DISTRIBUTED MULTI-AGENT DEEP REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Centralized training with decentralized execution (CTDE) has been the dominant paradigm in multi-agent reinforcement learning (MARL), but its reliance on global state information during training introduces scalability, robustness, and generalization bottlenecks. Moreover, in practical scenarios such as adding/dropping teammates or facing environment dynamics that differ from the training, CTDE methods can be brittle and costly to retrain, whereas distributed approaches allow agents to adapt using only local information and peer-to-peer communication. We present a distributed MARL framework that removes the need for centralized critics or global information. Firstly, we develop a novel Distributed Graph Attention Network (D-GAT) that performs global state inference through multi-hop communication, where agents integrate neighbor features via input-dependent attention weights in a fully distributed manner. Leveraging D-GAT, we develop the distributed graph-attention MAPPO (DG-MAPPO) – a distributed MARL framework where agents optimize local policies and value functions using local observations, multi-hop communication, and shared/averaged rewards. Empirical evaluation on the StarCraftII Multi-Agent Challenge (SMAC) demonstrates that our method consistently outperforms strong CTDE baselines, achieving superior coordination across a wide range of cooperative tasks with both homogeneous and heterogeneous teams. Our distributed MARL framework offers a principled and scalable solution for robust collaboration without requiring centralized training or global observability. To the best of our knowledge, DG-MAPPO appears to be the first to fully eliminate reliance on privileged centralized information, enabling agents to learn and act solely through peer-to-peer communication.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has emerged as a powerful framework for training multiple agents to learn cooperative and competitive behaviors in complex dynamic environments (Zhang et al., 2021). However, learning effective collaborative policies remains challenging, since each agent simultaneously seeks to maximize its own return, giving rise to the fundamental issue of non-stationarity, i.e., the environment is constantly changing due to the evolving behaviors of other agents from the perspective of any single agent. Recent works such as MAPPO (Yu et al., 2022), MADDPG (Lowe et al., 2017), and HAPPO/HATRPO (Kuba et al., 2021; Zhong et al., 2024) alleviate this challenge by using the *centralized training decentralized execution* (CTDE) framework, where agents assume access to global state information during training but rely on local information during execution. Although effective, the CTDE approach suffers from several drawbacks that limit its applicability in practical settings. First, it requires access to global information during training, which may be infeasible in large-scale systems due to communication bandwidth, latency, or privacy constraints. Second, reliance on centralized critics introduces scalability bottlenecks and potential single points of failure. Third, CTDE methods often suffer from a train–test mismatch, since agents learn with privileged information that is unavailable at execution time, leading to suboptimal generalization. These limitations highlight the need for distributed MARL approaches that enable agents to learn cooperative strategies using only local observations and peer-to-peer communication from neighboring agents.

However, fully distributed learning techniques remain relatively underexplored compared to the centralized training approaches, partly due to the inherent complexity of the problem. Existing studies in this domain typically retain some form of centralization. For instance, Zhang et al. (2018) proposed a decentralized multi-agent actor–critic algorithms that use average consensus protocols (Tsitsiklis, 1984) to approximate global returns and value functions via neighbor communication, while actors update their policies independently. Although effective, this approach relies on simple averaging of value functions, which can yield suboptimal performance in heterogeneous teams with non-i.i.d. dynamics. Moreover, their framework assumes access to global state information for advantage estimation and thus cannot directly handle partial observability. In parallel, graph-based methods have been introduced to better capture structured communication among agents. Jiang et al. (2018) proposed graph convolution RL (DGN), where agents are represented as nodes in a dynamic graph and leverage Graph Attention Networks (GATs) (Veličković et al., 2017) to process node-level observations and actions. However, DGN shares both the Q-network and GAT parameters among all agents, preventing fully distributed training.

