Efficient Information Sharing for Training Decentralized Multi-Agent World Models

Anonymous authors
Paper under double-blind review

Keywords: cooperative multi-agent reinforcement learning, decentralized world models, multi-agent communication.

Summary

World models, which were originally developed for single-agent reinforcement learning, have recently been extended to multi-agent settings. Due to unique challenges in multi-agent reinforcement learning, agents' independently training of their world models often leads to underperforming policies, and therefore existing work has largely been limited to the centralized training framework that requires excessive communication. As communication is key, we ask the question of how the agents should communicate efficiently to train and learn policies from their decentralized world models. We address this question progressively. We first allow the agents to communicate with unlimited bandwidth to identify which algorithmic components would benefit the most from what types of communication. Then, we restrict the inter-agent communication with a predetermined bandwidth limit to challenge the agents to communicate efficiently. Our algorithmic innovations develop a scheme that prioritizes important information to share while respecting the bandwidth limit. The resulting method yields superior sample efficiency, sometimes even over centralized training baselines, in a range of cooperative multi-agent reinforcement learning benchmarks.

Contribution(s)

1. This paper proposes a model-based MARL method that explicitly considers communication in both the world model and the actor-critic training stages, and analyzes the impact of communication bandwidth on decentralized training.
   Context: Previous work either bases on model-free MARL, discussing only experience sharing under different bandwidths with limited model and information diversity, or extracts shared agent information as features for centralized training, but omits these features during decentralized execution (Gerstgrasser et al., 2023; Venugopal et al., 2023).
2. Our experiment comprehensively studies information sharing under bandwidth limitations, and optimizes the efficiency of information transmission while guarantee the performance under the decentralized framework.
   Context: Existing method doesn't to effectively deal with communication bandwidth constraints, and rely on Euclidean distance constraints to filter communication neighbors (Toledo & Prorok, 2024).

Efficient Information Sharing for Training Decentralized Multi-Agent World Models

Anonymous authors
Paper under double-blind review

Abstract

World models, which were originally developed for single-agent reinforcement learning, have recently been extended to multi-agent settings. Due to unique challenges in multi-agent reinforcement learning, agents' independently training of their world models often leads to underperforming policies, and therefore existing work has largely been limited to the centralized training framework that requires excessive communication. As communication is key, we ask the question of how the agents should communicate efficiently to train and learn policies from their decentralized world models. We address this question progressively. We first allow the agents to communicate with unlimited bandwidth to identify which algorithmic components would benefit the most from what types of communication. Then, we restrict the inter-agent communication with a predetermined bandwidth limit to challenge the agents to communicate efficiently. Our algorithmic innovations develop a scheme that prioritizes important information to share while respecting the bandwidth limit. The resulting method yields superior sample efficiency, sometimes even over centralized training baselines, in a range of cooperative multi-agent reinforcement learning benchmarks.

1 Introduction

In model-based reinforcement learning (RL), the agent learns a world model that encode its raw observations to latent states in a way that effectively recovers/predicts the observations, rewards, and future latent state dynamics. This model-based framework has contributed algorithms that have been shown to greatly improve sample efficiency for single-agent RL (Hafner et al., 2019b;a; Schrittwieser et al., 2020; Ye et al., 2021). In this paper, we are interested in extending the success of world models to the setting of cooperative multi-agent reinforcement learning (MARL) where a team of agents collectively interact an environment to achieve a shared goal, which finds a wide range of applications such as video games (Vinyals et al., 2019), traffic and vehicle control (Chu et al., 2019; Dinneweth et al., 2022), and multi-robot systems (Corke et al., 2005). Such scenarios introduce additional challenges on top of single-agent RL, such as partial observability when the agents only partially observe the environment (Oliehoek et al., 2016) and non-stationarity as all agents concurrently update their policies during training, causing the environment dynamics to continuously change from the perspective of any individual agent (Hernandez-Leal et al., 2017). Due to these challenges, existing success in learning multi-agent world models has been largely relying on the centralized training approach, where a single world model is trained and shared by all agents (Egorov & Shpilman, 2022; Venugopal et al., 2023).

Although effective, centralized training requires excessive inter-agent communication, limiting its applicability and scalability. On the other hand, as confirmed in prior work (Toledo & Prorok, 2024) and this paper, independent learning of multi-agent world models without explicit communication results in ineffective multi-agent policies after planning with the world models. This raises a critical question: *How should the agents communicate efficiently to train and learn policies from their decentralized world models?* Addressing this question is particularly challenging, since world models

consist of intricate and inter-dependent components that involve various types of information for communication, which is further complicated by the bandwidth limit that often exists in practice. Adopting DreamerV2 (Hafner et al., 2020) as the architecture backbone of our decentralized world models, this work addresses the question with a two-stage study.

