Semantic-level Exploration for Multi-Agent Reinforcement Learning

Jiangjin Yin[1], **Zijian Ye**[1], **Rongbo Zhu**[1*], **Hangyu Mao**[2], **Zhiwei Xu**[3]
[1]College of Informatics, Huazhong Agricultural University, Key Laboratory of Smart Farming for Agricultural Animals, Hubei Engineering Technology Research Center of Agricultural Big Data, and Engineering Research Center of Intelligent Technology for Agriculture
[2]AI Center, Institute of Microelectronics, Chinese Academy of Sciences
[3]School of Artificial Intelligence, Shandong University

Abstract

Multi-Agent Reinforcement Learning (MARL) faces significant exploration challenges due to exponentially growing joint state-action spaces. Existing exploration methods operate directly in raw state-action spaces, which is inefficient and fails to exploit inherent semantic structure. This paper introduces semantic space into multi-agent exploration. We theoretically establish that the observation-action space can be partitioned into discrete semantic prototypes, providing a principled foundation for transferring exploration statistics across semantically similar situations. Building on this theory, we propose a semantic-level exploration mechanism that first compresses the high-dimensional observation–action space into discrete semantic prototypes via vector quantization, and then applies sliding-window count-based bonuses to enable efficient statistical transfer across semantically similar situations. Our approach is versatile and can be seamlessly integrated with existing value-based MARL frameworks. Extensive experiments demonstrate that our method outperforms state-of-the-art baselines across diverse multi-agent benchmarks in terms of both effectiveness and training efficiency.

Code: `https://github.com/hzau-yjj/SEMEX`.

1 Introduction

Multi-Agent Reinforcement Learning (MARL) has shown great potential in real-world applications, including autonomous driving [10, 42], energy system scheduling [18, 40, 47], and robotic swarms [14, 15]. In these applications, agents must learn to coordinate their behaviors to achieve a shared goal. This collaborative setting introduces several challenges, including credit assignment, non-stationarity, and partial observability. To mitigate this challenge, the popular Centralized Training with Decentralized Execution (CTDE) framework has been proposed [43], which allows agents to leverage global information during training while maintaining the ability to act independently based on local observations during execution. Under this framework, value-based methods such as VDN [33], QMIX [27], and QTRAN [30] have achieved remarkable performance. Despite these advances, these methods remain limited by the traditional ϵ-greedy strategy, which fails to facilitate the coordinated exploration required for complex tasks.

A wide range of exploration techniques have been developed in single-agent reinforcement learning, including count-based methods [4, 25], curiosity-based approaches [5, 26], and information-theoretic strategies [7, 9]. However, these methods cannot be directly extended to MARL, as exploration in multi-agent settings is particularly challenging due to the exponential growth of the joint state–action space and inherent partial observability. Recent advancements in MARL exploration have spanned several paradigms: social influence signals [37], behavioral diversity [17], and state-novelty rewards [45].

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, existing approaches typically operate directly within the massive, raw state-action space, which is not only inefficient but also overlooks the underlying semantic structure of the environment.

In practice, diverse observations across tasks often share similar high-level semantic features. If properly exploited, such semantic structure can enable efficient transfer of exploration statistics across functionally equivalent situations, substantially reducing redundant exploration while accelerating the discovery of genuinely novel coordination patterns. Guided by this insight, we introduce *Vanilla_SEMEX*, a baseline SEMantic EXploration approach that operates solely within the observation space. Specifically, we theoretically show that high-dimensional observation spaces can be partitioned into discrete semantic equivalence classes. To operationalize this insight, we leverage a Vector Quantized Variational Autoencoder (VQ-VAE) [35] to instantiate the proposed semantic decomposition in practice. We further introduce a count-based exploration bonus with a sliding-window mechanism, enabling shared exploration across semantically equivalent observations while adapting to the non-stationary dynamics of multi-agent environments.

However, Vanilla_SEMEX suffers from over-generalization: similar high-level observation semantics lead to shared exploration statistics, even when the same action results in fundamentally different outcomes. To address this issue, we propose SEMEX, which encodes observation–action pairs into joint semantics. This design allows SEMEX to better capture the functional impact of each action within its specific observational context. Consequently, the exploration policy can assess behavioral novelty more precisely, enabling more targeted and fine-grained exploration.

We evaluate our method through extensive experiments on the classical StarCraft Multi-Agent Challenge (SMAC and SMACv2) and Multi-Agent Coordination (MACO) scenarios. The results validate that our approach significantly outperforms state-of-the-art baselines, particularly in scenarios demanding complex coordination. Ablation studies further confirm the effectiveness of our method.

2 Related Work

Single-agent Exploration. Diverse exploration works have been investigated in single-agent scenarios. Count-based methods [3, 21], which have been widely adopted due to their theoretical guarantees, provide exploration bonuses inversely proportional to state-action visitation counts. Curiosity-driven methods [5, 26] provide intrinsic rewards based on prediction errors or novelty. Uncertainty-based approaches [2, 24] leverage epistemic uncertainty to guide exploration. Information-theoretic methods [7, 9] maximize information gain about the environment or policy. However, these approaches face fundamental challenges in multi-agent settings.

Multi-agent Exploration. Exploration in MARL presents challenges beyond single-agent settings due to the exponential growth of the joint state–action space and the need for coordinated exploration [8]. To navigate these challenges, researchers have proposed several classes of exploration mechanisms. One prominent line of work leverages influence as a signal for meaningful interaction. EITI and EDTI [37] measure the influence of an agent's behavior on the state transitions of other agents using mutual information and expected return, respectively. [12] further formalize the impact on other agents' behaviors as social influence and incorporate it as an intrinsic reward signal. Another class of methods focuses on promoting behavioral diversity among agents to encourage exploration. MAVEN [19] introduces a global latent variable to condition diverse joint policies, enabling multiple coordination strategies. CDS [17] explicitly regularizes policies to foster role emergence, while AIR [46] integrates individual identity with collective novelty to sustain long-term behavioral differentiation. A third research direction grounds exploration in state novelty and visitation-based signals. MULTI [11] proposes multiple intrinsic reward variants based on combining individual agents' curiosity, with a meta-policy dynamically selecting the reward type for each episode. EMC [45] adapts episodic memory to MARL by measuring novelty via similarity to past joint observations. In contrast, SMMAE [44] estimates local visitation density in a self-supervised manner to directly incentivize coverage of sparse regions. Beyond visitation-based novelty, SAME [41] prioritizes under-explored state subspaces through an uncertainty-weighted entropic objective, while TMAE [32] employs counterfactual causal inference to identify reward-relevant state subspaces for targeted exploration. From an action-space perspective, SEG [34] clusters actions with similar effects and adopts a bi-level ϵ-greedy strategy to explore action groups and individual actions within them. However, these methods operate within primitive observation spaces without explicitly modeling the latent semantic structure of the environment, often resulting in redundant exploration efforts.

Representation Learning in MARL. Representation learning provides a fundamental paradigm for extracting compact and informative latent structures from high-dimensional spaces. Variational autoencoder (VAE) [16] offers a foundational probabilistic framework for learning continuous latent representations via stochastic encoding and decoding. This framework has been extended to discrete latent representations by vector-quantized variational autoencoder (VQ-VAE) [35], which quantizes the latent space into a learned discrete codebook. The ability of these structured representations to mitigate multi-agent complexity has paved the way for research that explores representation learning to facilitate coordination and role specialization. NCC [20] leverages VAE to learn neighborhood cognitive consistency, encouraging similar representations among spatially proximate agents to promote coordinated behavior. ROMA [38] learns role-based continuous representations through action-space clustering, enabling agents to discover and specialize into different functional roles. CoS [28] employs VQ-VAE to learn discrete agent groupings for coordination. LAGMA [22] uses VQ-VAE to build a discrete latent space for generating goal-reaching trajectories. In contrast to previous works, this paper leverages representation learning to shape exploration in a compact semantic space, enabling statistical transfer across semantically similar situations and reducing redundant exploration in the raw observation–action space.

3 Preliminaries

Dec-POMDP. We consider a fully cooperative multi-agent scenario, which can be modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [23]. Formally, it is commonly defined as an eight-tuple $\langle N, S, \boldsymbol{A}, P, R, \boldsymbol{O}, Z, \gamma \rangle$, where N is the set of agents; S is the state space; $\boldsymbol{A} = A_1 \times \cdots \times A_N$ is the joint action space with A_i being the action space of agent i; $P(s' \mid s, \boldsymbol{a}) : S \times \boldsymbol{A} \times S \to [0, 1]$ is the state transition function; $\boldsymbol{O} = O_1 \times \cdots \times O_N$ is the joint observation space, where each agent i receives a local observation $o_i \in O_i$ according to the observation function $Z(\boldsymbol{o} \mid s, \boldsymbol{a}) : \boldsymbol{O} \times S \times \boldsymbol{A} \to [0, 1]$; $R(s, \boldsymbol{a}) : S \times \boldsymbol{A} \to \mathbb{R}$ is the shared reward function; and $\gamma \in [0, 1)$ is the discount factor. The objective is to find a joint policy $\boldsymbol{\pi} = \langle \pi_1, \ldots, \pi_N \rangle$ that maximizes the expected discounted return: $J(\boldsymbol{\pi}) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \boldsymbol{a}_t) \right]$.