From an optimization perspective, decentralized stochastic gradient descent (D-SGD) (Lian et al., 2017; Assran et al., 2019) and classical distributed averaging protocols (Nedić & Ozdaglar, 2009; Tsitsiklis, 1984) provide strong theoretical foundations for consensus optimization over networks. Building on these insights, we introduce **Distributed Graph Attention Networks (D-GATs)**, which couple the expressiveness of GATv2 (Brody et al., 2021) with neighbor-averaged parameter sharing inspired by D-SGD. This design preserves dynamic, input-dependent attention while promoting consensus among agents in a fully distributed setting. While related works, such as GATTA (Tian et al., 2023), have applied graph attention to distributed supervised learning and personalization, it comes with higher computational overhead that scales poorly as the number of agents increases—making it less suitable for multi-agent reinforcement learning settings where efficiency is critical. In contrast, our Distributed Graph Attention Network (D-GAT) is designed specifically for MARL, focusing on global state inference through lightweight, input-dependent attention mechanisms that remain tractable even in large teams.

Building on D-GAT, we introduce **distributed graph-attention MAPPO (DG-MAPPO)**, a principled distributed MARL framework that removes the need for centralized training or global observability. DG-MAPPO integrates agents' local observations with global state inference from D-GAT and a shared/averaged team reward to learn collaborative policies that naturally scale to large teams. Unlike CTDE approaches, our fully distributed framework enables agents to infer global state during both training and execution, yielding more robust coordination at test time. We evaluate DG-MAPPO on the StarCraftII Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019), a widely used benchmark for cooperative MARL, against strong CTDE baselines such as MAPPO, MAT-Dec (Wen et al., 2022), and HAPPO. SMAC scenarios demand coordination under partial observability, dynamic environments, and complex team interactions, making them a rigorous testbed for scalability and robustness. Our experiments show that DG-MAPPO achieves consistently strong performance across diverse tasks, demonstrating its ability to handle both homogeneous and heterogeneous settings, scale to large teams, and learn effective collaboration without centralized training or privileged information.

Our contributions are threefold:

- We propose D-GAT, an effective communication module that enables agents to infer global state representations from purely local communication over multiple hops.
- We propose DG-MAPPO, a distributed MARL framework that learns collaborative policies using only local observations, global state inference from D-GAT, and a shared/averaged team reward. Crucially, DG-MAPPO operates entirely without any centralized mechanism.
- We demonstrate empirically that our approach outperforms prior CTDE baselines on cooperative tasks, achieving robust coordination under limited communication and no privileged information.

2 PRELIMINARIES

We first establish the necessary background in this section. We begin by formulating the cooperative **multi-agent reinforcement learning** (MARL) problem as a **decentralized partially observable**

Markov decision process (Dec-POMDP). We then describe how graph neural networks, in particular graph attention networks (GATs), can be used to model agent communication and representation learning. Finally, we review the policy gradient theorem in the multi-agent setting, which forms the foundation for our optimization framework. Throughout the paper, we denote matrices by bold uppercase letters (e.g., $\boldsymbol{X}$), vectors by bold lowercase letters (e.g., $\boldsymbol{x}$), local data with superscript i (e.g., x^i), global data without superscript (e.g., x), and approximations with a hat (e.g., $\hat{x}$).

2.1 Problem Formulation

We consider a distributed cooperative MARL problem formulated as a Dec-POMDP, represented by the tuple $\langle \mathcal{N}, \{\mathcal{O}^i\}_{i=1}^n, \{\mathcal{A}^i\}_{i=1}^n, R, P, \gamma \rangle$. Here, $\mathcal{N} = 1, \ldots, n$ is the set of agents. Each agent $i \in \mathcal{N}$ has an observation space $\mathcal{O}^i \subset \mathbb{R}^p$, where p is the observation dimension, and an action space $\mathcal{A}^i \subset \mathbb{R}^q$, where q is the action dimension. The joint observation and action spaces are $\mathcal{O} = \prod_{i=1}^n \mathcal{O}^i$ and $\mathcal{A} = \prod_{i=1}^n \mathcal{A}^i$, respectively. The transition kernel $P : \mathcal{O} \times \mathcal{A} \times \mathcal{O} \to [0, 1]$ defines the environment dynamics, $R : \mathcal{O} \times \mathcal{A} \to [-R_{\max}, R_{\max}]$ is the shared reward function, and $\gamma \in [0, 1)$ is the discount factor.

