action space, V_π^t denotes the value function base on the global state aims to estimate the expected reward at t. It should be noted that the conventional input of behavior policy for each agent in POMDP is local observation without global information integration in the inference period [18,30]. In the case of UTOPIA, the local observations are denoised as inputs with a guided denoising model optimized from the global information such as the team reward. Learning to estimate the team reward typically helps improve the policy update, where the value module takes as input the global information predefined. In contrast, the value model in UTOPIA evaluates the task-oriented global state $\hat{s} = \hat{o}_{i=1\ldots n}$ concatenated by denoised observation embeddings. In the following section, we will introduce the corresponding objectives about how to optimize these presented modules.

4.3 Learning Objective

In this section, we first introduce how to define the noise information and how to learn to filter it from the local observations. Then the objective of the guided denoising model to learn to conditionally denoise each agent observation is introduced. Then we describe the objective of the multi-agent world model, including transition, reward, and other predictive models. Finally, based on the above world model, we introduce how to optimize the corresponding multi-agent action and value models are finally introduced.

Guided Denoised Model Learning. The denoised representations in latent space can be abstracted by the diffusion model beforementioned $p_\theta(\hat{o}_i^t|o_i^t, a_i^{t-1}, \hat{s}^{t-1})$. To simplify the training process, we send each agent observation as input and leverage the simple version of diffusion objective as follows:

$$\begin{aligned}\mathcal{L}_{denoising} = \sum_{i=1}^{n} \mathbb{E}_{\tau_i \sim \pi_i}[\sum_{d} ||z_d - z_\theta(\sqrt{\bar{\alpha_d}}\tau_i \\ + \sqrt{1-\bar{\alpha}_t} z_d, d)||], \quad \text{where } z_d \sim \mathcal{N}(0, \mathcal{I})\end{aligned} \tag{6}$$

where n denotes the agent number, τ_i denotes the trajectory of agent i, and z_θ denotes the output of diffusion model p_θ described in Fig. 2.

Based on the denoising model above, to filter the noise information for each agent, we define the noise as long as it does not affect the transitions and reward predictions like in Denoised MDPs [23]. The guided model takes as input the diffusion representations and divides them into three parts, denoting x, y, and z respectively, constructing the training set, where y and z denote the transition-irrelevant and reward-irrelevant noise. In terms of the two parts, which involve the global transition and reward, we set the following objective to train the guidance model:

$$\begin{aligned}\mathcal{L}_{guided} = I(p(\hat{o}_i^{'}|[x_i, y_i, z_i], a_i)||p(\hat{o}_i^{'}|[x_i, z_i], a_i))\\ + I(p(r|[x_i, y_i, z_i]_{i=1\dots n}, \boldsymbol{a})||p(r|[x_i, y_i, z_i]_{i=1\dots n}], \boldsymbol{a}))\end{aligned} \tag{7}$$

where the first term aims to minimize the distance of the posterior between the two conditions, and y_i denotes the uncontrollable part in each agent. Similarly, the second term aims to learn to separate the reward-irrelevant part z_i, where x denotes purified observation used to construct global states. When the objective is minimized, the noise y and z have no effect on the rollouts of the world model. Then we can generate x of each agent to construct the purified global states for the unified multi-agent world model.

World Model Learning. To further construct the world model based on the purified global states, we train the transition and prediction models for reward, available actions, and masks by reconstruction loss. We train a sequential model by taking as input the purified global state $\hat{s}$ concatenated from the diffusion outputs. The supervised learning process for the global world model can be optimized by minimizing the negative log-likelihood:

$$\mathcal{L}_{world} \triangleq -\mathbb{E}_{i,t}\big[\log p_\theta(\hat{s}_i^{t+1}, r^t, m_i^t, ava_i^t|\hat{s}_i^t, \boldsymbol{a}^t)\big] \tag{8}$$

where θ is the parameter of the global world transition model.

