

# EXPONENTIAL TOPOLOGY-ENABLED SCALABLE COMMUNICATION IN MULTI-AGENT REINFORCEMENT LEARNING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

In cooperative multi-agent reinforcement learning (MARL), well-designed communication protocols can effectively facilitate consensus among agents, thereby enhancing task performance. Moreover, in large-scale multi-agent systems commonly found in real-world applications, effective communication plays an even more critical role due to the escalated challenge of partial observability compared to smaller-scale setups. In this work, we endeavor to develop a scalable communication protocol for MARL. Unlike previous methods that focus on selecting optimal pairwise communication links—a task that becomes increasingly complex as the number of agents grows—we adopt a global perspective on communication topology design. Specifically, we propose utilizing the exponential topology to enable rapid information dissemination among agents by leveraging its small-diameter and small-size properties. This approach leads to a scalable communication protocol, named ExpoComm. To fully unlock the potential of exponential graphs as communication topologies, we employ memory-based message processors and auxiliary tasks to ground messages, ensuring that they reflect global information and benefit decision-making. Extensive experiments on large-scale cooperative benchmarks, including MAgent and Infrastructure Management Planning, demonstrate the superior performance and robust zero-shot transferability of ExpoComm compared to existing communication strategies.

## 1 INTRODUCTION

Cooperative multi-agent reinforcement learning (MARL) has recently emerged as a promising approach for complex decision-making tasks across diverse real-world applications, such as resource allocation (Ying & Dayong, 2005), package delivery (Seuken & Zilberstein, 2007), autonomous driving (Zhou et al., 2021), robot control (Swamy et al., 2020), and infrastructure management planning (Leroy et al., 2024). Under the widely adopted centralized training and decentralized execution (CTDE) paradigm (Kraemer & Banerjee, 2016; Lyu et al., 2021), algorithms like MADDPG (Lowe et al., 2017), COMA (Foerster et al., 2018), MATD3 (Ackermann et al., 2019), QMIX (Rashid et al., 2020), and MAPPO (Yu et al., 2022) have achieved notable success.

To enhance agent collaboration in partially observable scenarios, communication mechanisms have been incorporated into multi-agent systems (MASs) to assist in decentralized decision-making (Sukhbaatar et al., 2016). Enabling information exchange during execution helps MARL algorithms to address non-stationarity and partial observability prevalent in these environments. Building upon this foundation, researchers have devoted efforts to designing effective communication protocols, focusing on three core considerations: 1) *whom* the agents should communicate with (Ding et al., 2020; Hu et al., 2024); 2) *when* communication should occur (Hu et al., 2021; Kim et al., 2019); and 3) *how* the agents should design and utilize the communication messages effectively (Das et al., 2019; Guan et al., 2022). By leveraging tools such as attention and graph neural networks (GNNs), learnable and adaptive communication mechanisms have significantly advanced MARL performance.

Despite considerable success, most existing communication strategies are designed for small-scale MASs (Lowe et al., 2017; Samvelyan et al., 2019; Peng et al., 2021) and may struggle as systems

054 scale to dozens or even hundreds of agents, which are ubiquitous in real-world applications (Cui  
055 et al., 2022; Schmidt et al., 2022; Yang et al., 2023; Ma et al., 2024). In these *many-agent* systems,  
056 existing methods that learn pairwise connectivity among agents falter for two reasons: First, these  
057 methods often require agents to receive messages only from “useful” peers. However, identifying  
058 these peers becomes increasingly challenging as the number of agents grows, potentially compro-  
059 mising the effectiveness of communication protocols (Guan et al., 2022). Second, the overhead  
060 of these methods scales poorly. Specifically, training memory consumption quickly becomes pro-  
061 hibitively large, as shown in our empirical evaluation, and the communication overhead during exe-  
062 cution scales quadratically with the number of agents, which is infeasible for many-agent systems.

063 This motivates a fundamental rethinking of scalable MARL communication: Can we adopt a  
064 global perspective and design an overall topology that propagates information among all agents  
065 effectively and at low cost, rather than relying on finding task-specific pairwise connectivity? In this  
066 vein, we propose an exponential topology-enabled communication protocol, termed *ExpoComm*,  
067 as a scalable solution for MARL communication. Unlike previous works that seek to identify  
068 useful communication links at each timestep, ExpoComm draws inspiration from graph theory  
069 and leverages the small-diameter property of exponential topologies to ensure effective commu-  
070 nication by facilitating message flow across all agents within a limited number of timesteps. The  
071 inherent sparsity (small size) of exponential topologies allows ExpoComm’s communication cost  
072 to scale (near-)linearly with the number of agents. Moreover, to fully leverage the small-size and  
073 small-diameter properties of exponential graphs for efficient information dissemination, we employ  
074 memory-based blocks for message processing and auxiliary tasks to ground messages, ensuring  
075 that they effectively reflect global information. Extensive experiments across twelve scenarios on  
076 large-scale benchmarks, including MAgent (Zheng et al., 2018) and Infrastructure Management  
077 Planning (IMP) (Leroy et al., 2024), demonstrate the superior performance of ExpoComm com-  
078 pared to baseline algorithms when handling large numbers of agents up to a hundred. Additionally,  
079 owing to its global perspective without pairwise reliance, ExpoComm exhibits remarkable zero-shot  
080 transferability to larger numbers of agents during test time.

## 081 2 RELATED WORK

082  
083 **Communication in MASs** Communication among agents in MARL was first introduced by  
084 Sukhbaatar et al. (2016); Foerster et al. (2016) and has since become an active research area due to its  
085 potential to enhance cooperation and improve task performance. The flexibility of communication  
086 protocols makes finding effective solutions for the MARL paradigm challenging (Zhu et al., 2024).  
087 To address this difficulty, many studies have focused on optimizing communication components,  
088 such as message generators, message aggregators, and connectivity among agents, through end-to-  
089 end training (Peng et al., 2017). From the sender side, ToM2C (Wang et al., 2022) and MAIC (Yuan  
090 et al., 2022) enhance message generation through teammate modeling, while CACL (Lo et al.,  
091 2024) uses contrastive learning techniques to learn communication encoding in a decentralized  
092 training paradigm. From the receiver side, TarMAC (Das et al., 2019), G2ANet (Liu et al., 2020),  
093 and MASIA (Guan et al., 2022) improve message aggregation using attention-based strategies.

094 Recently, researchers have addressed challenges posed by real-world communication systems. No-  
095 tably, NDQ (Wang et al., 2020b) and TMC (Zhang et al., 2020) reduce communication costs by  
096 crafting succinct messages, while ATOC (Jiang & Lu, 2018), IC3 (Singh et al., 2019), I2C (Ding  
097 et al., 2020), and CommFormer (Hu et al., 2024) manage overhead by pruning unnecessary commu-  
098 nication links. Additionally, Freed et al. (2020) propose a stochastic encoding/decoding scheme to  
099 handle noisy channels, and DACOM (Yuan et al., 2023) introduces delay-aware communication to  
100 account for the high latency of wireless channels.

101 Despite these advancements, scalability in communication mechanisms has been largely overlooked,  
102 often due to the quadratically increasing communication cost associated with fully-connected graphs  
103 as the number of agents grows. Although few works explicitly address the scalability issue, efforts to  
104 design communication topologies among agents offer potential solutions. These can be categorized  
105 into fully-connected, rule-based, and learned topologies. Early works (Sukhbaatar et al., 2016;  
106 Foerster et al., 2016; Peng et al., 2017) typically adopt fully-connected topologies to demonstrate  
107 communication benefits, but at the cost of high bandwidth requirements. Later on, to reduce the  
overall communication overhead, Jiang et al. (2020) and Weil et al. (2024) restrict communication

to nearby neighbors based on distance, while NeurComm (Chu et al., 2020) limits communication to neighboring agents in networked MASs. In spite of achieving significant performance gains, their further applicability may be limited since they require extra information beyond local observation to determine the communication topology. In contrast, learned topology methods assume no such requirements and offer high flexibility. In particular, ATOC (Jiang & Lu, 2018), IC3 (Singh et al., 2019), I2C (Ding et al., 2020) locally deploy gates for agents to decide if they should engage in communication. However, these methods may result in uncontrollable overall communication costs due to individual control schemes. Alternatively, MAGIC (Niu et al., 2021) utilizes graph attention mechanisms to learn the communication topology, while CommFormer (Hu et al., 2024) extends the idea and enables control over the overall communication sparsity. Although effective in small-scale MASs, peer-wise connectivity becomes increasingly difficult to learn in large-scale MASs, and high sparsity may impair performance, as discussed by Hu et al. (2024).

Our proposed ExpoComm, which incorporates rule-based topologies for rapid information dissemination among all agents, complements existing efforts in MAS communication by explicitly addressing scalability challenges.

**Exponential Graphs** Exponential graphs are a class of graph topologies that exhibit strong scalability properties with respect to the number of nodes. They have been primarily used in distributed learning to periodically synchronize model updates across devices. Assran et al. (2019) investigate exponential graphs with gossip algorithms and achieve high consensus rates for decentralized learning. Follow-up works (Wang et al., 2020a; Ying et al., 2021; Kong et al., 2021; Yuan et al., 2021) build upon this topology, optimizing model weight update algorithms and providing empirical evidence and theoretical guarantees for the effectiveness of exponential graphs. Beyond distributed learning, exponential graphs also have applications in chip design (Wang et al., 2015; 2016). Overall, exponential graphs demonstrate efficient information dissemination across many nodes, making them a promising candidate topology for achieving scalable communication in MARL.

### 3 SCALABLE COMMUNICATION WITH EXPONENTIAL GRAPH IN MARL

In this section, we propose ExpoComm, which leverages exponential graphs as communication topologies among agents in MARL to enable scalable communication. We structure the following subsections to address three key questions: 1) Why and how should exponential graphs be adapted for agent communication? 2) How can the corresponding neural network architecture be designed to effectively utilize the messages transmitted through these topologies? 3) How can messages propagated among agents be grounded to ensure their usefulness?

