Structural Entropy Optimized Communication for Multi-Agent Reinforcement Learning

Wei Du[1,2] **Benyu Wu**[1] **Wei Guo**[1,2] **Zhongmin Yan**[1] **Guoxian Yu**[1,2] **Lizhen Cui**[1,2*]
[1]School of Software, Shandong University, China
[2]Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR), Shandong University, China
duwei@sdu.edu.cn, bywu@mail.sdu.edu.cn, {guowei, yzm, gxyu, clz}@sdu.edu.cn

Abstract

Communication learning is crucial for complex coordination in multi-agent reinforcement learning (MARL), yet existing methods often struggle with a fundamental dilemma between communication overhead and task performance. Typically designed without theoretical guidance, they produce unstructured and redundant communication graphs, limiting both efficiency and scalability. To address this, we propose Structural Entropy Optimized Communication (SEOC), a framework that directly optimizes inter-agent communication topology using the information-theoretic principle of Structural Entropy. SEOC guides the network to self-organize into a low-entropy state with sparse connectivity and distinct communities, enabling efficient and interpretable decentralized coordination. This is achieved through a unified mechanism that jointly enforces structural order and semantic conciseness, reducing topological complexity via a differentiable structural entropy loss while compressing messages with a node information bottleneck. Experiments on benchmarks such as the StarCraft Multi-Agent Challenge (SMACv2) demonstrate that SEOC significantly outperforms state-of-the-art methods.

1 Introduction

Multi-Agent Reinforcement Learning (MARL) has become a central paradigm for solving complex, decentralized control problems, with applications ranging from autonomous vehicle management [Xu et al., 2024, Jiang et al., 2024] to traffic signal control [Ma et al., 2024, Wang et al., 2025]. A fundamental challenge in cooperative MARL is the partial observability inherent in decentralized systems, where each agent possesses only a limited view of the global state. To overcome this, learning to communicate has emerged as a key solution, enabling agents to share information, infer intent, and formulate sophisticated collaborative strategies [Singh et al., 2019, Niu et al., 2021, Hu et al., 2024, Ding et al., 2024a].

A straightforward approach is to broadcast information of an agent to all others, forming a fully-connected communication graph. This leads to a quadratic increase ($O(N^2)$) in communication overhead, causing "information overload" that corrupts both scalability and performance [Sukhbaatar et al., 2016]. To address this, recent works have focused on learning a sparse communication topology. Some methods leverage attention (e.g., TarMAC, [Das et al., 2019]), gating mechanisms (e.g., ATOC, [Jiang and Lu, 2018]), or even Transformers (e.g., CommFormer, [Hu et al., 2024]) to dynamically prune connections. However, these methods, while achieving sparsity, are often heuristic and driven implicitly by the downstream task reward. They lack a principled, theoretical objective for what constitutes a "good" communication graph, often resulting in topologies that are sparse but structurally disordered (see Figure 1b).

*Corresponding author.

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

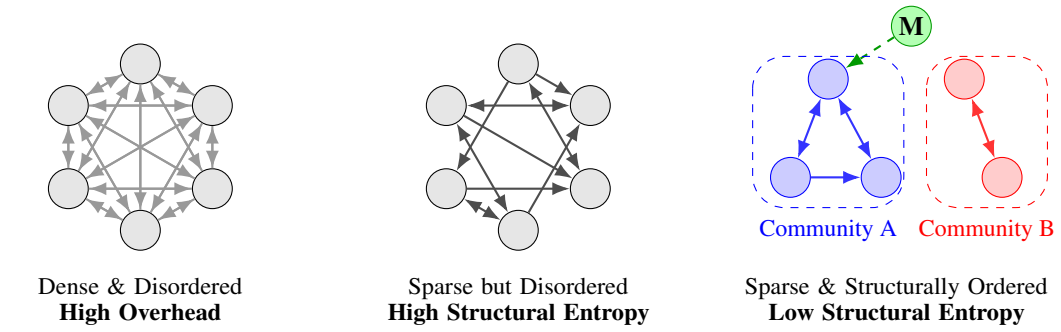


Figure 1: Conceptual comparison of communication topologies. (a) Full communication protocol is dense and suffers from high overhead. (b) Typical sparse communication methods prune edges but often result in a structurally disordered graph with high structural entropy. (c) SEOC explicitly optimizes for low structural entropy, leading to a sparse graph with clear, interpretable community structures (e.g., Community A and B) and targeted inter-community links (e.g., a Medivac agent 'M' communicating with specific agents in a community in SMAC scenario).

This leads to our core thesis: an efficient multi-agent communication network must be not only **sparse** but intrinsically **ordered**. We ground this idea in the principle of Structural Entropy (SE) from information theory [Su et al., 2025a], which measures the topological complexity of a network. Its minimization principle holds that natural systems evolve toward configurations of lower structural uncertainty, characterized by dense intra-community links and sparse inter-community connections.

We adopt SMAC [Samvelyan et al., 2019, Ellis et al., 2023] tasks to provide a clear, intuitive interpretation of this abstract principle. As illustrated by the SMAC scenario in Figure 1(c), guided by the objective of low SE, SEOC forms a sparse yet hierarchically organized communication graph. This graph exhibits clearly interpretable communities of agents performing cooperative sub-tasks (e.g., Community A and B), along with targeted inter-community links (An Medivac agent 'M' communicates only with specific low-health agents within a community without requiring full connectivity.) Such **ordered sparsity** corresponds precisely to low structural entropy.

This principle-driven design is reflected in our framework, which is motivated by a dual-level objective aimed at achieving both topological and semantic efficiency. At the **structural level** (*who to talk to*), we minimize SE to directly shape the communication topology. This compels agents to self-organize into communities with clear roles, thereby reducing structural complexity and preventing noisy, over-connected interactions. Here "noisy" refers to task-irrelevant information overload, i.e., redundant messages that hinder decision-making, rather than physical channel noise. At the **content level** (*what to say*), we complement this by employing a node information bottleneck to dynamically compress the messages themselves, ensuring that the information exchanged over the optimized topology is both concise and task-relevant.

Crucially, these two levels are jointly optimized: the sparse topology eliminates redundant pathways, while the content bottleneck enforces semantic focus. This synergy yields a lightweight yet expressive communication protocol, ensuring agents communicate with the right partners and exchange precisely the necessary information for coordination. This synergy stems from a unified design where structural entropy serves as the driving force to shape the communication topology, while message compression and graph aggregation are seamlessly integrated upon this discovered structure. This principled combination enables interpretable community emergence and efficient coordination without relying on task-specific heuristics.

The main contributions of this paper are: (i) We introduce the minimization of structural entropy as a principled, information-theoretic objective for learning communication topologies in MARL. (ii) We design a Structural Optimization Module featuring a differentiable SE proxy loss and a Content Optimization Module based on the Node Information Bottleneck, allowing for joint, end-to-end training. (iii) SEOC learns sparse and structurally-ordered communication graphs that are more efficient and interpretable.

2 Related Work

Attention-based Communication. To overcome the limitations of full communication, early advanced methods introduced dynamic selection mechanisms. TarMAC [Das et al., 2019] utilizes a soft attention mechanism to assign weights to messages, while ATOC [Jiang and Lu, 2018] and G2ANet [Liu et al., 2020] employ gating units to learn "hard" binary communication decisions, thereby enabling graph sparsification. This paradigm has recently evolved with the adoption of the Transformer in works like CommFormer [Hu et al., 2024], which uses a global self-attention mechanism to model all-pairs agent dependencies. While effective, both classic attention and modern Transformer-based approaches often come with high computational complexity ($O(N^2)$ in vanilla forms) and their attention maps lack an explicit structural prior, being guided implicitly by the downstream task reward. In contrast, SEOC provides a distinct yet complementary philosophy, using the information-theoretic principle of structural entropy, rather than attention weights to explicitly optimize the communication topology for structural order and sparsity.

GNN-based Communication. Modeling the multi-agent system as a graph and leveraging Graph Neural Networks (GNNs) [Wang et al., 2020, 2021] for message aggregation has become a dominant paradigm. The aforementioned methods, such as TarMAC and G2ANet, can be viewed through this graph lens. DGN [Jiang et al., 2020] and MAGIC [Niu et al., 2021] are representative methods in this paradigm, often coupling the learning of the communication graph with the GNN's message propagation process. Further structuring of GNN-based communication has led to hierarchical and sequential methods, such as LSC [Sheng et al., 2022], HAMA [Ryu et al., 2020], and SeqComm [Ding et al., 2024a]. For instance, SeqComm employs a multi-level scheme with asynchronous negotiation and launching phases, using priority-based ordering to determine when and what to communicate. While these methods effectively structure temporal dependencies and decision sequences, they often rely on predefined hierarchies or protocols. In contrast, SEOC directly optimizes the spatial communication topology (who communicates with whom) from a first-principles structural entropy objective. This makes SEOC orthogonal and complementary to these approaches, as it can be naturally integrated with temporal ordering mechanisms while removing the need for predefined structures.

Information-Theoretic Communication. Recently, information theory has provided a powerful lens for designing efficient multi-agent communication [Lin et al., 2023, Li et al., 2022]. Most relevant to our work is MAGI [Ding et al., 2024b], which applies the information bottleneck principle to enhance robustness in multi-agent communication. While MAGI focuses on compressing the information flow through a given graph to improve robustness, SEOC introduces a different objective: optimizing the topological order of the graph itself through structural entropy minimization. In other words, MAGI asks *"what to communicate given a structure"*, while SEOC asks *"what structure should emerge to facilitate efficient communication"*. The two are orthogonal and complementary: MAGI focuses on edge-level message content, while SEOC shapes connectivity patterns for low-entropy, modular organizations. This shift allows SEOC to directly address the structural complexity of multi-agent interaction, fostering interpretable community formation and scalable coordination, a dimension not captured by information compression alone.

Overall, SEOC provides a distinct alternative to existing methods by explicitly optimizing the topological order of communication graphs with structural entropy.

3 Preliminaries

3.1 Dec-POMDP

A cooperative multi-agent task can be modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [Oliehoek, 2012], defined by a tuple $\langle \mathcal{N}, S, A, P, R, \Omega, O \rangle$. Here, $\mathcal{N} = \{1, ..., N\}$ is the set of agents. At each timestep t, each agent $i \in \mathcal{N}$ receives a local observation $o_i \in \Omega$ based on observation function $O(s, i)$, and chooses an action $a_i \in A$. The environment transitions from state $s \in S$ to s' based on the joint action $\mathbf{a} \in A^N$ and the transition function $P(s'|s, \mathbf{a})$. All agents share a team reward $r = R(s, \mathbf{a})$. The goal is to learn a joint policy $\pi(\mathbf{a}|\mathbf{o})$ that maximizes the discounted cumulative reward $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$.

3.2 Structural Entropy

Structural entropy extends Shannon's information entropy to quantify the uncertainty embedded in a graph's structure [Zeng et al., 2024, Cao et al., 2024, Su et al., 2025b]. For a graph $G = (V, E)$ partitioned into communities $\mathcal{P} = \{C_1, ..., C_K\}$, the 2D structural entropy is defined as the minimum bits needed to describe a node's location, considering this hierarchy:

$$H_{\mathcal{P}}(G) = -\sum_{k=1}^{K} \frac{\text{Vol}(C_k)}{\text{Vol}(G)} \log_2 \left(\frac{\text{Vol}(C_k)}{\text{Vol}(G)} \right) - \sum_{k=1}^{K} \frac{\text{Vol}(C_k)}{\text{Vol}(G)} \sum_{i \in C_k} \frac{d_i}{\text{Vol}(C_k)} \log_2 \left(\frac{d_i}{\text{Vol}(C_k)} \right). \tag{1}$$

The first term is the inter-community entropy (uncertainty of community selection), and the second is the intra-community entropy (weighted average uncertainty of node selection within a community). The SE minimization principle posits that a natural network structure is one that minimizes this value. $\text{Vol}(C_k) = \sum_{i \in C_k} d_i$ is the volume of community C_k, and d_i is the degree of node i.