Action and Value Model Learning. UTOPIA trains the behavior policy $\pi_i(a_i|\hat{o}_i)$ for each agent i and value model $V(\hat{s})$ via backpropagating from additional roll-outs τ from the learned unified world model beforementioned. We adopt PPO-style algorithm for policy and value model optimization. To update the action and value models, we first compute the estimated state value $V_\psi^\pi(\hat{s}_\tau)$ for all states s_τ along the imagined trajectories. The objective for the action model $\pi(a_\tau|\hat{o}_\tau)$ is to predict actions that result in state trajectories with high-value estimates. The objective for the value model $V_\psi(\hat{s}_\tau)$, in turn, is to regress the cumulated reward. The objective of the action model is:

$$\max_\phi \sum_{i=1}^{n} \mathbb{E}_{q_\phi, q_\psi}\Big[\sum_{j=t}^{t+h} \min\big(\frac{\pi_\phi(\tau_i^j)}{\pi_{\phi old}(\tau_i^j)}, \sigma\big) A(\tau_j)\Big] \tag{9}$$

where $\sigma = \text{clip}(\frac{\pi_\phi(\tau_i^j)}{\pi_{\phi old}(\tau_i^j)}, 1-\epsilon, 1+\epsilon)$, h represents the rollout horizon and $\tau_i^j = \{\hat{o}_i^m\}_{m=0}^j$, $\tau_j = \{\hat{s}_m\}_{m=0}^j$. The objective of the value model is:

$$\min_\psi \mathbb{E}_{q_\phi, q_\psi}\Big[\sum_{j=t}^{t+h} \frac{1}{2}\Big\| \sum_{l=j}^{\infty} \gamma^l r_j - V_\psi^\pi(\tau_j))\Big\|^2\Big] \tag{10}$$

4.4 Training and Sampling

To describe the update procedure of each module described above, in this section, we demonstrate the training and sampling process in detail.

Training Process. To filter the task-oriented global states in latent space and utilize the purified states as behavior and value model inputs, we train the diffusion model first and then optimize the guided model with the objective in Eq. 6, 7. The world model is learned from the rollout trajectories denoised by the guided diffusion model, then we train the critic model to estimate the true reward on each step. To predict optimal actions, we learn the behavior policy to maximize the total discounted values estimated by the critic model.

Sampling Process. Then we derive the guided sampling in the presented denoising model with the explicit objective defined in Eq. 7. Let $\mathcal{O}_d$ be a binary random variable denoting the purity of timestep t of a trajectory, with $p(\mathcal{O}_d = 1) = \exp(-\mathcal{L}_{guided})$.We can derive the conditional probability as follows:

$$p(\hat{o}_d|\hat{o}_{d+1}, \mathcal{O}_d = 1) = p_{\theta,\phi}(\hat{o}_d|\hat{o}_{d+1}, \mathcal{O}_d = 1) \tag{11}$$

By separating the diffusion model and the guided model, we split the reverse process as follows:

$$p_\theta(\hat{o}_d|\hat{o}_{d+1}, \mathcal{O}_d = 1) = p_\theta(\hat{o}_d|\hat{o}_{d+1}) p_\phi(\mathcal{O}_t = 1|\hat{o}_t) \tag{12}$$

We ignore the superscript i for succinct notation. By defining three variables x, y, and z, we propose to discriminate noise that is y and z, which are irrelevant to the transition or reward. The guide sampling can be shown as follows:

$$\begin{aligned} &\log p_\theta(\mathcal{O}_t = 1|x_t) \approx \log p_\phi(\mathcal{O}_d = 1|\hat{o}_d)|_{\hat{o}_d=\mu} \\ &+ (\hat{o}_d - \mu)\nabla_{\hat{o}_d} \log p_\theta(\mathcal{O}_d = 1|\hat{o}_d)|_{\hat{o}_d=\mu} \\ &= (\hat{o}_d - \mu)g + C_1 \end{aligned} \tag{13}$$

where $g = \nabla_{\hat{o}_d} \log p_\theta(\mathcal{O}_d = 1|\hat{o}_d)|_{\hat{o}_d=\mu}$ is the gradients of the guided model at the specified value μ. With the sampling proceeding, each agent observation can be purified by objective-oriented for constructing the world model subsequently.