In Section 3.1, we outline the requirements for scalable communication: effective information dissemination among agents and low communication overhead. We translate these requirements into the challenge of identifying topologies with small diameters and sizes, key properties of exponential topologies. In Section 3.2, we discuss how memory-based message processors can enable meaningful message encoding, leveraging the small-diameter property over multiple timesteps within exponential topologies. In Section 3.3, we adopt a global perspective to ground messages using a global state reconstruction auxiliary task and contrastive learning, as ExpoComm aims to facilitate message flow across the entire graph rather than focusing on local features.

#### 3.1 EXPONENTIAL GRAPH AS THE COMMUNICATION TOPOLOGY

##### 3.1.1 PROBLEM SETTING

In this work, we consider a fully cooperative partially observable multi-agent task, which can be modeled as a decentralized partially observable Markov decision process (Dec-POMDP) (Oliehoek & Amato, 2016). The Dec-POMDP is defined by a tuple  $\mathcal{M} = \langle \mathcal{S}, A, P, R, \Omega, O, N, \gamma \rangle$  with  $N$  being the number of agents and  $\gamma \in (0, 1]$  being the discount factor. At each timestep  $t$ , with the global observation  $s^t \in \mathcal{S}$ , agent  $i$  receives a local observation  $o_i^t \in \Omega$  and then communicates with other agents. Upon receiving the messages from other agents, agent  $i$  then selects an action  $a_i^t \in A$  based on its local policy  $\pi_i$ . These individual actions collectively form a joint action  $\mathbf{a}^t \in A^N$ , leading to a transition to the next global observation  $s^{t+1} \sim P(s^{t+1}|s^t, \mathbf{a}^t)$  and inducing a global

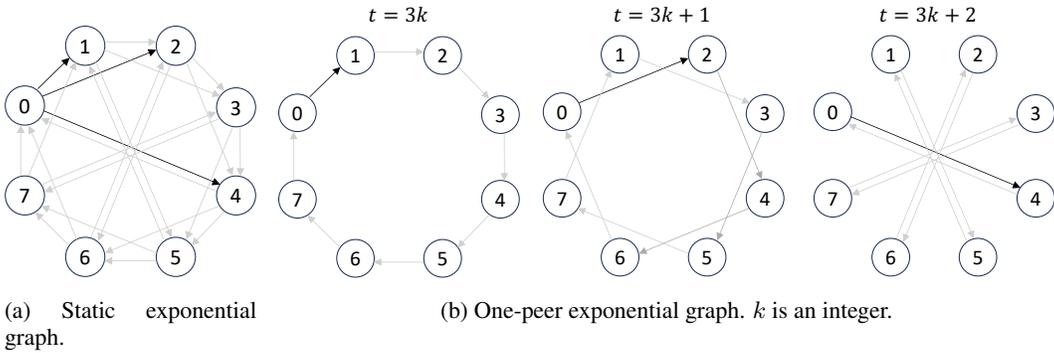


Figure 1: Illustration of exponential graphs with  $N = 8$ .

reward  $r^t = R(s^t, a^t)$ . The team objective is to learn the policies that maximize the expected discounted cumulative return  $G_t = \sum_t \gamma^t r^t$ .

### 3.1.2 COMMUNICATION TOPOLOGIES

To design an effective and scalable communication protocol in many-agent systems, it is essential to determine whom to communicate with, i.e., to construct the communication topology so that communication is both beneficial for decision-making and cost-effective. While previous work (Hu et al., 2024) assumes a static communication topology, we adopt a more flexible, time-varying directed graph  $\mathcal{G}^t = \langle \mathcal{V}, \mathcal{E}^t \rangle$ , where node  $v_i \in \mathcal{V}$  denotes agent  $i$  and edge  $e_{i \rightarrow j}^t \in \mathcal{E}^t$  indicates a communication link from agent  $i$  to agent  $j$  at timestep  $t$ .

From a graph perspective, we consider the following desiderata for the communication topology:

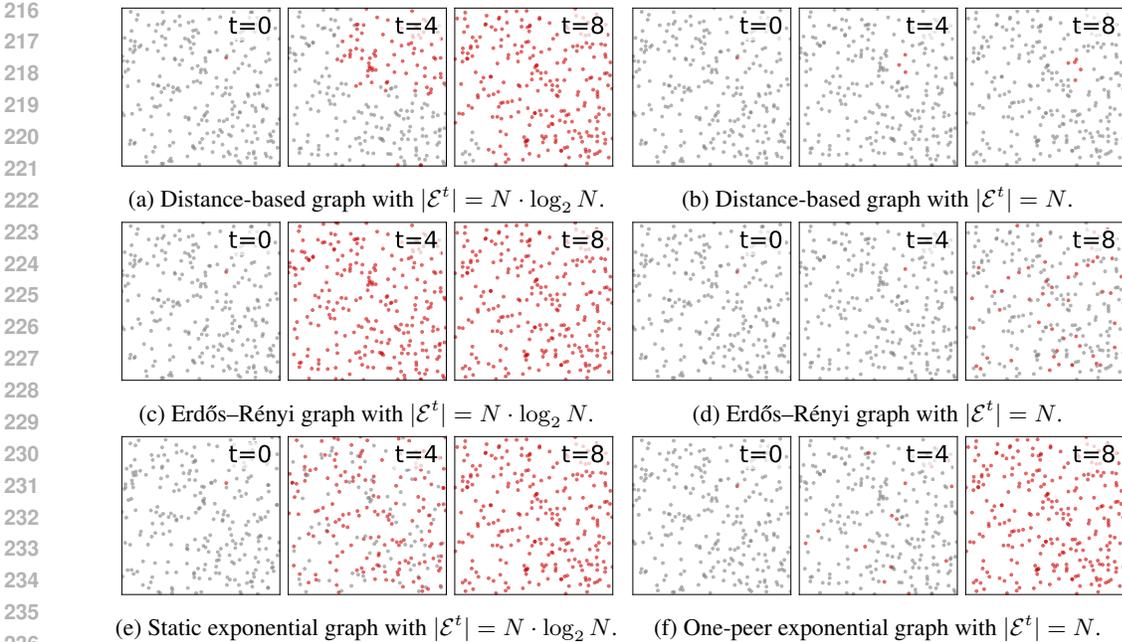
- **Small graph diameter for fast information dissemination:** Formally defined as  $\text{diameter}(\mathcal{G}^t) = \max_{v_i, v_j \in \mathcal{V}} d(v_i, v_j)$  with  $d(v_i, v_j)$  representing the shortest path distance from  $v_i$  to  $v_j$ , the graph diameter indicates how quickly messages travel through the graph. Since communication aids multi-agent decision-making by providing the locally observant agents with global information and alleviating the non-stationarity, a graph with a small diameter can expedite message exchange and is therefore desirable.
- **Small size for low communication overhead:** Formally defined as  $|\mathcal{E}^t|$ , the size of a graph denotes the total number of edges, corresponding to the number of communication links in an MAS. We assume that any message transmission incurs the same overhead, therefore the total overhead scales with the number of links. Given the high hardware requirement for communication modules and the potential delays induced by densely connected communication topologies, we prefer graphs with a small size in many-agent settings.

### 3.1.3 EXPONENTIAL GRAPHS

Based on the desiderata above for the communication topologies, we draw inspiration from graph literature and choose exponential graphs (Assran et al., 2019; Ying et al., 2021) as a promising candidate for communication topology in many-agent systems. Below, we introduce two variants of exponential graphs and demonstrate their small-diameter and small-size properties through an illustrative example.

**Static Exponential Graph** Assuming a randomly sequential ordering of agents  $0, 1, \dots, N - 1$  and the corresponding adjacency matrix  $E \in \{0, 1\}^{N \times N}$ , in the static exponential graph, each agent communicates with peers that are  $2^0, 2^1, \dots, 2^{\lfloor \log_2(N-1) \rfloor}$  hops away, which is illustrated by Figure 1a. Formally, we have

$$E_{ij}^{t(\text{stat})} = \begin{cases} 1 & \text{if } \log_2((j - i) \bmod N) \text{ is an integer or } i = j \\ 0 & \text{otherwise} \end{cases}. \quad (1)$$



237 Figure 2: A toy example to illustrate the message dissemination with different graph topologies. We  
238 demonstrate how the messages, represented by red dots, travel from a random agent to other agents  
239 over time, following different graph structures. In distance-based graphs (Jiang et al., 2020), agents  
240 are connected to top- $K$  nearest neighbors. In Erdős-Rényi random graphs (Erdos et al., 1960),  
241 the adjacency matrices are sampled uniformly from all the graphs satisfying the diameter and size  
242 conditions. In exponential graphs, the adjacency matrices follow Equations (1) and (2).  
243

244 **One-peer Exponential Graph** In the one-peer exponential graph, each agent iterates through dif-  
245 ferent peers that are  $2^0, 2^1, \dots, 2^{\lfloor \log_2 (N-1) \rfloor}$  hops away, which is illustrated by Figure 1b. Formally,  
246 we have

$$247 E_{ij}^{t(\text{one-peer})} = \begin{cases} 1 & \text{if } \log_2((j-i) \bmod N) = t \bmod \lfloor \log_2(N-1) \rfloor \text{ or } i = j \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

250  
251 **Properties** Using the adjacency matrices defined above, we verify that the graph diameter for both  
252 static and one-peer exponential graphs is  $\lceil \log_2(N-1) \rceil$  (see Appendix A for details). As discussed  
253 in Section 3.1.2, a small diameter facilitates efficient information dissemination, especially when the  
254 number of agents  $N$  is large.

255 Regarding communication costs, static exponential graphs have a size of  $N \cdot \lceil \log_2(N-1) \rceil$ , while  
256 one-peer exponential graphs have a size of  $N$ . Notably, the size of one-peer exponential graphs  
257 scales linearly with the number of agents, meaning the overall communication overhead also scales  
258 linearly.

259 To illustrate these properties, we provide a toy example in Figure 2. We visualize the message  
260 dissemination abilities of different communication topologies under varying communication bud-  
261 gets. In this example, with  $N = 256$  agents, graph sizes (communication budgets)  $|\mathcal{E}^t|$  are set to  
262  $N \cdot \log_2 N$  and  $N$ , respectively. In Figure 2, we observe that for each communication topology, re-  
263 ducing the graph sizes (as shown in Figures 2b, 2d and 2f) typically slows down dissemination speed  
264 due to increased graph diameters. This illustrates a trade-off between graph diameter and size, re-  
265 flecting the trade-off between communication performance and overhead in many-agent systems.  
266 Sparser graphs with smaller sizes result in slower message dissemination but lighter communication  
267 overhead. However, exponential topologies strike a balance in this trade-off, demonstrating strong  
268 information diffusion even with a minimal communication budget of  $N$ .

