Multi-Agent Model-Based Reinforcement Learning with Joint State-Action Learned Embeddings

Zhizun Wang
McGill University
Montreal, Canada
zhizun.wang@mail.mcgill.ca

David Meger
McGill University
Montreal, Canada
david.meger@mcgill.ca

ABSTRACT

Learning to coordinate many agents in partially observable and highly dynamic environments requires both informative representations and data-efficient training. To address this challenge, we present a novel model-based multi-agent reinforcement learning framework that unifies joint state-action representation learning with imaginative roll-outs. We design a world model trained with variational auto-encoders and augment the model using the state-action learned embedding (SALE). SALE is injected into both the imagination module that forecasts plausible future roll-outs and the joint agent network whose individual action values are combined through a mixing network to estimate the joint action-value function. By coupling imagined trajectories with SALE-based action values, the agents acquire a richer understanding of how their choices influence collective outcomes, leading to improved long-term planning and optimization under limited real-environment interactions. Empirical studies on well-established multi-agent benchmarks, including StarCraft II Micro-Management, Multi-Agent MuJoCo, and Level-Based Foraging challenges, demonstrate consistent gains of our method over baseline algorithms and highlight the effectiveness of joint state-action learned embeddings within a multi-agent model-based paradigm.

KEYWORDS

Multi-Agent Reinforcement Learning; Representation Learning

ACM Reference Format:
Zhizun Wang and David Meger. 2026. Multi-Agent Model-Based Reinforcement Learning with Joint State-Action Learned Embeddings. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 10 pages. https://doi.org/10.65109/IXMJ8234

1 INTRODUCTION

Reinforcement learning (RL) has been established as a fundamental approach for sequential decision-making in complex and uncertain environments. In particular, multi-agent reinforcement learning (MARL) extends RL to scenarios where multiple autonomous agents interact with each other in a shared environment, often requiring collaboration to accomplish a global objective [78, 81]. Due to partial observability, scalability, and non-stationarity issues posed by multi-agent systems [18, 76], model-free MARL algorithms may struggle to achieve sample efficiency and generalization ability [29, 39, 54, 66]. One promising solution to sample inefficiency in single-agent RL is feature learning [16, 17]. [16] proposes *state-action learned embeddings* (SALE), learning a predictive embedding space that captures how actions causally affect the states. SALE jointly constructs state and action representations, effectively learning the transition dynamics in a compact latent space. This allows for more structured representations and opens the door to imagination-based planning. As an alternative solution, model-based reinforcement learning (MBRL) has also demonstrated sample efficiency in solving complex single-agent tasks [23–25, 31, 74], since it requires a much smaller number of samples for training compared to model-free RL [40].

In this paper, we leverage joint state-action representation learning in model-based RL to address the sample complexity problem in multi-agent systems. More specifically, we incorporate SALE into a world model that learns the dynamics of the real environment and generates informative latent space roll-outs. We then present Multi-Agent Model-Based Framework with Joint State-Action Learned Embeddings (MMSA), a novel MARL framework in which the world model works as an imagination module. The predictions of the world model are learned by variational auto-encoders (VAEs). In this framework, we also propose applying SALE to the joint policy network and the joint agent network representing all agents in the environment. Aggregating the world model roll-outs with the global state and the individual action values from the agent network, we pass them into a mixing network [53] that employs value factorization for approximating joint action-value functions. Our framework benefits from SALE because it can capture the underlying dynamics between states and actions. It efficiently extracts meaningful features from the latent dynamics, leading to more effective and stable model learning. Moreover, our method uses the mixing network following the paradigm of centralized training with decentralized execution (CTDE) [34], ensuring efficient coordination among multiple agents in complex systems [53].

The key contributions of our paper are summarized as follows:

• **Dynamics model for sample efficiency.** We fuse two effective methods for reducing sample complexity, namely model-based RL and representation learning, by constructing a world model equipped with SALE. Reinforced by joint state-action learned features, the model can supply agents with structured and informative latent representations that markedly improve sample efficiency.

• **Unified MARL framework with imagination-based value decomposition.** SALE was initially proposed to improve TD7 [16], and the mixing network was originally part of QMIX [54], but both TD7 and QMIX are model-free RL algorithms. In contrast, we propose MMSA, a model-based framework that uses a world model

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/IXMJ8234

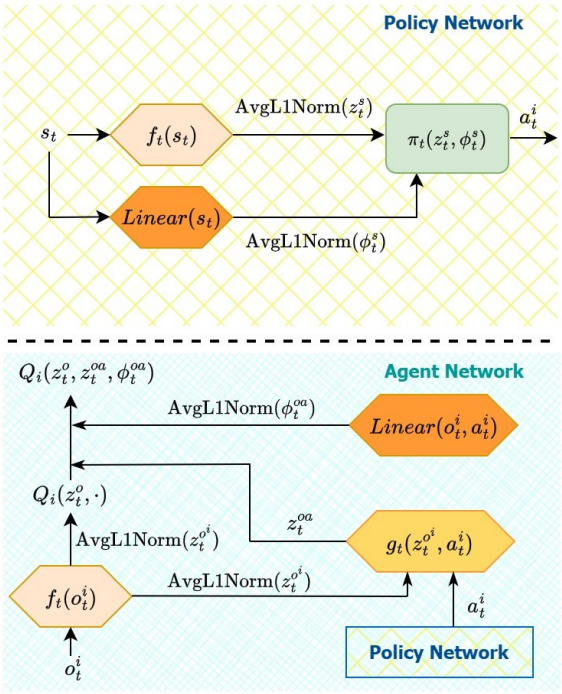

Figure 1: Architecture of the SALE-augmented policy and agent networks in MMSA. Top: the policy network in which the state s_t is encoded and passed into π_t to produce the action. Bottom: the agent network, which encodes the observation and action for computing $Q_i(z_t^o,\ z_t^{oa},\ \phi_t^{oa})$.

to augment the mixing network with simulated trajectories. Each individual agent in the joint agent network is also enhanced with SALE. This means the individual action-value function considers the learned representations from SALE in addition to the local action-observation history. The mixing network aggregates individual action values into a global estimate, preserving the CTDE paradigm.

• **Comprehensive benchmarking and ablation studies.** We evaluate MMSA on three widely used test beds: Multi-Agent MuJoCo [50], StarCraft II MARL benchmark [12, 56], and Level-Based Foraging [7]. Across a rich variety of environments, we have constantly observed the performance of MMSA matching or exceeding the competitive baselines. Moreover, we conduct in-depth design studies to explore the optimal design choices and present systematic ablations to demonstrate that every architectural ingredient contributes meaningfully to the overall performance gains.

2 RELATED WORK

Model-Based Reinforcement Learning (MBRL). MBRL represents a significant paradigm in RL, distinguished by its utilization of an explicit model of the environment [10, 40, 63]. The environment model refers to the abstraction of the environment dynamics, which can be formulated as a Markov decision process (MDP) [60, 64]. The model can function as a simulation module that helps to choose the correct actions from imagination [25, 74]. Unlike model-free approaches, which rely solely on direct interactions with the environment, MBRL predicts future transitions and rewards, enhancing learning efficiency and decision-making [8, 19, 31, 71, 75]. Built on the foundation of MBRL, the world model [21, 22] assists the agents with accurate knowledge representations and effective behavior learning. This line of work evolves into recurrent state-space models (RSSM) [25] and the Dreamer family [23, 24, 26, 27], highlighting the power of coupling representation learning, predictive modeling, and policy optimization within a single algorithmic framework. Recent efforts tackle limitations of MBRL frameworks and explore hybrid designs [6, 41, 58, 59]. [47] separates controllable factors from uncontrollable ones in visual control tasks. [51] combines a model-free policy path with a rollout encoder that takes simulated trajectories from a model-based path and employs a policy module to determine the imagination-augmented policy.

Multi-Agent Reinforcement Learning (MARL). MARL studies how multiple learners interact in a shared environment, where the agents are confronted with non-stationary dynamics, partial observability, and combinatorial complexities [5, 9, 18, 76, 81]. Analogous to the single-agent domain, two principal MARL algorithmic classes have emerged: value-based methods, which compute value function estimates of the agents [38, 45, 70], and policy gradient methods, which update the learning parameters along the direction of the gradient of specific metrics with respect to the policy parameter [7, 15, 29, 39, 50, 64, 78, 80]. Value-based MARL methods range from complete decentralization [66, 67] to full centralization [2, 20]. Recent studies focused on MARL algorithms that lie between the two extremes of centralization [52, 53, 61], according to the CTDE paradigm [34]. Value decomposition network (VDN) [62] constructs a factorizable joint value function Q_{tot} of the learning agents. QMIX [54] replaces the full factorization in VDN with the enforcement of monotonicity between Q_{tot} and individual $\tilde{Q}_i$, which enables it to represent a larger class of Q-functions than VDN.