Value Decomposition. MARL faces fundamental challenges that go beyond those in single-agent settings, such as credit assignment and scalability. To tackle these issues, value decomposition methods, such as QMIX [27] and QPLEX [36], decompose the global action-value function Q_{total} into individual agent utilities Q_i, which are then combined through a mixing function f: $Q_{\text{total}} = f(Q_1, \ldots, Q_n)$. Each agent i conditions its utility on its local action-observation history $\tau_i = (o_i^0, a_i^0, \ldots, o_i^t)$. These methods follow the Individual-Global-Max (IGM) principle to align individual and global optimality: $\forall i \in N,\ \arg\max_{a_i} Q_i(\tau_i, a_i) \in \arg\max_{\boldsymbol{a}} Q_{\text{total}}(\boldsymbol{\tau}, \boldsymbol{a})$. The model is trained by minimizing the temporal difference (TD) error:

$$\mathcal{L}_{\text{td}} = \left(r + \gamma \max_{\boldsymbol{a}'} Q^{-}_{\text{total}}(\boldsymbol{\tau}', \boldsymbol{a}') - Q_{\text{total}}(\boldsymbol{\tau}, \boldsymbol{a}) \right)^2, \tag{1}$$

where Q^{-}_{total} is a periodically updated target network of Q_{total}, r is the reward, and $\boldsymbol{\tau}'$ and $\boldsymbol{a}'$ denote the next-step joint action-observation histories and candidate actions, respectively.

VQ-VAE. VQ-VAE [35] employs vector quantization to learn discrete latent representations. It consists of three main components: an encoder f_ϕ, a learnable codebook $\mathcal{C} = \{e_k\}_{k=1}^{K}$ containing K embedding vectors $e_k \in \mathbb{R}^d$, and a decoder f_θ. Given an input x, the encoder maps it to a continuous latent representation $z_e = f_\phi(x) \in \mathbb{R}^d$. The continuous representation z_e is then quantized to the nearest codebook vector using nearest-neighbor lookup:

$$z_q = e_j, \quad \text{where} \quad j = \arg \min_{k \in \{1, \ldots, K\}} \|z_e - e_k\|_2. \tag{2}$$

The quantized representation z_q serves as the discrete semantic code, and the decoder reconstructs the input from this discrete code $\hat{x} = f_\theta(z_q)$. Because quantization is non-differentiable, gradients are passed from z_q to z_e as in the straight-through estimator. The VQ-VAE loss is written as

$$\mathcal{L}_{\text{VQ-VAE}} = \|x - \hat{x}\|_2^2 + \alpha \| \operatorname{sg}[z_e] - e_j \|_2^2 + \beta \| z_e - \operatorname{sg}[e_j] \|_2^2. \tag{3}$$

where sg denotes the stop-gradient operator, α and β are hyperparameters.

4 Method

In this section, we detail our semantic-level exploration method. Our method builds upon the classical value-based CTDE framework and introduces two core modules to enhance exploration efficiency: the Semantic Quantizer (SQ) and the Sliding Window Counting (SWC). The SQ module compresses high-dimensional observations or observation-action pairs into discrete semantic codes via a learned codebook. The SWC module maintains count-based exploration statistics on these semantic codes using a sliding window mechanism, providing exploration bonuses that are inversely proportional to visitation counts. These two modules are lightweight, parameter-sharing across agents, and can be seamlessly integrated with existing value decomposition methods.

Below, we first present our Vanilla_SEMEX method, which instantiates the semantic exploration framework using only observation semantics. Subsequently, we optimize Vanilla_SEMEX and propose SEMEX, which achieves joint observation-action semantic exploration.

4.1 Vanilla_SEMEX: Observation Semantics

4.1.1 Semantic Quantizer (SQ) module

We first illustrate the notion of semantic equivalence with Figure 1. As shown in the figure, each agent perceives a 3×3 egocentric grid, where each grid cell can be occupied by one of four discrete entity types (e.g., empty space, walls, apples, or other agents). Figure 1(a) and (c) are geometric transformations of Figure 1(b). For each agent, although the raw observations differ across the three views, they all originate from the same underlying semantic: agent A perceives two apples around, and agent B perceives one apple nearby. Such observations are functionally equivalent and should ideally be grouped together so that their exploration statistics can be shared. This example suggests that exploration should be conducted in the semantic space rather than in the low-dimensional space. To this end, the SQ module compresses high-dimensional observations into discrete semantics using a VQ-VAE model, enabling efficient count-based exploration in the semantic space.

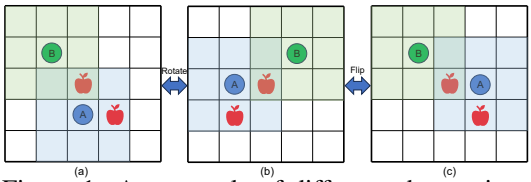


Figure 1: An example of different observations with the same semantic. (a) and (c) are obtained by rotating and flipping (b), respectively. Here, the observation semantic of agent A is two apples around, and that of agent B is one apple nearby.

Formally, we first partition observations into K semantic classes in Definition 1; we then introduce a mild Lipschitz assumption on the value network in Assumption 1 that bridges geometric and functional proximity; finally, we bound the within-class Q-value gap using Lemma 1 and Theorem 1, thereby justifying class-level aggregation of exploration statistics.

Definition 1. *The quantization scheme partitions the observation space O_i into K semantic observation classes $D_1, D_2, \ldots, D_K$, where each class consists of all observations assigned to the same code e_k, and every observation belongs to exactly one class.*

Assumption 1. *There exists a constant $\Lambda > 0$ such that for any action $a \in A_i$ and any $o, o' \in O_i$, $\left|Q_i(o,a) - Q_i(o',a)\right| \leq \Lambda \cdot \|o - o'\|_2$.*

Since Q_i is realized by a feed-forward network with 1-Lipschitz activations (e.g., ReLU, tanh) and bounded weights, its overall Lipschitz constant is bounded by $\Lambda \leq \prod_{l=1}^{H} \sigma_{\max}(W_l)$, where $\sigma_{\max}(W_l)$ denotes the spectral norm of the l-th weight matrix and H is the number of layers. Assumption 1 is therefore automatically satisfied by construction under standard regularization, and is widely adopted in the RL literature on smooth MDPs and value-function approximation [1].

Lemma 1. *Let p_k denote the probability that an observation is assigned to code e_k, with $\sum_{k=1}^{K} p_k = 1$. For a randomly chosen class e_k drawn with probability p_k, let o_1 and o_2 be two independent observations sampled from D_k. Then $\mathbb{E}[\|o_1 - o_2\|_2] \leq 2e_{\text{rec}}$, where $e_{\text{rec}}^2 = \mathbb{E}[\|o_1 - f_\theta(e_k)\|_2^2]$ is the reconstruction error of the SQ module.*

Proof. Since both o_1 and o_2 are assigned to e_k, the triangle inequality gives

$$\|o_1 - o_2\|_2 \leq \|o_1 - \hat{o}_k\|_2 + \|o_2 - \hat{o}_k\|_2. \tag{4}$$

Taking conditional expectations and applying Jensen's inequality yields

$$\mathbb{E}[\|o_1 - o_2\|_2 \mid k] \le 2\mathbb{E}[\|o_1 - \hat{o}_k\|_2 \mid k] \le 2\sqrt{\mathbb{E}[\|o_1 - \hat{o}_k\|_2^2 \mid k]}. \tag{5}$$

By the law of total expectation, and applying Jensen's inequality again, we obtain

$$\mathbb{E}[\|o_1 - o_2\|_2] \le 2\sum_{k=1}^{K} p_k \sqrt{\mathbb{E}[\|o_1 - \hat{o}_k\|_2^2 \mid k]} \le 2\sqrt{\sum_{k=1}^{K} p_k \mathbb{E}[\|o_1 - \hat{o}_k\|_2^2 \mid k]} = 2e_{\mathrm{rec}}. \tag{6}$$

□

Theorem 1. *Under Assumption 1, for any action $a \in A_i$, $\mathbb{E}\big[\,\big|Q_i(o_1, a) - Q_i(o_2, a)\big|\,\big] \;\le\; 2\,\Lambda \cdot e_{rec}$.*