269 Based on these observations, we conclude that exponential topologies are well-suited for many-agent  
communication because: 1) In exponential topologies, any two agents can exchange messages in

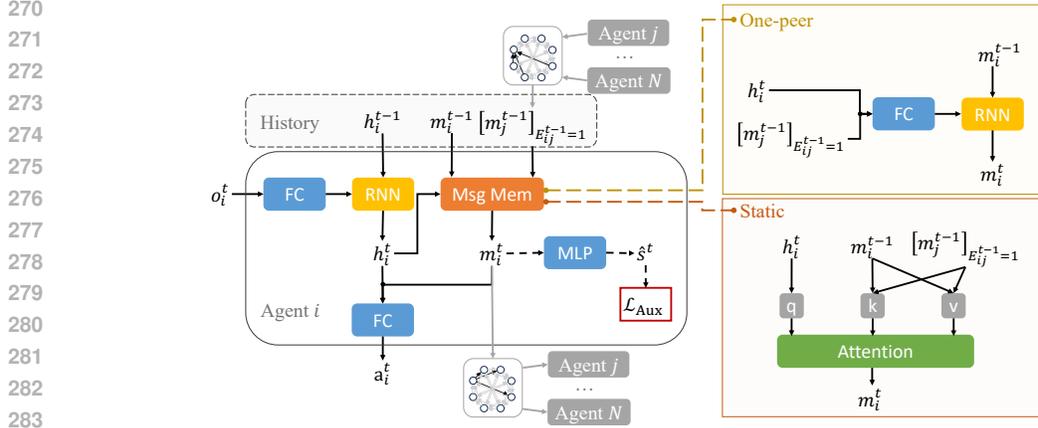


Figure 3: Neural network architecture for ExpoComm. For the static exponential topologies, attention blocks are used for message aggregation. For the one-peer exponential topologies, RNN blocks are used for message aggregation.

at most  $\lceil \log_2(N-1) \rceil$  timesteps, ensuring timely information exchange in decentralized decision-making problems. 2) The communication overhead scales nearly linearly with the number of agents, which is crucial for many-agent systems. 3) With a rule-based topology, exponential graphs are easy to deploy and adapt to systems with varying numbers of agents, as empirically verified in Section 4.2.

### 3.2 NEURAL NETWORK ARCHITECTURE DESIGN

With exponential graphs serving as the communication topology in ExpoComm, we elaborate on the neural network architecture to help agents utilize received messages for better decision-making. The overall architecture is illustrated in Figure 3. ExpoComm is based on the concept of facilitating message flow across all agents within a certain timeframe, where the graph diameter indicates the length of such timeframe. To capitalize on the small graph diameter of exponential graphs, the message-processing module at each agent should ideally preserve all information received within diameter( $\mathcal{G}^t$ ) timesteps. However, preserving all messages across multiple timesteps is not memory-efficient, so we employ sequential neural networks, such as attention blocks and recurrent neural networks (RNNs), for message processing.

### 3.3 TRAINING AND EXECUTION DETAILS

Following the QMIX (Rashid et al., 2020) algorithm, we update the network parameters  $\theta$  with the objective of minimizing the temporal difference (TD) error loss:

$$\mathcal{L}^{\text{TD}}(\theta) = \mathbb{E}_{(s^t, \mathbf{o}^t, \mathbf{a}^t, r^t, s^{t+1}, \mathbf{o}^{t+1}) \sim \mathcal{D}} \left[ (y^{tot} - Q_{tot}(s^t, \mathbf{o}^t, \mathbf{a}^t; \theta))^2 \right], \quad (3)$$

where  $y^{tot} = r + \gamma \max_{\mathbf{a}} Q_{tot}(s^{t+1}, \mathbf{o}^{t+1}, \mathbf{a}; \theta^-)$  and  $\theta^-$  represents the parameters of the target network as in DQN.

However, communication inevitably enlarges the policy space, making it more challenging to find the optimal policy **relying solely on** the MARL training objective (Li & Zhang, 2024). To facilitate learning meaningful messages, we introduce auxiliary tasks to restore global information from local messages. From a message perspective, **we aim for it to traverse among agents over multiple timesteps, accumulating new** information along the way, and ultimately reflecting global information useful for decision-making.

**Message grounding with the global state** In scenarios where the global state is available during training, the auxiliary loss is given by the prediction error of the current global state:

$$\mathcal{L}_{\text{pred}}^{\text{Aux}}(\theta, \phi) = \mathbb{E}_{(s^t, \mathbf{o}^t) \sim \mathcal{D}} [s^t - f(m_i^t; \phi)]^2, \quad (4)$$

where the **learnable** auxiliary network for prediction  $f(\cdot; \phi)$  is used to ground the messages and can be discarded after training.

**Algorithm 1** Training and Execution Procedure of ExpoComm

---

```

1: Init: Network parameters  $\theta, \phi, \mathcal{D} = \emptyset$ , step = 0,  $\theta^- = \theta$ 
2: while step < stepmax do
3:    $t = 0$ . Reset the environment.
4:   for  $t = 1, 2, \dots$ , episode_limit do
5:     // Decentralized execution at agent  $i$ 
6:     Update local history  $h_i^t$  based on current observation  $o_i^t$  and previous history  $h_i^{t-1}$ 
7:     Update agent  $i$ 's message  $m_i^t$  based on previous local message  $m_i^{t-1}$  and previously received messages  $[m_j^{t-1}]_{E_{ij}^{t-1}=1}$ 
8:     // Communication
9:     Send message  $m_i^t$  to peers  $\{j \mid E_{ij}^t = 1\}$  {▷ Equations (1) and (2)}
10:    // Action, which can happen concurrently with communication
11:    Sample action  $a_i^t$  based on current history  $h_i^t$  and current message  $m_i^t$ 
12:    Interact with the environment  $(s^{t+1}, \mathbf{o}^{t+1}, r^t) = \text{env.step}(\mathbf{a}^t)$ 
13:    Save the experience  $\mathcal{D} = \mathcal{D} \cup (s^t, \mathbf{o}^t, \mathbf{a}^t, r^t, s^{t+1}, \mathbf{o}^{t+1})$ 
14:  end for
15:  At some interval, update network parameters  $\theta, \phi$  and  $\theta^-$  {▷ Equation (6)}
16: end while
17: Output: Policy networks parameters  $\theta$ 

```

---

**Message grounding without the global state** Alternatively, when the global state is unavailable during training, we use contrastive learning for meaningful message encoding, similar to Lo et al. (2024). Specifically, we treat messages from different agents at the same timestep as positive pairs and messages with intervals larger than  $\text{diameter}(\mathcal{G}^t)$  as negative pairs, encouraging local messages  $m_i^t$  to reflect the current global latent state. The corresponding auxiliary loss is given as the InfoNCE loss (Oord et al., 2018):

$$\mathcal{L}_{\text{cont}}^{\text{Aux}}(\theta) = -\mathbb{E}_{i,j,t,t'} \left[ \log \frac{\exp(g(m_i^t) \cdot g(m_j^t)/\tau)}{\sum_{m \in \mathcal{M}} \exp(g(m_i^t) \cdot g(m)/\tau)} \right], \quad (5)$$

where  $i$  is uniformly sampled from  $\{0, \dots, N\}$ ,  $j$  is uniformly sampled from  $\{0, \dots, N : j \neq i\}$ ,  $\mathcal{M} = \{m_k^{t'} : k \in \{0, \dots, N\}, t' \notin [t - \text{diameter}(\mathcal{G}^t), t + \text{diameter}(\mathcal{G}^t)]\} \cup \{m_j^t\}$  with  $|\mathcal{M}| = M + 1$  and  $m$  is uniformly sampled from  $\mathcal{M}$ .  $g(\cdot)$  is the normalization function,  $M$  is the hyperparameter indicating the number of negative pairs and  $\tau$  is the temperature hyperparameter. The overall training loss is:

$$\mathcal{L}^{\text{TD}}(\theta) = \mathcal{L}^{\text{TD}}(\theta) + \alpha \cdot \mathcal{L}^{\text{Aux}}(\theta; \phi), \quad (6)$$

where  $\alpha$  is the hyperparameter and  $\mathcal{L}^{\text{Aux}}(\cdot)$  is the auxiliary loss defined by Equation (4) or Equation (5), depending on whether global information is available during training. The training and execution procedures are summarized in Algorithm 1.

## 4 EXPERIMENTAL RESULTS

In this section, we evaluate ExpoComm on two large-scale multi-agent benchmarks: MAgent (Zheng et al., 2018) and Infrastructure Management Planning (IMP) (Leroy et al., 2024). All experiments are averaged over five random seeds and the shaded areas represent the 95% confidence interval. Details on network architecture and the training hyperparameters are available in Appendix B.1.

### 4.1 EXPERIMENTAL SETUPS

**Environment descriptions** In this section, We test ExpoComm and baselines across twelve scenarios in two large-scale benchmarks, with the number of agents ranging from 20 to 100. Specifically, MAgent is a particle-based gridworld environment representative of the typical MARL gaming benchmarks. To expand the variety of tasks, we also include the IMP benchmark, with tasks oriented from real-world applications. More details regarding the environment settings are provided in Appendix B.2.