Model-based MARL (or equivalently, multi-agent MBRL) is an emerging discipline with huge potential in real-world applications [11, 40, 57, 72, 76]. Early development of multi-agent MBRL has mainly focused on theoretical analyses [3, 4], and existing algorithms may rely heavily on specific prior knowledge, such as global states and opponent policies [1, 48, 78]. [42] devises model-based decentralized policy optimization, reducing reliance on global communication while maintaining performance. [42] focuses on large-scale network control problems that involve traffic networks, whereas our application lies in the areas of multi-agent gaming and robotic control. [49] proposes M^3-UCRL, integrating mean-field game theory with model-based exploration to obtain sub-linear regret bounds. [77] conducts a rigorous theoretical analysis on model-based MARL but only focuses on zero-sum Markov games. [78] introduces a framework to improve the efficiency of multi-agent policy optimization in competitive settings. However, the framework requires prior knowledge about opponents and therefore risks sizeable generalization errors when the assumptions break.

Representation Learning. Early studies framed representation learning as state abstraction, where bisimulation metrics or MDP homomorphisms are used to collapse an MDP into a smaller decision process [8, 13, 35]. For high-dimensional spaces, representation learning embeds observation data, such as images, into compact

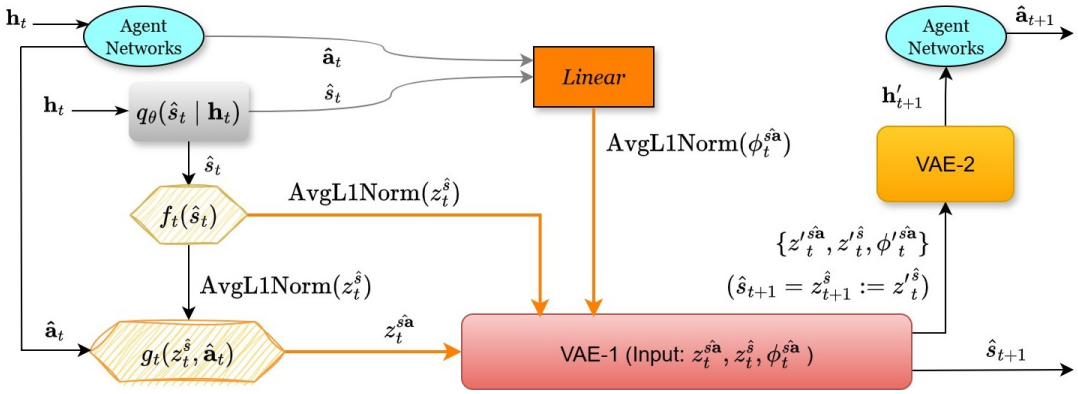

Figure 2: Illustration of the world model imagination in MMSA. The input $\mathbf{h}_t$ encapsulates the past information, including $\hat{s}_{t-1}$ and $\mathbf{a}_{t-1}$. The agent networks receive $\mathbf{h}_t$ and infer $\hat{\mathbf{a}}_t$. Taking the normalized joint state-action learned embeddings $(z_t^{\hat{s}\mathbf{a}}, z_t^{\hat{s}}, \phi_t^{\hat{s}\mathbf{a}})$ as input, VAE-1 reconstructs ${z'}_t^{\hat{s}\mathbf{a}}, {z'}_t^{\hat{s}}$, and ${\phi'}_t^{\hat{s}\mathbf{a}}$. The outputs are passed into VAE-2 to infer $\mathbf{h'}_{t+1}$.

latent vectors for control [14, 37, 43, 73]. Another mainstream interpretation is feature learning through auxiliary signals, which shape latent spaces towards the aspects of the environment most relevant for decision-making [30, 55, 65]. As a newly developed representation learning method built on OFENet [46], SALE [16] demonstrates the importance of learning low-level state-action representations in understanding the complexity of dynamical systems.

3 BACKGROUND

Dec-POMDP. The decentralized partially observable Markov decision process (Dec-POMDP) is appropriate for modeling collaborative agents in partially observable scenarios [44]. A Dec-POMDP is defined by a tuple $M := \langle \mathcal{S}, \mathcal{A}, \mathcal{N}, T, \Omega, O, R, \gamma \rangle$, where $\mathcal{S}$ is the state space of all agents, $\mathcal{A}$ is the joint action space of all agents, $\mathcal{N} := \{1, ..., N\}$ represents the set of N agents, T is the state transition function, Ω represents the observation space, O is the observation function, R is the reward function, and $\gamma \in [0, 1]$ represents the discount factor with respect to time. At time step t, each agent $i \in \mathcal{N}$ chooses an action a^i from its own action space $\mathcal{A}^i$ to form the joint action $\mathbf{a} \in \mathcal{A}$, where $\mathbf{a} := (a^1, ..., a^N)$ and $\mathcal{A} := \times_{i \in \mathcal{N}} \mathcal{A}^i$. Then the environment moves from s to s' based on $T(s'|s, \mathbf{a}) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$. Every agent i draws an observation $o \in \Omega$ according to $O(s, i) : \mathcal{S} \times \mathcal{N} \to \Omega$ because of partial observability. i has its own action-observation history, denoted by $\tau^i \in \mathcal{T}^i := (\Omega \times \mathcal{A}^i)^*$, and selects a^i by its policy $\pi^i(a^i|\tau^i) : \mathcal{T}^i \times \mathcal{A}^i \to [0, 1]$. The learning goal is to maximize the expected return by optimizing the joint policy $\pi := (\pi^1, ..., \pi^N)$. The joint action-value function of π is written as

$$Q^\pi(s_t, \mathbf{a}_t) = \mathbb{E}_{s_{t+1:\infty}, \mathbf{a}_{t+1:\infty}}[G_t | s_t, \mathbf{a}_t],$$

where $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the discounted return; r_{t+k} is the reward computed by R for all agents at time step $t + k$.

4 METHOD

We propose a multi-agent MBRL approach named Multi-Agent Model-Based Framework with Joint State-Action Learned Embeddings (MMSA). Our method couples a world model augmented by state-action representation learning and a value decomposition framework under the CTDE paradigm. We begin by formalizing the amortized variational inference problem in Dec-POMDPs, deriving the corresponding evidence lower bound (ELBO), and analyzing the optimization process for the ELBO. In the following subsection, we introduce two key components for MMSA: a policy network that maps state embeddings to actions and an agent value network that evaluates local action-value functions before they are combined by a monotonic mixing network. We then elaborate on the world model, which employs SALE encoders to generate roll-outs in latent space, providing synthetic experience without additional environment interaction. The details of the full MMSA workflow are presented next, showing how these modules interact under the CTDE paradigm. The section concludes with the unified learning objective, which involves loss functions for KL regularization, VAE reconstruction, temporal-difference (TD), and SALE prediction.

4.1 Deriving and Optimizing the Variational Lower Bound

To model the dynamics of multi-agent systems and perform model learning, we need to analyze how the evidence lower bound (ELBO) for the latent state inference in the world model can be deduced and optimized when the system is modeled as a Dec-POMDP. We first denote $\hat{s}_t$ as an abstraction of the agents' local observations $\mathbf{o}_t$, which implies $\mathbf{o}_t \sim p(\mathbf{o}_t|\hat{s}_t)$ by definition. Inspired by [28], we introduce two approximators $q_\theta(\cdot)$ and $q_\pi(\cdot)$. The function $q_\pi(\cdot)$ approximates the optimal policy, and $q_\theta(\cdot)$ is the inference function for the latent state space, where θ represents the learnable parameters. When $q_\theta(\cdot)$ is fixed, $q_\pi(\cdot)$ can be trained using vanilla Q-learning. When $q_\pi(\cdot)$ is fixed as the optimal policy, $q_\theta(\cdot)$ can be learned. Consider the joint observation $\mathbf{o} \in \boldsymbol{O}$ and joint action $\mathbf{a} \in \mathcal{A}$. We can define the approximate posterior as $q_\theta(\hat{s}_t|\hat{s}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t)$ at time step $t \in [0, T]$. This function is used to infer the latent states. We can then derive the ELBO of Dec-POMDP as follows.