In the first stage, we allow the agents with unlimited communication bandwidth so that we can focus on identifying which algorithmic components would benefit the most from what types of communication. Specifically, we separately allow the information shared in an unlimited manner between the agents for the decentralized training of their local world models and actor-critic networks, respectively. Evaluated on the cooperative MARL benchmark of SMAC, both types of information sharing yield multi-agent policies that outperform 1) the no communication baseline by a large margin and 2) the centralized training baseline some SMAC scenarios. This might be surprising as centralized training is widely considered as a performance upper bound. Encouraged by the results from the first stage, the second stage restricts the inter-agent communication with a predetermined bandwidth limit, which further challenges the agents to efficiently communicate with selective information. By experimenting with various bandwidths ranging from small to the largest (i.e., unlimited), our results show that there exists a relatively small bandwidth that works well for both types of information sharing, the performance of which is comparable or even better than that with unlimited bandwidth.

2 Related work

Single- and multi-agent world models. One of the earliest model-based RL algorithms is Dyna (Sutton, 1991), in which the agent alternates between learning a world model of state dynamics with reward signals and planning with it to take an action. Dyna's framework has been adopted in many recent model-based RL algorithms including the Dreamer family (Hafner et al., 2019a; 2020; 2023), which is known for their effectiveness in addressing partial observability and simplicity of training a policy from the learned world model. This paper and most recent works on multi-agent world models use Dreamer as the architecture backbone. Egorov & Shpilman (2022) develop MAMBA, a centralized training and centralized execution framework where all agents share their local observations in both the global world model and the local policies. Adapting MAMBA, Venugopal et al. (2023) introduce MABL that employs a bi-level hierarchy to enhance the agents' understanding of global information during centralized world model training, while enabling fully distributed local policies for execution. Xu et al. (2022) consider model-based cooperative MARL in the centralized training framework of value decomposition. Contrastive to these works, in this work each agent maintains a local world model and learns it in a decentralized manner with inter-agent communication.

Decentralized MARL with communication. Due to challenges such as partial observability and non-stationarity, effective training of cooperative MARL agents requires either centralization like a centralized critic (Lowe et al., 2017; Chu et al., 2019; Rashid et al., 2020) and a global world model (Egorov & Shpilman, 2022; Venugopal et al., 2023) or inter-agent communication of learnable parameters (Chen et al., 2022), experiences of local trajectories (Christianos et al., 2020; Gerstgrasser et al., 2023), intents (Kim et al., 2020), etc. Closest to our work is CoDreamer (Toledo & Prorok, 2024) where the agents communicate over a graph to train a centralized world model that encodes and averages the agents' observations and actions with a graph neural network, and therefore it falls into the centralized training framework. In contrast, our work trains decentralized world models where each agent maintains its local world model and policy. Moreover, CoDreamer operates on a predefined graph for communication while our work focuses on achieving efficient communication where agents selective choose what information to share.

3 Preliminaries

Coorperative multi-agent reinforcement learning. We formalize multi-agent reinforcement learning with a decentralized partially observable Markov decision process (Dec-POMDP), denoted as $\langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, P, R, \{O^i\}_{i\in\mathcal{N}}, \gamma \rangle$, where $\mathcal{N} := \{1, \ldots, N\}$ represents the set of agents, $\mathcal{S}$

the state space, $\mathcal{A}^i$ the action space of agent i. At each time step t, every agent i selects an action $a_t^i \in \mathcal{A}^i$ to form an joint action $a_t = (a_1, \cdots, a_N) \in \prod_{i=1}^n \mathcal{A}^i =: \mathcal{A}$. The next state follows the distribution given by the state transition function $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ as $s_{t+1} \sim P(\cdot \mid s_t, a_t)$, where $\Delta(\mathcal{X})$ is the set of probability distributions over set $\mathcal{X}$. All agents receive the same reward according to the reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ as $r_t := R(s_t, a_t)$. The observation function $O^i : \mathcal{S} \to \mathcal{O}^i$ generates an observation for agent i from its observation space $\mathcal{O}^i$, denoted as $o_t^i := O^i(s_t)$. Each agent chooses actions by sampling from its (fully decentralized) policy $\pi^i(a_t^i|\tau_t^i)$ that is conditioned on its action-observation trajectory $\tau_t^i := (o_0^i, a_0^i, o_1^i, a_1^i, ..., o_t^i)$. Agents' individual policies form the joint policy, $\pi := (\pi^1, ..., \pi^N)$, and their goal is to find π that maximizes expected discounted cumulative rewards $\mathbb{E}_\pi[\sum_{r=0}^{\infty} \gamma^t r_t]$.

Latent-space world models. Many recent works on world models rely on recurrent state-space models (RSSMs). We here review the core components of DreamerV2's RSSMs in the single-agent case. Initializing the latent state as h_0, an RSSM encodes the agent's action-observation trajectory $(o_0, a_0, ..., o_{t-1}, a_{t-1})$ into latent state h_t with a recurrent model f as $h_t = f(h_{t-1}, z_{t-1}, a_{t-1})$ where $z_t \sim q(z_t|h_t, o_t)$ is the encoded observation o_t (conditioned on h_t) by a (stochastic) representation model q. Paired with a (stochastic) transition model $\hat{z}_t \sim p(\hat{z}_t|h_t)$ and a (stochastic) observation model $\hat{o}_t \sim p(\hat{o}_t|h_t, z_t)$, all models f, q, and p are parameterized by neural networks and trained by maximizing the evidence lower bound (ELBO) of the log probability of $\log p(o_{0:T}|a_{0:T-1})$.