4 Methodology

SEOC is a general, plug-and-play communication protocol designed to be integrated into current MARL frameworks. Its core function is to enhance each agent's local representation by generating a sparse, information-theoretic communication graph and corresponding compressed messages. The communication protocol transforms an agent's local hidden state representation h_i into a communication-aware state h'_i. This protocol consists of three key components: a *Structural Optimization Module*, a *Content Optimization Module*, and a *GNN-based Aggregation Module*. These components are trained jointly with a downstream MARL algorithm. **The complete training procedure is summarized in Algorithm 1 provided in Appendix B.**

4.1 Overall Framework

Let the set of agents be $\mathcal{V} = \{1, \ldots, N\}$. At each timestep t, based on local observation o_i^t, each agent i possesses a local hidden state $h_i^t \in \mathbb{R}^d$ typically by an observation encoder such as a gated recurrent unit (GRU). SEOC protocol operates as follows: (i) The Structural Optimization Module takes $\mathbf{h}^t = \{h_1^t, \ldots, h_N^t\}$ and generates a sparse, differentiable adjacency matrix $\tilde{A}^t$ and a sparse, structurally-ordered communication graph; (ii) The Content Optimization Module encodes each h_i^t into an informative, compact message m_i^t; (iii) The messages $\{m_i^t\}$ are aggregated via a multi-layers Graph Neural Network (GNN) parameterized by the adjacency structure $\tilde{A}^t$, producing the final enhanced states $\tilde{\mathbf{h}}^t = \{\tilde{h}_1^t, \ldots, \tilde{h}_N^t\}$ for decision-making.

4.2 Structural Optimization Module

This module dynamically determines *who to communicate with* by generating a sparse, structurally-ordered communication graph. It uses a fully differentiable proxy to optimize the graph's SE.

4.2.1 Differentiable Graph Generation

First, we must generate a differentiable representation of the communication graph. Given the hidden states $\mathbf{h} = \{h_1, ..., h_N\}$ from an encoder (e.g., a GRU), we compute pairwise connection logits s_{ij} for all potential edges (i, j): $s_{ij} = f_{\theta_{\text{edge}}}([h_i, h_j])$, where $f_{\theta_{\text{edge}}}$ is a small MLP with parameters θ_{edge}, and $[\cdot, \cdot]$ denotes concatenation. These logits denote the unnormalized probability of an edge existing.

To obtain a discrete graph structure A while maintaining differentiability, we employ a binary Gumbel-Softmax trick. For each potential edge (i, j), we compute a logit s_{ij} for edge existence and sample a relaxed binary decision from a two-class Gumbel-Softmax distribution, with the "edge-off" logit fixed at 0 and the "edge-on" logit set to s_{ij}:

$$A_{ij} = \frac{\exp((s_{ij} + g_{ij,1})/\sigma)}{\exp((0 + g_{ij,0})/\sigma) + \exp((s_{ij} + g_{ij,1})/\sigma)}. \tag{2}$$

Here, $g_{ij,0}, g_{ij,1}$ are i.i.d. samples from a Gumbel(0,1) distribution, and σ is the temperature parameter. As $\sigma \to 0$ during training, A_{ij} converges to a binary value, yielding a discrete graph. We refer to this differentiable matrix as the soft adjacency matrix.

Although computing pairwise logits requires all agents' hidden states, this does not incur $O(N^2)$ communication. At each timestep, every agent broadcasts its hidden state once, totaling $O(N)$ low-dimensional communication. Each agent then locally computes all pairwise logits ($O(N^2)$ computation with zero communication) and independently samples its incoming edges via Gumbel-Softmax. Once the sparse neighbor set is determined, high-dimensional messages are transmitted only along active edges ($O(|E|)$ communication, where $|E| \ll N(N-1)$). Thus, $O(N^2)$ communication occurs only in fully connected protocols where every agent broadcasts high-dimensional messages to all others. The local $O(N^2)$ computation incurs no communication overhead, and the actual communication scales with the sparse active edges, not the total number of agents.

4.2.2 Differentiable Structural Entropy Proxy

The key innovation is a differentiable proxy for the SE of a graph, which favors graphs with well-organized community structures. To make SE a trainable objective, we define a differentiable proxy for each component in its formula.

We first need a differentiable mapping from an agent's state to a community assignment. We introduce a community assignment network f_{ϕ_C}, another small MLP, which maps each agent's hidden state h_i to a probability distribution over K latent communities:

$$\mathbf{c}_i = \text{Softmax}(f_{\phi_C}(h_i)) \in \Delta^K, \tag{3}$$

where $\mathbf{c}_i = [c_{i1}, ..., c_{iK}]$ is the soft community assignment vector for agent i, with $c_{ik} = p(\text{agent } i \in \text{community } k)$. With the soft adjacency matrix A and soft community assignments $\{\mathbf{c}_i\}$, we define differentiable proxies for graph-theoretic concepts. The soft degree of node i is the expected number of edges connected to it: $d_i^{\text{soft}} = \sum_{j \neq i} A_{ij}$. The soft volume of the G is the sum of all soft degrees:

$$\text{Vol}^{\text{soft}}(G) = \sum_{i=1}^{N} d_i^{\text{soft}} = \sum_{i=1}^{N} \sum_{j \neq i} A_{ij}. \tag{4}$$

The soft volume of a community k, $\text{Vol}^{\text{soft}}(C_k)$, is the expected sum of degrees of all nodes within that community, weighted by their probability of belonging to it:

$$\text{Vol}^{\text{soft}}(C_k) = \sum_{i=1}^{N} c_{i,k} \cdot d_i^{\text{soft}}. \tag{5}$$

The soft degree of node i within community k is defined as $d_{i|k}^{\text{soft}} = c_{i,k} \cdot d_i^{\text{soft}}$, where $c_{i,k}$ is the k-th component of $\mathbf{c}_i$. The two-dimensional structural entropy of a graph G is defined as $H^{(2)}(G) = -\sum_{k=1}^{K} \frac{\text{Vol}(C_k)}{\text{Vol}(G)} \sum_{i=1}^{N} \frac{d_{i|k}}{\text{Vol}(C_k)} \log_2 \frac{d_i}{\text{Vol}(C_k)}$. By substituting these differentiable quantities into the two components of the SE formula (Eq. 1), we derive final SE proxy loss. The differentiable inter-community entropy loss $\mathcal{L}_{\text{inter}}$ measures the uncertainty in selecting a community:

$$\mathcal{L}_{\text{inter}} = -\sum_{k=1}^{K} \frac{\text{Vol}^{\text{soft}}(C_k)}{\text{Vol}^{\text{soft}}(G)} \log_2 \left(\frac{\text{Vol}^{\text{soft}}(C_k)}{\text{Vol}^{\text{soft}}(G)} + \epsilon \right). \tag{6}$$

The differentiable intra-community entropy loss $\mathcal{L}_{\text{intra}}$ measures the average uncertainty of selecting a node within its assigned community:

$$\mathcal{L}_{\text{intra}} = -\sum_{k=1}^{K} \frac{\text{Vol}^{\text{soft}}(C_k)}{\text{Vol}^{\text{soft}}(G)} \sum_{i=1}^{N} \frac{d_{i|k}^{\text{soft}}}{\text{Vol}^{\text{soft}}(C_k)} \log_2 \left(\frac{d_{i|k}^{\text{soft}}}{\text{Vol}^{\text{soft}}(C_k)} + \epsilon \right). \tag{7}$$

Here, $d_{i|k}^{\text{soft}} = c_{i,k} \cdot d_i^{\text{soft}}$ is the soft degree of node i within community k, and ϵ is a small constant for numerical stability. The final SE loss is the weighted sum of these two components:

$$\mathcal{L}_{\text{SE}} = \lambda_{\text{inter}} \cdot \mathcal{L}_{\text{inter}} + \lambda_{\text{intra}} \cdot \mathcal{L}_{\text{intra}}. \tag{8}$$

Symmetrization for SE. The SE loss $\mathcal{L}_{\text{SE}}$ is computed using the symmetrized adjacency matrix $\tilde{A} = (A + A^\top)/2$. This ensures consistency with the undirected graph formulation in Eq. (1), thereby encouraging stable community structures, while the original directed graph A is retained for flexible, state-dependent message passing during forward execution in multi-agent communication.

By minimizing $\mathcal{L}_{\text{SE}}$ via gradient descent, we explicitly push the communication graph A and the community assignments $\{\mathbf{c}_i\}$ towards a configuration of low structural uncertainty, thereby encouraging the emergence of sparse, modular communication patterns.

4.3 Content Optimization Module

This module addresses *what to communicate* by learning to compress an agent's local state into an informative, compact message. We adopt the node information bottleneck (XIB) principle [Wu et al., 2020], which aims to learn a stochastic encoder $p_\psi(m_i|h_i)$ that maximizes the mutual information between the message m_i and a task-relevant target variable Y_i, while compressing the information about the raw input h_i. In our MARL setting, we set Y_i to be the global state value $V(s)$ estimated by centralized critic, thereby aligning the communication objective with the team's reward signal.

$$\max_{p_\psi(m_i|h_i)} I(m_i; V(s)) - \beta \cdot I(m_i; h_i), \tag{9}$$

where $\beta > 0$ controls the trade-off. Using the variational inference framework, we introduce a variational prior $r(m_i)$ (e.g., a standard Gaussian $\mathcal{N}(0, I)$) and a variational decoder $q_\xi(V(s)|m_i)$. This yields the following evidence lower bound, which we minimize as our content loss:

$$\mathcal{L}_{\text{XIB}} = \beta \cdot \mathbb{E}_{p(h_i)} \left[D_{\text{KL}} \left(p_\psi(m_i|h_i) \| r(m_i) \right) \right] - \mathbb{E}_{p(h_i, m_i)} \left[\log_2 q_\xi(V(s)|m_i) \right]. \tag{10}$$

The target $V(s)$ in Eq. (10) is detached from the centralized critic $V_\phi(s)$ to prevent circular optimization. The encoder p_ψ is a Gaussian $\mathcal{N}(\mu_\psi(h_i), \sigma^2_\psi(h_i)I)$ with reparameterized sampling. The decoder q_ξ regresses this detached target to predict $V(s)$, stabilizing training. The sampled compressed messages $\{m_i\}$ are then passed to the GNN aggregation module (Section 4.4) as part of the initial node representations. We adopt $V(s)$ for its stability and task relevance, deferring action-conditional signals to future work.

4.4 GNN-based Aggregation Module

Given the soft adjacency matrix A and the compressed messages $\{m_i\}$ sampled from the content optimization module (Section 4.3), we employ an L-layer Graph Neural Network (GNN) to perform iterative message passing. The compressed message m_i serves as the initial representation of agent i; i.e., we set $h_i^{(0)} = m_i$. For each layer $l = 1, \ldots, L$, we compute: $\hat{m}_i^{(l)} = \eta^{(l)}\big(h_i^{(l-1)}\big)$, $\bar{m}_i^{(l)} = \sum_{j=1}^{N} A_{ij} \cdot \hat{m}_j^{(l)}$, $h_i^{(l)} = \phi^{(l)}\big(h_i^{(l-1)} \| \hat{m}_i^{(l)} \| \bar{m}_i^{(l)}\big)$, where $\eta^{(l)}$ and $\phi^{(l)}$ are learnable MLPs specific to layer l, and $\|$ denotes vector concatenation. The final enhanced state for agent i is $h'_i = h_i^{(L)}$. This multi-layer architecture captures higher-order interactions and propagates information beyond immediate neighbors, fostering more sophisticated coordination.

4.5 Integration and Joint Optimization with MAPPO

We ground our method in the state-of-the-art MAPPO framework [Yu et al., 2022]. The enhanced state feature representations $\{h'_i\}$ produced by SEOC are fed into both the actor and critic networks.

4.5.1 Adaptation for MAPPO

Within MAPPO, we instantiate the target variable Y_i in Eq. (9) as the global state value $V_\phi(s_t)$ estimated by the centralized critic. This aligns the message compression objective with the global team reward signal, encouraging agents to communicate information that is valuable for value prediction. The networks are utilized as follows:

Decentralized Actor: Each agent's policy π_{θ_i} uses the enhanced local state h'_i to sample actions: $a_i^t \sim \pi_{\theta_i}(\cdot|h'_i)$. The Actor Loss is the clipped surrogate objective from PPO, computed using Generalized Advantage Estimation:

$$\mathcal{L}_{\text{p}}(\theta_i) = \mathbb{E}_t \left[\min \left(r_t(\theta_i)\hat{A}_t,\ \text{clip}(r_t(\theta_i),\ 1-\epsilon,\ 1+\epsilon) \cdot \hat{A}_t \right) \right], \tag{11}$$

where $r_t(\theta_i) = \frac{\pi_{\theta_i}(a_t|h'_t)}{\pi_{\theta_{i,old}}(a_t|h'_t)}$ is the probability ratio, $\hat{A}_t$ denotes the estimated advantage function.