We model the multi-agent interaction structure as a dynamic graph $G = (\mathcal{N}, \mathcal{E})$, where nodes $(\mathcal{N})$ correspond to agents and edges $(\mathcal{E})$ denote available communication links which can change in real-time. At each time step t, agent i receives a local observation $\boldsymbol{o}_t^i \in \mathcal{O}^i$ $\left(\boldsymbol{o}_t = [\boldsymbol{o}_t^1, \ldots, \boldsymbol{o}_t^n]^\top\right)$, communicates with nodes $j \in \mathcal{N}^i$, where $\mathcal{N}^i$ is some neighborhood of node i (including i) in the graph G over multiple-hops, and forms a local approximation of the global observation $\hat{\boldsymbol{o}}_t^i \in \hat{\mathcal{O}}^i$. Based on this, the agent selects an action a_t^i from its policy π^i, which is the i^{th} component of the joint policy $\pi = \prod_{i=1}^n \pi^i$. The transition kernel and the joint policy induce the marginal observation distribution $\rho_\pi(\cdot) = \sum_{t=0}^\infty \gamma^t Pr(\boldsymbol{o}_t \mid \pi)$ (Wen et al., 2022). All agents then receive a shared team reward $R(\boldsymbol{o}_t, \boldsymbol{a}_t)$ and observe $\boldsymbol{o}_{t+1}^i$.

We focus on the **fully cooperative** setting where all agents share the same reward signal. Settings with local reward functions can be incorporated by computing a consensus-based team reward (e.g., via average consensus (Tsitsiklis, 1984)). The goal is to learn local policies $\{\pi^i\}_{i=1}^n$ that maximize the expected discounted team return:

$$J(\pi) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R(o_t, a_t)\right] \tag{1}$$

2.2 Graph-Attention Networks

Graph neural networks (GNNs), such as GraphSAGE (Hamilton et al., 2017), learn node representations by aggregating information from local neighborhoods in a graph. At each layer, a node updates its embedding by combining its own features with those of its neighbors, typically using simple operations such as mean or sum. While effective, this uniform treatment of neighbors may fail to capture the varying importance of different connections.

GATs (Veličković et al., 2017; Brody et al., 2021) address this limitation by incorporating an attention mechanism into the aggregation process. Instead of assigning equal weight to all neighbors, GATs learn to adaptively highlight the most relevant nodes when computing new representations. Formally, for a node i with feature vector $\boldsymbol{h}^i \in \mathbb{R}^d$ and neighborhood $\mathcal{N}^i$, GAT defines a shared attention function $e : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ to measure the importance of a neighbor j:

$$e(\boldsymbol{h}^i, \boldsymbol{h}^j) = \text{LeakyReLU}\left(\boldsymbol{q}^\top \left[\boldsymbol{W}\boldsymbol{h}^i \| \boldsymbol{W}\boldsymbol{h}^j\right]\right), \tag{2}$$

where $\boldsymbol{W} \in \mathbb{R}^{d' \times d}$ is a learnable linear transformation matrix, $\boldsymbol{q} \in \mathbb{R}^{2d'}$ is a trainable weight vector, and $\|$ denotes concatenation. The parameters $\boldsymbol{W}$ and $\boldsymbol{q}$ are shared across all nodes. These scores are normalized with a softmax across all neighbors:

$$\alpha_{ij} = \frac{\exp\left(e(\boldsymbol{h}^i, \boldsymbol{h}^j)\right)}{\sum_{j' \in \mathcal{N}^i} \exp(e(\boldsymbol{h}^i, \boldsymbol{h}^{j'}))}. \tag{3}$$