Proof. By Assumption 1, we have $\big|Q_i(o_1, a) - Q_i(o_2, a)\big| \le \Lambda\,\|o_1 - o_2\|_2$. Taking expectation on both sides and applying Lemma 1,

$$\mathbb{E}\big[\,\big|Q_i(o_1, a) - Q_i(o_2, a)\big|\,\big] \;\le\; \Lambda \cdot \mathbb{E}\big[\,\|o_1 - o_2\|_2\,\big] \;\le\; 2\,\Lambda \cdot e_{\mathrm{rec}}. \qquad \square$$

This proves that observations falling into the same semantic class are functionally near-equivalent, with the deviation directly controlled by the VQ-VAE reconstruction loss. In our implementation, the Semantic Quantizer (SQ) removes the straight-through estimator and directly decodes from codebook vectors: the encoder learns $z_e^i = f_\phi^{\mathrm{obs}}(o_i)$, and the decoder learns $\hat{o}_i = f_\theta(e_c)$, where $c = \arg\min_k \|z_e^i - e_k\|_2$. Hence, the VQ-VAE loss is

$$\mathcal{L}_{\mathrm{VQ}} \;=\; \|o_i - \hat{o}_i\|_2^2 + \alpha\,\|\,\mathrm{sg}[z_e^i] - e_c\|_2^2 + \beta\,\|z_e^i - \mathrm{sg}[e_c]\|_2^2, \tag{7}$$

where $\mathrm{sg}[\cdot]$ is the stop-gradient operator, and α, β are hyperparameters. By minimizing this loss, the model is driven to learn a semantically coherent partitioning.

However, training a VQ-VAE codebook in non-stationary multi-agent environments introduces two practical challenges: some codebook entries may never be selected (dead codes), while active entries may collapse to a low-dimensional subspace. To maintain codebook health, we introduce two auxiliary mechanisms: dead-code resetting and prototype decorrelation.

Dead-Code Resetting. In practice, a subset of codebook entries may never be selected during training, becoming so-called dead codes. These unused prototypes waste representational capacity and degrade the effective resolution of the semantic partition. To address this, we periodically inspect the codebook utilization every T_{reset} forward passes. Let $\mathcal{S}_{\text{alive}}$ denote the set of active codes assigned to at least one sample in the current batch, and $\mathcal{S}_{\text{dead}}$ the set of dead codes. For each dead code $k \in \mathcal{S}_{\text{dead}}$, we reinitialize its embedding by copying a randomly chosen active code and adding small Gaussian noise:

$$e_k \leftarrow e_{k'} + \epsilon, \quad k' \sim \mathrm{Uniform}(\mathcal{S}_{\text{alive}}), \quad \epsilon \sim \mathcal{N}(0, \sigma_{\text{reset}}^2 I), \tag{8}$$

where σ_{reset} is a small constant. This procedure repositions dead codes near high-density regions of the latent space, allowing them to attract nearby encoder outputs in subsequent training steps and thereby improving codebook utilization.

Prototype Decorrelation. Even when all codebook entries are actively used, they may collapse to a low-dimensional subspace, reducing the expressiveness of the semantic partition. To encourage diversity among prototypes, we augment the SQ objective with a decorrelation regularizer. Specifically, let $\bar{e}_k = e_k / \|e_k\|_2$ denote the ℓ_2-normalized prototype. We penalize off-diagonal entries of the cosine similarity matrix:

$$\mathcal{L}_{\text{decorr}} = \frac{1}{K(K-1)} \sum_{j \neq k} \big(\bar{e}_j^\top \bar{e}_k\big)^2. \tag{9}$$

This drives pairwise cosine similarities toward zero, encouraging the prototypes to spread across orthogonal directions in the embedding space. As a result, the overall training objective becomes

$$\mathcal{L}_{\text{SQ}} = \mathcal{L}_{\text{VQ}} + \lambda_{\text{decorr}}\,\mathcal{L}_{\text{decorr}}, \tag{10}$$

where λ_{decorr} controls the strength of the decorrelation penalty.

4.1.2 Sliding Window Counting (SWC) module

The SWC module performs lightweight count-based exploration in the semantic space induced by the SQ module. We adopt a classical exploration bonus from [31], defined as $\frac{\eta}{\sqrt{M(s,a)}}$, where η is the exploration coefficient and $M(s,a)$ is the visit count of state–action pairs, encouraging exploration of infrequent regions. We extend this idea to multi-agent semantic exploration by redefining the count over semantic representations: for each agent i, we employ $M_i(e_k, a)$ to track the occurrences of taking action a under semantic code e_k.

To handle non-stationarity in multi-agent environments, we adopt a sliding-window strategy, where $M_i(e_k, a)$ is computed over the most recent W steps. This keeps the exploration bonus responsive to dynamic interactions. We detail the SWC module as follows.

Sliding-Window Count Table. For each agent i, we maintain a sliding-window count table $M_i \in \mathbb{R}^{K\times|A_i|}$, which records how often each action $a \in A_i$ is taken under each semantic code e_k ($k \in \{1, ..., K\}$) within the most recent W timesteps. Formally, the count is defined as

$$M_i^t(e_k, a) = \sum_{j=t-W+1}^{t} \mathbf{1}\left[e_i^j = e_k,\ a_i^j = a\right]. \tag{11}$$

Here, t denotes the current timestep, j indexes the historical timesteps within the sliding window, e_i^j and a_i^j denote the semantic code and action of agent i at timestep j, respectively.

Action Selection. Based on the count table M_i, we compute the exploration bonus for each action:

$$B_i(a) = \frac{\eta}{\sqrt{M_i^t(e_k, a) + 1}}. \tag{12}$$

Then each agent i selects greedy actions with exploration bonus:

$$a_i = \arg\max_{a\in A_i} \left[Q_i(o_i, a) + B_i(a)\right]. \tag{13}$$

This encourages exploration of infrequently taken actions under the current semantic e_c^i. Finally, the output Q-value $Q_i(o_i, a_i)$ is fed into the mixing network (by default following the QMIX architecture) to produce the joint action-value $Q_{total}(\boldsymbol{o}, \boldsymbol{a})$. Importantly, while the exploration bonus $B_i(a)$ guides decentralized action selection, it is not involved in the computation of Q_{total}. This prevents overestimation of Q_{total} and simultaneously enhances the agents' exploration behavior.

During training, we adopt a decoupled training strategy that separates the optimization of the agent networks from that of the semantic encoding. The network parameters of all agents are optimized based on the TD-loss L_{td} (Eq. (1)), while the semantic encoding is optimized using L_{SQ} (Eq. (10)). This decoupling avoids gradient interference and ensures training stability.

4.2 SEMEX: Observation-Action Joint Semantics

Although Vanilla_SEMEX enables the transfer of exploration statistics across similar semantics to reduce redundant exploration, it also suffers from over-generalization. As illustrated in Figure 1, agent A shares the same observation semantic in Figures 1(a)–(c). When agent A takes a rightward action in Figures 1(a)–(b), it successfully collects an apple. However, taking the same rightward action in Figure 1(c) yields no reward. In this case, because agent A in Figures 1(a)–(c) is associated with similar high-level observation semantics, Vanilla_SEMEX forces the exploration statistics to be shared across these situations, despite their different action outcomes. This leads to inappropriate over-generalization and is therefore undesirable.

In light of this, we propose SEMEX, which improves upon Vanilla_SEMEX by performing joint observation–action semantic exploration rather than relying on observations alone. As this modification only affects the SQ module, all other components remain identical to those in Vanilla_SEMEX. Therefore, we focus on the SQ module in the following.

SQ Module. For each action $a \in A_i$ of agent i, we first concatenate the observation encoding z_i^e with the action's one-hot representation, and feed it into the encoder to obtain the joint encoding,

$$z_i^{\text{joint}} = f_\phi^{\text{joint}}(\text{Concat}[z_i^e,\ \text{onehot}(a)]). \tag{14}$$