Centralized Critic: The critic V_ϕ receives the global state s_t (often a concatenation or mean pooling of all $\{h'_i\}$) to estimate the state value $V_\phi(s_t)$. The Critic Loss is the standard squared error loss for the value function, where G_t is the Monte-Carlo return:

$$\mathcal{L}_{\text{v}}(\phi) = \mathbb{E}_t \left[(V_\phi(s_t) - G_t)^2 \right]. \tag{12}$$

The final MAPPO loss function to be minimized is:

$$\mathcal{L}_{\text{ppo}} = -\sum_{i=1}^{N} \mathcal{L}_p(\theta_i) + c_1 \cdot \mathcal{L}_v(\phi) - c_2 \cdot S[\pi], \tag{13}$$

where $S[\pi]$ is an entropy bonus to encourage exploration, c_1, c_2 are PPO coefficients.

4.5.2 Joint Optimization with PPO Losses

The entire system, including the SEOC parameters and the MAPPO actor-critic networks, is trained end-to-end. The total loss function is a composition of the standard PPO losses and our communication-regularizing losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ppo}}(\theta, \phi) + \alpha_{\text{SE}} \cdot \mathcal{L}_{\text{SE}} + \alpha_{\text{XIB}} \cdot \mathcal{L}_{\text{XIB}}, \tag{14}$$

where $\alpha_{\text{SE}}, \alpha_{\text{XIB}} > 0$ are our hyperparameters balancing task completion with communication efficiency and structural order. This joint optimization enables SEOC to learn a sparse, structured communication graph and informative messages that directly contribute to maximizing the team's expected return. **The hyperparameter choices are detailed in Appendix D.2.**

As shown in the hyperparameter sensitivity analysis (Appendix D.2), SEOC's performance remains stable across a broad range of α_{SE} and α_{XIB} values, indicating that the optimization landscape is sufficiently well-behaved to eliminate the need for adaptive reweighting. Moreover, the joint convergence analysis in Appendix A.1 demonstrates that constant coefficients are adequate for reaching a stationary point, and introducing dynamically adjusted weights would add unnecessary complexity without empirical benefit.

Although the SE loss is purely topology-oriented, semantic alignment emerges from its joint optimization with PPO and the information bottleneck. The community assignment network takes as input the task-driven latent states h_i updated by PPO gradients, grounding communities in task-relevant representations. The XIB loss further aligns messages with the global state value $V(s)$, ensuring that what is communicated is task-relevant. The SE loss acts as a structural prior, organizing these semantically rich representations into sparse, modular communities. Optimizing each objective in isolation would fail: without PPO, communities would lack semantic meaning; without XIB, messages would carry task-irrelevant information; without SE, the topology would lack interpretable structure.

5 Experiments

In this section, we conduct a comprehensive evaluation of SEOC on the representative multi-agent benchmark SMACv2 [Ellis et al., 2023] and Traffic Signal Control task based on SUMO [Lopez et al., 2018]. **For comprehensive details regarding supplementary experiments, mathematical proofs, complete environment descriptions, baseline descriptions, and hyperparameter settings, please refer to the Appendix.** We design experiments to answer the following questions:

Q1 (Task performance): How does SEOC's performance compared with baseline methods?

Q2 (Communication Efficiency): How does SEOC's efficiency compared with baseline methods?

Q3 (Interpretability): To what extent are SEOC's communication topologies interpretable?

Q4 (Ablation Study): What are the contributions of SEOC's components to overall performance?

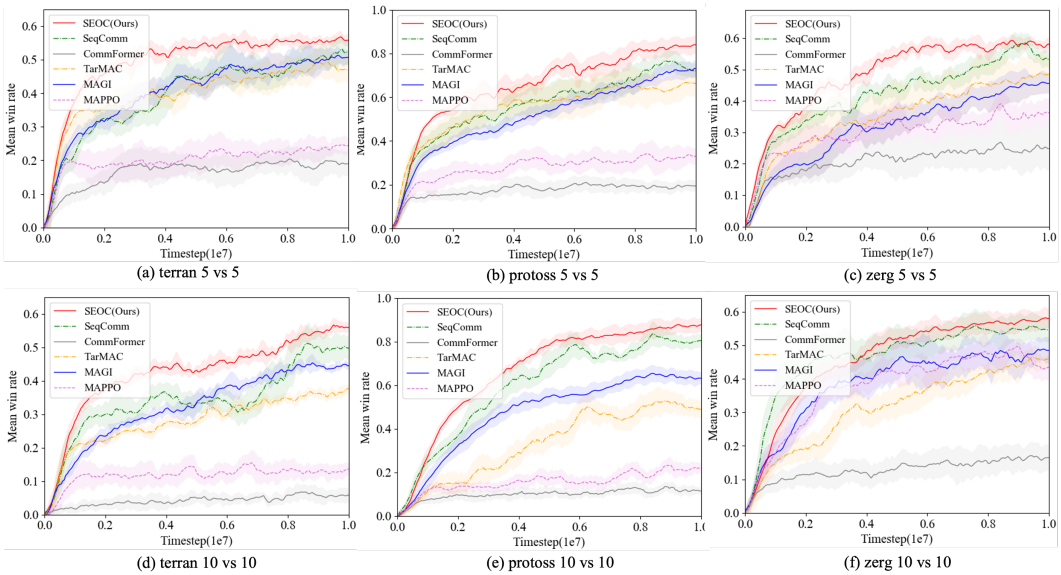
Q5 (Scalability): What is SEOC's performance trend as the number of agents increases?

SEOC is implemented based on MAPPO [Yu et al., 2022]. Baselines include SeqComm [Ding et al., 2024a], MAGI [Ding et al., 2024b], CommFormer [Hu et al., 2024], TarMAC [Das et al., 2019], and a non-communicating baseline MAPPO [Yu et al., 2022]. **A theoretical computational complexity analysis is provided in Appendix C.1.**

SMACv2 is more challenging than SMAC due to higher stochasticity (randomized compositions and starting positions) and larger scales, which cause a combinatorial explosion of coordination patterns. To further increase difficulty, we reduced each agent's sight range from 9 to 3 and disabled perception of allied units, making purely independent, local observation infeasible and thus necessitating structured multi-agent communication and cooperation. Experiments were run on RTX 4090 GPUs.

5.1 Task Performance (Q1).

As shown in Figure 2, across multiple scenarios in SMACv2, SEOC demonstrates a consistent and significant advantage over baselines in win-rate. Under severely limited observation, it enables rapid self-organization and situational awareness reconstruction, leading to faster win-rate growth in complex scenarios where baselines struggle with fine-grained heterogeneous coordination. By delivering higher-quality, decision-focused information, SEOC produces more precise and robust coordinated actions. SEOC's performance advantage stems from its synergistic optimization of communication structure and content. The framework jointly learns dynamic, structured communities through structural entropy minimization and extracts value-aligned messages via a node IB. This integration of topological sparsity and semantic compression enables efficient, adaptive coordination under partial observability. **Performance comparison on traffic signal control task and standard SMACv2 (sight range 9, ally states observable) are provided in Appendix E.1.** Additionally, we compare with the hierarchical LSC baseline and instantiate TarMAC/CommFormer on the same MAPPO backbone; SEOC consistently outperforms these variants, confirming the benefit of the SE objective (Appendix E.1).



(a) terran 5 vs 5 (b) protoss 5 vs 5 (c) zerg 5 vs 5

(d) terran 10 vs 10 (e) protoss 10 vs 10 (f) zerg 10 vs 10

Figure 2: Learning curves of SEOC and baseline methods in different SMACv2 scenarios. Shaded regions around each curve reflect standard errors derived from 5 random seeds.

5.2 Communication Efficiency (Q2).

To evaluate the communication efficiency, we present the test win rate against the average number of active edges on terran 10 vs 10. At test time, we use the hard, discrete graph obtained by applying $\arg\max$ to the Gumbel-Softmax outputs (Eq. 2) for message passing. This graph is also used to count active edges. All baselines obtain discrete adjacency matrices by thresholding their soft attention weights at 0.5 using the union over attention heads, from which active edges are counted. As shown in Figure 3 left, SEOC achieves the highest win rate with the fewest active edges per timestep.

Instead of maintaining persistent or dense communication links, SEOC continuously optimizes its interaction graph by grouping agents into functional communities. Connections are formed and pruned based on real-time necessity and role compatibility, such as linking a Medivac to a wounded unit or focusing fire on a high-priority target. This process ensures that each active edge is purposeful and carries high-value information, minimizing redundancy and noise.

More fundamentally, this adaptive structure enables the system to scale effectively. Due to structural entropy regularization, the number of active edges grows sub-quadratically in our scalability tests: approximately 6, 22, and 50 for $N = 5, 10, 20$, respectively, far below the fully connected upper bound $N(N-1) = 20, 90, 380$. This sub-quadratic scaling is a direct consequence of the SE objective,

which penalizes redundant connections and encourages community structures that naturally limit inter-group edges. Therefore, SEOC's ability to deliver strong performance with low communication cost validates its design principle and highlights its suitability for real-world applications where bandwidth or processing power may be constrained, such as in traffic signal control. **Appendix E.2 provides a more detailed quantification of communication bandwidth and training time efficiency.**

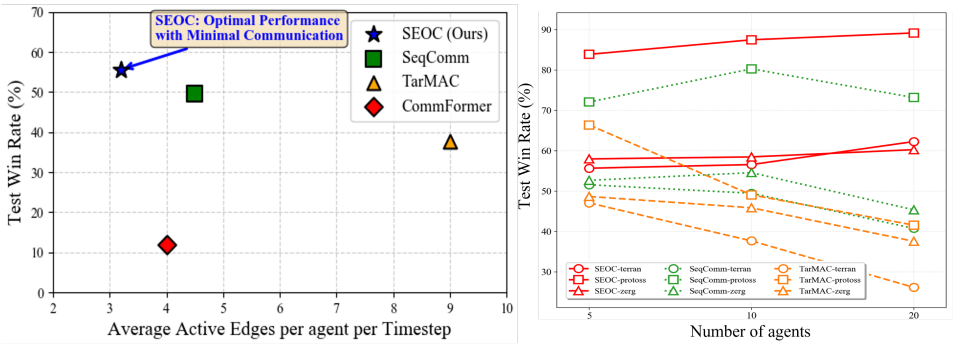


Figure 3: Left: Comparative analysis of communication efficiency. Right: Win rate of SEOC and baselines across different numbers of agents in terran, protoss, and zerg scenarios.

5.3 Interpretability (Q3).

SEOC not only achieves high task performance but also produces interpretable communication topologies that naturally align with task semantics. As illustrated in Figure 4, the learned graphs exhibit clear community structures and intuitive, role-aware connectivity, providing transparent insights into the underlying coordination mechanisms. A representative example occurs by timestep $t = 55$ in Figure 4, where a surviving Medivac selectively strengthens its connections to wounded units, while combat agents autonomously form tightly clustered squads for focused attacks. This state-driven, context-aware linking shows how SEOC prioritizes functionally critical interactions like healing and coordinated assault without predefined rules or uniform communication templates. These emergent patterns align with human-interpretable teamwork concepts and confirm the semantic consistency of the learned topology.

Besides, SEOC demonstrates adaptive reorganization in response to environmental changes. As agents are eliminated, the graph dynamically consolidates connectivity among survivors (Figure 4, $t = 10$ to $t = 55$), removing lost links and strengthening relevant ones to preserve coordination. The learned graphs also reveal hierarchical organization: dense bidirectional connections within groups enable local coordination, while sparse inter-group bridges enable strategic signaling. This human-like structure clarifies decision logic and bridges black-box models with interpretable MAS, supporting more reliable real-world deployment.