4 Method

4.1 Information sharing without bandwidth constraints

In the decentralized training and decentralized execution framework, each agent operates independently and accumulating distinct experiences. However, the constantly changing policies of other agents in the environment lead to inherent instability in the agent's learning process. To address this, we enable agents to exchange valuable experiences, so as to facilitate the optimization of their objective functions. Our approach first extends standard DreamerV2 to the MARL setting, then focusing on efficient information sharing and model optimization.

Centralized training. We establish an upper bound for information sharing using a centralized training and decentralized execution paradigm, where all agents share the same world model and actor-critic network parameters ϕ and θ. This approach is based on DreamerV2 by incorporating a Recurrent State Space Model (RSSM), reconstruction models and prediction models based on latent variables.

$$\text{RSSM:} \begin{cases} \text{Recurrent model:} & h_t^i = f_\phi(h_t^i \mid h_{t-1}^i, z_{t-1}^i, a_{t-1}^i) \\ \text{Representation model:} & z_t^i = q_\phi(z_t^i \mid o_t^i, h_t^i) \\ \text{Transition model:} & \hat{z}_t^i = p_\phi(\hat{z}_t^i \mid h_t^i) \end{cases}$$

where, the RSSM is a framework for learning latent dynamics, composed of a recurrent model maintains historical dependencies, a representation model infers posterior distributions, and a transition model predicts future states, facilitating synthetic trajectory generation.

In addition, we categorize the auxiliary components in the world model into two types: decoders and predictors. The decoders are responsible for decoding the latent representation back into actual behavior and perception, ensuring that the latent space contains sufficient environmental and agent-specific information. In contrast, predictors are used to infer future dynamics based on the current latent states.

$$\text{Decoders:} \begin{cases} \text{Observation decoder:} & \hat{o}_t^i = p_\phi(\hat{o}_t^i \mid h_t^i, z_t^i) \\ \text{Action decoder:} & \hat{a}_t^i = p_\phi(\hat{a}_t^i \mid h_t^i, z_t^i) \end{cases}$$

$$\text{Predictors:} \begin{cases} \text{Reward predictor:} & \hat{r}_t^i = p_\phi(\hat{r}_t^i \mid h_t^i, z_t^i) \\ \text{Termination predictor:} & \hat{\gamma}_t^i = p_\phi(\hat{\gamma}_t^i \mid h_t^i, z_t^i) \\ \text{Available action predictor:} & \hat{A}_t^i = p_\phi(\hat{A}_t^i \mid h_t^i, z_t^i) \end{cases}$$

To generate more effective synthetic trajectories for policy optimization, the predictors assist training by providing important feedback signals, including a reward predictor outputs a continuous reward value $\hat{r}_t^i$, while the termination predictor outputs a binary variable $\hat{\gamma}_t^i$ to indicate whether the current state is terminal. The available action predictor outputs a vector $\hat{A}_t^i$ of size A, where each element indicates whether the corresponding action is available at time step t. Thus, both the termination and available action predictors use Bernoulli distributions.

As described in previous section, we optimize the decoders and predictors using supervised learning, and optimize the RSSM model by maximizing the ELBO. For a trajectory of length T from agent i, the loss $L_{ELBO} = L_{\hat{o}t} + D_{KL}$ is computed as the expectation with respect to the posterior distribution $q_\phi(z_{1:T}^i \mid o_{1:T}^i, a_{1:T}^i)$, so as to maximize the reconstruction accuracy and align the prior distribution p_ϕ with the posterior distribution q_ϕ. The loss functions are:

$$L_{\hat{o}_t} = -\sum_i \sum_t \log p_\phi\big(\hat{o}_t^i \mid h_t^i, z_t^i\big), \quad L_{\hat{a}_t} = -\sum_i \sum_t \log p_\phi\big(\hat{a}_t^i \mid h_t^i, z_t^i\big) \tag{1}$$

$$L_{\hat{r}_t} = -\sum_i \sum_t \log p_\phi\big(\hat{r}_t^i \mid h_t^i, z_t^i\big), \quad L_{\hat{\gamma}_t} = -\sum_i \sum_t \log p_\phi\big(\hat{\gamma}_t^i \mid h_t^i, z_t^i\big) \tag{2}$$

$$L_{\hat{A}_t} = -\sum_i \sum_t \log p_\phi(\hat{A}_t^i \mid h_t^i, z_t^i), \quad D_{KL} = \sum_i \sum_t KL[q_\phi(z_t^i \mid o_t^i, h_t^i) \parallel p_\phi(\hat{z}_t^i \mid h_t^i)]. \tag{3}$$