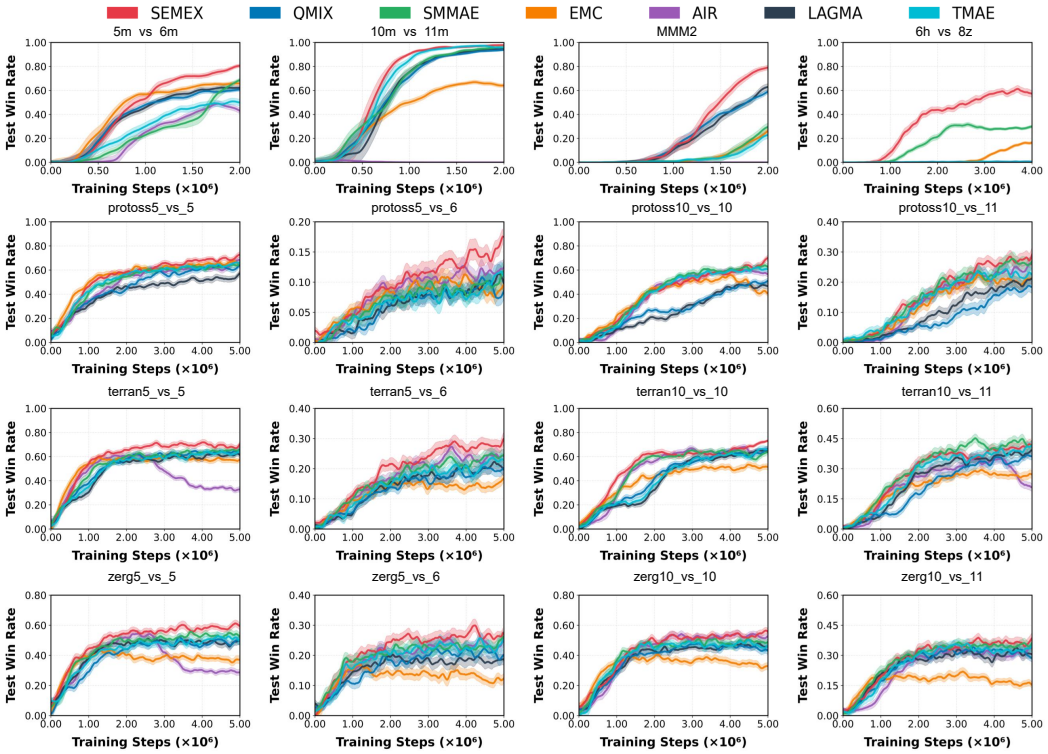


Figure 2: Performance comparison in DR_SMAC (first row) and SMACv2 (last three rows) tasks.

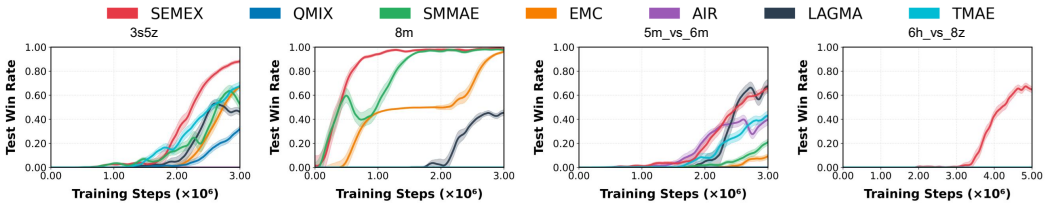


Figure 3: Performance comparison in different SR_SMAC tasks.

Then, the decoder reconstructs the observation encoding z_i^e rather than the joint encoding z_i^{joint} as

$$\hat{z}_i^e = f_\theta(e_k). \tag{15}$$

where $e_k = \arg\min_{e \in C} \|z_i^{\text{joint}} - e\|_2$, and $C = \{e_k\}_{k=1}^K$. This design ensures that the semantic codebook captures action-invariant observation features. Accordingly, the training objective of the SQ module is rewritten as

$$\mathcal{L}_{\text{SQ}} = \|z_e^i - \hat{z}_e^i\|_2^2 + \alpha \|\text{sg}[z_{\text{joint}}^i] - e_c\|_2^2 + \beta \|z_{\text{joint}}^i - \text{sg}[e_c]\|_2^2 + \lambda_{\text{decorr}} \mathcal{L}_{\text{decorr}}. \tag{16}$$

5 Experiments

We evaluate SEMEX on four challenging benchmarks: SMAC with dense reward (DR_SMAC) [29], SMACv2 [6], SMAC with sparse reward (SR_SMAC) [13] and Multi-Agent Coordination (MACO) [39]. We evaluate our method against six state-of-the-art baselines, all implemented using their official open-source code to ensure a fair comparison. All results are averaged over five independent runs with different random seeds.

5.1 Performance on Various Scenarios

DR_SMAC. SMAC is a StarCraft II micromanagement benchmark in which partially observable agents control allied units to defeat rule-based enemies. Figure 2 illustrates the performance of various

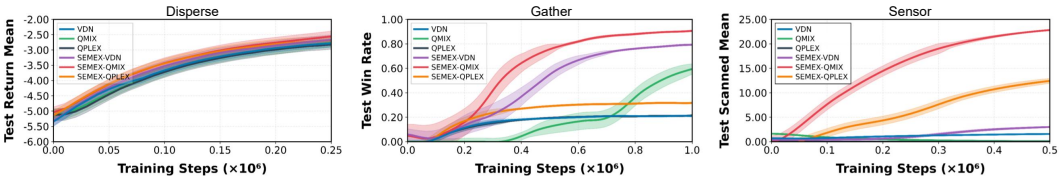


Figure 4: Performance comparison with baselines in different MACO scenarios.

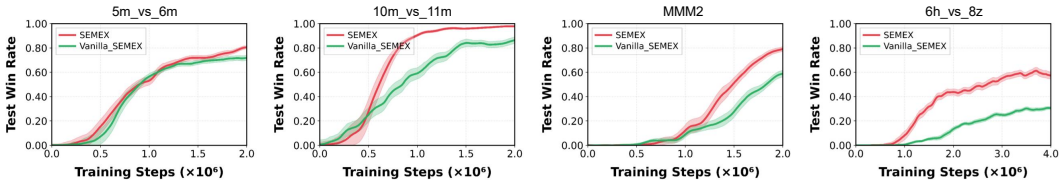


Figure 5: A comparison between SEMEX and Vanilla_SEMEX across four DR_SMAC tasks.

methods across four DR_SMAC tasks. It is evident that SEMEX consistently outperforms all methods in every task. This advantage is particularly pronounced in the more challenging *6h_vs_8z* task, where SEMEX achieves substantially higher win rates compared to all baselines. These results underscore that semantic-level exploration facilitates a more efficient discovery of effective coordination patterns, especially within high-complexity environments.

SMACv2. Compared to DR_SMAC, SMACv2 introduces greater diversity and stochasticity, demanding stronger generalization and adaptability from MARL algorithms. As illustrated in Figure 2, while the inherent randomness of these environments occasionally renders specific episodes intrinsically unsolvable, SEMEX consistently surpasses the baseline QMIX across all tested tasks. Furthermore, SEMEX demonstrates clear superiority over other state-of-the-art baselines in most scenarios, validating its capacity for robust, adaptive exploration within highly dynamic settings.

SR_SMAC. To further validate the robustness of our method in sparse-reward tasks, we conducted experiments in SR_SMAC tasks. As illustrated in Figure 3, SEMEX consistently achieves the highest win rates across all sparse-reward tasks. In particular, on the more challenging *6h_vs_8z* scenarios, most baselines struggle to learn effective policies under the sparse reward signal, whereas SEMEX converges reliably to high win rates. These results demonstrate that semantic-level exploration offers a clear advantage in low-reward settings, as the count-based bonus over semantic prototypes effectively guides agents toward rewarding coordination patterns that are difficult to discover through undirected exploration.

MACO. MACO scenario extends several classic cooperative problems by deliberately increasing their complexity through temporal extensions, stochastic dynamics, and larger team sizes. We evaluate the performance of classic value decomposition algorithms (VDN [33], QMIX [27], QPLEX [36]) and their SEMEX-enhanced variants across three classic MACO tasks: Gather, Disperse, and Sensor. The results are shown in Figure 4. SEMEX consistently improves performance over VDN, QMIX, and QPLEX on the *Gather* and *Sensor* tasks. On the *Disperse* task, where all methods achieve comparable results, SEMEX maintains baseline performance without degradation, indicating that it introduces no negative bias. Overall, SEMEX demonstrates strong plug-and-play compatibility as an effective enhancement for value-based MARL.

5.2 Ablation Study and Visualization Analysis

SEMEX vs. Vanilla_SEMEX. We compare SEMEX with Vanilla_SEMEX on DR_SMAC tasks (Figure 5). On relatively simple scenarios (5m_vs_6m and 10m_vs_11m), both methods perform well, suggesting that observation-only semantics can already provide sufficient exploration signals when optimal behaviors are less sensitive to action-context interactions. In contrast, on more challenging scenarios that require context-dependent action selection (MMM2 and 6h_vs_8z), SEMEX achieves substantially higher win rates than Vanilla_SEMEX. We attribute this gap to the over-generalization inherent in observation-only semantics. Overall, these ablation results support our key design insight: effective semantic exploration in MARL should operate at the granularity of observation–action pairs.