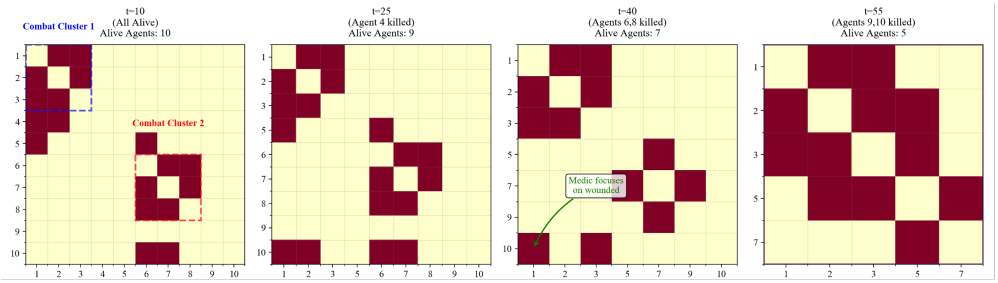


Figure 4: Visualization of learned communication graphs. Communication matrix dynamically adjusts size as agents are eliminated. Original 10 agents with random types (Marine, Marauder, Medivac). Agent numbers are 1-indexed, killed agents removed from matrix.

Quantitatively, SEOC achieves a Modularity of $Q = 0.63 \pm 0.05$ and a Normalized Mutual Information (NMI) of 0.52 ± 0.07 against ground-truth unit types, compared to $0.31/0.14$ for SEOC w/o SE and $0.28/0.11$ for TarMAC (Appendix E.3.2). These metrics confirm that the observed

community structures are not artifacts of the visualization choice, but reflect genuine, task-grounded topological organization. Besides, clarification on directed vs undirected optimization is provided in Appendix E.3.1.

5.4 Ablation Study (Q4).

We ablate SEOC's components (SE loss, IB loss, GNN aggregation) on protoss 10 vs 10, evaluating variants without SE, without XIB, and without GNN (mean pooling). Results in Table 1 show that removing the SE loss yields a denser, less structured graph and lower win rate, confirming its critical role in learning sparse, interpretable topologies. Ablating the information bottleneck also causes a noticeable performance drop, indicating that compressing task-relevant information while filtering noise benefits coordination. Replacing the multi-layer GNN with mean pooling leads to the most substantial decline, highlighting the necessity of iterative message passing for capturing higher-order dependencies. The full SEOC achieves the strongest results, demonstrating that synergy among structured topology, semantic compression, and powerful aggregation is key to its effectiveness.

Notably, SEOC w/o SE is a controlled comparison: the MAPPO backbone and the GNN aggregator are fixed, and the only change is the graph construction method. The gap between 87.5% and 74.2% therefore isolates the effect of the SE objective itself. Across ablations on protoss 10 vs 10, lower final SE correlates with fewer active edges and higher win rate: SEOC Full achieves 2.1 bits, 22.3 edges, and 87.5%, while SEOC w/o SE yields 3.2 bits, 38.6 edges, and 74.2% (Appendix E.4). This supports the causal chain that SE minimization prunes redundant cross-community edges, reduces information overload, and improves coordination. **More ablation studies on SE and topological level are provided in Appendix E.4.**

Table 1: Ablation study of SEOC components on protoss 10 vs 10.

Variant	Win Rate (%)	Avg. Active Edges
SEOC (Full)	$\mathbf{87.5 \pm 2.8}$	22.3 ± 1.7
SEOC w/o SE	74.2 ± 3.5	38.6 ± 2.4
SEOC w/o XIB	77.8 ± 3.1	24.1 ± 1.9
SEOC w/o GNN	69.4 ± 4.2	20.8 ± 1.5

5.5 Scalability (Q5).

We further evaluate the scalability of SEOC by comparing it against SeqComm and TarMAC, across varying numbers of cooperative agents in SMACv2 scenarios (terran, protoss and zerg). As shown in Figure 3 right, SEOC demonstrates consistently stronger and more stable scalability as the number of agents increases from 5 to 20 (5 vs 5, 10 vs 10, 20 vs 20). In all scenarios, SEOC not only sustains high initial performance but in several cases shows slight improvement with larger teams. These outcomes suggest that SEOC's structured communication mechanism, effectively supports scalable coordination in growing teams by preserving relevant inter-agent dependencies and filtering redundant interactions. SEOC is also robust to the number of communities: for $K \in [3, 6]$, the win-rate variation is below 3% across scenarios, and even at the extremes $K = 1$ and $K = 8$, SEOC maintains win rates above 79% on protoss 10 vs 10 (Appendix D.2). **To further evaluate the scalability of SEOC, we conduct experiments in a real-world urban road network, results are provided in Appendix E.5.**

6 Conclusion

This paper proposes a MARL framework that enhances multi-agent coordination by directly optimizing the communication topology through structural entropy minimization. SEOC jointly learns a sparse, community-structured interaction graph and compresses messages via a node information bottleneck, achieving efficient, interpretable, and scalable communication. Evaluated on the SMACv2 benchmark and traffic signal control task, SEOC outperforms state-of-the-art baselines. Future work includes applying SEOC to domains like autonomous vehicles and distributed robotics, integrating it with adversarial or cross-modal learning frameworks, and deepening the theoretical understanding of structural entropy to support human-agent teamwork and open-world applications.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2026YFE0101100, 2025KJHZ036), the National Natural Science Foundation of China (No. 92367202, 62606283, 62531013, 62272276 and 62432006), Shandong Provincial Natural Science Foundation (No. ZR2024JQ001, ZR2025LZH004 and ZR2024QF204).

References

Y. Cao, H. Peng, Z. Yu, and P. S. Yu. Hierarchical and incremental structural entropy minimization for unsupervised social event detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pages 8255–8264, 2024.

A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning (ICML)*, pages 1538–1546, 2019.

G. Ding, Z. Liu, Z. Fang, K. Su, L. Zhu, and Z. Lu. Multi-agent coordination via multi-level communication. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:118513–118539, 2024a.

S. Ding, W. Du, L. Ding, J. Zhang, L. Guo, and B. An. Robust multi-agent communication with graph information bottleneck optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(5):3096–3107, 2024b.

B. Ellis, J. Cook, S. Moalla, M. Samvelyan, M. Sun, A. Mahajan, J. Foerster, and S. Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:37567–37593, 2023.

S. Hu, L. Shen, Y. Zhang, and D. Tao. Learning multi-agent communication from graph modeling perspective. In *International Conference on Learning Representations (ICLR)*, 2024.

J. Jiang and Z. Lu. Learning attentional communication for multi-agent cooperation. *Advances in Neural Information Processing Systems (NeurIPS)*, 31, 2018.

J. Jiang, C. Dun, and T. Huang. Graph convolutional reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2020.

K. Jiang, X. Cai, Z. Cui, A. Li, Y. Ren, H. Yu, H. Yang, D. Fu, L. Wen, and P. Cai. Koma: Knowledge-driven multi-agent framework for autonomous driving with large language models. *IEEE Transactions on Intelligent Vehicles (TIV)*, 2024.

P. Li, H. Tang, T. Yang, X. Hao, T. Sang, Y. Zheng, J. Hao, M. E. Taylor, W. Tao, and Z. Wang. Pmic: Improving multi-agent reinforcement learning with progressive mutual information collaboration. In *International Conference on Machine Learning (ICML)*, pages 12979–12997. PMLR, 2022.

Y. Lin, W. Li, H. Zha, and B. Wang. Information design in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:25584–25597, 2023.

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

P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner. Microscopic traffic simulation using sumo. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 2575–2582. IEEE, 2018.

C. Ma, A. Li, Y. Du, H. Dong, and Y. Yang. Efficient and scalable reinforcement learning for large-scale network control. *Nature Machine Intelligence (NMI)*, 6(9):1006–1020, 2024.

Y. Niu, R. R. Paleja, and M. C. Gombolay. Multi-agent graph-attention communication and teaming. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2021.

F. A. Oliehoek. Decentralized pomdps. *Reinforcement Learning*, 2012.

H. Ryu, H. Shin, and J. Park. Multi-agent actor-critic with hierarchical graph attention network. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 7236–7243, 2020.

M. Samvelyan, T. Rashid, C. Schroeder de Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, and S. Whiteson. The starcraft multi-agent challenge. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 2186–2188, 2019.

J. Sheng, X. Wang, B. Jin, J. Yan, W. Li, T.-H. Chang, J. Wang, and H. Zha. Learning structured communication for multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems (AAMAS)*, 36(2):50, 2022.

A. Singh, T. Jain, and S. Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *International Conference on Learning Representations (ICLR)*, 2019.

D. Su, H. Peng, Y. Pan, and A. Li. A survey of structural entropy: theory, methods, and applications. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 10660–10668, 2025a.

D. Su, H. Peng, G. Zeng, P. Li, A. Li, and Y. Pan. Emergence of cooperation in multi-agent reinforcement learning via coalition labeling and structural entropy. In *SIAM International Conference on Data Mining (SDM)*, pages 507–515. SIAM, 2025b.

S. Sukhbaatar, A. Szlam, and R. Fergus. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

K. Wang, Z. Shen, Z. Lei, X. Liu, and T. Zhang. Towards multi-agent reinforcement learning based traffic signal control through spatio-temporal hypergraphs. *IEEE Transactions on Mobile Computing (TMC)*, 2025.

X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei. Am-gcn: Adaptive multi-channel graph convolutional networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1243–1253, 2020.

X. Wang, N. Liu, H. Han, and C. Shi. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1726–1736, 2021.

T. Wu, H. Ren, P. Li, and J. Leskovec. Graph information bottleneck. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:20437–20448, 2020.

Y. Xu, Y. Shi, X. Tong, S. Chen, and Y. Ge. A multi-agent reinforcement learning based control method for cavs in a mixed platoon. *IEEE Transactions on Vehicular Technology (TVT)*, 73(11): 16160–16172, 2024.

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

X. Zeng, H. Peng, and A. Li. Effective exploration based on the structural information principles. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:95291–95325, 2024.

A Proof

In this section, we provide a convergence analysis of the total loss function used in SEOC, focusing on the interplay between the PPO objective and the two regularization terms: the structural entropy loss $\mathcal{L}_{\text{SE}}$ and the information bottleneck loss $\mathcal{L}_{\text{XIB}}$. We establish basic properties (boundedness, monotonicity), prove convergence of the total loss to a stationary point under standard stochastic optimization assumptions, and discuss the qualitative roles of the two regularizers in optimizing the communication topology and message representations.

A.1 Convergence Analysis of SEOC's Dual Regularization

We analyze how the two regularization terms, namely structural entropy loss $\mathcal{L}_{\text{SE}}$ and information bottleneck loss $\mathcal{L}_{\text{XIB}}$, interact with the PPO objective during optimization. Our analysis focuses on their **boundedness, monotonicity, and effect on convergence**.

A.1.1 Basic Properties of the Two Regularizers

Property 1 (Boundedness of $\mathcal{L}_{\text{SE}}$). For any soft adjacency matrix $A \in [0,1]^{N\times N}$ and soft community assignments $\mathbf{c}_i \in \Delta^K$, the loss $\mathcal{L}_{\text{SE}}$ satisfies $\mathcal{L}_{\text{SE}} \le \lambda_{\text{inter}} \log K + \lambda_{\text{intra}} \log N$, and is empirically non-negative in our experiments.

Proof. $\mathcal{L}_{\text{inter}}$ is the entropy of a K-categorical distribution, hence $\le \log K$. For $\mathcal{L}_{\text{intra}}$, each term is $-\frac{d^{\text{soft}}_{i|k}}{\text{Vol}^{\text{soft}}(C_k)} \log\big(\frac{d^{\text{soft}}_{i}}{\text{Vol}^{\text{soft}}(C_k)} + \epsilon\big)$. Since $\frac{d^{\text{soft}}_{i|k}}{\text{Vol}^{\text{soft}}(C_k)} \le 1$ and the logarithm argument is at most N (ratio of a degree to the smallest possible volume), the term is bounded by $-\log(1/N) = \log N$. Weighted summation preserves the bound. The lower bound follows from the finite ranges; empirically $\mathcal{L}_{\text{SE}} \ge 0$ in all runs.

Property 2 (Monotonicity of $\mathcal{L}_{\text{XIB}}$ under decoder improvement). For fixed encoder p_ψ, if the variational decoder q_ξ improves its estimate of $V(s)$ (i.e., $\mathbb{E}[\log_2 q_{\xi^{(t+1)}}(V(s)|m_i)] \ge \mathbb{E}[\log_2 q_{\xi^{(t)}}(V(s)|m_i)]$), then $\mathcal{L}^{(t+1)}_{\text{XIB}} \le \mathcal{L}^{(t)}_{\text{XIB}}$.