The model-based approach effectively decouples model learning from policy learning in MARL. Once the world model is trained, it generates imagined trajectories for policy optimization. We employ an actor-critic framework to enhance agents' decision-making and coordination. Each agent utilizes the shared parameter policy network π_θ, and the objective is to maximize the cumulative MARL returns, thereby learning an optimal policy. Specifically, at time step t, the agent selects an action based on the following:

$$a_t^i \sim \pi_\theta(a_t^i \mid \hat{z}_t^i, h_t^i). \tag{4}$$

Here, the agent performs policy inference based on its own hidden state vector h_t^i and the inferred prior distribution $\hat{z}_t^i$. And the shared critic V_ϕ estimates each agent's value function to guide policy optimization based on:

$$\hat{V}_t^i \sim V_\phi(\hat{z}_t^i, h_t^i). \tag{5}$$

In this centralized training method, all agents share the parameters of the world models, actor and critic networks, enabling efficient learning and coordination for multi-agent systems.

Independent training. In order to verify the impact of information sharing in fully decentralized MARL, we introduce a lower bound method, which follows an independent training paradigm. Initially, all agents have the identical architectures and optimizers of world model and the actor-critic models, along with the same parameters ϕ_0^i and θ_0^i. During training, each agent treats other agents as part of their environment, builds its own world model and optimizes policy without sharing parameters or information, and follows the previously introduced loss functions but iterating solely over time steps.

Sharing experiences across the RSSMs and predictors. We first propose the RSSM+Predictors method, in which part of the world models share experiences while others remain independent. Specifically, the RSSM and predictors are trained by sampling from a shared experience replay buffer used by all agents. This global perspective allows each agent to not only rely on its own experience for training, but also capture global dynamics and environmental features across agents, thereby enhancing the understanding of environmental patterns. On the other hand, each agent's decoders depend entirely on its own independent experience, ensuring that each agent can optimize its

decoders based on its own perspective and behavioral patterns, thus preventing a decline in decoding accuracy due to differences in experiences. Through this design, the RSSM and predictors focus on learning general latent-space representations and predicting accurate environmental dynamics, while the decoders focus on generating predictions of observations and actions based on the agent's own experience.

Algorithm 1 illustrates how experiences are shared for training the RSSM and predictor models. The method contains two types of buffers: one shared experience buffer and N independent experience buffers. During the experience collection phase, each agent stores its observations, actions, rewards, and other feedback information into the shared buffer and its independent buffer. During training, the RSSM and predictors update using samples from the shared buffer, while decoders optimize only with agent-specific buffer.

Algorithm 1 Training World Model with Shared and Individual Experience Buffers

Initialize SharedReplayBuffer
Initialize N IndividualReplayBuffers
for $t = 1$ **to** T **do**
  for each agent $i = 1$ **to** N **do**
    SharedReplayBuffer.Add($(o_t^i, a_t^i, r_t^i, \gamma_t^i, A_t^i)$)
    IndividualReplayBuffers$[i]$.Add($(o_t^i, a_t^i, r_t^i, \gamma_t^i, A_t^i)$)
  end for
  for each agent $i = 1$ **to** N **do**
    Train RSSM and predictors using shared experience:
      GradientStep(ϕ^i, Eq 1) on samples from SharedReplayBuffer
    Train individual decoders using agent-specific experience:
      GradientStep(ϕ^i, Equation 2 to 3) on samples from IndividualReplayBuffers$[i]$
    Train individual actor and critic using agent-specific experience:
      IndividualReplayBuffers$[i]$.Sample(batch_size)
      $\hat{a}_{1:H}^i, \hat{A}_{1:H}^i, logit_{1:H}^i, \hat{r}_{1:H}^i, h_{1:H}^i, z_1^i, \hat{z}_{2:H}^i$ = ImaginationRollout$(o_t^i, a_t^i, \gamma_t^i)$
      GradientStep[(θ^i, Equation 4), (ϕ^i, Equation 5)]
  end for
end for

Sharing rollouts across the actor-critic networks. In this method, the optimization of the actor and critic networks is achieved through rollouts sharing. Specifically, each agent trains its independent world model based on its own experience and generates synthetic trajectories over a certain horizon with length H. As shown in Algorithm 2, these trajectories include not only sequences of actions $\hat{a}_{1:H}^i$ and rewards $\hat{r}_{1:H}^i$, but also incorporate latent state information $h_{1:H}^i$, z_1^i, and $\hat{z}_{2:H}^i$, and auxiliary information $\hat{A}_{1:H}^i$ and $logit_{1:H}^i$, where the $logit_h^i$ denotes the logits corresponding to $\hat{a}_h^i$, ensuring differentiability. Subsequently, all agents share their generated synthetic trajectories and the aggregated multi-source trajectory data is used for training their actor-critic networks.

On one hand, fully utilizing model-generated data alleviates the issue of high interaction costs, improving sample efficiency and exploration capability. On the other hand, this sharing mechanism enables agents to perform policy learning on a wider data distribution, thereby enhancing decision-making effects and accelerating the convergence of the training process.