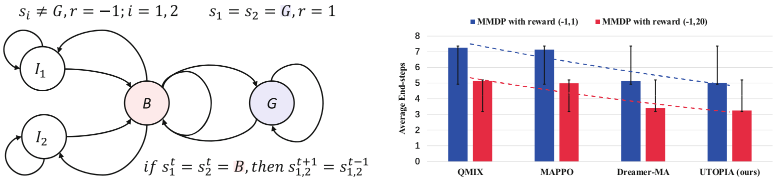
5 Experiments

On top of the theoretic analysis, we tested UTOPIA on various environments. To illustrate the necessity of our method even in a simple case, we validated our method on a toy example, a multi-agent MDP (MMDP) (see Fig. 3(a)). To verify our method, we validate UTOPIA on both easy and (super-)hard maps of the StarCraft multi-agent challenge (SMAC) for efficient cooperation. All experimental results are obtained compared with popular multi-agent model-free and model-based MARL in terms of sample efficiency and performance. We mainly answer the following two research questions: **RQ1**: Can the world model help improve the sample efficiency of cooperative multi-agent tasks? **RQ2**: Can our method construct a purified world model to help multi-agent cooperation? Therefore, we set up different experiments to answer these two questions.

Baselines. We chose our baselines in MARL from two groups, namely model-free and model-based approaches. Since methods that worked well in POMDP belong to the model-free class, we compared several strong and representative model-free methods as our baselines. In the model-free group, the baselines include two policy and value-based methods, MAPPO and QMIX [18,30] that achieve state-of-the-art performance on the well-known challenging task, SMAC. In the model-based group, learning from multi-agent observations to construct the world model has been little addressed. We further extend the single-agent observation encoder used in Dreamer [4] by sharing it across agents for the global state abstraction as our model-based baseline. The latent global states are concatenated directly from the shared encoder outputs. The concatenated states conduct a world model for model-based MARL without denoising like that in Dreamer [4]. We named this approach Dreamer-MA as the model-based baseline.

Evaluation Metrics. On the toy example, we set up the end step as the evaluation metric, which denotes the maximum interactions of two players used to reach the goal state in the MMDP example, the end step can be intuitively evaluated the sample efficiency of all methods. It is also proportional to the reward defined in Fig. 3(a). On the metric of SMAC, in contrast to using the win rate, to compare these methods with a more fine-grained metric, we use average returns, which are normalized from 0 to 20 and proportion to the win rate. Therefore, we leverage *Average Returns* at each timestep in an episode as the evaluation metric to validate our method and baselines in SMAC.

Implementation Details. Our implementation uses PyTorch libarary [16]. All experiments run in 10 random seeds. We use a single Nvidia A100 GPU and 256 CPU cores for each training run. The baseline implementations are based on the official open source code. The world models are learned via reconstruction in Dreamer-MA. We reuse the architecture for estimating the world model in UTOPIA to the official implementation from Dreamer [4]. Concretely, we borrow the neural network architecture from QMIX, MAPPO, and Dreamer [4,18,30] as our baselines and train the behavior policy and value models as PPO methods. All of the other compared methods adopt exactly the same structure as the feature embedding from vectors of UTOPIA. For the hyperparameter selection, we have fine-tuned the key hyperparameters for our method and baselines in each experiment. In this section, we include details of our experiments.

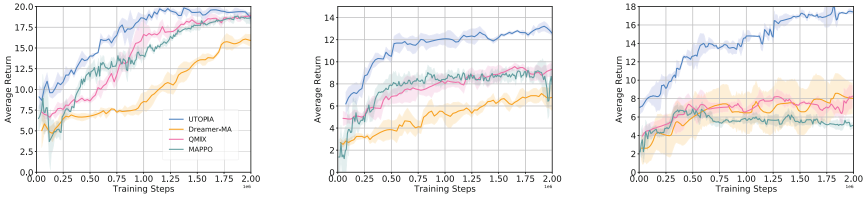


(a) (b)

Fig. 3. (a) An illustration of Multi-agent MDP (MMDP): I_i denotes the initial state of agent i, B denotes the rebound state, S denotes a normal state, and G denotes the terminal goal state. Two players cooperate to reach the goal state together, rewarded with the above rule. (b) End-steps in MMDP after 50 interactions for our method and baselines. The blue and red data means the final reward is 1 or 20.