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\mathbf{a}_{0:T}, \mathbf{o}_{1:T}) &= \log p(\mathbf{a}_{0:T}, \mathbf{o}_{1:T}) \\ &= \log \mathbb{E}_{q_\theta(\hat{s}_{1:T}|\mathbf{a}_{0:T}, \mathbf{o}_{1:T})} \left[\frac{p(\hat{s}_{1:T}, \mathbf{a}_{0:T}, \mathbf{o}_{1:T})}{q_\theta(\hat{s}_{1:T}|\mathbf{a}_{0:T}, \mathbf{o}_{1:T})} \right]\end{aligned}$$

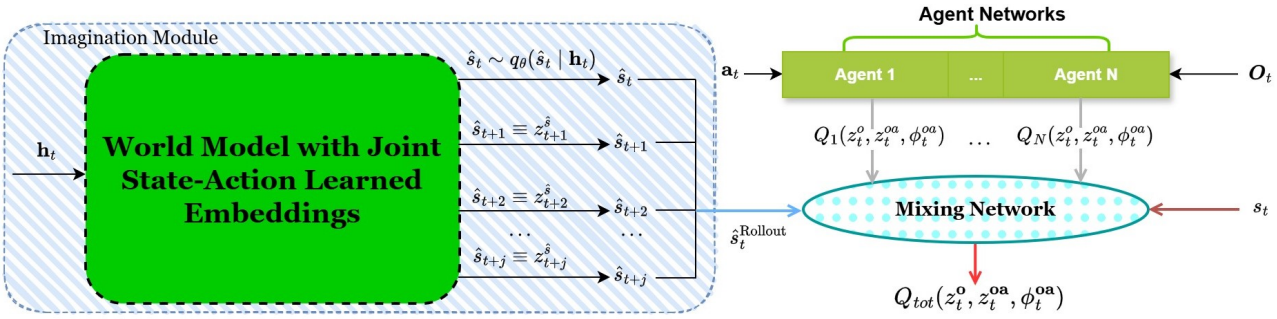

Figure 3: An overview of the MMSA method that illustrates how our model-based MARL framework weaves together (1) a learned world model with state-action learned embeddings, (2) decentralized agent value networks equipped with SALE, and (3) a QMIX-style mixing network under the CTDE paradigm. The learning process of the world model is shown in Figure 2.

$$\approx \sum_{t=1}^{T} \{\log\left[p(\mathbf{a}_t|\mathbf{o}_t)\right] + \log\left[p(\mathbf{o}_t|\hat{s}_t)\right] - \mathcal{D}_{\mathrm{KL}}\left[q_\theta(\hat{s}_t|\hat{s}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t) \parallel p(\hat{s}_t|\hat{s}_{t-1}, \mathbf{a}_{t-1})\right]\} \quad (1)$$

We derive $\mathcal{L}_{\mathrm{ELBO}}$ with respect to the joint action and joint observation to solve the amortized variational inference problem in MARL. In contrast, [28] only studies the POMDP in single-agent domains.

The term $\log\left[p(\mathbf{a}_t|\mathbf{o}_t)\right]$ in Eq.1 represents the joint policy, which is irrelevant to the state inference problem. The last term of $\mathcal{L}_{\mathrm{ELBO}}$ denotes the negative Kullback-Leibler (KL) divergence, which implies that the KL distance between the posterior and prior approximates should be minimized. Since the actual prior distribution $p(\hat{s}_t|\hat{s}_{t-1}, \mathbf{a}_{t-1})$ is unknown, we introduce a generative model $p_\theta^{\mathrm{prior}}(\hat{s}_t|\hat{s}_{t-1}, \mathbf{a}_{t-1})$ to estimate the prior. We use variational auto-encoders (VAEs) to learn $p_\theta^{\mathrm{prior}}(\hat{s}_t|\cdot)$ and $q_\theta(\hat{s}_t|\cdot)$, which we will elaborate in the following subsections.

4.2 Policy Network and Agent Network

In Figure 1, we show the policy network (top diagram) and the agent network (bottom diagram) in the MMSA method. We adapt the policy and Q-function of SALE [16] to model-based MARL settings. Originally, SALE consists of two encoders: a state encoder f and a state-action encoder g, where $z^s := f(s)$ is the embedding of s, and $z^{sa} := g(z^s, a)$ is the joint state-action embedding [16]. In our framework, each agent's policy network builds on the SALE state encoder. At time t, s_t is passed through the encoder that outputs $z_t^s = f_t(s_t)$. To stabilize downstream learning, we apply AvgL1Norm [16], a normalization function that rescales the input vector and preserves the relative scale of the embedding throughout learning. It can be expressed as

$$\mathrm{AvgL1Norm}(z_t^s) = \frac{z_t^s}{\frac{1}{N}\sum_{i=1}^{N}|z_{t,i}^s|},$$

assuming that $z_{t,i}^s$ is the i-th dimension of an N-dimensional vector z_t^s. In parallel, the state is mapped through a linear layer to produce $\phi_t^s = \mathrm{Linear}(s_t)$. The normalized SALE embedding and the learned feature vector are then concatenated and fed into the policy head π_t, which outputs each agent's action distribution $a_t^i \sim \pi_t(z_t^s,\ \phi_t^s)$.

By decoupling the SALE encoder from the training of the policy, the network benefits from stable state representations [16].

For Q-function estimation, each agent uses its local observation o_t^i and its own action a_t^i at time t. First, the SALE state encoder yields $z_t^o = f_t(o_t^i)$, which is normalized via AvgL1Norm. Then, the state-action encoder computes a joint embedding $z_t^{oa} = g_t(z_t^{o^i},\ a_t^i)$. In addition, a direct feature mapping of o_t^i and a_t^i is learned via a linear layer $\phi_t^{oa} = \mathrm{Linear}(o_t^i,\ a_t^i)$ and normalized. Lastly, the Q-function for agent i is computed by $Q_i(z_t^o,\ z_t^{oa},\ \phi_t^{oa})$ given the inputs above. Because the joint state-action embeddings are provided alongside conventional representations, the agent network can exploit rich transition information learned by SALE while maintaining stable training dynamics.

We use recurrent neural networks (RNNs) to implement the agent network in practice. We denote the hidden outputs of the RNN for agent i and for the whole network as h_t^i and $\mathbf{h}_t$, respectively. We interpret h_t^i as the past knowledge specific to agent i and assume that $\mathbf{h}_t$ collectively encapsulates all the past information of the environment. By this assumption, we can reformulate the approximate posterior $q_\theta(\cdot|\hat{s}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t)$ as $q_\theta(\cdot|\mathbf{h}_t)$. $\mathbf{h}_t$ can be considered as a function of $\hat{s}_{t-1}$, $\mathbf{a}_{t-1}$, and $\mathbf{o}_t$. Initially, the posterior latent state is $\hat{s}_t \sim q_\theta(\hat{s}_t|\hat{s}_{t-1}, \mathbf{a}_{t-1}, \mathbf{o}_t)$. With the reparameterization, it can be transformed into $\hat{s}_t \sim q_\theta(\hat{s}_t|\mathbf{h}_t)$.

4.3 Integrating World Model and State-Action Learned Embeddings

Based on the variational inference analysis and the agent network discussed above, we propose a novel world model that incorporates joint state-action learned embeddings to grasp the underlying dynamics between states and actions. The world model is displayed in Figure 2. It can be considered as the imagination module in our MMSA framework. This world model significantly enhances sample efficiency by generating imagined roll-outs without the need for interaction with the real environment, thereby saving valuable resources and time. Additionally, the VAEs that estimate prior and posterior distributions of the latent states are integrated into the world model, providing great robustness for probabilistic inference. Because we use SALE in our model, the prior distribution learned by the VAE should be written as $\hat{s}_t \sim p_\theta^{\mathrm{prior}}(\hat{s}_t|z_{t-1}^{\hat{sa}}, z_{t-1}^{\hat{s}}, \phi_{t-1}^{\hat{sa}})$.