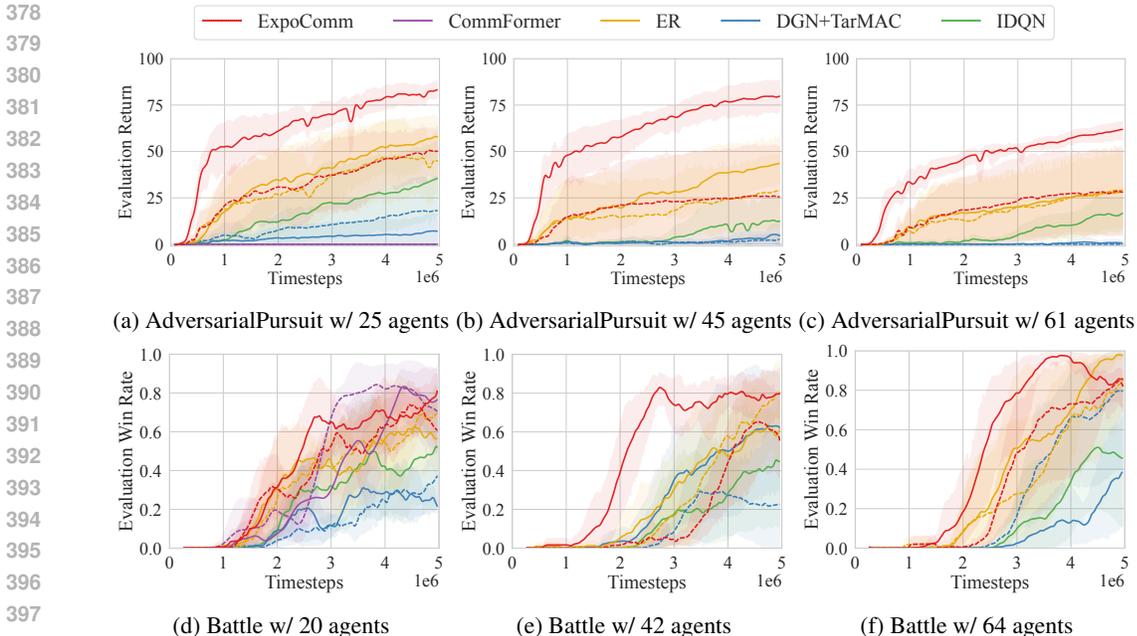


Figure 4: Performance comparison with baselines on MAgent tasks. Solid lines represent communication budgets of  $K = 1$ , while dashed lines represent budgets of  $K = \lceil \log_2 N \rceil$ . Runs requiring more than 40 GB of GPU memory are excluded due to extreme training costs compared to other methods.

**Communication Budgets** Denoting the number of agents each agent communicates to by  $K$ , for each baseline in each scenario, we test two communication budgets:  $K = \lceil \log_2 N \rceil$  and  $K = 1$ , where  $N$  is the number of agents in the systems.

**Baselines** In the following, we compare our proposed ExpoComm with four baselines: (i) *IDQN/QMIX* (Rashid et al., 2020): Base algorithms without communication; (ii) *DGN+TarMAC* (Jiang et al., 2020; Das et al., 2019): Position-based communication topologies in which agents communicate with their nearest neighbors and use TarMAC structure to aggregate messages; (iii) *ER*: ExpoComm with the exponential graph topologies replaced by random graph communication topologies following the Erdős–Rényi model; (iv) *CommFormer* (Hu et al., 2024): Learned communication topologies using GNN. For ExpoComm, we use the static exponential graph variant for  $K = \lceil \log_2 N \rceil$  and the one-peer exponential graph variant for  $K = 1$ . For DGN+TarMAC, agents communicate to top- $K$  nearest neighbors. For ER, communication graphs are sampled uniformly from all the  $K$ -in-regular directed graphs. CommFormer uses constraints with varying sparsity levels for different communication budgets. Official implementations of these baselines are utilized wherever available; otherwise, we closely follow the descriptions from their respective papers, integrating them into the base algorithms. More implementation details can be found in Appendix B.3.

## 4.2 RESULTS

**Benchmark results** We present the comparative performance of ExpoComm and baselines in MAgent and IMP environments with Figure 4 and Table 1, respectively. Overall, ExpoComm demonstrates superior performance in these large-scale benchmarks under both communication budgets, underscoring the scalability and robustness of ExpoComm strategies. Notably, the one-peer version of ExpoComm achieves the best performance in most scenarios, despite communication costs that only grow linearly with the number of agents. This makes it the most suitable method for handling large-scale MARL communication problems under very low communication budgets. [Additional visualization results to illustrate the learned policies are provided in Appendix C.1.](#)

**Zero-shot transfer** Similar to the experimental settings suggested by Wang et al. (2022), we test the zero-shot transfer ability of our proposed ExpoComm and the baseline methods, reporting the



Table 1: Performance comparison with baselines on IMP tasks. Results are reported as the mean and standard deviation of the percentage of normalized discounted rewards relative to expert-based heuristic policies, following Leroy et al. (2024), with details in Appendix B.2. The best-performing method is indicated in **bold**, and the second best is underlined.

Scenario	QMIX	ER		ExpoComm	
	$K = 0$	$K = 1$	$K = \lceil \log_2 N \rceil$	$K = 1$	$K = \lceil \log_2 N \rceil$
$N = 50$					
Uncorrelated	26.42 (3.43)	24.91 (3.77)	26.62 (2.03)	<u>27.31</u> (2.26)	<b>28.26</b> (2.51)
Correlated	24.81 (4.16)	34.63 (9.72)	34.76 (5.07)	<b>43.82</b> (6.33)	<u>40.01</u> (3.19)
OWF	62.45 (3.46)	62.99 (3.02)	61.70 (4.62)	<u>64.66</u> (0.26)	<b>65.19</b> (0.51)
$N = 100$					
Uncorrelated	12.86 (6.88)	21.94 (5.97)	18.36 (12.92)	<u>27.34</u> (13.32)	<b>27.81</b> (5.71)
Correlated	-40.20 (96.35)	-65.14 (65.08)	9.84 (32.27)	<b>19.17</b> (23.94)	<u>17.25</u> (22.70)
OWF	65.55 (0.53)	<b>66.70</b> (0.50)	65.92 (0.87)	65.26 (1.34)	<u>66.23</u> (0.38)

<sup>1</sup> DGN+TarMAC is not suitable for this benchmark because it requires the physical positions of agents, which are not available in this environment.

<sup>2</sup> Methods that require more than 40 GB GPU memory are excluded from comparison due to the extreme training cost compared to other methods.

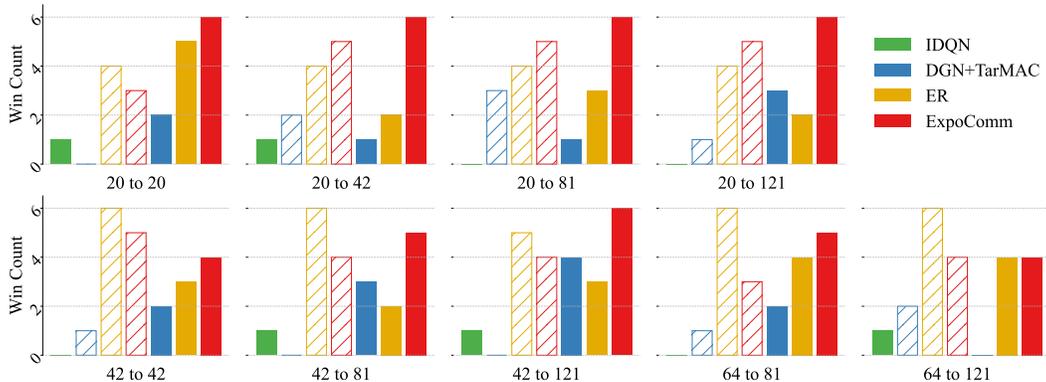


Figure 5: Zero-shot transfer results on Battle scenario. The subtitle “X to Y” indicates that methods are trained with X agents and tested with Y agents. Filled bars represent communication budgets of  $K = \lceil \log_2 N \rceil$ , while hatched bars represent budgets of  $K = 1$ . Baseline CommFormer is not excluded in this experiment because it learns a fixed peer-to-peer communication topology among agents in a specific scenario and it is non-trivial to transfer such topology to scenarios with different numbers of agents.

results in Figure 5. Specifically, we train the agent policies and their corresponding communication policies in scenarios with smaller numbers of agents and directly test these policies against each other in larger agent scenarios in the competitive task Battle. We test each pair of methods over 200 games, record the method with more wins as the winner, and summarize the results in Figure 5. We observe that both ER and ExpoComm demonstrate good transfer ability compared to other baselines, with ExpoComm performing better under smaller communication budgets. The superior transfer ability of ER and ExpoComm may be attributed to the grounding of messages, which reflects global information.

**Ablation studies** We conduct ablation studies to assess the impact of various design elements in ExpoComm, with results presented in Figure 6. In particular, we compare ExpoComm with two ablations: (i) *ExpoComm w/o mem*, in which the message generators are not memory-based as described in Section 3.2. (ii) *ExpoComm w/o aux*, which lacks the auxiliary loss term described in Section 3.3. From the results, we see that removing the memory blocks from the message generators hinders effective message generation, especially in scenarios with strong time correlation such as

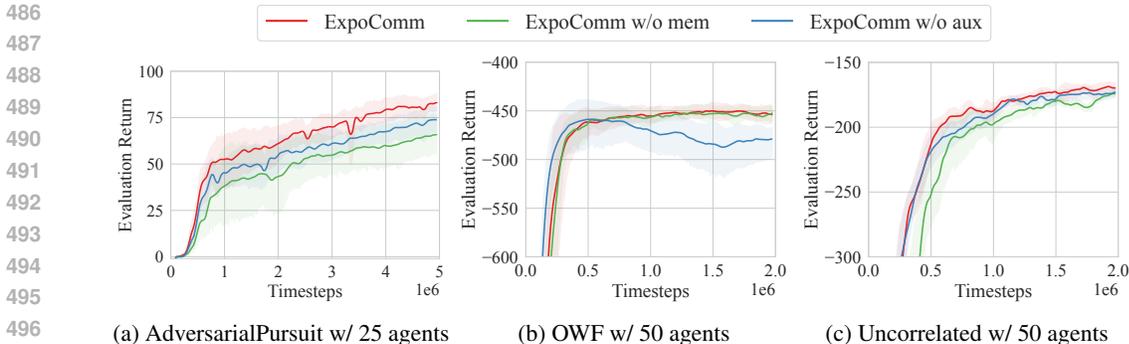


Figure 6: Ablation studies on MAgent and IMP benchmarks.

AdversarialPursuit. Auxiliary tasks primarily aid in grounding the messages, without which the messages could lack guidance and even be detrimental to decision-making.