5.1 Toy Example

To illustrate the necessity of our method even in a simple multi-agent cooperation problem, we designed a two-player Multi-agent Markov decision process (MMDP) as a toy example, as shown in Fig. 3(a). We conduct a two-player multi-agent MDP to illustrate the necessity of our method, even in this simple case. This setup is arguably easier but more practical. Two players following the MMDP depicted in Fig. 3(a) cooperate to reach the same goal state together. This example can be regarded as a simplified version of many practical multi-agent cooperation problems, like vehicle collision avoidance. Each agent i is initialized in the state I_i and moves along the side in MMDP each timestep. Once two agents are headed to the rebound state B, they will return to their original states in the next step. The maximum episode length is set at 20 steps. The game ends when two agents reach the goal state G together or the timestep exceeds the episode limit. Each agent receives a -1 reward if the two agents are still not in the goal state. To fully validate the effectiveness of our method compared with baselines, we set two different final rewards as 1 and 20 if they reach it together and denote the two MMDP as $(-1, 1)$ and $(-1, 20)$. The average end-steps of 10 experiments are shown in Fig. 3(b) on these two variants. In terms of the rule, the two agents need to plan together to avoid the collision in the rebound state to save the step to finish the game. Therefore, for this simple example, we see that the model-based method can achieve lower end-steps at the fixed rollout samples than the model-free methods. This indicates that planning in the latent space is crucial to improving the sample efficiency, even in a simple setting. When we dive into the model-based methods in Dreamer-MA, where the observation of each agent is shared across teammates to build a multi-agent shared world model, the noise can be included in the global states. By denoising task-irrelevant information, UTOPIA outperforms the naive model-based method, which also indicates the effectiveness of our method in Fig. 3(b). We analyze the reason for the limited improvement by comparing our method with Dreamer-MA mainly because of the finite noise that existed in this toy example.



(a) so many baneling (easy)

(b) 5m vs. 6m (hard)

(c) 3s5z vs. 3s6z (super hard)

Fig. 4. The average returns of model-free and model-based MARL on three maps.

5.2 Cooperative Multi-agent Tasks on SMAC

Another more challenging benchmark we used as the testbed is StarCraft multi-agent challenge (SMAC) [19], where multiple agents are masked with the observation function and receive local observations, which can be modeled as a partially observed Markov decision process (POMDP). This partially observed property prevents multiple agents from collaborating to beat the opponents. Different maps include various entities with different observation and action spaces, which own a team reward normalized in $[0, 20]$. Multiple agents require discrete actions to cooperate to beat the enemy in the POMDP. The state-based input makes learning without denoising more difficult due to the high-level information in observations [18]. We compare UTOPIA with state-of-the-art model-free methods and an extension from single-agent model-based methods as described in baselines on three maps (easy, hard, and super hard) from SMAC. The results are shown in Fig. 4. Results suggest our method outperforms the other three baselines in terms of sample efficiency across all three difficulty maps. We also find that Dreamer-MA cannot compete with the performance of the other two model-free baselines. Dreamer-MA decreases the performance in the more complicated tasks, caused by the abundant noise with the increasing agent number and randomness of the environment. This also verifies the necessity of denoising the task-irrelevant information for each agent in UTOPIA.