Using Figure 2, we explain how a complete step of simulation in the world model works. At time step t, $\mathbf{h}_t$ encapsulates all past information, including the joint action-observation histories $\boldsymbol{\tau}_t$, as mentioned in Section 4.2. This $\mathbf{h}_t$ is passed into both the posterior VAE and the joint agent network (also referred to as the agent networks). The posterior latent state $\hat{s}_t \sim q_\theta(\hat{s}_t|\mathbf{h}_t)$ is inferred. $\hat{s}_t$ is encoded into a state embedding $z_t^{\hat{\mathbf{s}}} = f_t(\hat{s}_t)$ and normalized via AvgL1Norm. On the other hand, the joint agent network is a collection of all agents in the environment, and it implements the policy network that yields the joint policy $\pi := (\pi^1, ..., \pi^N)$ for all agents. Given $\mathbf{h}_t$, the joint agent network outputs the joint action $\hat{\mathbf{a}}_t \sim \pi(\cdot|\boldsymbol{\tau}_t)$. $\hat{\mathbf{a}}_t$ is passed together with $z_t^{\hat{\mathbf{s}}}$ into the state-action encoder to obtain $z_t^{\hat{\mathbf{s}}\hat{\mathbf{a}}} = g_t(z_t^{\hat{\mathbf{s}}}, \hat{\mathbf{a}}_t)$. Meanwhile, a lightweight linear layer also processes $\hat{s}_t$ and $\hat{\mathbf{a}}_t$ jointly to produce $\phi_t^{\hat{\mathbf{s}}\hat{\mathbf{a}}} = \text{Linear}(\hat{s}_t, \hat{\mathbf{a}}_t)$. The embeddings $(z_t^{\hat{\mathbf{s}}\hat{\mathbf{a}}}, z_t^{\hat{\mathbf{s}}}, \phi_t^{\hat{\mathbf{s}}\hat{\mathbf{a}}})$ are passed as inputs to VAE-1, whose decoder predicts the next-step embeddings $(z'^{\hat{\mathbf{s}}\hat{\mathbf{a}}}_t, z'^{\hat{\mathbf{s}}}_t, \phi'^{\hat{\mathbf{s}}\hat{\mathbf{a}}}_t)$. Then, we set $\hat{s}_{t+1} = z_{t+1}^{\hat{\mathbf{s}}} := z'^{\hat{\mathbf{s}}}_t$, and feed the reconstructed representations $(z'^{\hat{\mathbf{s}}\hat{\mathbf{a}}}_t, z'^{\hat{\mathbf{s}}}_t, \phi'^{\hat{\mathbf{s}}\hat{\mathbf{a}}}_t)$ into VAE-2. Because these representations embody rich information about the joint state and action at time step t, they are sufficient for VAE-2 to derive $\mathbf{h}'_{t+1}$. Now, we have both the imagined state $\hat{s}_{t+1}$ and $\mathbf{h}'_{t+1}$ that contains the imagined action-observation histories $\boldsymbol{\tau}'_{t+1}$. Therefore, the one step of imagination is complete, and we are ready to repeat the procedures at time $t+1$.

4.4 MARL Framework with Joint State-Action Representation Learning

Figure 3 presents an overview on the architecture of MMSA. The joint agent network is displayed on the top right, where each agent computes an individual value estimate. Although trained centrally, each agent's network uses only its own observation and action at execution time. The module on the left shows the world model we described in Section 4.3, with the roll-out horizon set to $j \in \mathbb{Z}$. The roll-out horizon is a tunable hyperparameter. When we perform j simulated roll-outs in the imagination module, we obtain a series of latent states $\{\hat{s}_t, z_{t+1}^{\hat{\mathbf{s}}}, ..., z_{t+j}^{\hat{\mathbf{s}}}\}$, or equivalently, $\{\hat{s}_t, ..., \hat{s}_{t+j}\}$. They are aggregated to form the roll-out state $\hat{s}_t^{\text{Rollout}}$. By supplying the aggregated roll-outs to the mixing network, we build a bridge between world model learning and multi-agent value decomposition.

Combining the world model with the mixing network makes the model applicable in complex multi-agent systems. The MMSA framework operates under the condition of Individual-Global-Max [61]. Receiving outputs of the joint agent network and merging them monotonically, the mixing network reconciles individual Q-function estimates with the team objective. It models the joint action-value function $Q_{tot}(z_t^{\mathbf{o}}, z_t^{\mathbf{oa}}, \phi_t^{\mathbf{oa}})$. This function is consistent with the Q-function under SALE [16], but there is a small difference. Because each of the individual agents in the joint agent network only performs decision-making based on local observation, the individual Q-function should be denoted as $Q_i(z_t^o, z_t^{oa}, \phi_t^{oa}), \forall i \in \mathcal{N}$. Because MMSA follows the CTDE paradigm, the joint action-value function can be decomposed into individual Q-functions to calculate the expected returns. Therefore, Q_{tot} should also maintain consistency with the Q_i's and be a function of joint observation and action, i.e., $z_t^{\mathbf{o}}$, $z_t^{\mathbf{oa}}$, and $\phi_t^{\mathbf{oa}}$.

The mixing network receives three different types of inputs. The first type consists of individual action-value functions from the agent networks, the second is the real global state s_t, and the last is $\hat{s}_t^{\text{Rollout}}$. Because $\hat{s}_t^{\text{Rollout}}$ incorporates information about potential future observations as well as underlying dynamics between states and actions, it can assist the agents greatly in decision-making.

The overall learning objective consists of four distinct components. First, we minimize the KL divergence between the prior and posterior distributions, $p_\theta^{\text{prior}}(\cdot)$ and $q_\theta(\cdot)$. Because $p_\theta^{\text{prior}}(\cdot)$ is a parameterized surrogate for estimating the true prior, we also need to minimize its divergence from an isotropic Gaussian $p_0(\hat{s}_t) := \mathcal{N}(0, \mathbf{I})$. Combining the two KL terms, we obtain:

$$\mathcal{L}_{\text{KL}}(\theta) = \mathcal{D}_{\text{KL}}\left[p_\theta^{\text{prior}}\left(\hat{s}_t \mid z_{t-1}^{\hat{\mathbf{s}}\hat{\mathbf{a}}}, z_{t-1}^{\hat{\mathbf{s}}}, \phi_{t-1}^{\hat{\mathbf{s}}\hat{\mathbf{a}}}\right) \| p_0(\hat{s}_t)\right] + \mathcal{D}_{\text{KL}}\left[q_\theta(\hat{s}_t|\mathbf{h}_t) \| p_\theta^{\text{prior}}\left(\hat{s}_t \mid z_{t-1}^{\hat{\mathbf{s}}\hat{\mathbf{a}}}, z_{t-1}^{\hat{\mathbf{s}}}, \phi_{t-1}^{\hat{\mathbf{s}}\hat{\mathbf{a}}}\right)\right].$$

We implement the KL balancing technique in the optimization of $\mathcal{L}_{\text{KL}}(\theta)$ because when the learned prior is inaccurate, forcing the posterior to match it aggressively can lead to poor representations [26]. KL balancing allows the prior to mature quickly while preventing the posterior from being over-regularized. Let $\alpha \in [0, 1]$ be the learning rate. KL balancing can then be defined as:

$$\mathcal{D}_{\text{KLB}}\left[q_\theta(\cdot) \| p_\theta^{\text{prior}}(\cdot)\right] = \alpha \mathcal{D}_{\text{KL}}\left[q_\theta(\cdot) \parallel |p_\theta^{\text{prior}}(\cdot)|_\times\right] + (1-\alpha)\mathcal{D}_{\text{KL}}\left[|q_\theta(\cdot)|_\times \parallel p_\theta^{\text{prior}}(\cdot)\right],$$

where $|\cdot|_\times$ denotes the stop-gradient operation.