Codebook Maintenance Mechanisms. We evaluate the two auxiliary codebook maintenance mechanisms by considering two ablation variants: (1) SEMEX w/o Decorr (removing prototype

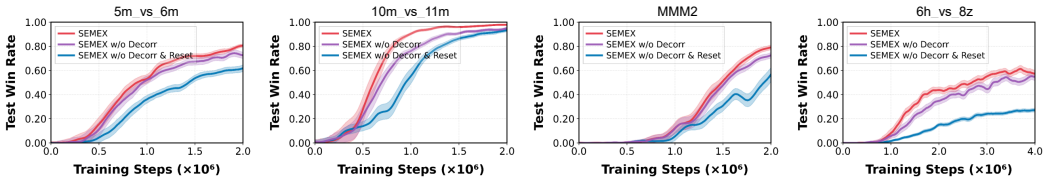


Figure 6: Ablation study on codebook maintenance mechanisms across DR_SMAC tasks.

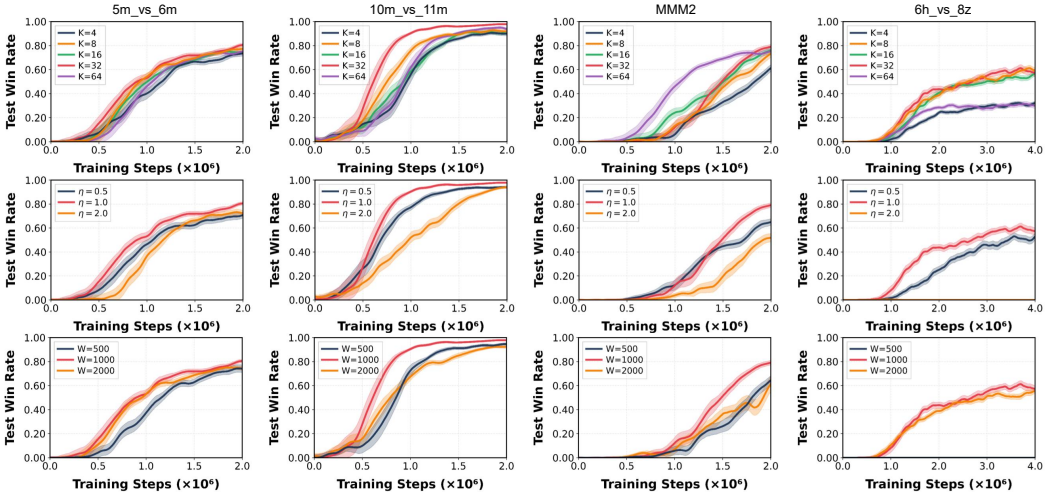


Figure 7: Hyperparameter sensitivity analysis across DR_SMAC tasks.

decorrelation), and (2) SEMEX w/o Decorr & Reset (removing both prototype decorrelation and dead-code resetting). As shown in Figure 6, removing decorrelation alone results in only a modest performance decline, since dead-code resetting can partially compensate by repositioning unused entries near active regions. However, removing both mechanisms leads to significant degradation, indicating that the two provide complementary benefits: dead-code resetting prevents codebook collapse by reviving inactive entries, while decorrelation prevents representational collapse by diversifying active entries.

Hyperparameter Sensitivity. We investigate the sensitivity of three key hyperparameters—codebook size K, exploration coefficient η, and sliding window size W—across DR_SMAC tasks, with results shown in Figure 7. For $K \in \{4, 8, 16, 32, 64\}$, performance improves as K increases from 4 to 32 but degrades at 64: too few codes merge distinct semantics, while too many inhibit statistical transfer. For $\eta \in \{0.5, 1.0, 2.0\}$, a small η leads to under-exploration, whereas a large η causes the bonus to dominate Q-values; $\eta = 1.0$ achieves the best balance between exploration and exploitation. For $W \in \{500, 1000, 2000\}$, a short window causes forgetting of visited regions, leading to redundant re-exploration, while a long window retains stale counts that lag behind non-stationary dynamics; $W = 1000$ optimally trades off historical coverage and adaptiveness. We adopt $K = 32$, $\eta = 1.0$, and $W = 1000$ as default settings.

Interpretability of the learned semantics. Beyond aggregate performance, we examined whether the learned semantic prototypes correspond to coherent behavioral contexts rather than arbitrary discrete identifiers. Appendix C provides two complementary analyses.

5.3 Training Efficiency

To evaluate computational efficiency, Table 1 reports the average training times across eight SMACv2 tasks. As shown, SEMEX achieves the shortest training time among all compared MARL methods, with only a modest overhead over the backbone QMIX. The underlying reason is that SMMAE and EMC introduce per-agent

Table 1: Time cost (h) of different algorithms, where P and T are the abbreviations for Protoss and Terran respectively.

Algorithm	P5v5	P5v6	P10v10	P10v11	T5v5	T5v6	T10v10	T10v11	**Avg.**
QMIX	21.4	21.2	35.1	33.0	20.3	27.7	37.7	38.7	**29.3**
SMMAE	33.0	35.6	70.0	65.3	41.0	42.1	71.7	71.3	**53.75**
EMC	95.0	103.9	212.7	160.5	140.9	136.7	170.6	195.8	**152.0**
AIR	22.6	27.5	42.2	44.2	30.0	30.4	45.2	51.2	**36.7**
LAGMA	29.9	36.8	42.4	44.4	36.8	34.1	50.0	51.3	**40.7**
TMAE	35.6	36.8	46.2	49.5	37.4	40.1	48.3	49.3	**42.9**
SEMEX	22.4	25.5	35.3	39.6	27.8	28.8	40.6	39.5	**32.46**

modules, significantly increasing computational and memory costs. In contrast, AIR relies on a shared identity classifier with relatively low overhead. SEMEX instead performs exploration in a compact semantic space induced by a shared VQ-VAE encoder, enabling all agents to share a lightweight representation and exploration model. Although LAGMA also adopts VQ-VAE, its extended codebook and trajectory management introduce additional overhead. TMAE introduces additional computation by training a reward model and periodically estimating the causal effects of state subspaces through counterfactual interventions to guide targeted exploration.

5.4 Scalability in Large-Scale Scenarios

To further evaluate the performance of SEMEX in large-scale multi-agent scenarios, we conducted experiments on a custom SMACv2 task featuring 50 Marines vs. 50 Marines (referred to as *M50v50*). Due to the extremely high training cost and large memory footprint of EMC, SMMAE, TMAE and LAGMA in the *M50v50* scenario, we were unable to successfully run these methods and encountered out-of-memory errors. Therefore, for this scenario, we only report comparisons with AIR and QMIX. As demonstrated in Figure 8, SEMEX achieves a win rate of 80%, substantially outperforming both AIR and QMIX. These results validate the superior scalability of SEMEX in handling large-scale tasks.

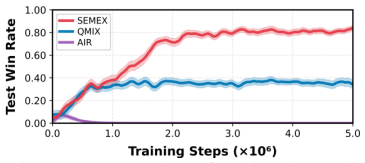


Figure 8: Results of M50v50 task.

6 Conclusion

This paper proposes a semantic exploration method called SEMEX to address the fundamental challenge of exponentially growing joint state-action spaces in MARL. SEMEX instantiates semantic-level exploration through vector quantization that compresses high-dimensional observation-action spaces into discrete semantic prototypes, and sliding-window count-based bonuses that enable efficient statistical transfer across semantically similar situations. Extensive experiments on challenging MARL environments demonstrate that SEMEX outperforms state-of-the-art baselines.

7 Limitation

SEMEX has several limitations. First, our evaluation focuses on cooperative benchmarks with structured observations, and its applicability to image-based observations settings remains unverified. Second, the semantic codebook has a fixed capacity, requiring its size to be selected according to environment complexity. Although the evaluated tasks perform well with a small codebook, too few codes may merge behaviorally distinct experiences, whereas too many may fragment visitation statistics and reduce statistical sharing. Future work could investigate adaptive codebook sizing and extend semantic exploration to visual multi-agent tasks.

Acknowledgments and Disclosure of Funding

The authors thank the anonymous area chair and the reviewers for their insightful comments. This work was supported in part by the National Natural Science Foundation of China (Grant No. 62302189 and No. 62506210), the Key Research and Development Project of Hubei Province, China (Grant No. 2024BAB070), and the CAS Pioneer Hundred Talents Program (Grant No. 2026000063/E6YB01T and No. 2026000065/E6SR01T).

References

[1] Kavosh Asadi, Dipendra Misra, and Michael L. Littman. Lipschitz continuity in model-based reinforcement learning. In *International Conference on Machine Learning*, pages 264–273. PMLR, 2018.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2):235–256, 2002.

[3] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andew Bolt, et al. Never give up: Learning directed exploration strategies. *arXiv preprint arXiv:2002.06038*, 2020.

[4] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[5] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[6] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:37567–37593, 2023.

[7] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

[8] Jianye Hao, Tianpei Yang, Hongyao Tang, Chenjia Bai, Jinyi Liu, Zhaopeng Meng, Peng Liu, and Zhen Wang. Exploration in deep reinforcement learning: From single-agent to multiagent domain. *IEEE Transactions on Neural Networks and Learning Systems*, 35(7):8762–8782, 2023.