Proof. $\mathcal{L}_{\text{XIB}} = \text{KL}(p_\psi(m_i|h_i)\|r(m_i)) - \beta\mathbb{E}[\log_2 q_\xi(V(s)|m_i)]$. The KL term does not depend on ξ. Thus, improving the decoder only (weakly) decreases $\mathcal{L}_{\text{XIB}}$. □

This implies that $\mathcal{L}_{\text{XIB}}$ naturally decreases as the XIB decoder q_ξ becomes more accurate at predicting $V(s)$, requiring no manual annealing.

A.1.2 Convergence of the Total Loss

The total loss is: $\mathcal{L}_{\text{total}}(\Theta) = \mathcal{L}_{\text{PPO}}(\theta,\phi) + \alpha_{\text{SE}}\mathcal{L}_{\text{SE}}(\theta_{\text{edge}},\phi_C) + \alpha_{\text{XIB}}\mathcal{L}_{\text{XIB}}(\psi,\xi)$. We adopt standard stochastic optimization assumptions: unbiased gradient estimates, bounded variance, step sizes satisfying $\sum \eta_t = \infty,\ \sum \eta_t^2 < \infty$, and bounded iterates (satisfied due to PPO's clipping mechanism and bounded losses).

Lemma 1 (Descent in expectation). Under the above assumptions, for sufficiently small η_t:

$$\mathbb{E}[\mathcal{L}_{\text{total}}(\Theta_{t+1}) \mid \mathcal{F}_t] \le \mathcal{L}_{\text{total}}(\Theta_t) - \eta_t\|\nabla\mathcal{L}_{\text{total}}(\Theta_t)\|^2 + \frac{L\eta_t^2}{2}\big(\|\nabla\mathcal{L}_{\text{total}}(\Theta_t)\|^2 + \sigma^2\big),$$

where L is the Lipschitz constant of $\nabla\mathcal{L}_{\text{total}}$. Since $\sum \eta_t^2 < \infty$ and gradients are bounded, the quadratic term is summable.

Proof. Standard Taylor expansion and conditional expectation, omitted for brevity. □

Theorem 1 (Convergence to a stationary point). Let $\mathcal{L}_{\text{total}}$ be bounded below. Then:

$$\liminf_{t\to\infty} \mathbb{E}[\|\nabla\mathcal{L}_{\text{total}}(\Theta_t)\|^2] = 0.$$

Proof. Take expectation of Lemma 1 and telescope:

$$\sum_{t=1}^{T} \eta_t \mathbb{E}[\|\nabla\mathcal{L}_{\text{total}}(\Theta_t)\|^2] \le \mathbb{E}[\mathcal{L}_{\text{total}}(\Theta_1)] - \mathbb{E}[\mathcal{L}_{\text{total}}(\Theta_{T+1})] + \frac{L\sigma^2}{2}\sum_{t=1}^{T}\eta_t^2.$$

Boundedness below and $\sum \eta_t^2 < \infty$ imply the RHS converges as $T \to \infty$. Since $\sum \eta_t = \infty$, we must have $\liminf_{t\to\infty} \mathbb{E}[\|\nabla\mathcal{L}_{\text{total}}(\Theta_t)\|^2] = 0$. □

A.1.3 Role of the Two Regularizers

The convergence analysis above treats the total loss as a general differentiable function. Here we briefly discuss how each regularizer may influence optimization beyond the standard guarantees.

Structural entropy loss $\mathcal{L}_{\text{SE}}$. Minimizing $\mathcal{L}_{\text{SE}}$ encourages the communication graph to form a clear community structure. By Cheeger's inequality, such a structure has a *small* spectral gap, not a large one (the original version contained a typo). This small spectral gap indicates the presence of bottlenecks between communities, which can be beneficial for localizing GNN message passing.

Information bottleneck loss $\mathcal{L}_{\text{XIB}}$. By compressing the raw observation h_i into a minimal sufficient message m_i, $\mathcal{L}_{\text{XIB}}$ reduces the variance of the downstream policy gradient. Specifically, the mutual information term $I(m_i; h_i)$ penalizes encoding of observation noise, leading to a more concentrated gradient estimate. While not altering the convergence guarantee, this variance reduction can improve sample efficiency and final performance, as observed empirically in Section 5.

We note that a formal characterization of these effects, such as bounding the reduction in gradient variance or the improvement in GNN mixing time, is an interesting direction for future work.

B Algorithm Description

Algorithm 1 outlines the iterative training of SEOC with MAPPO. Each cycle alternates between trajectory collection and parameter updates. During collection, SEOC generates a sparse, differentiable communication graph via Gumbel-Softmax sampling, while agents compress observations into messages via a variational encoder. A graph network aggregates these messages to form enhanced representations for decision-making. The update phase optimizes a combined loss and all parameters are jointly updated.

Algorithm 1 Training SEOC with MAPPO

Initialize actor networks $\{\pi_{\theta_i}\}$, critic network V_ϕ, and SEOC-Net parameters $(\theta_{\text{edge}}, \phi_C, \psi, \xi)$.
Initialize replay buffer $\mathcal{D}$.
for each training iteration **do**
    Initialize hidden states $\{h_i\}$ for all agents.
    for $t = 1$ to T **do**
        SEOC Communication
        For each pair (i, j), compute logits $s_{ij} = f_{\theta_{\text{edge}}}([h_i, h_j])$.
        Sample soft adjacency matrix A_t using Gumbel-Softmax.
        For each agent i, sample message $m_i \sim p_\psi(m_i|h_i)$.
        Set initial GNN representations as $[h_i; m_i]$
        Aggregate via GNN using A_t to obtain enhanced states $\{h'_i\}$
        MAPPO Action Selection
        For each agent i, select action $a_i \sim \pi_{\theta_i}(a_i|h'_i)$.
        Execute joint action, observe reward r and next observations.
        Store transition in trajectory buffer.
        Update hidden states $\{h_i\}$ with encoders.
    end for
    Add trajectory to replay buffer $\mathcal{D}$.
    for each PPO update epoch **do**
        Sample a batch of trajectories from $\mathcal{D}$.
        Compute advantages $\hat{A}_t$ and returns G_t.
        Compute policy loss $\mathcal{L}_{policy}$ and value loss $\mathcal{L}_{value}$.
        Compute structural entropy loss $\mathcal{L}_{SE}$.
        Compute content bottleneck loss $\mathcal{L}_{XIB}$.
        Compute total loss $\mathcal{L}_{total}$.
        Update all parameters using Adam on $\mathcal{L}_{total}$.
    end for
end for

C Baseline Implementation Details.

To ensure a fair comparison, all baseline methods use the same observation space and encoder architecture where applicable. All methods were trained for the same number of steps with parameter sharing enabled. Regarding backbones: SeqComm is explicitly built on MAPPO, and MAGI has been integrated into MAPPO. TarMAC and CommFormer use their official implementations, which are not based on MAPPO; however, both are communication architectures that can be instantiated on different MARL backbones. To provide a controlled same-backbone comparison, we additionally implement TarMAC and CommFormer within our MAPPO framework (Table 5). Specific implementation details are as follows:

- **SeqComm**: A multi-level communication scheme with negotiation and launching phases. Agents use a learned world model to simulate future trajectories and dynamically determine decision-making priorities based on state-value estimates, enabling explicit coordination by communicating actual actions from higher-priority to lower-priority agents. *Implementation*: We implemented SeqComm based on the original paper description, using a 2-layer MLP world model (hidden size 128), 1 communication round per timestep, and a priority threshold of 0.6. The communication graph is fully connected during the negotiation phase.
- **CommFormer**: A Transformer-based method that uses multi-round message passing with self-attention to dynamically weight inter-agent messages. Focuses on learning what to communicate from a graph modeling perspective without explicit structural constraints. *Implementation*: We use the official implementation with default hyperparameters: 2 attention heads, 2 communication rounds, message dimension 64, and dropout rate 0.1.
- **TarMAC**: An attention-based gating mechanism for directed communication. Each agent learns to generate keys and queries to select and send targeted messages, resulting in differentiable but often dense communication graphs. *Implementation*: We use the official implementation with key/query dimension 64, softmax temperature 1.0, and a fully connected communication graph as in the original paper.
- **MAGI**: A robust communication method applying Graph Information Bottleneck (GIB) to compress edge-level messages. MAGI operates on a fixed or attention-generated graph, focusing on message compactness rather than topology optimization. We integrate MAGI into MAPPO for fair comparison. Implementation: 2-layer MLP encoder (hidden 64), Gaussian prior, IB coefficient $\beta = 0.01$, attention-based underlying graph (same as TarMAC).
- **MAPPO**: A CTDE baseline with centralized value function and decentralized actors, without explicit communication. Provides a performance baseline for methods without communication. *Implementation*: We use the popular open-source implementation with standard PPO hyperparameters: clip range 0.2, entropy coefficient 0.01, and GAE λ=0.95.

C.1 Computational complexity analysis.

To complement our empirical efficiency evaluation, we provide a theoretical per-forward-pass complexity analysis. Let N be the number of agents, d the hidden dimension, L the number of GNN layers in SEOC, and K the fixed number of communities.

- **MAPPO (No Comm):** $O(Nd^2)$, dominated by the policy network inference.
- **TarMAC:** $O(N^2d)$ due to the pairwise key-query attention.
- **CommFormer:** $O(N^2d + Nd^2)$ per communication round from multi-head self-attention and feedforward layers.
- **SeqComm:** $O(N^2 + Nd^2)$, mainly from world model simulation and priority sorting.
- **MAGI:** $O(N^2d)$, from edge-wise GIB encoding and attention-based graph construction.
- **SEOC (Ours):** $O(N^2d)$ for pairwise edge-logit computation, $O(N^2)$ for Gumbel-Softmax sampling, $O(NKd)$ for differentiable community assignment (with $K \ll N$), and $O(L(N^2d + Nd^2))$ for multi-layer GNN aggregation. Overall, the dominant term is $O(N^2Ld)$.

All methods that rely on pairwise interactions (TarMAC, MAGI, CommFormer, and SEOC) share a quadratic dependence on N. The complexity of SEOC is $O(L(N^2d + Nd^2))$; the term Nd^2 arises

from the per-layer MLP transformations, which is common to all neural-network-based methods. For typical settings where $N \leq 61$ and $d = 128$, the Nd^2 term is larger than N^2d, but as N grows large (e.g., hundreds of agents), the N^2d term dominates. The asymptotic complexity of SEOC is comparable to that of TarMAC and MAGI (up to a constant factor L), and no method incurs an asymptotically distinct overhead.

SEOC does not claim asymptotic computational savings: pairwise edge-logit computation and Gumbel-Softmax sampling remain $O(N^2)$ per timestep, and are performed locally by each agent. The contribution is communication efficiency: high-dimensional messages are transmitted only along sparse active edges ($O(|E|)$), rather than over a fully connected graph ($O(N^2)$). Topology discovery requires only an $O(N)$ low-dimensional broadcast of hidden states, while heavy message passing is restricted to the sparse graph. This distinction matters for real-world deployment, such as autonomous driving and wireless networks, where communication bandwidth is often the primary bottleneck and computation can be parallelized locally.

D Experimental Description

D.1 Experimental Setup

This section provides comprehensive details of our experimental setup, including environment configuration, network architectures, hyperparameters, and training procedures to ensure reproducibility.

We conduct experiments on the SMACv2. To create a challenging setting where communication is essential for coordination, we modify the default environment with the following changes: (1) Sight Range: Reduced from 9 to 3 for all unit types; (2) Ally Visibility: Agents cannot perceive any information about allied units (health, position, type) even within sight range; (3) Stochasticity: Randomized army compositions and starting positions per episode (default SMACv2 setting). All other environment parameters remain at their default values.

Table 2 summarizes the key scenario configurations used in our experiments. Probabilities denote the sampling likelihood of each unit type in randomized army compositions.