Second, both the prior and the posterior VAEs are trained to reconstruct the inputs given. We therefore add a loss function measuring how well each VAE recovers the original variables. Using mean-squared error (MSE), the reconstruction loss is written as:

$$\mathcal{L}_{\text{REC}}(\theta) = \text{MSE}(z_t^{\hat{\mathbf{s}}\hat{\mathbf{a}}}, z'^{\hat{\mathbf{s}}\hat{\mathbf{a}}}_t; \theta) + \text{MSE}(z_t^{\hat{\mathbf{s}}}, z'^{\hat{\mathbf{s}}}_t; \theta) + \text{MSE}(\phi_t^{\hat{\mathbf{s}}\hat{\mathbf{a}}}, \phi'^{\hat{\mathbf{s}}\hat{\mathbf{a}}}_t; \theta) + \text{MSE}(\mathbf{h}_t, \mathbf{h}'_t; \theta).$$

Third, we denote the learnable parameters of the value decomposition framework as ψ. Analogous to [53], we define the TD target y^{tot} and TD loss $\mathcal{L}_{\text{TD}}(\psi)$ as follows:

$$y^{tot} = r_t + \gamma \max_{\mathbf{a}_{t+1}} Q_{tot}(z_{t+1}^{\mathbf{o}}, z_{t+1}^{\mathbf{oa}}, \phi_{t+1}^{\mathbf{oa}}, s_{t+1}, \hat{s}_{t+1}^{\text{Rollout}}; \psi^-),$$
$$\mathcal{L}_{\text{TD}}(\psi) = \left(y^{tot} - Q_{tot}(z_t^{\mathbf{o}}, z_t^{\mathbf{oa}}, \phi_t^{\mathbf{oa}}, s_t, \hat{s}_t^{\text{Rollout}}; \psi)\right)^2,$$

where ψ^- denotes the parameters of the target network.

Fourth, we adapt the MSE loss between the state-action embedding and the embedding of the next state [16]. It is applied to train the SALE encoders:

$$\mathcal{L}(f, g) := \left(g_t(f_t(\hat{s}_t), \hat{\mathbf{a}}_t) - |f_{t+1}(\hat{s}_{t+1})|_\times\right)^2 = \left(z_t^{\hat{\mathbf{s}}\hat{\mathbf{a}}} - |z_{t+1}^{\hat{\mathbf{s}}}|_\times\right)^2.$$

Finally, the overall loss function for MMSA can be written as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{KL}}(\theta) + \mathcal{L}_{\text{REC}}(\theta) + \mathcal{L}_{\text{TD}}(\psi) + \mathcal{L}(f, g).$$

Through unified optimization of the loss components, MMSA learns to generate accurate latent-state predictions and improved policies, ultimately maximizing the cumulative return.

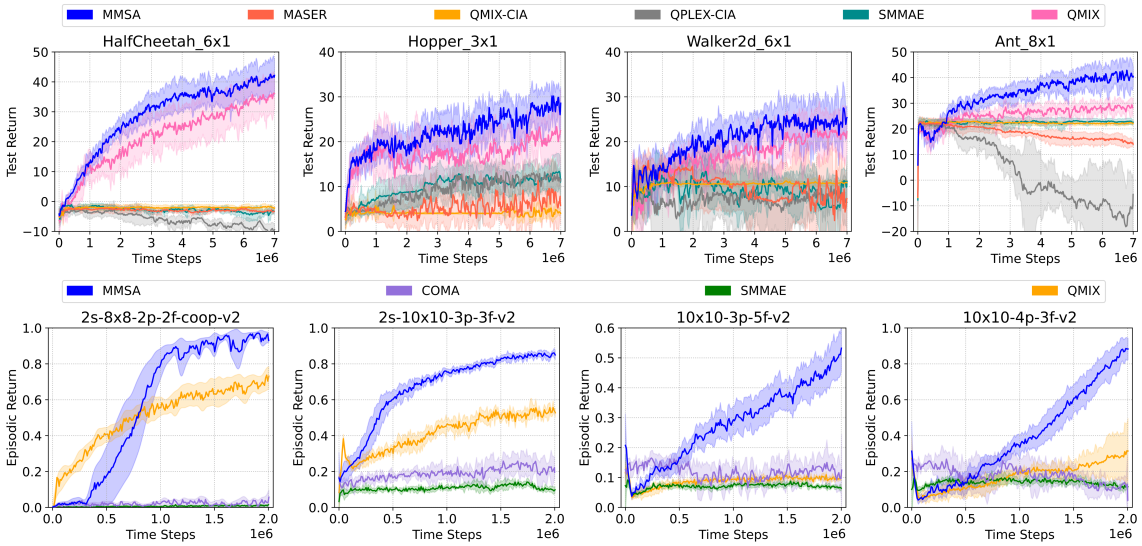

Figure 4: Performance of MMSA in Multi-Agent MuJoCo (top row) and in Level-Based Foraging (bottom row). The shaded region captures a 95% confidence interval around the average performance. Top: Comparison of the average episodic return of MMSA with competing MARL algorithms in Multi-Agent MuJoCo tasks. The return is scaled for clear plotting. Experiments are run for 7M time steps. Bottom: The mean episodic return of MMSA compared to other MARL methods in Level-Based Foraging. Each run lasts 2M time steps. In both MARL benchmarks, MMSA excels the competitors in all of the environments.

5 EMPIRICAL EVALUATION

To evaluate the effectiveness and generalization ability of MMSA, we benchmark our method on various MARL test beds: Multi-Agent MuJoCo (MAMuJoCo) [50], Level-Based Foraging (LBF) [7], and StarCraft Multi-Agent Challenges, including SMAC [56] and SMACv2 [12]. MMSA is compared against a broad suite of model-free MARL algorithms (VDN [62], COMA [15], QMIX [53], SET-QMIX [36], MASER [32], QMIX-CIA [38], QPLEX-CIA [38], SMMAE [79], HPN-QMIX [33], and HPN-VDN [33]) and model-based MARL methods (MAG [74], MAMBA [11], and MABL [69]). The competing methods range from the classic value-based methods to the newest MARL representatives. All experiments are performed with five different seeds.

We then conduct a rigorous design study on the MMSA framework to identify the design elements that most strongly affect the performance. Moreover, we carry out ablation experiments to analyze the impact of different components of MMSA, involving SALE, world model imagination, KL balancing, and the global state. We verify that all components are critical to the competence of MMSA.

Multi-Agent MuJoCo. To foster the research interest for multi-agent robotic control, [50] extends the original MuJoCo suite [68] to Multi-Agent MuJoCo (MAMuJoCo) by decomposing a single robot into disjoint sub-graphs. Each of them represents an agent and needs to cooperate to solve continuous control tasks. As Figure 4 shows, four MAMuJoCo learning tasks with partial observability are completed[1]. Across all four MAMuJoCo domains, MMSA markedly outperforms the baselines. Several methods, including QPLEX-CIA and MASER, perform poorly and even incur negative returns. This indicates that the robots fail to advance during training. By comparison, MMSA reliably learns forward-moving behaviors for the robot and achieves the highest average return in MAMuJoCo. This superior performance stems from the use of roll-outs generated by the SALE-augmented world model. By simulating and evaluating future joint state-action sequences in latent space, MMSA employs coordinated control strategies that other algorithms, which rely solely on real trajectories, are unable to develop.

Level-Based Foraging. Level-Based Foraging (LBF) [7] is an MARL benchmark that blends cooperation and competition. The general setting of LBF consists of agents and food items. Each of them is assigned an integer level. The agents must coordinate their efforts to collect items whose levels exceed that of any single agent. In our study, we define four distinct LBF environments that vary in the number of agents, the item count, cooperation requirements, partial observability, and the world size.

Figure 4 displays that MMSA rapidly rises above competing algorithms on every scenario, achieving higher average episodic returns within fewer training episodes. In the task of *10x10-3p-5f-v2*, there are three players with five items to collect, more than in any other world. The players need to spend more time gaining rewards. However, MMSA still achieves remarkable progress given the limited time steps. In contrast, QMIX plateaus at lower returns, while other methods, such as MASER and SMMAE, often struggle to coordinate sufficient joint effort and exhibit flatter performance. The results demonstrate that MMSA's combination of latent imagination, joint representations, and monotonic mixing yields more effective cooperation in the LBF domains.

[1] *Ant_8×1* is the Ant partitioned into 8 agents, *Walker2d_6×1* is the Walker partitioned into 6 agents, *HalfCheetah_6×1* is the Half Cheetah partitioned into 6 agents, and *Hopper_3×1* is the Hopper partitioned into 3 agents. In our settings, each agent is constrained to observe only the two nearest joints.