4.2 Information sharing with bandwidth constraints

Even though communication between agents can be unlimited, considering communication efficiency and information priority, we focus on adjusting the priority of information transmission to avoid redundant information that may delay or hinder model training. In the previous section, we investigated the effectiveness of information sharing in model-based MARL. In this section, we fur-

Algorithm 2 Training Actor-Critic with Shared Imagination Buffers

Initialize N IndividualReplayBuffers
Initialize SharedImaginationBuffer
for $t = 1$ **to** T **do**
  for each agent $i = 1$ **to** N **do**
    Train individual world model using agent-specific experience:
      IndividualReplayBuffers$[i]$.Sample(batch_size)
      GradientStep(ϕ^i)
  end for
  for each agent $i = 1$ **to** N **do**
    Aggregate imagination rollouts from all agents:
      IndividualReplayBuffers$[i]$.Sample(batch_size)
      $\hat{a}^i_{1:H}, \hat{A}^i_{1:H}, logit^i_{1:H}, \hat{r}^i_{1:H}, h^i_{1:H}, z^i_1, \hat{z}^i_{2:H}$ = ImaginationRollout$(o^i_t, a^i_t, \gamma^i_t)$
      SharedImaginationBuffer.Add$(\hat{a}^i_{1:H}, \hat{A}^i_{1:H}, logit^i_{1:H}, \hat{r}^i_{1:H}, h^i_{1:H}, z^i_1, \hat{z}^i_{2:H})$
  end for
  for each agent $i = 1$ **to** N **do**
    Train actor and critic using shared imagination:
      SharedImaginationBuffer.Sample(batch_size)
      GradientStep$[(\theta^i$, Equation 4$), (\phi^i$, Equation 5$)]$
  end for
end for

ther consider information selection under bandwidth constraints to enhance data-sharing efficiency and optimize communication resource utilization.

During world model training, experience sharing involves four key component models. However, due to significant variations in the loss distribution and scale across different models, applying a unified standard directly would lead to imbalanced sharing criteria. Therefore, we first use a multi-model loss normalization and aggregation strategy, ensuring fair and stable sample selection. At each training step, we first compute the loss values for the four models:$L_{\hat{A}_t}, L_{\hat{\gamma}_t}, L_{\hat{r}_t}, D_{KL}$. Then we maintain historical statistics for each loss function within a sliding window with length K, incuding the mean and standard deviation:

$$\mu_m = \frac{1}{K}\sum_{k=0}^{K} L_k, \qquad \sigma_m = \sqrt{\frac{1}{K}\sum_{k=0}^{K}\left(L_k - \mu_m\right)^2}.$$

Each loss is normalized as $L'_m = \frac{L_m - \mu_m}{\sigma_m + \epsilon}$ where ϵ is a small constant to prevent division by zero.

After normalization, we aggregate the four loss values into a single composite loss L_M for sample selection: $L_M = \max(L'_{\hat{A}_t}, L'_{\hat{\gamma}_t}, L'_{\hat{r}_t}, D'_{KL})$. This strategy ensures that if any single model exhibits an abnormally high loss, the corresponding sample is selected for sharing, reducing the risk of selection being dominated by a single model's loss and ensuring balanced multi-model learning. To adaptively control the number of shared samples, we apply the deterministic Gaussian experience selection based on a sliding window (Gerstgrasser et al., 2023). Specifically, we maintain the most recent K samples to track the distribution of the composite loss L_M, including its mean μ_M and standard deviation σ_M, then share the experience when:

$$L_M \geq \mu_M + c \cdot \sigma_M$$

where c is a constant determined based on the target bandwidth β, satisfying $1 - \text{cdf}N(c) = \beta$.

Experimental results (Figure 3) demonstrate that critic loss exhibits significant variation across different methods (Actor-Critic, Centralized Training, Independent Training), indicating that training strategies mainly impact the critic network. Therefore, critic loss is a reliable indicator of agent learning progress and policy evaluation accuracy. In contrast, actor loss exhibits less variation in

both magnitude and trend across different methods. Based on these observations, we adopt a critic loss based selection method in actor-critic training process. During training, we apply a similar strategy as used for the world model but exclusively utilize critic loss for selecting high informative samples. The statistics of critic loss in the sliding window are calculated as:

$$\mu_{\text{critic}} = \frac{1}{K}\sum_{k=0}^{K} L_k, \qquad \sigma_{\text{critic}} = \sqrt{\frac{1}{K}\sum_{k=0}^{K}\left(L_k - \mu_{\text{critic}}\right)^2}.$$

Then it applies sample sharing based on:

$$L_{\text{critic}} \geq \mu_{\text{critic}} + c \cdot \sigma_{\text{critic}}.$$

5 Experiments

To evaluate decentralized cooperation in multi-agent systems with experience and imagination rollout sharing, we use the StarCraft Multi-Agent Challenge (SMAC) benchmark, based on StarCraft II (Vinyals et al., 2017; Samvelyan et al., 2019). Each agent independently executes its policy while coordinating with other agents to defeat enemy units. Specifically, we conduct experiments on two micro-trick scenarios in which two agents face a single enemy (2s_vs_1sc and 2m_vs_1z), a homogeneous and symmetric scenario where both armies consist of three Marines (3m), and a heterogeneous and symmetric scenario where the allied team consists of two Stalkers and three Zealots, matching the enemy composition (2s3z). In each scenario, we perform ten independent runs for each method with an equal number of training steps. To ensure fair comparison, we ensure that the architecture and setup of all models are identical across all methods.