[9] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

[10] Min Hua, Xinda Qi, Dong Chen, Kun Jiang, Zemin Eitan Liu, Hongyu Sun, Quan Zhou, and Hongming Xu. Multi-agent reinforcement learning for connected and automated vehicles control: Recent advancements and future prospects. *IEEE Transactions on Automation Science and Engineering*, 2025.

[11] Shariq Iqbal and Fei Sha. Coordinated exploration via intrinsic rewards for multi-agent reinforcement learning. *arXiv preprint arXiv:1905.12127*, 2019.

[12] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pages 3040–3049. PMLR, 2019.

[13] Jeewon Jeon, Woojun Kim, Whiyoung Jung, and Youngchul Sung. Maser: Multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *International conference on machine learning*, pages 10041–10052. PMLR, 2022.

[14] Yonghyeon Jo, Sunwoo Lee, and Seungyul Han. Retaining suboptimal actions to follow shifting optima in multi-agent reinforcement learning. *arXiv preprint arXiv:2602.17062*, 2026.

[15] Dilshani Karunarathna, Nuwan Jaliyagoda, Ganindu Jayalath, Janaka Alawatugoda, Roshan Ragel, and Isuru Nawinne. Mixed-reality based multi-agent robotics framework for artificial swarm intelligence experiments. *IEEE Access*, 11:105718–105733, 2023.

[16] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] Chenghao Li, Tonghan Wang, Chengjie Wu, Qianchuan Zhao, Jun Yang, and Chongjie Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:3991–4002, 2021.

[18] Pei Li, Yongkang Zhang, Zhonglin Lv, jinming Zhu, and Jiangjin Yin. CAIC: Congestion-aware intent communication for multi-agent reinforcement learning. In *Forty-Second Annual Conference on Uncertainty in Artificial Intelligence*, 2026. URL `https://openreview.net/forum?id=n2YWRp39tM`.

[19] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. Maven: Multi-agent variational exploration. *Advances in neural information processing systems*, 32, 2019.

[20] Hangyu Mao, Wulong Liu, Jianye Hao, Jun Luo, Dong Li, Zhengchao Zhang, Jun Wang, and Zhen Xiao. Neighborhood cognition consistent multi-agent reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7219–7226, 2020.

[21] Jarryd Martin, Suraj Narayanan Sasikumar, Tom Everitt, and Marcus Hutter. Count-based exploration in feature space for reinforcement learning. *arXiv preprint arXiv:1706.08090*, 2017.

[22] Hyungho Na and Il-chul Moon. Lagma: Latent goal-guided multi-agent reinforcement learning. *arXiv preprint arXiv:2405.19998*, 2024.

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

[24] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016.

[25] Georg Ostrovski, Marc G Bellemare, Aäron van den Oord, and Rémi Munos. Count-based exploration with neural density models. In *International Conference on Machine Learning*, pages 2721–2730. PMLR, 2017.

[26] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[27] 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.

[28] Jingqing Ruan, Xiaotian Hao, Dong Li, and Hangyu Mao. Learning to collaborate by grouping: A consensus-oriented strategy for multi-agent reinforcement learning. *arXiv preprint arXiv:2307.15530*, 2023.

[29] 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.

[30] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pages 5887–5896. PMLR, 2019.

[31] Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

[32] Chuxiong Sun, Dunqi Yao, Rui Wang, Wenwen Qiang, Changwen Zheng, and Jiangmeng Li. Tmae: Learning targeted multi-agent exploration via causal inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 25682–25690, 2026.

[33] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

[34] Hon Tik Tse and Ho-fung Leung. Exploiting semantic epsilon greedy exploration strategy in multi-agent reinforcement learning. *arXiv preprint arXiv:2201.10803*, 2022. URL `https://arxiv.org/abs/2201.10803`.

[35] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

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

[37] Tonghan Wang, Jianhao Wang, Yi Wu, and Chongjie Zhang. Influence-based multi-agent exploration. *arXiv preprint arXiv:1910.05512*, 2019.

[38] Tonghan Wang, Heng Dong, Victor Lesser, and Chongjie Zhang. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*, 2020.

[39] Tonghan Wang, Liang Zeng, Weijun Dong, Qianlan Yang, Yang Yu, and Chongjie Zhang. Context-aware sparse deep coordination graphs. *arXiv preprint arXiv:2106.02886*, 2021.

[40] Zhizun Wang and David Meger. Multi-agent model-based reinforcement learning with joint state-action learned embeddings. *arXiv preprint arXiv:2602.12520*, 2026.

[41] Pei Xu, Junge Zhang, Qiyue Yin, Chao Yu, Yaodong Yang, and Kaiqi Huang. Subspace-aware exploration for sparse-reward multi-agent tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11717–11725, 2023. doi: 10.1609/aaai.v37i10.26384.

[42] Zhiwei Xu, Kun Hu, Xin Xin, Weiliang Meng, Yiwei Shi, Hangyu Mao, Bin Zhang, Dapeng Li, and Jiangjin Yin. Reidentify: Context-aware identity generation for contextual multi-agent reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

[43] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

[44] Shaowei Zhang, Jiahan Cao, Lei Yuan, Yang Yu, and De-Chuan Zhan. Self-motivated multi-agent exploration. *arXiv preprint arXiv:2301.02083*, 2023.

[45] Lulu Zheng, Jiarui Chen, Jianhao Wang, Jiamin He, Yujing Hu, Yingfeng Chen, Changjie Fan, Yang Gao, and Chongjie Zhang. Episodic multi-agent reinforcement learning with curiosity-driven exploration. *Advances in Neural Information Processing Systems*, 34:3757–3769, 2021.

[46] Guangchong Zhou, Zeren Zhang, and Guoliang Fan. Air: Unifying individual and collective exploration in cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22919–22927, 2025.

[47] Quan Zhou, Yonggui Li, and Yingtao Niu. Intelligent anti-jamming communication for wireless sensor networks: A multi-agent reinforcement learning approach. *IEEE Open Journal of the Communications Society*, 2:775–784, 2021.

A Algorithmic Description

Algorithm 1 presents the pseudo-code of SEMEX, and the source code can be found in the supplementary material.

Algorithm 1 The training procedure of our SEMEX algorithm.

Require: Codebook $C = \{e_k\}_{k=1}^{K}$, window size W, exploration coefficient η, and reset interval T_{reset}

Initialize the QMIX networks, SQ module, codebook C, and count table $M_i^t(e_k, a)$
Initialize the timestep counter $t \leftarrow 0$
for each training episode **do**
  Reset the environment
  while the episode is not terminated **do**
    for each agent i **do**
      Obtain the local observation o_i and compute $Q_i(o_i, a)$ for each available action $a \in A_i$
      Encode the observation: $z_i^e = f_\phi^{\text{obs}}(o_i)$
      for each available action a **do**
        Compute the joint encoding: $z_i^{\text{joint}} = f_\phi^{\text{joint}}\big(\text{Concat}[z_i^e, \text{onehot}(a)]\big)$
        Assign it to the nearest code: $e_k = \arg\min_{e \in C} \|z_i^{\text{joint}} - e\|_2$
        Compute the exploration bonus: $B_i(a) = \eta / \sqrt{M_i^t(e_k, a) + 1}$
      end for
      Select $a_i = \arg\max_{a \in A_i} [Q_i(o_i, a) + B_i(a)]$
    end for
    Execute the joint action and store the transition
    $t \leftarrow t + 1$
    Update $M_i^t(e_k, a)$ using the most recent W timesteps according to Eq.(11)
    Update the agent and mixing networks using $\mathcal{L}_{\text{td}}$
    if $t \bmod 5 = 0$ **then**
      Update the SQ module and codebook using $\mathcal{L}_{\text{SQ}}$
    end if
    if the number of forward passes reaches T_{reset} **then**
      Identify $\mathcal{S}_{\text{alive}}$ and $\mathcal{S}_{\text{dead}}$
      Reset each e_k, $k \in \mathcal{S}_{\text{dead}}$, according to Eq.(8)
    end if
  end while
end for

B Hyperparameters

All experiments in this paper are conducted on the machine with the following configuration: NVIDIA GeForce RTX 4090 D and AMD EPYC 9654 96-Core Processor. Table 2 shows the hyperparameters used in our algorithms.

The open-source implementations for all baseline methods are provided below.

- VDN: *https://github.com/Louiii/ValueDecomposition*
- QMIX: *https://github.com/starry-sky6688/MARL-Algorithms*
- QPLEX: *https://github.com/wjh720/QPLEX*
- SMMAE: *https://github.com/Zhang-Shaowei/SMMAE*
- EMC: *https://github.com/kikojay/EMC*
- AIR: *https://github.com/Jugg1er/AIR*
- LAGMA: *https://github.com/aailabkaist/LAGMA*

Table 2: Hyperparameters used in the three experimental scenarios.