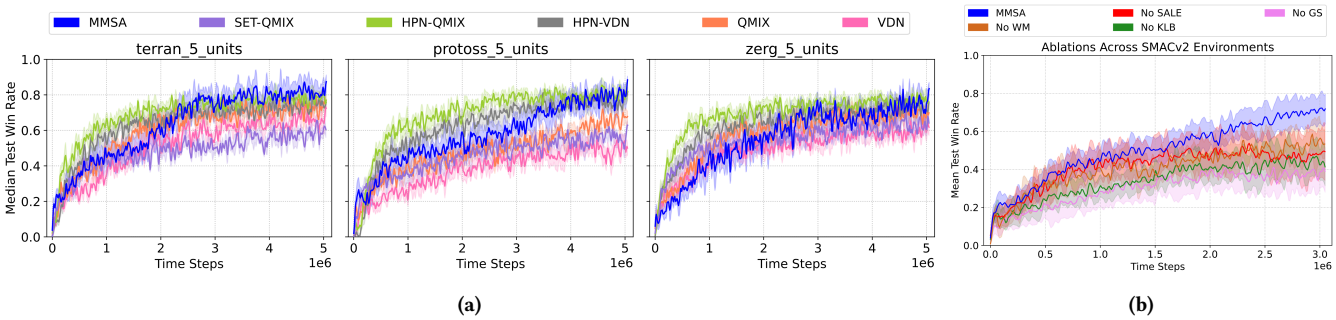

Figure 5: Performance of MMSA compared with MARL baselines and ablations in SMACv2. (a) Test win rates of MMSA compared with top-performing methods in SMACv2. We plot the median test win rates with the 25% − 75% percentiles, as in [33]. Each run lasts 5M time steps. Although MMSA shows a slow start, it gradually outruns the baselines such as VDN and QMIX. It exhibits an overall performance matching that of HPN-QMIX, the best-competing method. (b) Ablations for the MMSA architecture. MMSA is compared against the variants in which the world model, SALE, KL balancing, or global state is removed, respectively (No-WM, No-SALE, No-KLB, or No-GS). Performance is averaged over all SMACv2 challenges. Each run lasts 3M time steps.

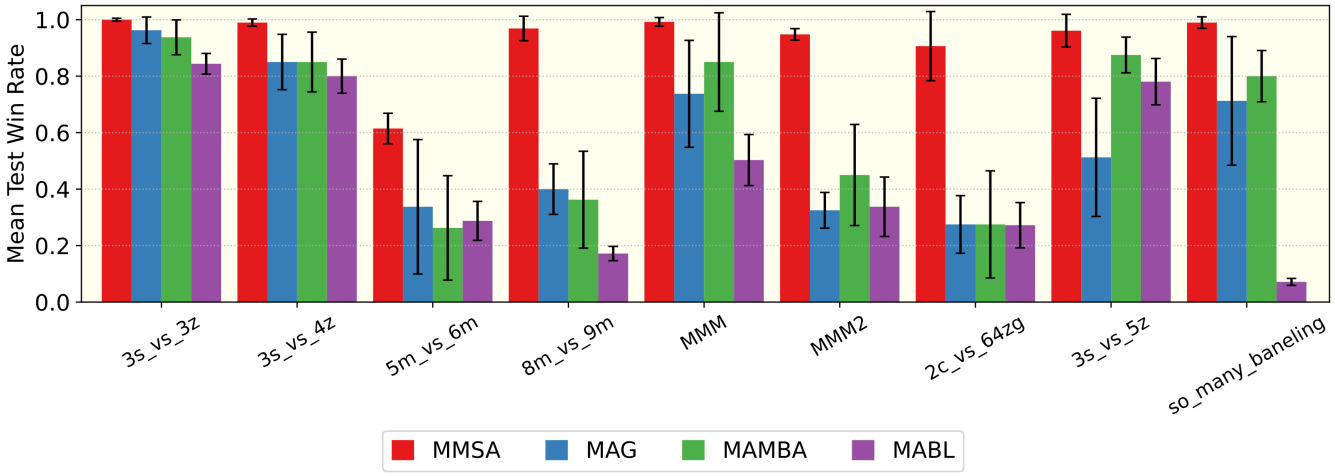

Figure 6: Comparing the performance of MMSA with model-based MARL algorithms MAG [74], MAMBA [11], and MABL [69] in StarCraft Multi-Agent Challenges. The error bars represent the 95% confidence intervals around the mean test win rates.

SMAC and SMACv2. Developed on top of the StarCraft II API, SMAC (SMACv1) comprises a diverse set of battle scenarios corresponding to an extensive range of learning tasks [56]. SMAC targets the problem of micro-management, wherein each agent is responsible for the fine-grained control of an individual unit and selects actions independently. We conduct experiments on the SMAC benchmark to compare the performance of MMSA with MAMBA (Multi-Agent Model-Based Approach) [11], MABL (Multi-Agent Bi-Level world model) [69], and MAG (Models as AGents) [74], three model-based MARL algorithms that were recently proposed. Figure 6 displays the mean and the 95% confidence interval of the test win rate over five different seeds for each method in SMAC maps. During the experiments, the total number of roll-outs in a single run is the same (2M) for all methods, ensuring that a fair comparison is made. From the results, we can observe that MMSA achieves the highest win rates in every SMAC environment. Furthermore, MMSA exhibits consistency across independent runs, because it yields very small confidence intervals in eight of the nine scenarios.

SMACv2 [12] is a stochastic extension of SMAC [56]. Each episode randomizes the allies' and enemies' spawn locations, posing challenges for the allied units to beat the enemies that approach from multiple angles simultaneously. The unit compositions are also randomized, with three unit types per race sampled according to predefined probabilities. Figure 5a shows that MMSA climbs more slowly than the baselines, which could be due to the extra effort required to train the world model with SALE in randomized scenarios. However, MMSA gradually approaches and matches HPN-QMIX across all the SMACv2 maps, surpassing the other methods. MMSA's sustained ascent stems from the use of imagined roll-outs. By refining the latent dynamics model and SALE representations, MMSA uncovers more effective coordination strategies over time.

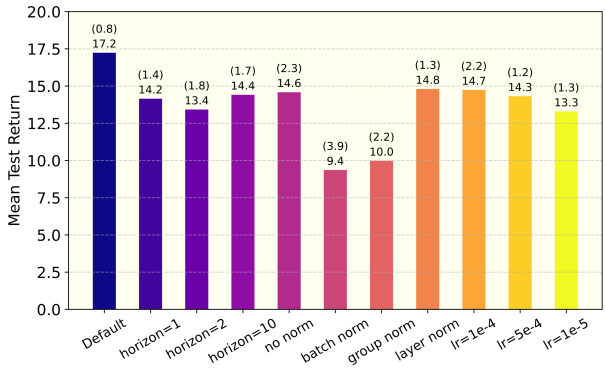

Figure 7: We study the design space of the MMSA framework. In the default setting, the roll-out horizon, learning rate, and normalizer are set to 3, 1e-3, and AvgL1Norm, respectively. The mean test return is shown above the bars. Bracketed values stand for the range of the 95% confidence interval around the mean. Experiments are conducted on SMACv2 environments. Each run lasts 3M time steps. The performance is averaged over the SMACv2 scenarios.

Design Studies. We present detailed studies on three key design elements of MMSA: roll-out horizon, learning rate, and normalization function. Figure 7 implies that our default setting attains the highest average return, demonstrating both strong performance and low variance. Reducing the horizon to 1 or 2 steps causes a drop in performance, which indicates that agents may not be able to exploit the full strength of the imagination module with shorter horizons. Increasing the horizon to 10 steps also has a negative impact on the method. Although long roll-outs enable agents to anticipate long-term consequences, the model error can compound over time, leading to unrealistic trajectory predictions as the horizon becomes large. The results suggest that a three-step roll-out should be applied. For normalization, AvgL1Norm proves critical. Omitting normalization or using layer normalization still gives a reasonable return. However, the other normalizers drastically underperform. Lastly, deviating from the default learning rate of 1e-3 yields lower returns. The reason could be that the agents' learning is sensitive to step-size, and smaller learning rates can lead to slower convergence. The default setting proves to be the most effective and reliable design choice.

Ablation Studies. Moreover, to study the contributions of each architectural component, we compare the complete MMSA framework against four ablated variants: without the world model (No-WM), without using SALE (No-SALE), without KL balancing (No-KLB), and without global state in the mixing network (No-GS). The results are displayed in Figure 5b. Firstly, removing the learned world model caps the win rate around 0.5, which indicates that the world model's roll-out predictions are vital for pushing the agents beyond baseline behaviors. Secondly, omitting the SALE mechanism yields a lower asymptotic performance than No-WM, which highlights the importance of modeling the underlying structure of the environment and capturing the interactions between states and actions. Thirdly, the learning curve of No-KLB reveals the significance of KL balancing in preventing posterior collapse and maintaining robust representation learning. Finally, the evident performance drop of No-GS implies that access to the centralized state during training is critical for effective coordination in SMACv2 scenarios. By combining these key components, MMSA achieves the greatest efficiency in early learning and the highest asymptotic win rate.