5.3 Ablation Studies

To better analyze the importance of each component in UTOPIA, we have conducted an extensive ablation study as shown in Table 1. To clarify the ablation study, we claim several modifications in our method by dropping a module. We explain the modification in detail for the rows in Table 1. In the first row, we replace the behavior policy's objective from the clipped weighted advantage loss in PPO-style to the actor-critic method used in Dreamer [4], where the critic is to precisely estimate the expected rewards and the actor aims to maximize the critic value. In the second row, we use the generative model VAEs employed in Denoised MDP [23] to extract the global state via reconstruction loss and share this observation encoding network among agents. In the third row, we drop the guided denoising in UTOPIA to validate the effectiveness of the diffusion and guided models. In the last row, we use the fixed planning horizon for the latent roll-outs of the world model, where we claim the fixed horizon will limit behavior policy learning. The results in each row are obtained by replacing one component of UTOPIA and keeping other components unchanged. The first row shows the advantage of PPO in optimizing the behavior policy. The second row shows the degradation when our method shares the VAE encoder across agents, which is validated to be harmful to abstracting the global states and constructing the world model. The third row validates the effectiveness of the guided denoising models. Planning based on the fixed horizon step also struggles to compete with our method. We guess many hard tasks need more planning steps than the fixed horizon, which degrades behavior policy learning.

Table 1. Average returns of agents after training from 1M timesteps of UTOPIA variants by dropping different modules on two SMAC maps. All quantities are provided on a scale of 0.01, averaged under 10 random seeds. Standard deviations over these random seeds are provided in brackets.

Method	3m (easy)	MMM2 (super hard)
UTOPIA (our method)	**19.66 (±0.25)**	**15.23 (±2.37)**
with actor-critic	17.26 (±0.15)	14.47 (±1.72)
with VAE observation encoder	15.74 (±0.62)	11.37 (±2.68)
without guided denoising	10.25 (±1.27)	8.46 (±2.33)
without dynamic planning horizon	8.26 (±2.15)	7.14 (±3.43)

6 Conclusion

In this paper, we presented UTOPIA, a new model-based MARL that improves sample efficiency by learning to filter multi-agent observations and building a purified world model. In a multi-agent setting, we highlight the necessity of denoising each agent observation differently and defining noise based on teammates' contributions to transitions and rewards. Considering the explicit definition, we encode each agent observation using a guided diffusion model to generate purified global states. Furthermore, our method shares an easily scalable condition-dependent guided diffusion model for each agent. By constructing a world model from purified states, we plan the trajectory in latent space, providing a buffer for learning policy and value models. Experimental results on a toy example and challenging tasks show that our method can outperform existing state-of-the-art mode-free methods and a multi-agent extension of the single-agent model-based approach. In summary, this method provides a feasible direction for model-based MARL and can be applied to many real-world tasks requiring planning with expensive interactions, such as robotic control tasks. In the future, we hope that the world model and behavior policy in multi-agent can be optimized together and avoid the extrapolation errors caused by insufficiently trained modules.

Acknowledgements. This work is supported by the National Key R&D Program of China (No. 2022ZD0116405), the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant (No. XDA27030300), and the Program for National Nature Science Foundation of China (62073324).