Hyperparameter	**DR_SMAC**	**SR_SMAC**	**SMACv2**	**MACO**
Optimizer	Adam	Adam	Adam	Adam
Learning rate	5×10^{-4}	5×10^{-4}	5×10^{-4}	5×10^{-4}
Learning rate for VQ-VAE	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Hidden dimension	64	64	64	64
VQ-VAE embedding dimension	32	32	32	32
Activation function	ReLU	ReLU	ReLU	ReLU
Buffer size	5000	5000	5000	5000
α	0.25	0.25	0.25	0.25
β	1.0	1.0	1.0	1.0
η	1.0	1.0	1.0	1.0
K	32	32	32	16
W	1000	1000	1000	1000
λ_{decorr}	0.01	0.01	0.01	0.01

C Visualization Analysis

C.1 Behavioral Interpretation of Learned Semantic

We rolled out the trained model on 5m_vs_6m and collected approximately 35,000 records together with their semantic-code assignments. Considering that presenting all code features would be overly redundant, we selected three representative codes for detailed analysis.

Table 3: Behavioral profiles and semantic interpretations of representative semantic codes.

Code	**Observation (Mean Value)**	**Action**	**Semantic**
13	Own health: 0.59; enemies in sight: 2.9	**60.3%** move, **39.7%** attack	Kiting
14	Own health: 0.28; enemies in sight: 1.2	**100%** stop	Holding the line
28	Own health: 0.54; enemies in sight: 3.5	**93.3%** attack	Proactive engagement

As shown in Table 3, **Code** denotes semantic-code index, **Observation** reports the mean statistics of the local observations assigned to that code, **Action** summarizes the corresponding action distribution, and **Semantic** provides a post-hoc interpretation of the resulting observation–action pattern.

Code 13 — Kiting. Code 13 is associated with agents that retain moderate health and observe several nearby enemies. Its action distribution contains both movement and attack actions, rather than being dominated by either one. This mixed pattern is consistent with kiting behavior, in which an agent alternates between repositioning and engaging the enemy instead of continuously attacking from a fixed position.

Code 14 — Holding the line. Code 14 is associated with agents that have the lowest average health among the three codes and observe relatively few enemies. Its action distribution is entirely dominated by the stop action. This combination corresponds to holding the line, where the agent maintains its position under an unfavorable local condition rather than moving or initiating an attack.

Code 28 — Proactive engagement. Code 28 is associated with agents that retain moderate health and observe the largest number of enemies among the three codes. Its action distribution is strongly dominated by attack actions, accounting for 93.3% of the assigned records. This combination corresponds to proactive engagement, where the agent directly attacks under a dense enemy presence rather than repositioning or waiting.

The comparison among these codes is also informative. Codes 13 and 28 have similar average health, but they differ substantially in both the number of visible enemies and the action distribution: Code 13 mixes movement and attack, whereas Code 28 is strongly attack-dominated. This suggests that the codebook is not merely partitioning semantics according to health. These results provide empirical evidence that the learned codebook organizes observation–action experiences according to coherent behavioral patterns. Observation–action pairs assigned to the same code exhibit similar local

conditions and action tendencies, whereas different codes correspond to distinct tactical responses. Therefore, the learned codes are not arbitrary quantization bins, but represent high-level semantics in multi-agent decision-making.

C.2 Observation Semantic versus Observation–Action Semantic

To analyze the difference between the observation semantic and the observation-action semantic, we further conducted a visualization in the Gather environment. As shown in Fig. 9(a), the agent is located at $(1, 3)$, while the randomly selected goal is located at $(1, 4)$, as indicated by the green box. The agent can move up, down, left, or right, or remain stationary, but moving right is the only action that directly reaches the goal.

As shown in Fig. 9(b), Vanilla_SEMEX constructs semantics from the observation alone. Under the same observation, all candidate actions are shown in the same color, indicating that they share one semantic code regardless of whether they reach the goal, leave the agent in place, or move it away. In contrast, Fig. 9(c) uses different colors to indicate the semantic codes assigned by SEMEX to observation–action pairs. Although moving up, down, and left are different atomic actions, they share a code because they all move the agent away from the goal. Moving right and remaining stationary receive different codes because one reaches the goal whereas the other leaves the agent in place.

Together, these analyses provide empirical evidence that SEMEX groups observation–action pairs and captures the functional impact of each action within its specific observational context, thereby avoiding the over-generalization exhibited by Vanilla_SEMEX.

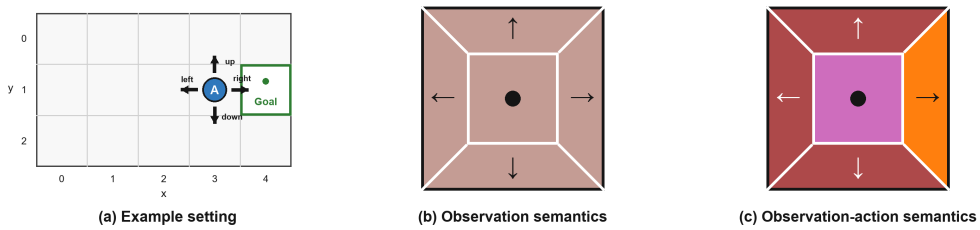


Figure 9: Observation-only and observation–action semantic assignments in the Gather environment. **(a)** Example state and candidate actions. **(b)** Observation semantics produced by Vanilla_SEMEX. **(c)** Observation–action semantics produced by SEMEX.

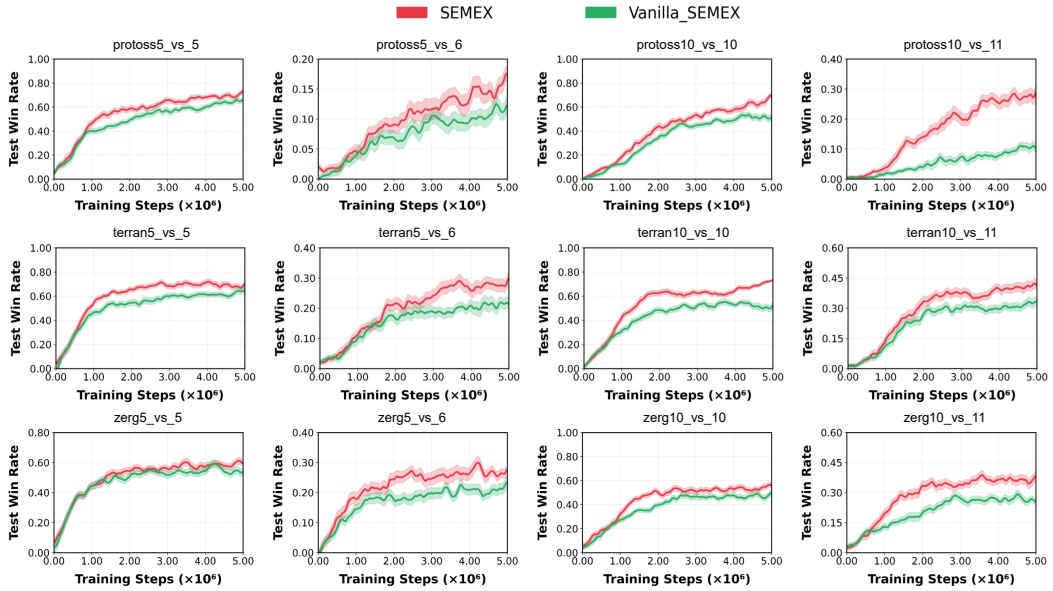


Figure 10: Comparison of SEMEX and Vanilla_SEMEX in SMACv2 tasks.

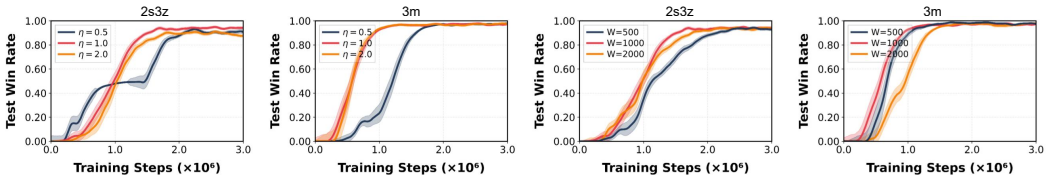


Figure 11: Hyperparameter sensitivity analysis across SR_SMAC tasks.

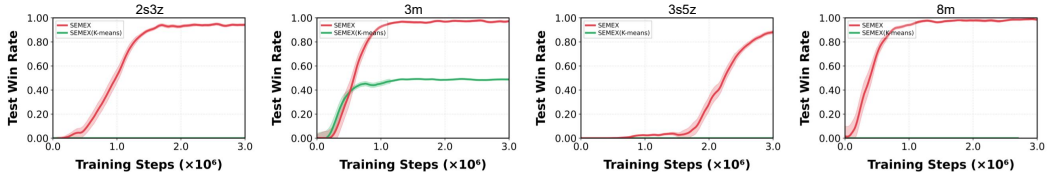


Figure 12: Comparison of SEMEX and SEMEX (K-means) across SR_SMAC tasks.