Table 1: An overview of the performance of MMSA at the end of training compared to competing methods across different MARL benchmarks. For SMAC (SMACv1) and SMACv2, we use the mean test win rate averaged over the environments, instead of the median win rate. For MAMuJoCo and LBF, the episodic returns are averaged over all tasks within the benchmarks. MMSA leads the group among MAMuJoCo, LBF, and SMAC. On SMACv2, MMSA ties with HPN-QMIX and outperforms the other MARL approaches.

Environments	MMSA	MASER	SMMAE
MAMuJoCo Tasks	**36.94**	5.94	11.01
Environments	QMIX-CIA	QPLEX-CIA	QMIX
MAMuJoCo Tasks	8.71	1.85	28.69

Environments	MMSA	SMMAE	QMIX	COMA
LBF Environments	**0.80**	0.08	0.42	0.11

Environments	MMSA	MAG	MAMBA	MABL
SMACv1 Challenges	**0.93**	0.57	0.63	0.45

Environments	MMSA	SET-QMIX	HPN-QMIX
SMACv2 Challenges	**0.81**	0.64	**0.81**
Environments	HPN-VDN	QMIX	VDN
SMACv2 Challenges	0.78	0.72	0.58

6 CONCLUSION

This paper presents MMSA, a model-based MARL method that fuses a value factorization framework with joint state-action representation learning, amortized variational inference, and an imagination module. MMSA is able to produce faithful latent roll-outs, preserve well-scaled embeddings, and learn decentralized policies from real and imagined experience. Experiments on various MARL benchmarks demonstrate the outstanding performance and generalizability of our approach. Design studies justify the design choices for the MMSA method. Ablation studies confirm the positive impact and indispensability of different MMSA components[2].

A promising avenue for future research is the systematic mitigation of model error, which is a long-standing problem for model-based MARL. The discrepancy between imagined roll-outs and real-world dynamics may accumulate and ultimately misguide cooperative policies. Extending MMSA with an ensemble of models could mitigate the impact of the errors, as model ensembles have proven to be effective in reducing model uncertainties. Alternatively, incorporating uncertainty estimation techniques and applying regularization schemes could prevent the model from overfitting and promote generalization to unseen states.

[2]The supplementary material can be found at https://arxiv.org/abs/2602.12520.

REFERENCES

[1] Yu Bai and Chi Jin. 2020. Provable self-play algorithms for competitive reinforcement learning. In *International conference on machine learning*. PMLR, 551–560.

[2] Craig Boutilier. 1996. Planning, Learning and Coordination in Multiagent Decision Processes. In *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*.

[3] Ronen Brafman and Moshe Tennenholtz. 1999. A near-optimal poly-time algorithm for learning in a class of stochastic games. *IJCAI* (1999), 734–739.

[4] Ronen Brafman and Moshe Tennenholtz. 2003. R-max–A General Polynomial Time Algorithm for Near-Optimal Reinforcement Learning. *Journal of Machine Learning Research* 3, 2 (2003), 213.

[5] Georgios Chalkiadakis and Craig Boutilier. 2003. Coordination in Multiagent Reinforcement Learning: A Bayesian Approach. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems* (Melbourne, Australia) *(AAMAS '03)*. Association for Computing Machinery, 709–716. https://doi.org/10.1145/860575.860689

[6] Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, and Sergey Levine. 2021. Actionable Models: Unsupervised Offline Reinforcement Learning of Robotic Skills. *arXiv:2104.07749 [cs]* (April 2021). http://arxiv.org/abs/2104.07749 arXiv: 2104.07749.

[7] Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2020. Shared Experience Actor-Critic for Multi-Agent Reinforcement Learning. *Advances in neural information processing systems* 33 (2020), 10707–10717.

[8] Kurtland Chua, Roberto Calandra, Rowan Thomas McAllister, and Sergey Levine. 2018. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. In *Neural Information Processing Systems*.

[9] Caroline Claus and Craig Boutilier. 1998. The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. In *AAAI/IAAI*.

[10] Marc Peter Deisenroth and Carl Edward Rasmussen. 2011. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *International Conference on Machine Learning*.

[11] Vladimir Egorov and Alexei Shpilman. 2022. Scalable Multi-Agent Model-Based Reinforcement Learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 381–390.

[12] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Nicolaus Foerster, and Shimon Whiteson. 2023. SMACv2: An Improved Benchmark for Cooperative Multi-Agent Reinforcement Learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. https://openreview.net/forum?id=5OjLGiJW3u

[13] Norm Ferns, Prakash Panangaden, and Doina Precup. 2011. Bisimulation Metrics for Continuous Markov Decision Processes. *SIAM J. Comput.* 40, 6 (2011), 1662–1714. https://doi.org/10.1137/10080484X

[14] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. 2016. Deep Spatial Autoencoders for Visuomotor Learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 512–519. https://doi.org/10.1109/ICRA.2016.7487173

[15] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual Multi-Agent Policy Gradients. *AAAI* 32 (2018). https://doi.org/10.1609/aaai.v32i1.11794

[16] Scott Fujimoto, Wei-Di Chang, Edward Smith, Shixiang Shane Gu, Doina Precup, and David Meger. 2024. For sale: State-action representation learning for deep reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2024).

[17] Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. 2022. Why Should I Trust You, Bellman? The Bellman Error is a Poor Replacement for Value Error. In *International Conference on Machine Learning*. PMLR, 6918–6943.

[18] Sven Gronauer and Klaus Diepold. 2021. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review* (2021). https://doi.org/10.1007/s10462-021-09996-w

[19] Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. 2016. Continuous deep q-learning with model-based acceleration. In *ICML*.

[20] Carlos Guestrin, Michail G. Lagoudakis, and Ronald E. Parr. 2002. Coordinated Reinforcement Learning. In *International Conference on Machine Learning*.

[21] David Ha and Jürgen Schmidhuber. 2018. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems*, Vol. 31.

[22] David R Ha and Jürgen Schmidhuber. 2018. World Models. *ArXiv* abs/1803.10122 (2018).

[23] Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. 2022. Deep Hierarchical Planning from Pixels. In *Advances in Neural Information Processing Systems*, Vol. 35. 26091–26104.

[24] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2020. Dream to Control: Learning Behaviors by Latent Imagination. In *International Conference on Learning Representations*.

[25] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019. Learning Latent Dynamics for Planning from Pixels. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 2555–2565.

[26] Danijar Hafner, Timothy P Lillicrap, Mohammad Norouzi, and Jimmy Ba. 2021. Mastering Atari with Discrete World Models. In *International Conference on Learning Representations*.

[27] Danijar Hafner, J. Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. 2023. Mastering Diverse Domains through World Models. *arXiv* abs/2301.04104 (2023).

[28] Shiyu Huang, Hang Su, Jun Zhu, and Tingling Chen. 2020. SVQN: Sequential Variational Soft Q-Learning Networks. In *ICLR*.

[29] Shariq Iqbal and Fei Sha. 2018. Actor-Attention-Critic for Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning*.

[30] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. 2017. Reinforcement Learning with Unsupervised Auxiliary Tasks. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[31] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. 2019. When to Trust Your Model: Model-Based Policy Optimization. In *Advances in Neural Information Processing Systems*, Vol. 32.

[32] Jeewon Jeon, Woojun Kim, Whiyoung Jung, and Youngchul Sung. 2022. MASER: Multi-Agent Reinforcement Learning with Subgoals Generated from Experience Replay Buffer. In *PMLR*, Vol. 162. 10041–10052.

[33] HAO Jianye, Xiaotian Hao, Hangyu Mao, Weixun Wang, Yaodong Yang, Dong Li, Yan Zheng, and Zhen Wang. 2023. Boosting multiagent reinforcement learning via permutation invariant and permutation equivariant networks. In *The Eleventh International Conference on Learning Representations*.

[34] Landon Kraemer and Bikramjit Banerjee. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing* (2016), 82–94.