Baselines: In our experiments, we compare the proposed method with the independent training baseline to assess the impact of information sharing in decentralized cooperation and also to find out how it compares to the centralized training baseline.

5.1 Performance comparison without bandwidth constraints

To evaluate the effectiveness of two information sharing strategies, we first compare the performance of different methods across four SMAC scenarios without considering bandwidth constraints. Figure 1 presents the win rate curves of these four methods over training steps in different scenarios. To further analyze the underlying factors contributing to performance improvements, Figure 2 and Figure 3 present the loss curves for RSSM+Predictors and Actor-Critic models of different methods, respectively, in the 2s_vs_1sc scenario. And we choose to discuss only the loss curves of Agent-0 for the RSSM+Predictor models, as all agents exhibit similar patterns in the allied team.

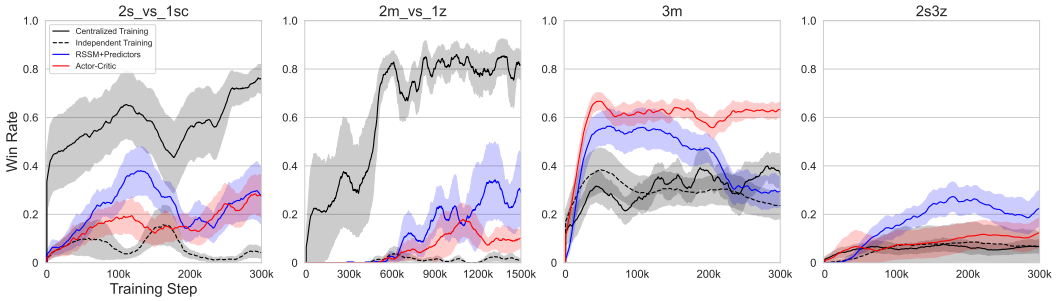


Figure 1: Win rate curves.

As shown in Figure 1, we can see that RSSM+Predictors (experience-sharing) and Actor-Critic (imagination rollouts sharing) methods consistently outperform Independent Training across all scenarios, achieving higher win rates and faster convergence. This improvement highlights the effectiveness of information sharing in facilitating superior coordination strategies, whereas Independent

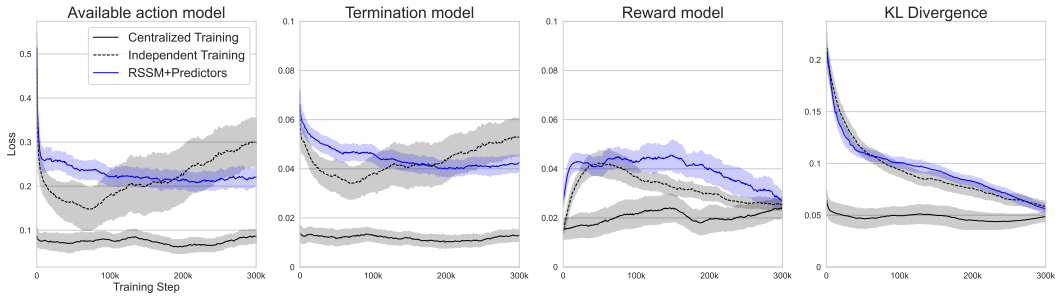


Figure 2: Loss curves of an agent on 2s_vs_1sc of different methods for RSSM+Predictor models.

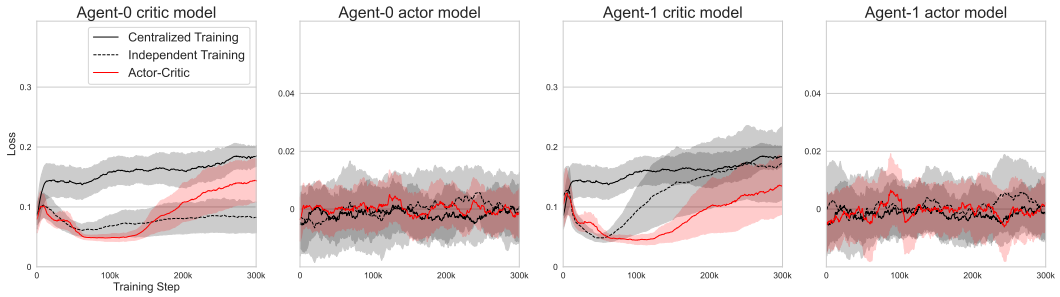


Figure 3: Loss curves on SMAC of different methods for Actor-Critic models.