The attention coefficients α_{ij} encode the relative contribution of neighbor j to node i. The updated representation of node i is then computed as

$$\hat{\boldsymbol{h}}^i = \sigma\left(\sum_{j\in\mathcal{N}^i} \alpha_{ij}\boldsymbol{W}\boldsymbol{h}^j\right), \tag{4}$$

where $\sigma : \mathbb{R}^{d'} \to \mathbb{R}^{d'}$ is a nonlinear activation. By learning these attention weights, GATs provide a more flexible and expressive aggregation scheme than traditional GNNs, enabling the model to prioritize informative neighbors and downplay less relevant ones. In the MARL setting, this enables each agent to selectively integrate neighbor information when constructing a local approximation of the global state.

2.3 Policy Gradient Theorem for Multi-Agent Reinforcement Learning

Policy gradient methods provide a principled approach to optimizing parameterized policies by estimating the gradient of the expected return with respect to policy parameters. Extending this idea to the multi-agent setting raises unique challenges: agents act simultaneously, rewards are often observed only locally, and in the fully decentralized case, no central controller is available to aggregate global information.

Zhang et al. (2018) established a multi-agent policy gradient theorem for fully decentralized MARL under the assumption that the global states and actions are observable to all agents, while rewards remain local. This result forms the theoretical basis of their decentralized actor–critic algorithms.

Theorem 1 (Policy Gradient for MARL (Zhang et al., 2018)) *Consider N agents with local policies $\pi^i_{\theta^i}$ parameterized by θ^i, and let the joint policy be $\pi_\theta = \prod_{i=1}^N \pi^i_{\theta^i}, \theta = [\theta^1, \ldots, \theta^n]^\top$. The collective objective is to maximize the globally averaged return $J(\pi_\theta)$ defined in Equation (1). Then, for each agent i, the policy gradient with respect to θ^i is given by*

$$\nabla_{\theta^i} J(\pi_\theta) = \mathbb{E}_{\boldsymbol{o}\sim\rho_{\pi_\theta},\boldsymbol{a}\sim\pi_\theta}\Big[\nabla_{\theta^i} \log \pi^i_{\theta^i}(\boldsymbol{o}_t, \boldsymbol{a}^i_t) A_\theta(\boldsymbol{o}_t, \boldsymbol{a}_t)\Big], \tag{5}$$

where $A_\theta(\boldsymbol{o}_t, \boldsymbol{a}_t) = R(\boldsymbol{o}_t, \boldsymbol{a}_t) + \gamma V_\phi(\boldsymbol{o}_{t+1}) - V_\phi(\boldsymbol{o}_t)$ is the global advantage function, and $V_\phi : \mathcal{O} \to \mathbb{R}$ is the global state-value function parameterized by ϕ.

This theorem shows that each agent can compute its policy gradient update using only its local policy parameters and an estimate of the global advantage function. Each agent learns its local value function parameters ϕ^i by minimizing the local TD-error δ^i, and then they run the average consensus protocol to find the global value function parameters ϕ:

$$\begin{aligned}
\delta^i &= R(\boldsymbol{o}_t, \boldsymbol{a}_t) + \gamma V_{\phi^i}(\boldsymbol{o}_{t+1}) - V_{\phi^i}(\boldsymbol{o}_t),\\
\tilde{\phi}^i &= \phi^i + \beta_t \delta^i_t \nabla_{\phi^i} V_{\phi^i}(\boldsymbol{o}_t),\\
\phi &= \sum_{j\in\mathcal{N}^i} c_t(i,j)\tilde{\phi}^i
\end{aligned} \tag{6}$$

where $\beta_t > 0$ is a learning rate, $\mathcal{N}^i$ is the neighborhood of agent i in the communication graph, and $c_t(i,j)$ are consensus weights. Through this consensus process, local value estimates propagate across the network, enabling each agent to converge to the global value function over time and hence compute an unbiased policy gradient update.