Table 2: SMACv2 Scenario Configurations

Race	Unit Types	Probability	Scenarios
terran	Marine	0.45	terran 5 vs 5
	Marauder	0.45	terran 10 vs 10
	Medivac	0.10	terran 20 vs 20
protoss	Stalker	0.45	protoss 5 vs 5
	Zealot	0.45	protoss 10 vs 10
	Colossus	0.10	protoss 20 vs 20
zerg	Zergling	0.45	zerg 5 vs 5
	Baneling	0.10	zerg 10 vs 10
	Hydralisk	0.45	zerg 20 vs 20

We evaluate SEOC in urban traffic signal control tasks using SUMO [Lopez et al., 2018], a widely adopted microscopic traffic simulator. In these tasks, each traffic intersection is controlled by an independent agent, with the objective of minimizing overall congestion through coordinated signal timing adjustments. The environment presents several key challenges: (1) partial observability, where agents only have access to local intersection information; (2) dynamic traffic patterns with stochastic vehicle generation; and (3) heterogeneous intersection configurations.

The task is configured as follows. We utilize real-world road networks from two cities: Jinan with approximately 32 intersections and Qingdao with about 61 intersections. All agents function as homogeneous traffic signal controllers. Each agent's observation consists of per-lane features, including the count of waiting vehicles, traffic volume, cumulative delay, and the current signal phase state. The action space allows the selection of any feasible signal phase, with 4 predefined phases for standard crossroads and 3 for T-junctions. To guide the learning process, the reward is designed as a weighted combination of multiple objectives: minimizing queue length and delay, maximizing

vehicle throughput, and penalizing excessive signal phase switching. Performance is assessed using 2 key metrics: Queue Length, defined as the number of waiting vehicles per lane, and Delay, defined as the average additional time spent by vehicles due to congestion and signal controls.

D.2 Hyperparameter Sensitivity Analysis

We conduct a sensitivity analysis on four key hyperparameters: the number of communities K, the structural entropy weight α_{SE}, the information bottleneck weight α_{XIB}, and the number of GNN layers L. The experiments are performed on three representative SMACv2 scenarios (terran 10 vs 10, protoss 10 vs 10, and zerg 10 vs 10) with 5 random seeds each. Table 3 summarizes the mean win rates under different hyperparameter settings, with default values highlighted in bold.

As shown in Table 3, SEOC exhibits robust performance across a wide range of hyperparameter values. The win rate remains relatively stable when K ranges from 3 to 6, α_{SE} from 0.01 to 0.2, α_{XIB} from 0.1 to 2.0, or L from 1 to 3, indicating that SEOC is not overly sensitive to these hyperparameters and can be tuned reliably in practice. The default values chosen in our experiments ($K = 4$, $\alpha_{\text{SE}} = 0.05$, $\alpha_{\text{XIB}} = 0.5$, $L = 2$) strike a good balance between expressiveness, stability, and performance across all scenarios. Besides, to assess sensitivity to the binarization threshold (0.5), we swept the threshold from 0.1 to 0.9 on terran 10 vs 10; the win rate of all methods varied by less than 2%, confirming our efficiency conclusions are robust to this choice.

Table 3: Hyperparameter sensitivity analysis on three SMACv2 scenarios.

HP	Value	terran	protoss	zerg
K	3	54.8 ± 2.9	86.1 ± 3.2	56.2 ± 3.1
	4	$\mathbf{56.6 \pm 3.2}$	$\mathbf{87.5 \pm 2.8}$	$\mathbf{58.5 \pm 2.9}$
	5	55.9 ± 3.0	86.3 ± 3.0	57.8 ± 3.3
	6	54.2 ± 3.4	84.7 ± 3.5	56.9 ± 3.5
α_{SE}	0.01	50.5 ± 3.6	80.9 ± 3.7	51.2 ± 4.2
	0.05	55.2 ± 3.0	$\mathbf{87.5 \pm 2.8}$	$\mathbf{58.5 \pm 2.9}$
	0.1	$\mathbf{56.6 \pm 3.2}$	86.1 ± 3.1	57.9 ± 3.2
	0.2	54.9 ± 3.3	85.2 ± 3.4	56.5 ± 3.6
α_{XIB}	0.1	55.2 ± 3.4	85.8 ± 3.2	57.2 ± 3.3
	0.5	$\mathbf{56.6 \pm 3.2}$	$\mathbf{87.5 \pm 2.8}$	$\mathbf{58.5 \pm 2.9}$
	1.0	55.7 ± 3.3	86.4 ± 3.0	57.8 ± 3.1
	2.0	54.6 ± 3.6	85.0 ± 3.5	56.9 ± 3.5
L	1	53.8 ± 3.7	84.2 ± 3.6	55.9 ± 3.8
	2	$\mathbf{56.6 \pm 3.2}$	$\mathbf{87.5 \pm 2.8}$	$\mathbf{58.5 \pm 2.9}$
	3	56.1 ± 3.0	86.9 ± 2.9	58.0 ± 3.0

To illustrate our hyperparameter selection, consider the number of communities K, which directly influences structural organization and coordination efficiency. Without prior knowledge of specific roles, we determined K via systematic validation. Comparing $K \in \{3, 4, 5, 6\}$ across representative scenarios, we found that $K = 4$ achieves the best or near-best trade-off between win rate and communication efficiency, and thus uniformly set $K = 4$ in all experiments. This cross-scenario validation demonstrates SEOC's ability to adaptively form structured communication in unseen tasks. We also tested extreme values $K = 1$ and $K = 8$ on protoss 10 vs 10; win rates were 79.3% and 83.1% (default 87.5%), still above many baselines, showing robustness even at extremes. The remaining hyperparameters follow standard MARL practices and were kept consistent with the MAPPO baseline for fair comparison. Table 4 summarizes the key hyperparameters for SEOC and the shared training setting.

Discussion on adaptive community number While we fix $K = 4$ across all scenarios based on the sensitivity analysis above, a more general design would allow the system to automatically determine K from the task structure. Building on the information-theoretic foundation of structural entropy, one promising direction is to incorporate a minimum description length (MDL) principle to penalize communities with negligible volume, thereby encouraging the model to prune redundant dimensions.

We leave the precise formulation and empirical validation of such an MDL-based regularization to future work.

D.3 Exact Formulation of Differentiable Structural Entropy Proxy

To ensure numerical stability and reproducibility, we provide the exact implementation of the differentiable structural entropy proxy introduced in Section 4.2.2. This addresses potential ambiguities in the indexing and the handling of near-zero probabilities.

Definitions with explicit indices. The soft degree of agent i is defined as $d_i^{\text{soft}} = \sum_{j \neq i} A_{ij}$, where A_{ij} is the soft adjacency matrix from Eq. (2). The total soft volume is $\text{Vol}^{\text{soft}}(G) = \sum_i d_i^{\text{soft}}$. The soft volume of community k is $\text{Vol}^{\text{soft}}(C_k) = \sum_i c_{i,k} \cdot d_i^{\text{soft}}$, where $c_{i,k} = \text{Softmax}(f_{\phi_C}(h_i))_k$ is the soft assignment probability. The soft degree of agent i conditioned on being in community k is $d_{i|k}^{\text{soft}} = c_{i,k} \cdot d_i^{\text{soft}}$.

Differentiable inter- and intra-community losses. The two components are computed as:

$$\mathcal{L}_{\text{inter}} = -\sum_{k=1}^{K} \frac{\text{Vol}^{\text{soft}}(C_k)}{\text{Vol}^{\text{soft}}(G)} \log_2 \left(\frac{\text{Vol}^{\text{soft}}(C_k)}{\text{Vol}^{\text{soft}}(G)} + \epsilon \right),$$

$$\mathcal{L}_{\text{intra}} = -\sum_{k=1}^{K} \frac{\text{Vol}^{\text{soft}}(C_k)}{\text{Vol}^{\text{soft}}(G)} \sum_{i=1}^{N} \frac{d_{i|k}^{\text{soft}}}{\text{Vol}^{\text{soft}}(C_k)} \log_2 \left(\frac{d_i^{\text{soft}}}{\text{Vol}^{\text{soft}}(C_k)} + \epsilon \right).$$

Here, we have simplified the argument of the logarithm using $d_{i|k}^{\text{soft}} / c_{i,k} = d_i^{\text{soft}}$, which holds for $c_{i,k} > 0$. In practice, we add a small constant $\epsilon = 10^{-8}$ inside the logarithm to avoid numerical issues when the argument approaches zero. No additional masking or normalization is applied beyond the softmax operation that guarantees $c_{i,k} \in (0, 1)$ and $\sum_k c_{i,k} = 1$.

Handling of near-zero community probabilities. Although the softmax output never yields exactly zero, it can produce extremely small values (e.g., $< 10^{-7}$). In the rare case that $\text{Vol}^{\text{soft}}(C_k)$ becomes negligible (smaller than ϵ), the term $\frac{\text{Vol}^{\text{soft}}(C_k)}{\text{Vol}^{\text{soft}}(G)} \log_2(\ldots)$ tends to zero because $x \log_2 x \to 0$ as $x \to 0$. We therefore clamp the volume to a small value (e.g., $\max(\text{Vol}^{\text{soft}}(C_k), 10^{-8})$) before normalization, which does not affect gradients in practice. The default PyTorch implementation automatically handles such cases without gradient explosion.

Symmetrization. The SE loss $\mathcal{L}_{\text{SE}}$ is computed on the symmetrized adjacency matrix $\tilde{A} = (A + A^\top)/2$ to align with the undirected graph assumption in Eq. (1). The directed matrix A is still used for message propagation in the GNN aggregation (Section 4.4). This separation ensures that the topological regularization encourages symmetric, modular communities while preserving flexible directed communication during execution.

E Additional Experimental Analysis

E.1 Task Performance (Q1)

Results on Traffic Signal Control task. As shown in Figure 5, SEOC demonstrates significant performance advantages in the traffic signal control task. The experiments were conducted on two real-world urban road networks (Qingdao with 61 intersections and Jinan with 32 intersections) using the SUMO simulation platform with stochastic and dynamic traffic flows. Each intersection is controlled by an independent agent with only local observational information, and the objective is to minimize overall congestion through coordination. Key evaluation metrics are average delay time (seconds) and average queue length (vehicles per lane). The results show that SEOC achieves the best performance across all compared methods. CommFormer and MAGI are omitted from this experiment due to computational constraints; their relative performance compared to SEOC is already demonstrated on SMACv2 in Table 6.

Table 4: Hyperparameter settings for SEOC.

Parameter	Value
SEOC-specific Parameters	
Gumbel-Softmax temperature (σ)	Anneal from 1.0 to 0.1
Structural entropy weights (λ_{inter}, λ_{intra})	0.05, 0.1
Node Information Bottleneck (β)	0.01
Total loss coefficients (α_{SE}, α_{XIB})	0.05, 0.5
Number of communities (K)	4
GNN layers (L)	2
Message dimension (dim(m_i))	64
Post-GRU MLP hidden size (dim(h_i))	128
MAPPO Training Parameters (Shared)	
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
PPO clip range (ϵ)	0.2
Learning rate	5×10^{-4}
Value loss coefficient (c_1)	0.5
Entropy bonus coefficient (c_2)	0.01
Parallel environments (num_envs)	8
Steps per environment	128
Total timesteps per rollout	1024
PPO epochs per update	5
Minibatches per epoch	4
Minibatch size (timesteps)	256
Gradient clipping norm	10.0

The superiority of SEOC stems from its ability to learn a structured, hierarchical communication topology that dynamically adapts to real-time traffic states. *In practice, the discovered communities consistently align with traffic corridors and arterial clusters, such as spatially adjacent intersections along major flows, enabling dense information exchange within functionally similar intersections while preserving necessary cross-community links.* Compared to baselines including SeqComm, TarMAC, and MAPPO, SEOC's entropy-minimized topology not only reduces the number of connections but also yields a sparse yet semantically organized communication pattern. This allows the system to respond swiftly to local congestion and coordinate global signal strategies, significantly reducing average delay and queue length. These results validate SEOC's effectiveness in complex cooperative decision-making and support its applicability in real-world large-scale traffic management.

Comparison with LSC and same-backbone baselines. We additionally compare with the hierarchical LSC baseline and instantiate TarMAC/CommFormer on the same MAPPO backbone under the hard-mode setting (sight range 3, no ally perception, 10M steps, 5 seeds). As shown in Table 5, SEOC consistently outperforms all variants.

Table 5: Comparison with LSC and same-backbone TarMAC/CommFormer on hard-mode SMACv2 (mean $\pm$ std, 5 seeds).