**Discussion** As discussed in Section 2, existing methods suitable for large-scale multi-agent communication fall into two categories: position-based methods (e.g., DGN+TarMAC) and GNN-based methods (e.g., CommFormer). We propose a third category based on specific graph structures. In this category, ExpoComm utilizes exponential topologies, and we construct the baseline ER using Erdős–Rényi random graphs. Based on our analysis and experiments, the advantages of ExpoComm are as follows:

- **Superior task performance:** Due to the small diameter of exponential graphs and memory-based message generators, ExpoComm facilitates fast message dissemination among all agents. It efficiently collects and carries local information from all agents, aiding decentralized decision-making. This advantage is supported by experiments shown in Figure 4 and Table 1.
- **Low communication costs:** The compact size of exponential graphs ensures that ExpoComm’s communication costs scale (near-)linearly with the number of agents  $N$ , crucial for managing communication costs in multi-agent systems. Unlike position-based methods or ER, which can only control the number of in-edges or out-edges without global scheduling, ExpoComm naturally balances communication overhead across agents.
- **Versatile adaptability:** ExpoComm shows strong transferability across different numbers of agents, as seen in Figure 5. This is due to its global message dissemination strategies, which focus on overall communication rather than pairwise relationships, allowing it to adapt to more agents. Additionally, ExpoComm handles a wide range of tasks, regardless of the task nature or agent count. Unlike position-based methods, which may struggle with non-gridworld tasks like IMP due to assumptions about knowledge of agent locations, ExpoComm makes no such assumptions. Moreover, while learning effective pairwise communication topologies using GNNs can lead to significant GPU memory consumption, ExpoComm bypasses these challenges. It does not rely on the expensive task of learning a scenario-specific communication topology guided by the MARL task itself but instead uses a well-designed rule-based topology based on the communication desiderata analyzed in Section 3.1.2.

## 5 CONCLUSIONS

In this work, we explored scalable communication strategies in MARL and introduced ExpoComm, an exponential topology-enabled communication protocol. We proposed a framework with communication topologies featuring small diameters for fast information dissemination and small graph sizes for low communication overhead. This framework is complemented by memory-based message processors and message grounding through auxiliary objectives to achieve effective global information representation. Despite requiring only (near-)linear communication costs relative to the number of agents, ExpoComm demonstrated superior performance and strong transferability on large-scale benchmarks like MAgent and IMP. This study highlights the potential for enhancing the scalability of MARL communication strategies through the explicit design of communication topologies.

## 540 REPRODUCIBILITY STATEMENT

541

542 Method and implementation details are provided in Section 3, Appendix B.1, and Appendix B.3.  
 543 Experiment settings and details are described in Section 4.1 and Appendix B.2. Information about  
 544 the experimental infrastructure is available in Appendix B.4. The code will be released publicly  
 545 after the review process.

546

## 547 REFERENCES

548

549 Johannes Ackermann, Volker Gabler, Takayuki Osa, and Masashi Sugiyama. Reducing over-  
 550 estimation bias in multi-agent domains using double centralized critics. *arXiv preprint*  
 551 *arXiv:1910.01465*, 2019.

552 Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push  
 553 for distributed deep learning. In *Proceedings of the 36th International Conference on Machine*  
 554 *Learning*, pp. 344–353. PMLR, 2019.

555 Tianshu Chu, Sandeep Chinchali, and Sachin Katti. Multi-agent reinforcement learning for net-  
 556 worked system control. In *Proceedings of the 8th International Conference on Learning Repre-*  
 557 *sentations*, 2020.

559 Kai Cui, Anam Tahir, Gizem Ekinci, Ahmed Elshamhory, Yannick Eich, Mengguang Li, and  
 560 Heinz Koeppl. A survey on large-population systems and scalable multi-agent reinforcement  
 561 learning. *arXiv preprint arXiv:2209.03859*, 2022.

562 Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and  
 563 Joelle Pineau. Tarmac: Targeted multi-agent communication. In *Proceedings of the 36th Interna-*  
 564 *tional Conference on Machine Learning*, pp. 1538–1546. PMLR, 2019.

566 Ziluo Ding, Tiejun Huang, and Zongqing Lu. Learning individually inferred communication for  
 567 multi-agent cooperation. In *Advances in Neural Information Processing Systems*, volume 33, pp.  
 568 22069–22079, 2020.

569 Paul Erdos, Alfréd Rényi, et al. On the evolution of random graphs. *Publ. math. inst. hung. acad.*  
 570 *sci*, 5(1):17–60, 1960.

572 Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to  
 573 communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information*  
 574 *Processing Systems*, volume 29, 2016.

575 Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson.  
 576 Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial*  
 577 *Intelligence*, volume 32, 2018.

579 Benjamin Freed, Guillaume Sartoretti, Jiaheng Hu, and Howie Choset. Communication learning via  
 580 backpropagation in discrete channels with unknown noise. In *Proceedings of the AAAI conference*  
 581 *on artificial intelligence*, volume 34, pp. 7160–7168, 2020.

582 Cong Guan, Feng Chen, Lei Yuan, Chenghe Wang, Hao Yin, Zongzhang Zhang, and Yang Yu.  
 583 Efficient multi-agent communication via self-supervised information aggregation. In *Advances in*  
 584 *Neural Information Processing Systems*, volume 35, pp. 1020–1033, 2022.

585 Guangzheng Hu, Yuanheng Zhu, Dongbin Zhao, Mengchen Zhao, and Jianye Hao. Event-triggered  
 586 communication network with limited-bandwidth constraint for multi-agent reinforcement learn-  
 587 ing. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

589 Shengchao Hu, Li Shen, Ya Zhang, and Dacheng Tao. Learning multi-agent communication from  
 590 graph modeling perspective. In *Proceedings of the 12th International Conference on Learning*  
 591 *Representations*, 2024.

592 Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation.  
 593 In *Advances in Neural Information Processing Systems*, volume 31, 2018.

- 594 Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. Graph convolutional reinforcement  
595 learning. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.  
596
- 597 Daewoo Kim, Sangwoo Moon, David Hostallero, Wan Ju Kang, Taeyoung Lee, Kyunghwan Son,  
598 and Yung Yi. Learning to schedule communication in multi-agent reinforcement learning. In  
599 *Proceedings of the 7th International Conference on Learning Representations*, 2019.
- 600 Lingjing Kong, Tao Lin, Anastasia Koloskova, Martin Jaggi, and Sebastian Stich. Consensus control  
601 for decentralized deep learning. In *Proceedings of the 38th International Conference on Machine*  
602 *Learning*, pp. 5686–5696, 2021.
- 603 Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for  
604 decentralized planning. *Neurocomputing*, 190:82–94, 2016.  
605
- 606 Pascal Leroy, Pablo G Morato, Jonathan Pisane, Athanasios Kolios, and Damien Ernst. IMP-MARL:  
607 a suite of environments for large-scale infrastructure management planning via MARL. In *Ad-*  
608 *vances in Neural Information Processing Systems*, volume 36, 2024.
- 609 Xinran Li and Jun Zhang. Context-aware communication for multi-agent reinforcement learning. In  
610 *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*,  
611 pp. 1156–1164, 2024.  
612
- 613 Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. Multi-agent game  
614 abstraction via graph attention neural network. In *Proceedings of the AAAI conference on artificial*  
615 *intelligence*, volume 34, pp. 7211–7218, 2020.
- 616 Yat Long Lo, Biswa Sengupta, Jakob Nicolaus Foerster, and Michael Noukhovitch. Learning multi-  
617 agent communication with contrastive learning. In *Proceedings of the 12th International Confer-*  
618 *ence on Learning Representations*, 2024.
- 619 Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-  
620 critic for mixed cooperative-competitive environments. In *Advances in Neural Information Pro-*  
621 *cessing Systems*, volume 30, 2017.  
622
- 623 Xueguang Lyu, Yuchen Xiao, Brett Daley, and Christopher Amato. Contrasting centralized and de-  
624 centralized critics in multi-agent reinforcement learning. In *Proceedings of the 20th International*  
625 *Conference on Autonomous Agents and Multiagent Systems*, pp. 844–852, 2021.
- 626 Chengdong Ma, Aming Li, Yali Du, Hao Dong, and Yaodong Yang. Efficient and scalable rein-  
627 forcement learning for large-scale network control. *Nature Machine Intelligence*, pp. 1–15, 2024.  
628
- 629 Yaru Niu, Rohan R Paleja, and Matthew C Gombolay. Multi-agent graph-attention communication  
630 and teaming. In *Proceedings of the 20th International Conference on Autonomous Agents and*  
631 *MultiAgent Systems*, volume 21, pp. 964–97, 2021.
- 632 Frans A Oliehoek and Christopher Amato. *A concise introduction to decentralized POMDPs*.  
633 Springer, 2016.  
634
- 635 Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predic-  
636 tive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- 637 Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. Benchmarking  
638 multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Proceedings of the*  
639 *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. URL <http://arxiv.org/abs/2006.07869>.
- 640  
641 Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr,  
642 Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gra-  
643 dients. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12208–12221,  
644 2021.  
645
- 646 Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang.  
647 Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning  
to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