References

1. Bard, N.: The hanabi challenge: a new frontier for AI research. Artif. Intell. **280**, 103216 (2020)
2. Dhariwal, P., Nichol, A.: Diffusion models beat GANs on image synthesis. Adv. Neural. Inf. Process. Syst. **34**, 8780–8794 (2021)
3. Fang, K., Yin, P., Nair, A., Levine, S.: Planning to practice: efficient online fine-tuning by composing goals in latent space. arXiv preprint arXiv:2205.08129 (2022)
4. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: learning behaviors by latent imagination. arXiv preprint arXiv:1912.01603 (2019)
5. Hafner, D., et al.: Learning latent dynamics for planning from pixels. arXiv preprint arXiv:1811.04551 (2018)
6. Hafner, D., Lillicrap, T., Norouzi, M., Ba, J.: Mastering Atari with discrete world models. arXiv preprint arXiv:2010.02193 (2020)
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
8. Hu, H., Foerster, J.N.: Simplified action decoder for deep multi-agent reinforcement learning. arXiv preprint arXiv:1912.02288 (2019)
9. Janner, M., Du, Y., Tenenbaum, J.B., Levine, S.: Planning with diffusion for flexible behavior synthesis. arXiv preprint arXiv:2205.09991 (2022)
10. Kuba, J.G., et al.: Trust region policy optimisation in multi-agent reinforcement learning. arXiv preprint arXiv:2109.11251 (2021)
11. Levine, S.: Reinforcement learning and control as probabilistic inference: Tutorial and review. arXiv preprint arXiv:1805.00909 (2018)
12. Mao, H., et al.: Neighborhood cognition consistent multi-agent reinforcement learning. In: AAAI (2020)
13. Mendonca, R., Rybkin, O., Daniilidis, K., Hafner, D., Pathak, D.: Discovering and achieving goals via world models. Adv. Neural. Inf. Process. Syst. **34**, 24379–24391 (2021)
14. Oliehoek, F.A., Amato, C.: A Concise Introduction to Decentralized POMDPs. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-28929-8
15. OroojlooyJadid, A., Hajinezhad, D.: A review of cooperative multi-agent deep reinforcement learning. arXiv preprint arXiv:1908.03963 (2019)
16. Paszke, A., et al.: Pytorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
17. Pasztor, B., Bogunovic, I., Krause, A.: Efficient model-based multi-agent mean-field reinforcement learning. arXiv preprint arXiv:2107.04050 (2021)
18. Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., Whiteson, S.: Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In: International Conference on Machine Learning, pp. 4295–4304. PMLR (2018)

19. Samvelyan, M., et al.: The starcraft multi-agent challenge. arXiv preprint arXiv:1902.04043 (2019)
20. Sunehag, P., et al.: Value-decomposition networks for cooperative multi-agent learning. arXiv preprint arXiv:1706.05296 (2017)
21. Wang, T., Dong, H., Lesser, V.R., Zhang, C.: Roma: multi-agent reinforcement learning with emergent roles. arXiv:abs/2003.08039 (2020)
22. Wang, T., Gupta, T., Mahajan, A., Peng, B., Whiteson, S., Zhang, C.: Rode: learning roles to decompose multi-agent tasks. arXiv:abs/2010.01523 (2021)
23. Wang, T., Du, S.S., Torralba, A., Isola, P., Zhang, A., Tian, Y.: Denoised MDPs: learning world models better than the world itself. arXiv preprint arXiv:2206.15477 (2022)
24. Wang, X., Zhang, Z., Zhang, W.: Model-based multi-agent reinforcement learning: Recent progress and prospects. arXiv preprint arXiv:2203.10603 (2022)
25. Wen, M., et al.: Multi-agent reinforcement learning is a sequence modeling problem. arXiv preprint arXiv:2205.14953 (2022)
26. Wu, P., Escontrela, A., Hafner, D., Goldberg, K., Abbeel, P.: Daydreamer: world models for physical robot learning. arXiv preprint arXiv:2206.14176 (2022)
27. Xu, Y., et al.: Learning general world models in a handful of reward-free deployments. arXiv preprint arXiv:2210.12719 (2022)
28. Yarats, D., Zhang, A., Kostrikov, I., Amos, B., Pineau, J., Fergus, R.: Improving sample efficiency in model-free reinforcement learning from images. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 10674–10681 (2021)
29. Ye, Z., Chen, Y., Jiang, X., Song, G., Yang, B., Fan, S.: Improving sample efficiency in multi-agent actor-critic methods. Appl. Intell. **52**(4), 3691–3704 (2022)
30. Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., Wu, Y.: The surprising effectiveness of PPO in cooperative, multi-agent games. arXiv preprint arXiv:2103.01955 (2021)
31. Zhang, K., Yang, Z., Başar, T.: Multi-agent reinforcement learning: a selective overview of theories and algorithms. In: Handbook of Reinforcement Learning and Control, pp. 321–384 (2021)
32. Zhou, M., et al.: Smarts: scalable multi-agent reinforcement learning training school for autonomous driving. arXiv preprint arXiv:2010.09776 (2020)