Method	terran 10v10	protoss 10v10	zerg 10v10
SEOC (Ours)	56.6 ± 3.2	87.5 ± 2.8	58.5 ± 2.9
LSC	48.3 ± 4.1	79.6 ± 3.8	50.7 ± 4.3
TarMAC + MAPPO	47.5 ± 4.2	72.3 ± 3.8	44.8 ± 4.5
CommFormer + MAPPO	50.8 ± 3.8	76.9 ± 3.5	48.2 ± 4.1

Results on standard SMACv2. In summary, SEOC achieves 73.5% (terran 10v10), 86.8% (protoss 10v10), and 65.2% (zerg 10v10) on standard SMACv2, outperforming the strongest baseline MAGI by 8.2%, 17.6%, and 5.4%, respectively. These results confirm that SEOC's effectiveness is not

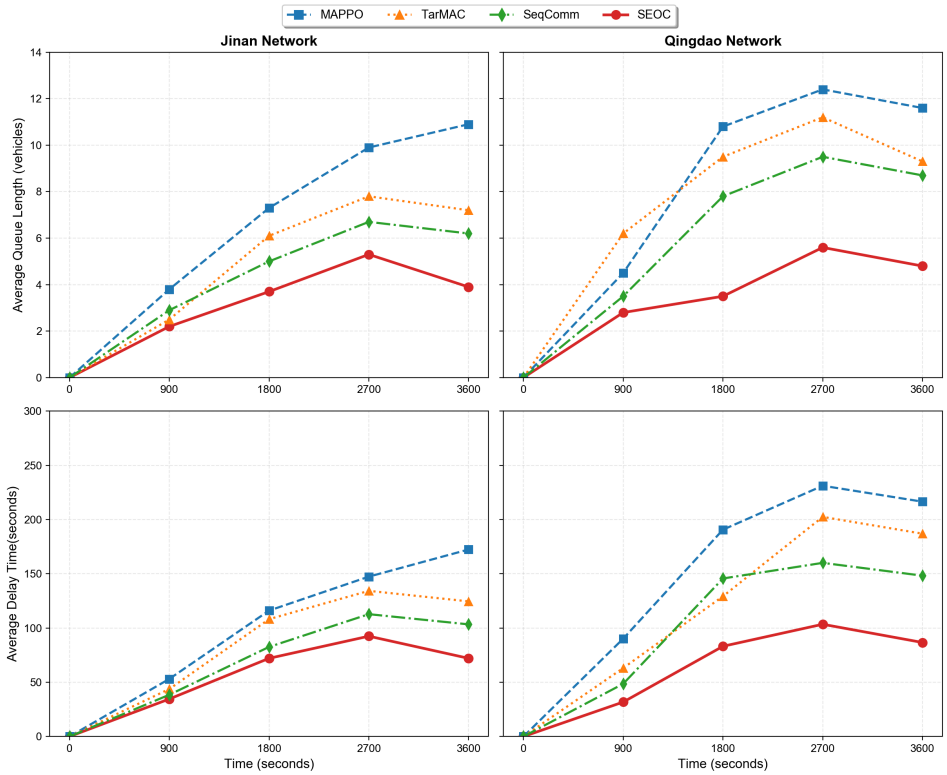


Figure 5: Evaluation results on large-scale traffic signal control task (Jinan and Qingdao Network).

contingent on the hard-mode modifications. To verify that SEOC generalizes beyond our modified hard-mode setting (sight range 3, no ally perception), we additionally evaluate all methods on the **standard SMACv2** configuration (sight range 9, ally states observable). We select three representative scenarios covering different races and scales: terran 10 vs 10, protoss 10 vs 10, and zerg 10 vs 10. Each result is averaged over 5 random seeds, reporting the mean win rate (%) after 10M training steps.

As shown in Table 6, SEOC consistently outperforms all communication baselines across all three scenarios, with significant margins especially in the hardest zerg 10 vs 10 setting. Notably, even when the environment provides richer observations, the structured low-entropy communication topology learned by SEOC remains beneficial, leading to faster coordination and higher final performance. MAGI, while outperforming SeqComm and TarMAC, still falls behind SEOC, confirming that optimizing the communication topology yields additional gains beyond compressing messages on a fixed graph. Independent two-sample t-tests (Welch's t-test, one-tailed, with Holm-Bonferroni correction for multiple comparisons) confirm that SEOC significantly outperforms each baseline at $\alpha = 0.05$ in all scenarios. The uncorrected p-values for all comparisons are below 0.01. These results substantiate the robustness of SEOC's advantage.

Table 6: Test win rate (%) on standard SMACv2 scenarios (mean $\pm$ std, 5 seeds). * $p < 0.01$ (uncorrected, two-sample t-test vs. SEOC).

Method	**terran 10 vs 10**	**protoss 10 vs 10**	**zerg 10 vs 10**
SEOC (Ours)	73.5 ± 3.2	86.8 ± 3.5	65.2 ± 3.8
MAGI	$65.3 \pm 3.8^*$	$69.2 \pm 3.6^*$	$59.8 \pm 4.1^*$
SeqComm	$60.6 \pm 4.1^*$	$63.5 \pm 4.2^*$	$53.4 \pm 4.5^*$
CommFormer	$63.2 \pm 3.6^*$	$67.4 \pm 3.8^*$	$56.7 \pm 4.0^*$
TarMAC	$56.8 \pm 4.3^*$	$60.3 \pm 4.4^*$	$49.5 \pm 4.7^*$
MAPPO	$53.5 \pm 4.5^*$	$46.7 \pm 4.6^*$	$36.3 \pm 5.0^*$

E.2 Communication Efficiency (Q2)

We conduct a systematic evaluation of communication efficiency by measuring communication bandwidth and training time for SEOC and baselines on SMACv2 tasks. All methods are trained for 10M timesteps. As shown in Table 7, SEOC demonstrates superior communication efficiency on both key metrics. Its bandwidth is significantly lower than that of other communication-based methods, indicating that the SE optimization successfully learns a highly sparse, community-aware topology that eliminates redundant links. Notably, SEOC attains the shortest training time (18.5 h) among all communication-based methods. Its sparse, structured topology not only accelerates policy convergence and reduces message-aggregation cost, but also counters the notion that learned communication must incur high training overhead. Compared to non-communicating baselines, SEOC introduces only minimal runtime overhead while enabling essential coordination through structured information exchange. This balance of low bandwidth, fast training, and strong task performance renders SEOC a practical and scalable communication paradigm for real-world MAS.

Formally, the bandwidth metric in Table 7 is defined as:

$$\text{Bandwidth (KB/s)} = \frac{1}{M}\sum_{m=1}^{M}\frac{\sum_{t=1}^{T_m}\sum_{i=1}^{N}\text{bytes_sent}_{i,t}}{1024\cdot T_m}, \tag{15}$$

where M is the number of evaluation episodes, T_m is the episode duration in seconds (steps $\times$ step duration), and $\text{bytes_sent}_{i,t}$ is the bytes transmitted by agent i at timestep t, including both hidden-state broadcasts for topology discovery and messages along active edges. This captures all communication over the network.

Table 7: Communication Efficiency (terran 5 vs 5)

Method	Bandwidth (KB/s)	Training Time (h)
CommFormer	192.0	24.1
TarMAC	76.8	26.2
SeqComm	68.5	20.3
MAGI	44.6	19.2
SEOC	**31.2**	18.5
MAPPO	0	12.6

E.3 Interpretablility (Q3)

E.3.1 Clarification on Directed Communication and Undirected SE Optimization.

The learned communication graphs in Figure 4 exhibit asymmetric connections because the forward-execution protocol samples directed edges. This enables asymmetric, state-dependent communication, such as a Medivac receiving status from many units while sending targeted responses to only a few. The structural-entropy loss, however, is computed on a symmetrized adjacency matrix, aligning with the undirected formalism in Eq. (1). This design separates instantaneous asymmetric decisions from long-term structural organization. Entropy minimization encourages symmetric, modular connection probabilities, promoting stable communities, while forward sampling preserves flexibility for transient coordination. Thus, snapshots show real-time adaptive communication, whereas the learned topology converges to interpretable, community-structured patterns. We also computed structural entropy directly on the directed graph; modularity was slightly lower and training variance increased, therefore we keep the symmetrization as in Eq. (1).

E.3.2 Quantitative Interpretability Analysis

To complement the qualitative visualizations in Section 5.3, we quantitatively evaluate the interpretability of SEOC's learned communication topologies using two metrics:

Modularity (Q). A standard graph-theoretic measure that quantifies the quality of community structure. For a graph partitioned into communities, modularity is defined as the fraction of intra-community edges minus the expected fraction under a random graph with the same degree distribution.

Higher modularity indicates clearer, more cohesive communities. We compute modularity on the hard communication graph (obtained via argmax of the Gumbel-Softmax output) using the learned hard community assignments.

Normalized Mutual Information (NMI). Measures the agreement between the learned hard community assignment and the ground-truth unit types (e.g., Marine, Marauder, Medivac in Terran scenarios). NMI ranges from 0 (random assignment) to 1 (perfect alignment). Higher NMI indicates that the emergent communities correspond to functionally distinct roles.

We report both metrics on the protoss 10 vs 10 scenario over 5 random seeds, comparing SEOC against its variant without the structural entropy loss (SEOC w/o SE) and the attention-based baseline TarMAC. Results are averaged across evaluation episodes after training convergence.

Table 8: Quantitative interpretability metrics (mean $\pm$ std, 5 seeds).

Method	**Modularity (Q)**	**NMI (vs. unit types)**
SEOC (Full)	0.63 ± 0.05	0.52 ± 0.07
SEOC w/o SE	0.31 ± 0.08	0.14 ± 0.06
TarMAC	0.28 ± 0.07	0.11 ± 0.05

As shown in Table 8, SEOC achieves substantially higher modularity (0.63) compared to both the variant without structural entropy (0.31) and TarMAC (0.28), confirming that the structural entropy minimization effectively drives the emergence of clear, cohesive communities. Furthermore, the NMI score of SEOC (0.52) indicates a strong alignment between learned communities and true unit roles, whereas both baselines produce near-random alignments (NMI ≈ 0.1). These quantitative results substantiate that SEOC's communication topology is not only sparse and efficient but also intrinsically interpretable, with communities naturally mirroring functional specializations.

E.4 Ablation Study (Q4)

Structural Entropy Analysis. We further analyze the SE of the learned communication graphs across ablated variants (Table 9). A lower SE indicates a more ordered, modular topology. The full SEOC achieves the *lowest* SE, confirming its convergence to a highly organized, community-structured graph. In contrast, removing the SE loss (w/o SE) results in *significantly higher* SE, showing $\mathcal{L}_{\text{SE}}$ is essential for driving the topology toward a low-entropy state. The variants without the information bottleneck (w/o XIB) or GNN aggregation (w/o GNN) exhibit *intermediate* SE, underscoring that both components are necessary auxiliary mechanisms for achieving an optimal low-entropy structure. Overall, the win rate of each variant shows a clear *negative correlation* with its final SE, providing empirical structural evidence for the effectiveness of entropy minimization in optimizing multi-agent communication.

Table 9: Comparison of Average and Final Structural Entropy in SEOC Ablation Study (protoss 10 vs 10 scenario)

Variant	**Avg. SE (bits)**	**Final SE (bits)**
SEOC (Full)	2.8 ± 0.2	2.1 ± 0.2
SEOC w/o SE	3.4 ± 0.4	3.2 ± 0.3
SEOC w/o XIB	3.0 ± 0.3	2.4 ± 0.2
SEOC w/o GNN	3.2 ± 0.3	2.8 ± 0.3

Effect of Structural Entropy Loss Components To address whether minimizing inter-community entropy leads to community collapse (e.g., a single dominant community), we evaluate SEOC on the protoss 10 vs 10 hard-mode scenario with varying ratios of the two SE loss coefficients, λ_{inter} and λ_{intra}. The default values are $\lambda_{\text{inter}} = 0.05, \lambda_{\text{intra}} = 0.1$ (ratio 0.5). We test three additional configurations: (i) $\lambda_{\text{inter}} = 0.1, \lambda_{\text{intra}} = 0.05$ (ratio 2.0, emphasizing inter-community separation), (ii) $\lambda_{\text{inter}} = 0.02, \lambda_{\text{intra}} = 0.1$ (ratio 0.2, de-emphasizing inter-community penalty), and (iii) $\lambda_{\text{inter}} = 0, \lambda_{\text{intra}} = 0.1$ (no inter-community entropy). Each configuration is run over 5 seeds; we report the win rate and the number of non-empty communities (effective K) at convergence.