Training struggles to achieve high win rates under fully decentralized learning. In the micro-trick asymmetric scenarios (2s_vs_1sc and 2m_vs_1z), agents need precise coordination to attack fewer enemies than themselves. Centralized Training enables all agents to learn and perform best through parameter sharing, thus shows excellent performance. However, in RSSM+Predictors and Actor-Critic methods, agents only share information and still optimize strategies independently, which is difficult to fully match the overall collaboration ability of Centralized Training, resulting in sub-optimal performance. Independent Training, due to its complete decentralization, agents cannot learn to cooperate effectively, resulting in the lowest win rate. For the homogeneous and heterogeneous symmetric scenarios (3m and 2s3z), we conjecture that Centralized Training may lead to unnecessary synchronous behavior due to parameter sharing. This could result in all agents attacking the same target simultaneously or retreating at the same time, potentially reducing combat efficiency. In contrast, RSSM+Predictors and Actor-Critics methods allow agents share information while independently optimize their policies, this might make them more adaptable when executed in a decentralized manner, thus surpassing Centralized Training. As shown in Figure 2, in 2s_vs_1sc scenario, the loss curves for the available action model, termination model, reward model and the KL divergence curves maintain lower loss values throughout training compared to Independent Training, even approaching the loss levels of Centralized Training. This indicates that experience-sharing strategy enhances the stability of the learning process and the accuracy of the prediction. Specifically, the KL divergence results show that RSSM+Predictors achieves better latent space alignment, demonstrating the effectiveness of experience sharing in approximating centralized training.

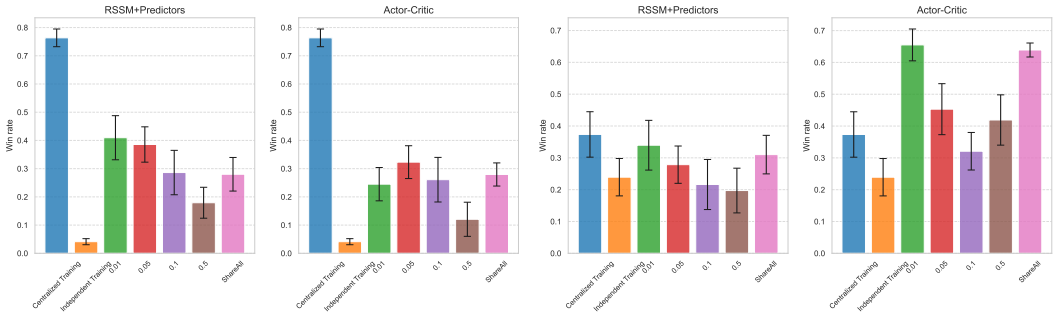
The actor-critic loss curves shown in Figure 3 provide another insight into the training dynamics of our method. For 2s_vs_1sc scenario, the critic loss curves for both Agent-0 and Agent-1 show distinct trends, whereas the actor loss curves remain similar in magnitude and trend across all methods. Specifically, the Independent Training method maintains a relatively lower critic loss while Centralized Training method exhibits a rising critic loss over time, indicating potential instability in value estimation due to indiscriminate sharing of imagination rollouts. And our Actor-Critic method shows an intermediate performance between Centralized Training and Independent Training methods that aligns with the win rate trends shown in Figure 1. For the actor, the loss curves show negligible differences across methods and maintain relatively stable training trends. This sug-

gests that behavior learning is less affected by the of rollouts sharing compared to value function estimation.

Overall, these results indicate that both experience and imagination rollouts sharing can benefit the multi-agent coordination in decentralized training.

5.2 Performance comparison with bandwidth constraints

In this section, we further investigate the impact of information sharing under different bandwidth constraints. And we compare the performance of RSSM+Predictors and Actor-Critic methods across various target bandwidths in the 2s_vs_1sc and 3m scenarios. As shown in Figure 4, the Centralized Training method achieves the highest performance, as it benefits from parameter sharing. In comparison, Independent Training performs as a lower bound, emphasizing the necessity of communication in multi-agent system. In particular, considering the limited bandwidth, we can observe a clear peak in RSSM+Predictors and Actor-Critic methods around the target bandwidth of 0.01 and 0.05 respectively for 2s_vs_1sc scenario. The results indicate that selective experience sharing can significantly promote learning process compare to Independent Training. Particularly, the ShareAll setting, where all information is shared without selection, does not necessarily lead to better performance compared to intermediate bandwidth values. This suggests that excessive communication may introduce redundant information, potentially hindering learning efficiency.



(a) 2s_vs_1sc (b) 3m

Figure 4: Comparison of RSSM+Predictors and Actor-Critic with different bandwidths.