[35] Lihong Li, Thomas J. Walsh, and Michael L. Littman. 2006. Towards a Unified Theory of State Abstraction for MDPs. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics (ISAIM)*. https://dblp.org/rec/conf/isaim/LiWL06

[36] Yan Li, Lingxiao Wang, Jiachen Yang, Ethan Wang, Zhaoran Wang, Tuo Zhao, and Hongyuan Zha. 2021. Permutation Invariant Policy Optimization for Mean-Field Multi-Agent Reinforcement Learning: A Principled Approach. *arXiv preprint arXiv:2105.08268* (2021). https://arxiv.org/abs/2105.08268

[37] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *CoRR* abs/1509.02971 (2015).

[38] Shunyu Liu, Yihe Zhou, Jie Song, Tongya Zheng, Kaixuan Chen, Tongtian Zhu, Zunlei Feng, and Mingli Song. 2023. Contrastive Identity-Aware Learning for Multi-Agent Value Decomposition. In *AAAI*, Vol. 37. https://doi.org/10.1609/aaai.v37i10.26370

[39] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. *CoRR* abs/1706.02275 (2017). arXiv:1706.02275 http://arxiv.org/abs/1706.02275

[40] Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. 2024. A survey on model-based reinforcement learning. *Science China Information Sciences* 67, 2 (2024), 121101.

[41] Xufang Luo and Yunhong Wang. 2020. PMA-DRL: A parallel model-augmented framework for deep reinforcement learning algorithms. *Neurocomputing* 403 (2020), 109–120.

[42] Chengdong Ma, Aming Li, Yali Du, Hao Dong, and Yaodong Yang. 2024. Efficient and scalable reinforcement learning for large-scale network control. *Nature Machine Intelligence* 6, 9 (2024), 1006–1020.

[43] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (Feb. 2015), 529–533. https://doi.org/10.1038/nature14236 Number: 7540 Publisher: Nature Publishing Group.

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

[45] Frans A. Oliehoek, Matthijs T. J. Spaan, and Nikos A. Vlassis. 2008. Optimal and Approximate Q-value Functions for Decentralized POMDPs. *J. Artif. Intell. Res.* 32 (2008), 289–353.

[46] Kei Ota, Tomoaki Oiki, Devesh K. Jha, Toshisada Mariyama, and Daniel Nikovski. 2020. Can Increasing Input Dimensionality Improve Deep Reinforcement Learning?. In *Proceedings of the 37th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 7424–7433.

[47] Minting Pan, Xiangming Zhu, Yunbo Wang, and Xiaokang Yang. 2022. Iso-Dream: Isolating and Leveraging Noncontrollable Visual Dynamics in World Models. In *Advances in Neural Information Processing Systems*, Vol. 35. 23178–23191.

[48] Jiwoong Park, Minsik Lee, Hyung Jin Chang, Kyuewang Lee, and Jin Young Choi. 2019. Symmetric Graph Convolutional Autoencoder for Unsupervised Graph Representation Learning. *ICCV* (2019), 6518–6527.

[49] Barna Pasztor, Ilija Bogunovic, and Andreas Krause. 2021. Efficient model-based multi-agent mean-field reinforcement learning. *arXiv:2107.04050* (2021).

[50] Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Boehmer, and Shimon Whiteson. 2021. FACMAC: Factored Multi-Agent Centralised Policy Gradients. In *NeurIPS*.

[51] Sebastien Racaniere, Theophane Weber, David P Reichert, Lars Buesing, Arthur Guez, Danilo Rezende, Adria Puigdomenech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. 2017. Imagination-augmented agents for deep reinforcement learning. In *NeurIPS*.

[52] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. 2020. Weighted QMIX: Expanding Monotonic Value Function Factorisation. *CoRR* abs/2006.10800 (2020). arXiv:2006.10800 https://arxiv.org/abs/2006.10800

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

[54] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *PMLR*.

[55] Martin Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degrave, Tom Van de Wiele, Volodymyr Mnih, Nicolas Heess, and Jost Tobias Springenberg. 2018. Learning by Playing: Solving Sparse Reward Tasks from Scratch. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 4344–4353.

[56] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. *arXiv preprint arXiv:1902.04043* (2019). arXiv:1902.04043 [cs.LG]

[57] Pier Giuseppe Sessa, Maryam Kamgarpour, and Andreas Krause. 2022. Efficient model-based multi-agent reinforcement learning via optimistic equilibrium computation. In *International Conference on Machine Learning*. PMLR, 19580–19597.

[58] Jian Shen, Han Zhao, Weinan Zhang, and Yong Yu. 2020. Model-based Policy Optimization with Unsupervised Model Adaptation. In *Advances in Neural Information Processing Systems*, Vol. 33. 2823–2834.

[59] Jian Shen, Han Zhao, Weinan Zhang, and Yong Yu. 2020. Model-Based Policy Optimization with Unsupervised Model Adaptation. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (Vancouver, BC, Canada) *(NIPS'20)*. Curran Associates Inc.

[60] David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. 2021. Reward is enough. *Artificial Intelligence* 299 (Oct. 2021), 103535.

[61] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. In *PMLR*. 5887–5896.

[62] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2017. Value-Decomposition Networks For Cooperative Multi-Agent Learning. *arXiv preprint arXiv:1706.05296* (2017). arXiv:1706.05296.

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

[64] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

[65] Richard S. Sutton, Joseph Modayil, Michael Delp, Thomas Degris, Patrick M. Pilarski, Adam White, and Doina Precup. 2011. Horde: A Scalable Real-time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IFAAMAS, 761–768.

[66] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PLOS ONE* (2017).

[67] Ming Tan. 1997. Multi-Agent Reinforcement Learning: Independent versus Cooperative Agents. In *ICML*.

[68] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *IROS*. https://doi.org/10.1109/IROS.2012.6386109

[69] Aravind Venugopal, Stephanie Milani, Fei Fang, and Balaraman Ravindran. 2024. MABL: Bi-Level Latent-Variable World Model for Sample-Efficient Multi-Agent Reinforcement Learning. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS '24)*. International Foundation for Autonomous Agents and Multiagent Systems, 1865–1873.

[70] Jianhao Wang, Zhizhou Ren, Beining Han, and Chongjie Zhang. 2020. Towards Understanding Linear Value Decomposition in Cooperative Multi-Agent Q-Learning. *CoRR* abs/2006.00587 (2020). arXiv:2006.00587 https://arxiv.org/abs/2006.00587

[71] Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric D. Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. 2019. Benchmarking Model-Based Reinforcement Learning. *CoRR* abs/1907.02057 (2019). arXiv:1907.02057

[72] Xihuai Wang, Zhicheng Zhang, and Weinan Zhang. 2022. Model-based Multi-agent Reinforcement Learning: Recent Progress and Prospects. https://doi.org/10.48550/arXiv.2203.10603 arXiv:2203.10603 [cs].

[73] Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. 2015. Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images. In *Advances in Neural Information Processing Systems 28 (NeurIPS 2015)*. 2746–2754.

[74] Zifan Wu, Chao Yu, Chen Chen, Jianye Hao, and Hankz Hankui Zhuo. 2023. Models as agents: optimizing multi-step predictions of interactive local models in model-based multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 10435–10443.

[75] Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. 2018. Algorithmic Framework for Model-based Reinforcement Learning with Theoretical Guarantees. *arXiv* abs/1807.03858 (2018).

[76] Yaodong Yang and Jun Wang. 2020. An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective. *CoRR* (2020). arXiv:2011.00583 https://arxiv.org/abs/2011.00583

[77] Kaiqing Zhang, Sham Kakade, Tamer Basar, and Lin Yang. 2020. Model-Based Multi-Agent RL in Zero-Sum Markov Games with Near-Optimal Sample Complexity. In *NeurIPS*.

[78] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. https://doi.org/10.48550/arXiv.1911.10635 arXiv:1911.10635 [cs, stat].

[79] Shaowei Zhang, Jiahan Cao, Lei Yuan, Yang Yu, and De-Chuan Zhan. 2023. Self-Motivated Multi-Agent Exploration. In *AAMAS*. 476–484.

[80] Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. 2020. Learning Implicit Credit Assignment for Multi-Agent Actor-Critic. *ArXiv* abs/2007.02529 (2020).

[81] Changxi Zhu, Mehdi M. Dastani, and Shihan Wang. 2022. A Survey of Multi-Agent Reinforcement Learning with Communication. *ArXiv* abs/2203.08975 (2022).