Table 10: Ablation of λ_{inter} and λ_{intra} coefficients on protoss 10 vs 10 (mean $\pm$ std, 5 seeds).

λ_{inter}	λ_{intra}	Ratio ($\lambda_{\text{inter}}/\lambda_{\text{intra}}$)	Win Rate (%)	Effective K
0.05 (default)	0.1 (default)	0.5	87.5 ± 2.8	3.6 ± 0.4
0.1	0.05	2.0	84.6 ± 3.0	2.9 ± 0.5
0.02	0.1	0.2	86.2 ± 2.9	3.2 ± 0.3
0	0.1	0 (inter removed)	80.1 ± 3.4	3.4 ± 0.2

As shown in Table 10, completely removing the inter-community entropy yields an effective number of communities (3.4 ± 0.2) that is slightly lower than the default (3.6 ± 0.4), but the drop is not statistically significant. The win rate, however, decreases notably, indicating that the inter-community penalty is crucial for maintaining high-quality community structures. Increasing the emphasis on inter-community separation (ratio 2.0) reduces the effective K to 2.9 ± 0.5, confirming that stronger separation pressure encourages community merging, while performance remains close to the default. In all configurations, the graph does not collapse into a single dominant community, as the effective K stays above 2.5. *No near-empty graphs are observed*; explicit edge-budget constraints are unnecessary because the SE objective naturally balances intra- and inter-community terms. This suggests that the joint minimization of inter- and intra-community entropies, as formulated in Eq. (8), naturally balances separation and cohesion, preventing degenerate solutions. The default ratio (0.5) achieves the best trade-off, delivering both a high win rate and a moderate number of functional communities.

Ablation Study on topological level. We conduct another ablation experiment to evaluate the contribution of other key components in SEOC. As shown in Figure 6, we design three ablation variants: Full Connection (all-to-all communication), Random Connection (random connections with similar density), and Fixed-size Matrix (maintaining original matrix size with dead agents zeroed out). The performance comparison across multiple dimensions highlights the distinct roles and synergistic effects of SEOC's modules.

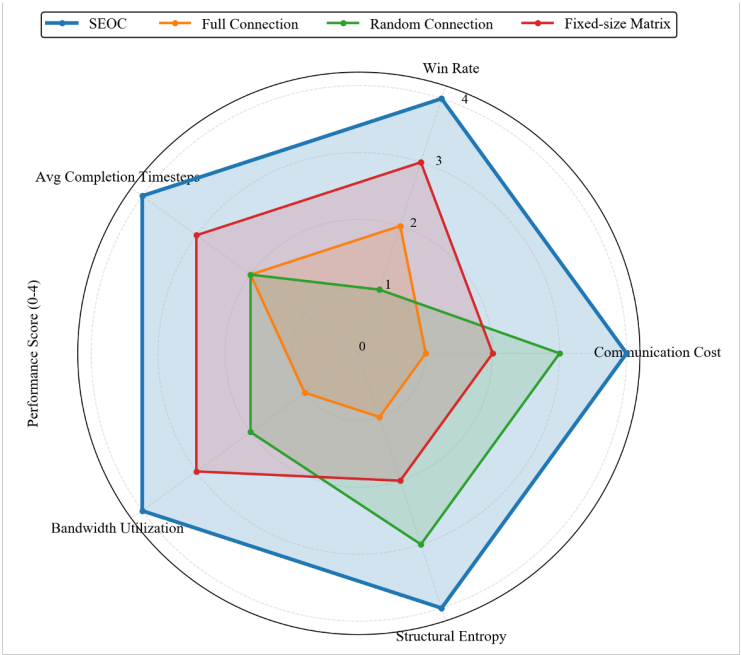


Figure 6: Comparing SEOC and variants across five performance metrics, including Win Rate, Communication Cost, Structural Entropy, Bandwidth Utilization, and Average Completion Timesteps. Metrics are normalized by min-max scaling; lower-is-better metrics are inverted before scaling. Higher scores always indicate better performance.

Compared to the ablated variants, SEOC achieves a balanced and superior performance. It drastically reduces communication overhead versus Full Connection while maintaining high win rates, proving

structured sparsity prevents information overload. Against Random Connection, SEOC yields better performance with similar link density, showing that purposeful, role-aware connections are more valuable than arbitrary ones. Its dynamic graph resizing also outperforms the Fixed-size Matrix baseline, proving adaptive topology management is essential. These results confirm SEOC's strength comes from integrating three principles: structured sparsity for efficiency, role-aware linking for coordination, and dynamic adaptation for scalability. This synergy enables the optimal balance between effective coordination and efficient communication. *We observe that minimizing $\mathcal{L}_{SE}$ correlates with reduced active edges (Pearson's $r = 0.73$) and increased modularity ($r = -0.81$), confirming that the SE proxy actively promotes sparsity and structure, not just order.*

E.5 Scalability (Q5)

To further evaluate the scalability of SEOC, we conduct experiments on real-world urban road networks from two cities: Qingdao with 61 traffic signal agents and Jinan with 32 traffic signal agents, each controlling an intersection. The simulation is implemented in SUMO [Lopez et al., 2018], where vehicles are generated randomly with dynamic traffic flows. Agents operate under partial observability with local intersection states and aim to minimize overall congestion.

We compare SEOC with three baselines: SeqComm, TarMAC, and MAPPO. We evaluate model performance based on two primary metrics concerning traffic congestion and vehicle efficiency at intersections: *average delay time* (seconds) and *average queue length* (vehicles per lane). Higher values for either metric correspond to poorer performance of the control method.

Table 11 presents the results on the Qingdao network with 61 agents. SEOC achieves the lowest delay time (86.72 s) and the shortest queue length (4.85 vehicles), substantially outperforming all baselines. Table 12 shows the results on the Jinan network with 32 agents. Again, SEOC achieves the best performance with a delay time of 72.15 s and a queue length of 3.92 vehicles. Across both network scales, SEOC consistently delivers the lowest congestion metrics. Queue length and delay improve by 28–40%, with no physical neighbor constraints and fully learned topologies.

We further implement a distance-based communication baseline in SUMO, where each intersection communicates with its k nearest neighbors ($k = 3, 5$). While distance-based communication improves over no communication, it still falls significantly short of SEOC: on the Qingdao network, SEOC achieves 86.72s delay and 4.85 queue length, versus 132.41s/7.23 for $k = 3$ and 118.68s/6.40 for $k = 5$, while MAPPO without communication yields 216.45s/11.62. This confirms that learning task-aware structured sparsity via structural entropy is more effective than heuristic spatial proximity; SEOC's communities align with functional traffic patterns rather than just spatial adjacency, capturing dynamic dependencies such as coordinating with upstream flows during congestion.

These results demonstrate that SEOC maintains strong scalability across different team sizes. The structured communication enables efficient coordination across many intersections without excessive communication overhead, confirming its suitability for large-scale urban traffic management. Moreover, the consistent relative improvement over baselines in both networks indicates that SEOC effectively adapts to different scales and traffic patterns. This positions SEOC as a robust and scalable framework for multi-agent collaboration, particularly in environments where team size may dynamically change or where coordination must be maintained under expanding agent populations.

Table 11: Performance comparison in the Qingdao network with 61 traffic signal agents.

Method	**Delay Time (s)**	**Queue Length**
MAPPO	216.45	11.62
TarMAC	187.03	9.31
SeqComm	148.34	8.74
Distance-based ($k = 5$)	118.68	6.40
Distance-based ($k = 3$)	132.41	7.23
SEOC	**86.72**	**4.85**

Table 12: Performance comparison in the Jinan network with 32 traffic signal agents.

Method	Delay Time (s)	Queue Length
MAPPO	172.36	10.94
TarMAC	124.57	7.25
SeqComm	103.42	6.18
SEOC	**72.15**	**3.92**

F Impact Statement

This work presents a principled framework for learning structured communication in multi-agent systems by minimizing the topological complexity of interaction networks. Our approach advances multi-agent coordination theory and yields significant empirical gains in efficiency, performance, and interpretability. The resulting sparse, modular communication graphs are highly efficient, enabling scalable coordination in resource-constrained settings such as large-scale traffic control or distributed robotics. Their human-interpretable structure further enhances transparency and trust, supporting deployment in safety-critical domains where understanding agent behavior is essential.

We recognize the broader societal implications of more capable multi-agent systems. Although developed for cooperative settings, the framework could potentially be adapted for competitive or adversarial scenarios. We therefore encourage its responsible application in alignment with ethical AI principles. By bridging information theory and reinforcement learning, this work provides a foundation for building more efficient, adaptive, and explainable collaborative AI systems.

G Limitations and Future Work

One limitation of our approach is the inconsistency between training and inference objectives for structural entropy optimization. Specifically, the differentiable proxy $\mathcal{L}_{\text{SE}}$ is computed using a soft adjacency matrix (via Gumbel-Softmax) and soft community assignments, while inference uses discrete, hard structures. Although the Gumbel-Softmax temperature anneals to near-zero during training, making the soft adjacency approximately binary, the community assignments remain continuous. This discrepancy means that the precise discrete structural entropy is not directly minimized. Empirically, we observe that the impact on final performance is negligible (as evidenced by the strong results in Section 5), and the learned graphs still exhibit clean, interpretable communities. A principled extension would be to apply straight-through estimators or Gumbel-Softmax to discretize both the graph and the assignments end-to-end. We leave this to future work.

Another limitation concerns the number of communities K. In the current framework, K is fixed via cross-scenario validation rather than being adaptively determined by the task structure and agent interdependencies. While this choice yields strong empirical performance across the tested benchmarks, a more general design would allow the system to infer the intrinsic collaborative granularity on the fly. In Appendix D.2, we outline a promising MDL-based regularization scheme that could enable automatic discovery of K without manual presetting, which we leave as future work. Despite this limitation, our hyperparameter analysis (Table 3) shows that SEOC is robust to a range of K values (3 to 6), with negligible performance degradation.

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 core claims, and these claims are supported by experimental results in Section 5 and Appendix E.

   Guidelines:

   - The answer NA 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 NA 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: The limitations of the work are discussed in Section 6 and Appendix G, and the future directions are also provided.

   Guidelines:

   - The answer NA 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: Appendix A provides complete theoretical results (convergence, sample complexity, coordination error bounds) with all assumptions (Assumptions 1-10) and detailed proofs.

Guidelines:

- The answer NA 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: Appendix C gives the full algorithm (Algorithm 1), Appendix D provides hyperparameters (Table 4) and environment configurations, ensuring reproducibility.

Guidelines:

- The answer NA 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: [No]

   Justification: The paper does not provide a public link to code or data, nor does it state that they will be released; thus reproducibility relies on the described details only.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 and Appendix D specify all training and test details, including sight range modifications, reward design, hyperparameter sensitivity analysis, and optimizer settings.

   Guidelines:

   - The answer NA 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: Learning curves (Figure 2) show shaded regions representing standard errors over 5 seeds; tables report mean ± std, and error bars are explained. Statistical significance is reported in Appendix E.1

   Guidelines:

   - The answer NA 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: Section 5 states experiments were run on an RTX 4090 GPU with 24GB memory; training time per method is reported in Appendix E.2 (Table 7).

   Guidelines:

   - The answer NA 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 conforms to the NeurIPS Code of Ethics; no human subjects or unethical data usage are involved, and anonymity is preserved.

   Guidelines:

   - The answer NA 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: [Yes]

Justification: Section F Impact Statement discusses positive societal impacts (efficient, interpretable MAS for traffic control, safety-critical domains) and potential misuse (competitive scenarios), urging responsible application.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

Answer: [Yes]

Justification: The paper does not release any models or datasets with high risk for misuse; no safeguards are needed.

Guidelines:

- The answer NA 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: [N/A]

Justification: The paper cites the original papers for SMACv2, SUMO, and baseline methods but does not explicitly mention their licenses; no assets are redistributed.

Guidelines:

- The answer NA 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: [Yes]

    Justification: All third-party assets used in the paper, including code and datasets, are properly credited by citing the original sources.

    Guidelines:

    - The answer NA 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 research involves only simulated multi-agent environments and does not use crowdsourcing or human subjects.

    Guidelines:

    - The answer NA 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: No human subjects research is conducted; thus IRB approval is not applicable.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Large language models were not used as an important, original, or non-standard component of the core methodology; any writing assistance does not impact scientific rigor.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.