Theorem 1 provides the theoretical foundation for distributed MARL, upon which we build our distributed MARL algorithm in the next section.

3 Method

In this section, we present DG-MAPPO, a distributed MARL algorithm that is both simple and scalable. Unlike the widely adopted CTDE paradigm, our approach does not assume access to global state information during either training or execution. Instead, coordination emerges organically through multi-hop message passing over a connected communication graph.

Assumption 1 (Connected Communication Graph) *The communication graph G is* connected*; that is, for any two distinct agents $i \neq j$, there exists at least one path from i to j in G.*

This assumption significantly relaxes the stronger requirement of centralized information sharing commonly made in prior MARL frameworks. Building on this, we formalize the notion of distributed MARL as follows:

Definition 1 (Distributed MARL) *Consider a system of n agents operating on a connected graph G. Each agent i observes only its local information $\boldsymbol{o}^i$ and can communicate it with its neighbors. A distributed MARL algorithm requires each agent to learn both its policy and value function using solely its local observations and peer-to-peer communication, without relying on centralized training or access to the global state.*

This definition distinguishes distributed approaches from purely decentralized ones: the former leverage communication among agents to enable collaboration, whereas the latter operate independently without inter-agent communication (e.g., the decentralized execution in CTDE). In what follows, we first introduce D-GAT, our communication module that enables global state inference via multi-hop message passing in a fully distributed manner. We then introduce our DG-MAPPO algorithm which learns collaborative policies entirely from local observations and peer-to-peer communication.

3.1 Distributed Graph Attention Networks

GATs (Veličković et al., 2017) are a powerful tool for learning from graph-structured data, but their standard formulation relies on globally shared attention parameters, preventing deployment in fully distributed settings. We address this by introducing D-GAT, where each agent independently maintains and updates its own local attention parameters. This design ensures that message aggregation remains attention-driven while fully respecting the real-time communication constraints. In addition, we adopt the dynamic attention formulation of GATv2 (Brody et al., 2021), which extends the original GAT by enabling input-dependent query–key interactions, thereby enhancing representational expressiveness. The overall framework of D-GAT is illustrated in Figure 1.

A single D-GAT layer for node i operates as follows. For a node i with feature vector $\boldsymbol{h}^i \in \mathbb{R}^d$ and neighborhood $\mathcal{N}^i$, we define a local attention function $e^i : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ that measures the importance of neighbor j as:

$$e^i(\boldsymbol{h}^i, \boldsymbol{h}^j) = {\boldsymbol{q}^i}^\top \text{LeakyReLU}\left(\boldsymbol{W}^i[\boldsymbol{h}^i \| \boldsymbol{h}^j]\right). \tag{7}$$

where $\boldsymbol{W}^i \in \mathbb{R}^{d' \times 2d}$ is a learnable linear projection of agent i, $\boldsymbol{q}^i \in \mathbb{R}^{d'}$ is a trainable weight vector of agent i, and $\|$ is the concatenation operator. We then perform score normalization and feature aggregation as:

$$\alpha_{ij} = \frac{\exp\left(e^i(\boldsymbol{h}^i, \boldsymbol{h}^j)\right)}{\sum_{j' \in \mathcal{N}^i} \exp(e^i(\boldsymbol{h}^i, \boldsymbol{h}^{j'}))}, \quad \hat{\boldsymbol{h}}^i = \sigma\left(\sum_{j \in \mathcal{N}^i} \alpha_{ij} \boldsymbol{h}^j\right), \tag{8}$$

where $\hat{\boldsymbol{h}}^i$ is the updated vector representation of agent i computed as an attention-weighted aggregation of its neighbors' features, followed by a nonlinear activation $\sigma(\cdot)$. We stack n such layers (equal to the number of agents) to facilitate multi-hop communication, ensuring that every agent $i \in \mathcal{N}$ can exchange information with all other agents $j \in \mathcal{N}$.