- 648 Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster,  
649 and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement  
650 learning. *The Journal of Machine Learning Research*, 21(1):7234–7284, 2020.
- 651 Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas  
652 Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon White-  
653 son. The starcraft multi-agent challenge. In *Proceedings of the 18th International Conference on*  
654 *Autonomous Agents and MultiAgent Systems*, pp. 2186–2188, 2019.
- 655 Lukas M Schmidt, Johanna Brosig, Axel Plinge, Bjoern M Eskofier, and Christopher Mutschler. An  
656 introduction to multi-agent reinforcement learning and review of its application to autonomous  
657 mobility. In *IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*,  
658 pp. 1342–1349. IEEE, 2022.
- 659 Sven Seuken and Shlomo Zilberstein. Improved memory-bounded dynamic programming for de-  
660 centralized pomdps. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial*  
661 *Intelligence*, pp. 344–351, 2007.
- 662 Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in  
663 multiagent cooperative and competitive tasks. In *Proceedings of the 7th International Conference*  
664 *on Learning Representations*, 2019.
- 665 Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropaga-  
666 tion. In *Advances in Neural Information Processing Systems*, volume 29, 2016.
- 667 Gokul Swamy, Siddharth Reddy, Sergey Levine, and Anca D Dragan. Scaled autonomy: Enabling  
668 human operators to control robot fleets. In *2020 IEEE International Conference on Robotics and*  
669 *Automation*, pp. 5942–5948. IEEE, 2020.
- 670 Jordan K Terry, Benjamin Black, and Mario Jayakumar. Magent. [https://github.com/](https://github.com/Farama-Foundation/MAgent)  
671 [Farama-Foundation/MAgent](https://github.com/Farama-Foundation/MAgent), 2020. GitHub repository.
- 672 Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael G. Rabbat. Slowmo: Improving  
673 communication-efficient distributed SGD with slow momentum. In *Proceedings of 8th Inter-*  
674 *national Conference on Learning Representations*, 2020a.
- 675 Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. Learning nearly decom-  
676 posable value functions via communication minimization. In *Proceedings of the International*  
677 *Conference on Learning Representations*, 2020b.
- 678 Xiaolu Wang, Huaxi Gu, Yintang Yang, Kun Wang, and Qinfen Hao. RPNOC: A ring-based packet-  
679 switched optical network-on-chip. *IEEE Photonics Technology Letters*, 27(4):423–426, 2015.
- 680 Xiaolu Wang, Huaxi Gu, Yintang Yang, Kun Wang, and Qinfen Hao. A highly scalable optical  
681 network-on-chip with small network diameter and deadlock freedom. *IEEE Transactions on Very*  
682 *Large Scale Integration (VLSI) Systems*, 24(12):3424–3436, 2016.
- 683 Yuanfei Wang, Fangwei Zhong, Jing Xu, and Yizhou Wang. ToM2C: Target-oriented multi-agent  
684 communication and cooperation with theory of mind. In *Proceedings of the 10th International*  
685 *Conference on Learning Representations*, 2022.
- 686 Jannis Weil, Zhenghua Bao, Osama Abboud, and Tobias Meuser. Towards generalizability of multi-  
687 agent reinforcement learning in graphs with recurrent message passing. In *Proceedings of the*  
688 *23rd International Conference on Autonomous Agents and Multiagent Systems*, pp. 1919–1927,  
689 2024.
- 690 Xianliang Yang, Zhihao Liu, Wei Jiang, Chuheng Zhang, Li Zhao, Lei Song, and Jiang Bian. A ver-  
691 satile multi-agent reinforcement learning benchmark for inventory management. *arXiv preprint*  
692 *arXiv:2306.07542*, 2023.
- 693 Bicheng Ying, Kun Yuan, Yiming Chen, Hanbin Hu, Pan Pan, and Wotao Yin. Exponential graph is  
694 provably efficient for decentralized deep training. In *Advances in Neural Information Processing*  
695 *Systems*, volume 34, pp. 13975–13987, 2021.

- 702 Wang Ying and Sang Dayong. Multi-agent framework for third party logistics in E-commerce.  
703 *Expert Systems with Applications*, 29(2):431–436, 2005.  
704
- 705 Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu.  
706 The surprising effectiveness of ppo in cooperative multi-agent games. In *Advances in Neural*  
707 *Information Processing Systems*, volume 35, pp. 24611–24624, 2022.
- 708 Kun Yuan, Yiming Chen, Xinmeng Huang, Yingya Zhang, Pan Pan, Yinghui Xu, and Wotao Yin.  
709 Decentlam: Decentralized momentum sgd for large-batch deep training. In *Proceedings of the*  
710 *IEEE/CVF International Conference on Computer Vision*, pp. 3029–3039, 2021.  
711
- 712 Lei Yuan, Jianhao Wang, Fuxiang Zhang, Chenghe Wang, Zongzhang Zhang, Yang Yu, and  
713 Chongjie Zhang. Multi-agent incentive communication via decentralized teammate modeling.  
714 In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 9466–9474,  
715 2022.
- 716 Tingting Yuan, Hwei-Ming Chung, Jie Yuan, and Xiaoming Fu. Dacom: Learning delay-aware  
717 communication for multi-agent reinforcement learning. In *Proceedings of the AAAI Conference*  
718 *on Artificial Intelligence*, volume 37, pp. 11763–11771, 2023.
- 719 Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Fully decentralized multi-  
720 agent reinforcement learning with networked agents. In *Proceedings of the 35th International*  
721 *Conference on Machine Learning*, pp. 5872–5881. PMLR, 2018.  
722
- 723 Sai Qian Zhang, Qi Zhang, and Jieyu Lin. Succinct and robust multi-agent communication with  
724 temporal message control. In *Advances in Neural Information Processing Systems*, volume 33,  
725 pp. 17271–17282, 2020.
- 726 Lianmin Zheng, Jiacheng Yang, Han Cai, Ming Zhou, Weinan Zhang, Jun Wang, and Yong Yu.  
727 Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In  
728 *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.  
729
- 730 Ming Zhou, Jun Luo, Julian Villella, Yaodong Yang, David Rusu, Jiayu Miao, Weinan Zhang, Mont-  
731 gomery Alban, Iman Fadakar, Zheng Chen, et al. Smarts: An open-source scalable multi-agent  
732 RL training school for autonomous driving. In *Proceedings of the Conference on Robot Learning*,  
733 pp. 264–285. PMLR, 2021.
- 734 Changxi Zhu, Mehdi Dastani, and Shihan Wang. A survey of multi-agent deep reinforcement learn-  
735 ing with communication. In Mehdi Dastani, Jaime Simão Sichman, Natasha Alechina, and Vir-  
736 ginia Dignum (eds.), *Proceedings of the 23rd International Conference on Autonomous Agents*  
737 *and MultiAgent Systems*, pp. 2845–2847, 2024.  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755

## A THEORETICAL ANALYSIS

In this section, we analyze the communication effect of exponential topologies.

**Theorem 1.** Suppose  $E_{ij}^t$  is defined by Equation (2) and let  $\tau = \lceil \log_2(N - 1) \rceil$ . Then, the following holds:

$$E_{ij}^t \times_b E_{ij}^{t+1} \times_b \dots E_{ij}^{t+\tau-1} = \mathbb{1}\mathbb{1}^T, \quad (7)$$

where  $\times_b$  denotes logical (Boolean) matrix multiplication.

**Remark 1.** If the information at each agent remains valid within  $\tau$  timesteps and there is no information loss during aggregation, the one-peer exponential topology ensures information exchange between any two agents in the system with  $\tau$  timesteps.

**Remark 2.** Static exponential topologies exhibit a similar communication effect as described in Theorem 1. Specifically,  $\forall i, j$  that  $E_{ij}^{t(\text{one-peer})} = 1$ , it holds that  $E_{ij}^{t(\text{stat})} = 1$ .

*Proof.*

**Lemma 1.** Define the function  $Z(x)$  for non-negative  $x$  as

$$Z(x) = \begin{cases} 1 & , x > 0 \\ 0 & , x = 0 \end{cases}. \quad (8)$$

Then, the following equivalence holds:

$$xy + wv = 0 \iff (Z(x) \times_b Z(y)) +_b (Z(u) \times_b Z(v)) = 0, \quad (9)$$

where  $\times_b$  denotes logical (Boolean) And, and  $+_b$  denotes logical (Boolean) Or.

Now, consider the connection between  $Z(x)$  and the structure of an all-ones matrix. For a non-negative matrix  $X$ , it holds that  $X_{ij} \in \mathbb{R}^+, \forall i, j \iff Z(X) = \mathbb{1}\mathbb{1}^T$ .

Therefore, by applying Lemma 1 to  $E_{ij}^t E_{ij}^{t+1} \dots E_{ij}^{t+\tau-1} = \frac{2^\tau}{N} \mathbb{1}\mathbb{1}^T$  (Ying et al., 2021), we have  $E_{ij}^t \times_b E_{ij}^{t+1} \times_b \dots E_{ij}^{t+\tau-1} = \mathbb{1}\mathbb{1}^T$ .  $\square$

## B EXPERIMENT DETAILS

### B.1 NETWORK ARCHITECTURE AND HYPERPARAMETERS

**Codebase** Our implementation of ExpoComm and baseline algorithms is based on the following codebase:

- EPyMARRL (Papoudakis et al., 2021): <https://github.com/uoel-agents/epymar1>
- CommFormer Hu et al. (2024): <https://github.com/charleshsc/CommFormer>
- CommNet (Sukhbaatar et al., 2016): <https://github.com/ispltze/MAProj>

The code for ExpoComm will be open-sourced upon publication of the paper.

**Neural network architecture** Following previous work Papoudakis et al. (2021), we employ deep neural networks consisting of multilayer perceptrons (MLPs) with rectified linear unit (ReLU) activation functions and gated recurrent units (GRUs) to parameterize the agent networks. In ExpoComm, the message memory blocks described in Section 3.2 are implemented using a single GRU or an attention block. **The prediction network  $f(\cdot; \phi)$  described in Section 3.3 is implemented using a two-layer MLP.**

**Hyperparameters** To ensure a fair comparison, we implement our method and self-constructed baselines using the same codebase with the same set of hyperparameters, with the exception of method-specific ones and the learning rate. In general, we follow the common settings provided by Papoudakis et al. (2021) for MAgent benchmark and adopt the settings in IMP paper (Leroy et al., 2024) for the IMP benchmark. The common hyperparameters are listed in Table 2. The ExpoComm-specific hyperparameters are provided in Table 3. For learning rate, we search among (0.0001, 0.0005) for ExpoComm and baselines. We use the value of 0.0005 for base algorithms without communication; 0.0005 for DGN+TarMAC in MAgent and 0.0001 for DGN+TarMAC in IMP; 0.0001 for ExpoComm in IMP with 50 agents and 0.0001 for other scenarios. For CommFormer, we adopt the optimal value in its official implementation.

Table 2: Common hyperparameters.

Hyperparameter	Benchmark	Value
Hidden sizes	-	64
Discount factor $\gamma$	MAgent	0.99
	IMP	0.95
Batch size	MAgent	32
	IMP	64
Replay buffer size	-	2000
Number of environment steps	MAgent	$5 \times 10^6$
	IMP	$2 \times 10^6$
Epsilon anneal steps	MAgent	$5 \times 10^5$
	IMP	$5 \times 10^3$
Test interval steps	MAgent	$5 \times 10^4$
	IMP	$2.5 \times 10^4$
Number of test episode	-	100

Table 3: Hyperparameters used for ExpoComm.