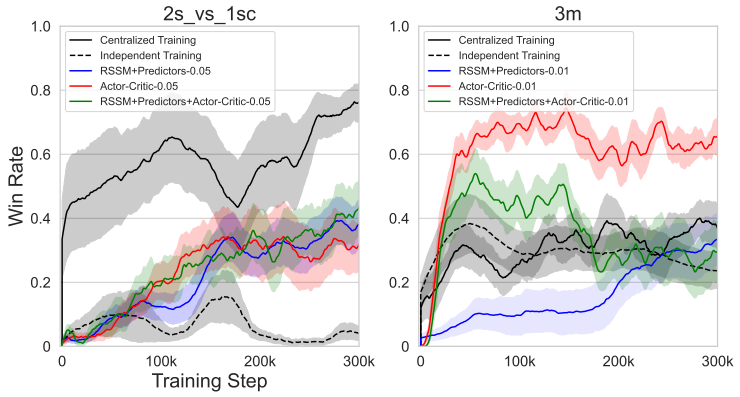


Figure 5: Comparison of RSSM+Predictors and Actor-Critic with optimal target bandwidths.

Figure 5 further shows the performance of different methods in 2s_vs_1sc and 3m scenarios. Specifically, based on Figure 4a and Figure 4b, we adopt the optimal target bandwidth of 0.05 and 0.01 for the 2s_vs_1sc and 3m scenarios, respectively. In 2s_vs_1sc, RSSM+Predictors-0.05

and Actor-Critic-0.05 outperform Independent Training, demonstrating the benefits of decentralized experience sharing, although they are still inferior to Centralized Training. The combination of RSSM+Predictors-0.05 and Actor-Critic-0.05 only provides similar performance, indicating an overlap of strengths rather than complementarity. In 3m scenario, RSSM+Predictors-0.01 and Actor-Critic-0.01 surpass Independent Training, with Actor-Critic-0.01 achieving the highest win rate, indicating that imagination rollouts sharing are particularly effective in symmetric multi-agent system.

Overall, our results demonstrate that Gaussian information selection significantly enhances performance compared to Independent Training, even under bandwidth constraints. Moreover, the presence of a pronounced peak suggests that excessive experience sharing does not always yield better performances, and an optimal balance between communication and independent learning is crucial for effective multi-agent coordination.

6 Conclusion

In this paper, we investigate information-sharing strategies in model-based MARL. Under a decentralized training and decentralized execution framework, we explore the effectiveness of selective communication in both the world model and actor-critic training process. Our methods first decompose the world model, where the RSSM and predictors leverage shared experiences for training, while decoders learn agent-specific features. Then we examine the performance of RSSM+Predictors and Actor-Critic methods under varying target bandwidths. Experimental results in four SMAC scenarios demonstrate that, compared to independent training, both RSSM+Predictors and Actor-Critic achieve superior performance across different bandwidth settings. Notably, in 2s_vs_1sc and 3m scenarios, the peak performance is observed at target bandwidths of 0.05 and 0.01 respectively, showing that full sharing is not always optimal. Moreover, applying both RSSM+Predictors and Actor-Critic with optimal bandwidth simultaneously does not simply lead to additive improvements but instead results in an overlapping. Our experiments highlight that selective information sharing can effectively enhance decentralized multi-agent cooperation.

References

Dingyang Chen, Yile Li, and Qi Zhang. Communication-efficient actor-critic methods for homogeneous markov games. *arXiv preprint arXiv:2202.09422*, 2022.

Filippos Christianos, Lukas Schäfer, and Stefano Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10707–10717, 2020.

Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE transactions on intelligent transportation systems*, 21(3): 1086–1095, 2019.

Peter Corke, Ron Peterson, and Daniela Rus. Networked robots: Flying robot navigation using a sensor net. In *Robotics research. The eleventh international symposium*, pp. 234–243. Springer, 2005.

Joris Dinneweth, Abderrahmane Boubezoul, René Mandiau, and Stéphane Espié. Multi-agent reinforcement learning for autonomous vehicles: A survey. *Autonomous Intelligent Systems*, 2(1):27, 2022.

Vladimir Egorov and Aleksei Shpilman. Scalable multi-agent model-based reinforcement learning. *arXiv preprint arXiv:2205.15023*, 2022.

Matthias Gerstgrasser, Tom Danino, and Sarah Keren. Selectively sharing experiences improves multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36: 59543–59565, 2023.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019a.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019b.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz De Cote. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*, 2017.

Woojun Kim, Jongeui Park, and Youngchul Sung. Communication in multi-agent reinforcement learning: Intention sharing. In *International conference on learning representations*, 2020.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Edan Toledo and Amanda Prorok. Codreamer: Communication-based decentralised world models. *arXiv preprint arXiv:2406.13600*, 2024.

Aravind Venugopal, Stephanie Milani, Fei Fang, and Balaraman Ravindran. Mabl: Bi-level latent-variable world model for sample-efficient multi-agent reinforcement learning. *arXiv preprint arXiv:2304.06011*, 2023.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

Zhiwei Xu, Bin Zhang, Yuan Zhan, Yunpeng Baiia, Guoliang Fan, et al. Mingling foresight with imagination: Model-based cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:11327–11340, 2022.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Advances in neural information processing systems*, 34:25476–25488, 2021.