The distributed design of D-GAT introduces a fundamental challenge in MARL: each agent updates its local attention parameters solely to maximize its own performance. Such locally selfish updates can impede the formation of a coherent global representation, which is essential for effective coordination. Consequently, agents may struggle to approximate the global state, leading to suboptimal joint performance. To mitigate this issue, we propose a two-step solution. Firstly, inspired by decentralized stochastic gradient descent (D-SGD) (Lian et al., 2017), we update each agent's graph network parameters via local SGD and then average them with its immediate neighbors. Mathematically, it is given as:

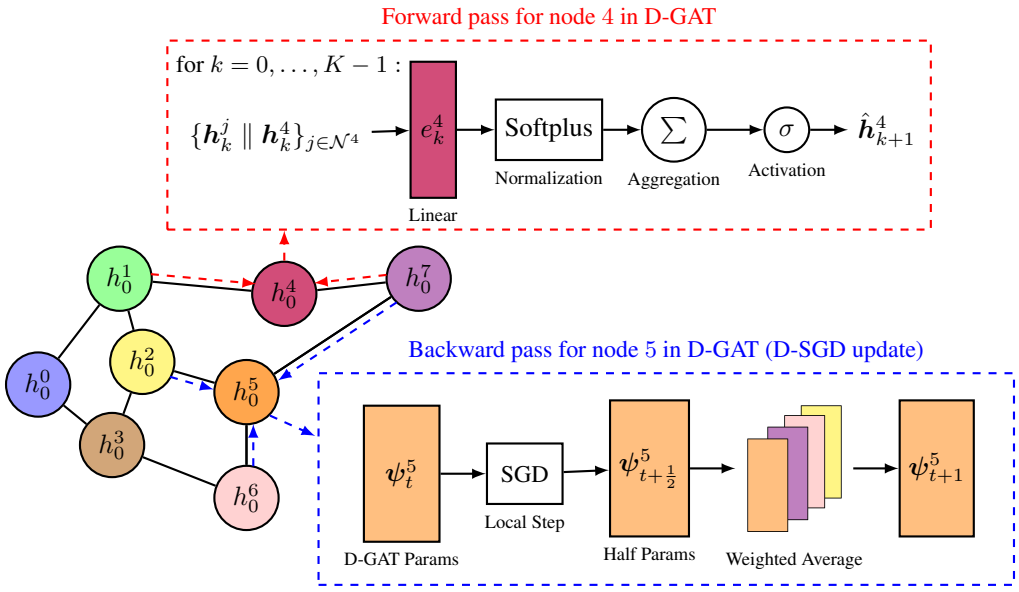


Figure 1: Illustration of the forward and backward passes in the proposed D-GAT framework. **1) Forward pass (top, red):** At each layer k, node 4 aggregates information from its neighbors $\{h_k^j \| h_k^4\}_{j\in\mathcal{N}^4}$ by applying a linear transformation, Softplus normalization, summation-based aggregation, and a nonlinearity σ to produce the updated embedding $\hat{\boldsymbol{h}}_{k+1}^4$. This process is repeated for $k = 0, \ldots, K-1$, where K is the predefined number of hops. **2) Backward pass (bottom, blue):** Node 5 updates its local D-GAT parameters $\boldsymbol{\psi}^5$ via decentralized stochastic gradient descent (D-SGD). First, a local step computes the half-step parameters $\boldsymbol{\psi}_{t+\frac{1}{2}}^5$ using the local update step of Equation (9). Next, a neighbor averaging step mixes parameters from neighbors via the neighbor averaging step of Equation (9) to get the updated D-GAT parameters $\boldsymbol{\psi}_t^5$, enabling distributed training without a central coordinator.