Hyperparameter	Value
Auxillary loss coefficient $\alpha$	0.1
Temperature $\tau$	0.07
Number of negative pairs $M$	20

## B.2 ENVIRONMENTAL DETAILS

**Codebase** The environments used in this work are listed below with descriptions in Table 4.

- MAgent (Zheng et al., 2018; Terry et al., 2020): <https://github.com/Farama-Foundation/MAgent2>
- IMP (Leroy et al., 2024): [https://github.com/moratodpg/imp\\_marl](https://github.com/moratodpg/imp_marl)

**MAgent** MAgent is a highly scalable gridworld gaming benchmark shown in Figure 7. In AdversarialPursuit, agents aim to tag adversaries while adversaries try to escape. Agents can choose actions from move, tag or do nothing. Agents are rewarded for successfully tagging an adversary and penalized for unsuccessful tagging attempts. In Battle, agents attempt to attack and eliminate adversaries, with the same goal for adversaries. Agents can choose actions from move, attack or do nothing. A team wins by eliminating all opponents or having more surviving agents when the episode ends.

Following the official implementation (Terry et al., 2020), we use an individual reward setting with IDQN as the base algorithm. Due to the high-dimensional observations in MAgent, storing experience in a replay buffer can be challenging because of hardware constraints. We adopt a preprocessing procedure following Jiang et al. (2020), compressing observations by concatenating [my\_team\_hp -

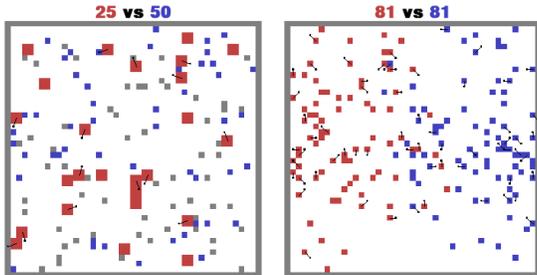


Table 4: Environments details.

Environment	Scenarios	Number of agents
MAgent	Adversarial Pursuit	(25, 45, 61) <sup>1</sup>
	Battle	(20, 42, 64) <sup>2</sup>
IMP	Uncorrelated: uncorrelated k-out-of-n; campaign cost	(50, 100)
	Correlated: correlated k-out-of-n; campaign cost	(50, 100)
	OWF: offshore wind farm; campaign cost	(50, 100)

<sup>1</sup> The number of agents in this scenario is determined by setting the map size to 25, 35, 40, respectively.

<sup>2</sup> The number of agents in this scenario is determined by setting the map size to 45, 60, 70, respectively.



(a) AdversarialPursuit.

(b) Battle.

Figure 7: Environments from the MAgent benchmark suite (Terry et al., 2020). In each scenario, the MARL algorithms control the red agents, while the blue adversary agents are controlled by pretrained policies.

obstacle/off the map, other\_team\_hp - obstacle/off the map]. To facilitate the use of communication, we use small view ranges (8 for AdversarialPursuit and 7 for Battle). In both scenarios, we pretrain the adversary policies using the IDQN algorithm with a self-play scheme and use these pretrained policies to test the performance of different algorithms.

**IMP** IMP is a platform for benchmarking the scalability of cooperative MARL methods in real-world engineering applications, as illustrated in Figure 8. This environment simulates an infrastructure management planning problem with agents controlling different components. Agents can choose actions from inspection, repair or do nothing. In different scenarios, the correlation between agent deterioration processes and the system failure function are defined differently, posing unique challenges.

Following the official implementation of IMP, we use a global reward setting and choose QMIX as the base algorithm due to its stable performance across scenarios. We adopt the campaign cost setting, which requires higher cooperation among agents. As recommended by Leroy et al. (2024), results are normalized with respect to expert-based heuristic policies using  $(x - H)/|H|$ , where  $x$  is the discounted rewards of the tested algorithm, and  $H$  is the discounted rewards achieved by heuristic policies listed in Table 5.

### B.3 IMPLEMENTATION DETAILS

In MAgent, we implement our proposed ExpoComm along with baselines DGN+TarMAC and ER on top of the IDQN base algorithm. In IMP, these are implemented on top of QMIX. For ExpoComm, we use Equation (4) for MAgent benchmark because the global state is provided in this environment, and Equation (5) for IMP, as the global state is a concatenation of all observations and is not compact or suitable for message grounding.

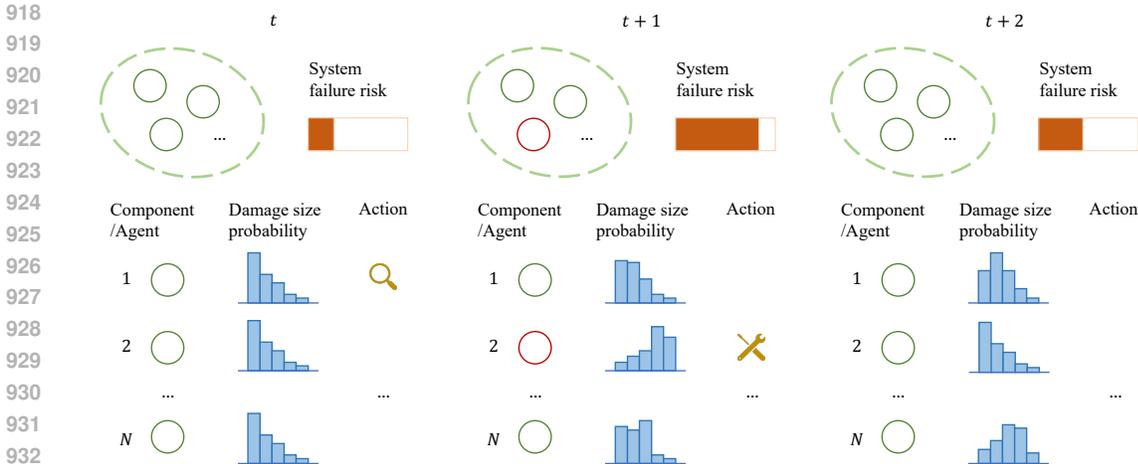


Figure 8: IMP environment (Leroy et al., 2024). This environment simulates an engineering system with multiple components controlled by agents. The objective is to minimize overall system failure risk at low costs. The system risk depends on component damage probabilities, which evolve over time and can be influenced by agent inspection or repair actions.

Table 5: Heuristic policies performance on the IMP benchmark.

Scenario	Number of agents $N$	Discounted reward $H$
Uncorrelated	50	-232.7
	100	-231.5
Correlated	50	-211.0
	100	-194.0
OWF	50	-1248.2
	100	-2436.3

#### B.4 EXPERIMENTAL INFRASTRUCTURE

The experiments were conducted using NVIDIA GeForce RTX 3080 GPUs and NVIDIA A100GPUs. Each experimental run required less than 2 days to complete.

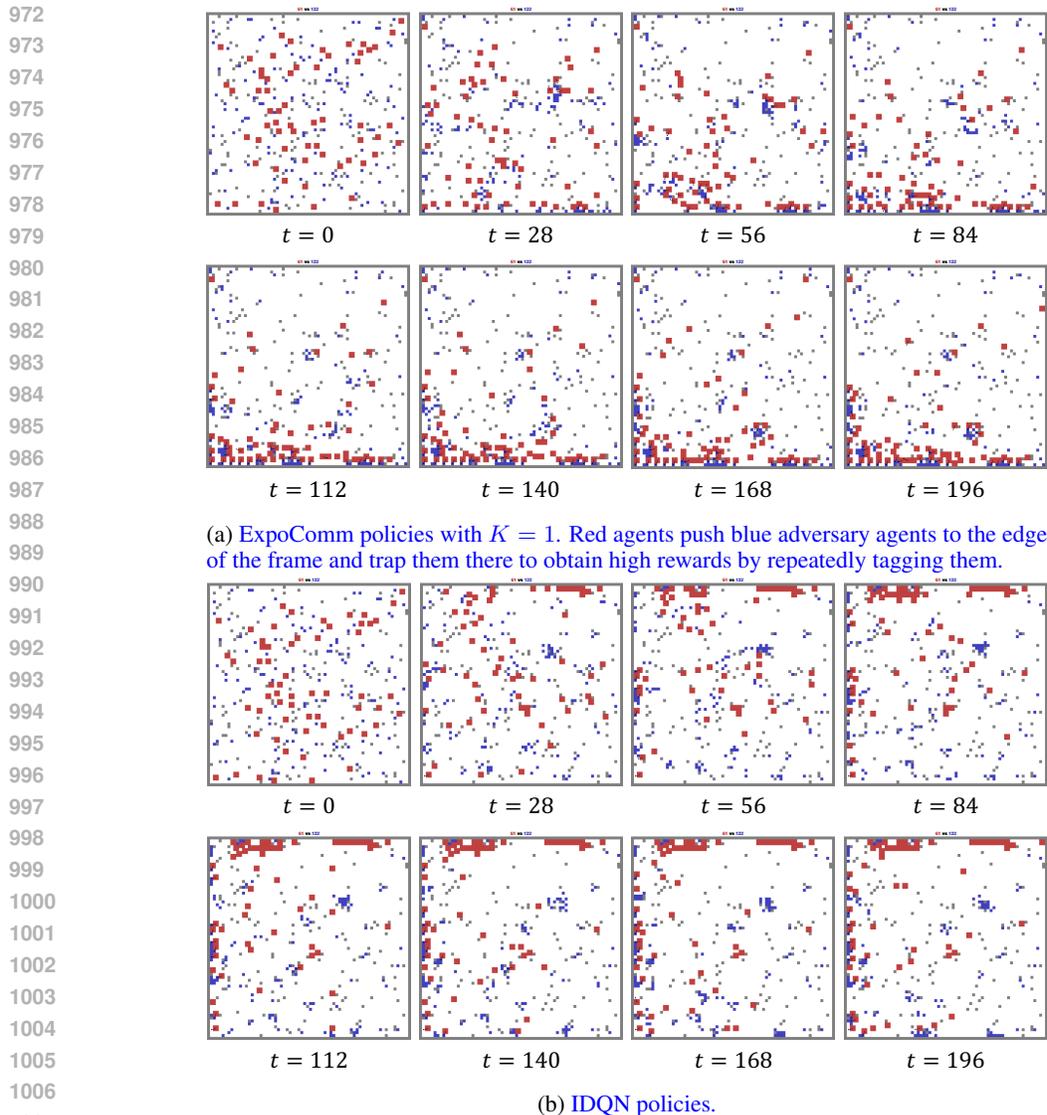
### C MORE RESULTS AND DISCUSSION

#### C.1 VISUALIZATION RESULTS

We visualize the final trained policies of ExpoComm and IDQN in `dversarialPursuit` and `Battle` with Figure 9 and Figure 10 respectively to demonstrate how ExpoComm enhances cooperation among agents. As shown in Figure 9a and Figure 10a, agents adopt a global perspective and act cooperatively with ExpoComm policies, demonstrating effectiveness even under extreme communication budgets ( $K = 1$ ). In comparison, IDQN agents focus only on local observations and often become trapped in suboptimal solutions due to lack of coordination.