D Additional Experimental Results

Figure 10 shows the results of SEMEX and Vanilla_SEMEX in SMACv2 scenario. It can be seen that SEMEX consistently outperforms Vanilla_SEMEX across all twelve tasks. This confirms the necessity for semantic exploration to operate at the granularity of observation-action pairs.

Figure 11 shows the sensitivity of SEMEX to two key hyperparameters on SR_SMAC: the exploration bonus coefficient η and the window size W. We thus adopt $\eta = 1.0$ and $W = 1000$ as defaults.

Figure 12 compares SEMEX with a variant that replaces the SQ module with K-means in SR_SMAC scenario. SEMEX consistently outperforms the K-means variant, which exhibits substantial performance degradation and fails to learn effective policies in most evaluated scenarios.

E Environments

E.1 StarCraft Multi-Agent Challenge (SMAC)

The StarCraft Multi-Agent Challenge (SMAC) is a benchmark environment specifically designed to evaluate and advance Multi-Agent Reinforcement Learning (MARL) algorithms within the complex context of the real-time strategy game StarCraft II. It emphasizes micromanagement tasks, where independent learning agents control individual units and must coordinate their actions based solely on local observations. As depicted in Figure 13(a), agents are able to observe various attributes such as distance, position, health, shield, and type within their sight range, with the ultimate goal of defeating enemy units. The environment enables centralized training while supporting decentralized execution, offering additional state information during training to facilitate algorithm development. The details of the experimental tasks used in our study are provided in Table 4. Table 5 presents the reward settings for SR_SMAC which follows the sparse reward settings from [13].

E.2 SMACv2

The SMACv2 is a benchmark environment that extends SMAC to evaluate MARL algorithms under heightened stochasticity and diversity within the StarCraft II micromanagement domain. As shown in Figure 13(b), in each episode, procedurally generated team compositions, randomized allied and enemy starting positions—either mirrored or encircling—and unit-specific attack and sight ranges sampled from the original game mechanics jointly create a previously unknown battlefield. As a result, agents must coordinate while coping with uncertainty in their own unit types, initial locations, and relative engagement distances, all inferred solely from local observations of distance, position, health, shield, and unit type within their sight range. SMACv2 retains the CTDE paradigm, providing extra state information during training to foster development of robust, adaptive strategies that generalize beyond fixed unit line-ups and deterministic spawn patterns. The details of the experimental tasks used in our study are provided in Table 6.

E.3 MACO

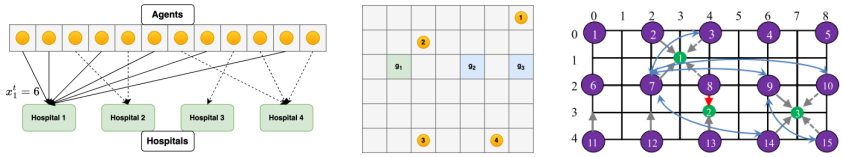
MACO extends several classic cooperative problems by intentionally enhancing their complexity through temporal extension, stochastic dynamics, and larger team sizes. We evaluate three MACO tasks: Disperse, Gather, and Sensor.

- Disperse task (see Figure 14(a)) features twelve agents repeatedly selecting among four hospitals at each timestep, with only one hospital requiring a non-zero and dynamically changing number of agents. Agents incur penalties for either under-staffing or over-staffing, compelling them to adaptively redistribute themselves based on limited and time-varying local information.
- Gather task (see Figure 14(b)) is a temporally extended, partially observable version of the classic Climb Game, involving five agents. At the start of each episode, one of three possible goal locations is randomly selected as the optimal target; however, only agents initially spawned in its vicinity are aware of the correct choice. To receive a team reward, all agents must navigate to and arrive simultaneously at the same goal. The reward is high if everyone converges on the optimal goal, moderate if they all select a suboptimal one, and severely penalized if they split across different goals.
- Sensors task (see Figure 14(c)) constructs a multi-agent partially observable environment, in which a group of fixed sensor agents must collaboratively detect multiple targets moving randomly on a grid map through active perception actions with limited fields of view. Each agent can only observe its local neighborhood and must decide at each time step whether to perform a scanning action and, if so, in which direction. Each scan incurs a cost. The system is designed such that a positive reward is granted only when at least two agents simultaneously observe the same target within the same time step.



(a) SMAC (b) SMACv2

Figure 13: SMAC and SMACv2 scenarios.



(a) Disperse (b) Gather (c) Sensor

Figure 14: MACO scenarios.

Table 4: Different tasks in SMAC scenarios.

Task	Ally Units	Enemy Units	Type
3m	3 Marines	3 Marines	homogeneous
8m	8 Marines	8 Marines	homogeneous
2s3z	2 Stalkers 3 Zealots	2 Stalkers 3 Zealots	heterogeneous
3s5z	3 Stalkers 5 Zealots	3 Stalkers 5 Zealots	heterogeneous
5m_vs_6m	5 Marines	6 Marines	homogeneous & asymmetric
10m_vs_11m	10 Marines	11 Marines	homogeneous & asymmetric
MMM2	1 Medivac 2 Marauders 7 Marines	1 Medivac 3 Marauders 8 Marines	heterogeneous & asymmetric
6h_vs_8z	6 Hydralisks	8 Zealots	micro-trick: focus fire

Table 5: Reward settings for SR_SMAC.

Condition	Sparse reward
All enemies die (Win)	+200
Each enemy dies	+10
Each ally dies	-5

Table 6: Different map tasks in SMACv2 scenarios.

Tasks	Ally count	Enemy count	Race	Unit types	Probability of generation
protoss5_vs_5	5	5	Protoss	Stalker, Zealot, Colossus	0.45, 0.45, 0.1
protoss5_vs_6	5	6			
protoss10_vs_10	10	10			
protoss10_vs_11	10	11			
terran5_vs_5	5	5	Terran	Marine, Marauder, Medivac	0.45, 0.45, 0.1
terran5_vs_6	5	6			
terran10_vs_10	10	10			
terran10_vs_11	10	11			
zerg5_vs_5	5	5	Zerg	Zergling, Baneling, Hydralisk	0.45, 0.1, 0.45
zerg5_vs_6	5	6			
zerg10_vs_10	10	10			
zerg10_vs_11	10	11			
terran50_ vs_50	50	50	Terran	Marine, Marauder, Medivac	1.0, 0.0, 0.0

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the contributions: (i) introducing semantic space into multi-agent exploration with a theoretical foundation, (ii) the SEMEX method combining a Semantic Quantizer and Sliding Window Counting, and (iii) superior performance over state-of-the-art baselines on SMAC, SMACv2, and MACO benchmarks. All claims are supported by the theoretical and experimental results in Sections 4 and 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Limitaion (Section 7).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Definition 1 formalizes the VQ-VAE-induced semantic partition. Assumption 1 is explicitly stated and accompanied by a remark explaining its automatic satisfaction for neural-network parameterizations of Q_i with bounded weights. Figure 1 provides an illustrative example of the motivating intuition. Lemma 1 and Theorem 1 are stated in Section 4.1.1.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 4 specifies the full architecture and algorithm. Section 5 provides experimental settings, and the supplementary material includes detailed hyperparameter configurations for all tasks. All baseline methods follow their official open-source implementations with original hyperparameter settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The experiments are conducted on public MARL benchmarks, and the implementation of the proposed method is provided in the supplementary material with the necessary instructions and hyperparameter settings for reproduction. The baseline implementations are also credited with their corresponding public repositories.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details including hyperparameters (codebook size K, exploration coefficient η, window size W, learning rate, batch size) are provided in Section 5 and the supplementary material. Environment configurations follow standard benchmark settings.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: All learning curves report the mean performance over five independent runs with different random seeds. Shaded regions in all figures denote the standard deviation across seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Table 1 reports per-task training times for all algorithms across eight SMACv2 tasks. Hardware specifications are provided in the supplementary material.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research involves only simulated multi-agent reinforcement learning benchmarks with no human subjects, personal data, or sensitive content, and conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational algorithmic research on exploration in multi-agent reinforcement learning and is not tied to any specific deployment or application. We do not identify a direct path to negative societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any high-risk data or models; the studied artifacts are MARL policies trained on public simulation benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All benchmarks (SMAC, SMACv2, MACO) and baseline algorithms (QMIX, SMMAE, EMC, AIR, LAGMA) are publicly available academic assets cited at their original publications throughout the paper. Our implementation builds on publicly released MARL frameworks.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: No new datasets or pre-trained models are released with the submission.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as a component of the proposed method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.