#### C.2 COMPARISON WITH PROXY-BASED COMMUNICATION

Although we primarily focus on decentralized communication-based MASs without centralized proxies, we also compare ExpoComm against the proxy-based CommNet (Sukhbaatar et al., 2016). As seen in Figure 11 and Table 6, ExpoComm outperforms CommNet in most scenarios, especially in IMP benchmarks. However, CommNet achieves comparable performance on the `AdversarialPursuit` tasks. This implies that a global perspective is more crucial for success



1008 Figure 9: Visualization in AdversarialPursuit w/ 61 agents.

1009  
1010  
1011 in these scenarios, possibly explaining ExpoComm’s larger advantage over other baselines in this  
1012 scenario.

### 1013 C.3 LIMITATIONS AND FUTURE WORK

1014  
1015 While ExpoComm demonstrates strong performance and scalability in cooperative multi-agent  
1016 tasks, some limitations remain.

1017  
1018 First, ExpoComm does not explicitly incorporate agent heterogeneity or properties of the underlying  
1019 environmental MDP when constructing the communication topology. This could result in suboptimal  
1020 performance in scenarios requiring targeted messaging between specific agents (Yuan et al.,  
1021 2022) or in networked MDPs (Zhang et al., 2018; Ma et al., 2024). Therefore, Incorporating factors  
1022 like agent identities or relationships presents a promising direction for further improvements in such  
1023 settings.

1024  
1025 Second, we evaluated ExpoComm primarily in fully cooperative tasks. Partially competitive settings  
requiring agents to learn to communicate only when necessary remain challenging. Examining

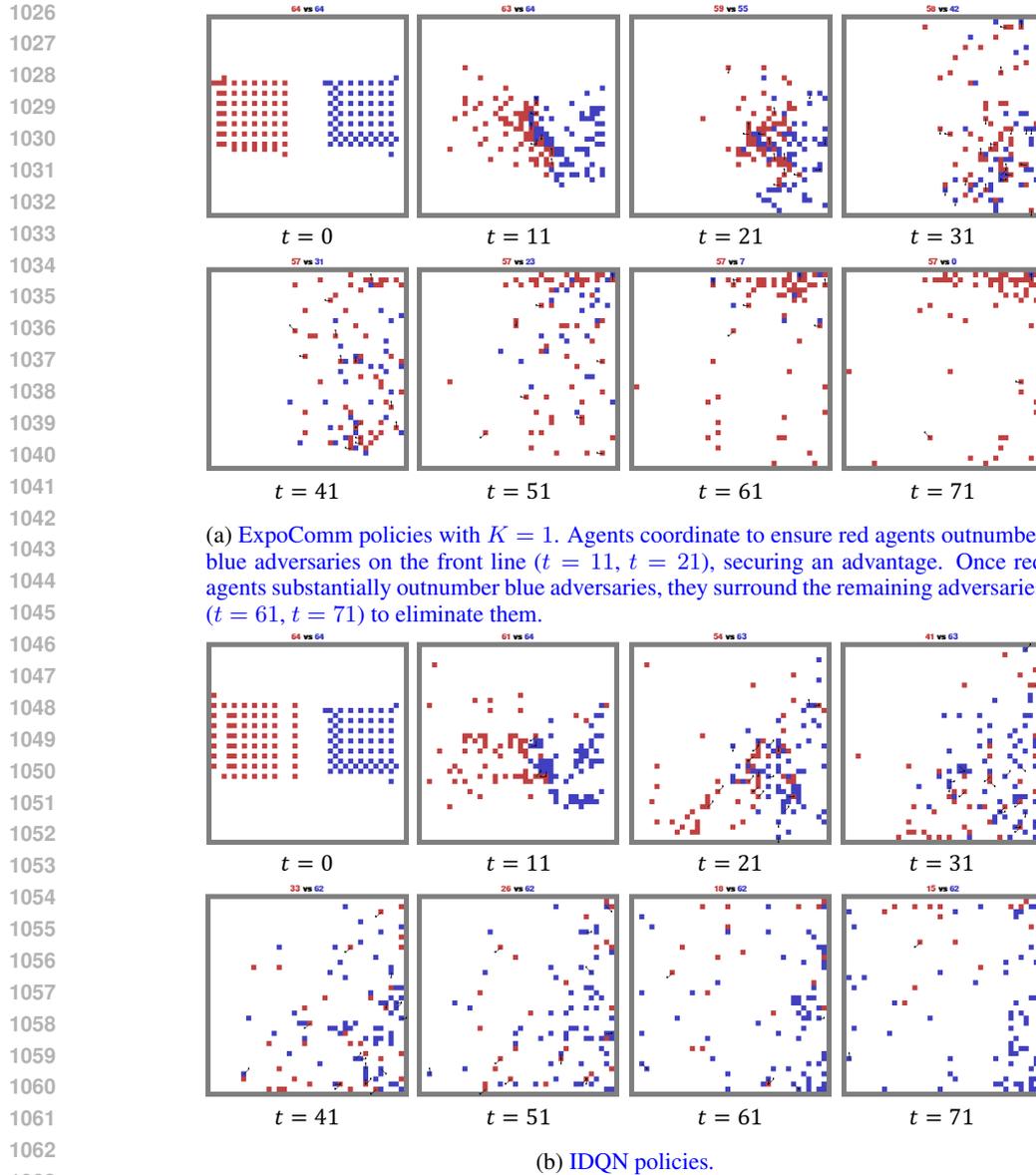


Figure 10: Visualization in Battle w/ 64 agents.

ExpoComm’s capabilities and limitations in such partially competitive tasks presents an important avenue for future work.

Finally, communication scalability in multi-agent systems remains an under-explored area despite the attempt of this work. For instance, incorporating finer graph topologies beyond exponential graphs may enhance performance, and exploiting temporal communication sparsity could further reduce costs. There are still many open questions in scaling communication efficiently.

1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133

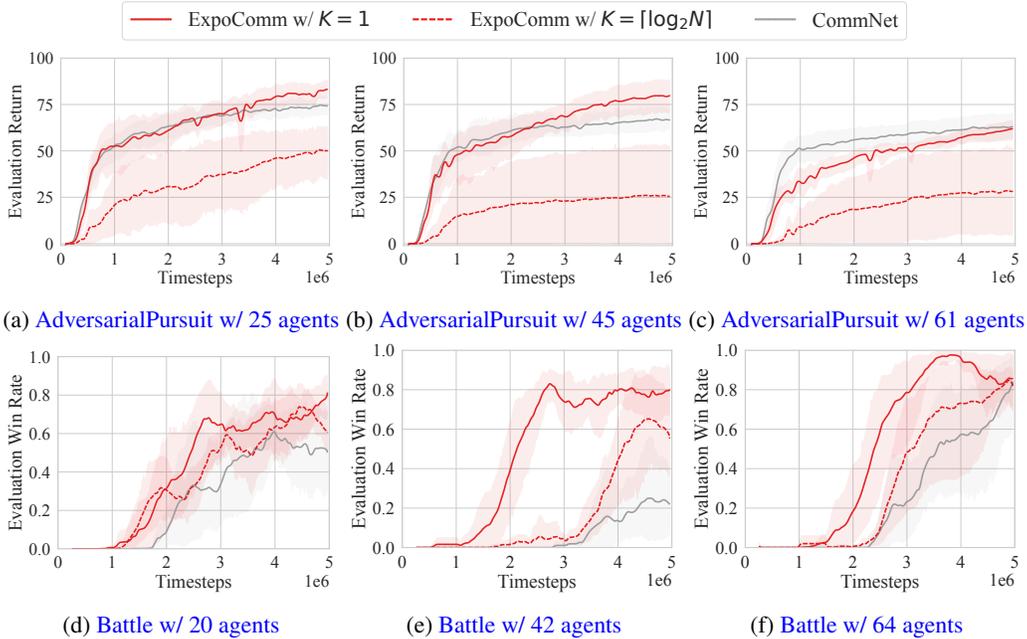


Figure 11: Performance comparison with proxy-based baselines on MAgent tasks.

Table 6: Performance comparison with proxy-based baselines on IMP tasks. Results are reported as the mean and standard deviation of the percentage of normalized discounted rewards relative to expert-based heuristic policies, following Leroy et al. (2024), with details in Appendix B.2. The best-performing method is indicated in **bold**, and the second best is underlined.

Scenario	CommNet	ExpoComm	
	with communication proxy	$K = 1$	$K = \lceil \log_2 N \rceil$
$N = 50$			
Uncorrelated	26.07 (6.82)	<u>27.31</u> (2.26)	<b>28.26</b> (2.51)
Correlated	26.14 (16.87)	<b>43.82</b> (6.33)	40.01 (3.19)
OWF	53.71 (1.27)	<u>64.66</u> (0.26)	<b>65.19</b> (0.51)
$N = 100$			
Uncorrelated	-65.92 (125.03)	<u>27.34</u> (13.32)	<b>27.81</b> (5.71)
Correlated	-82.76 (48.62)	<b>19.17</b> (23.94)	17.25 (22.70)
OWF	34.71 (5.34)	<u>65.26</u> (1.34)	<b>66.23</b> (0.38)