$$\begin{aligned} &\textbf{(Local step)} && \boldsymbol{\psi}_{t+\frac{1}{2}}^i = \boldsymbol{\psi}_t^i - \eta_t \,\widehat{\nabla}\ell^i\big(\boldsymbol{\psi}_t^i;\, \xi_t^i\big)\,, \\ &\textbf{(Neighbor averaging)} && \boldsymbol{\psi}_{t+1}^i = \sum_{j\in\mathcal{N}^i} c(i,j)\, \boldsymbol{\psi}_{t+\frac{1}{2}}^j, \qquad i = 1, \ldots, n, \end{aligned} \tag{9}$$

where $\boldsymbol{\psi}^i = \{\boldsymbol{W}^i, \boldsymbol{q}^i\}$ are the local graph attention network parameters, $\widehat{\nabla}\ell^i\big(\boldsymbol{\psi}_t^i;\, \xi_t^i\big)$ is a stochastic gradient computed from local data/minibatch ξ_t^i, η_t is the learning step-size, and $c(i,j)$ is the consensus weight between agent i and j consistent with the communication graph given by,

$$c(i,j) = \begin{cases} \dfrac{1}{|\mathcal{N}^i|}, & \text{if } j \in \mathcal{N}^i, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Here, $|\mathcal{N}^i|$ is the degree of node i. This D-SGD procedure is equivalent to an average consensus step over the local D-GAT parameters, ensuring that agents gradually align their representations with those of their neighbors. By doing so, the network parameters are regularized across the graph, which improves the generalizability of the learned representations. Importantly, since the local updates still follow the GATv2 architecture, agents can compute dynamic, input-dependent attention weights as in Brody et al. (2021), thereby preserving the expressive power of attention mechanisms while operating in a distributed setting.

Moreover, to facilitate global state inference, we introduce a consensus regularization objective for D-GAT that explicitly encourages neighboring agents to align their learned representations. Concretely, in addition to the value (Equation (14)) and policy losses (Equation (15)), each agent also

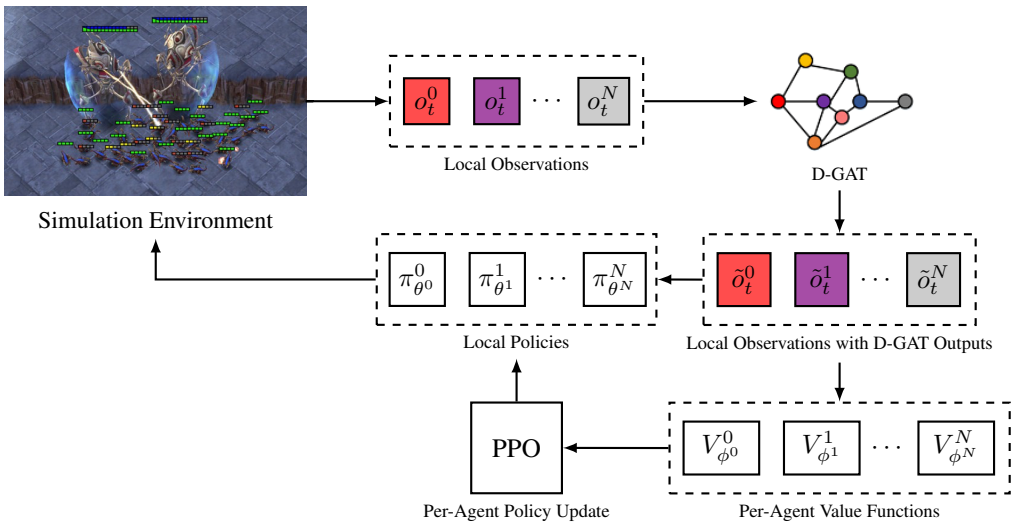


Figure 2: **DG-MAPPO Framework.** Each agent receives raw local observations $\boldsymbol{o}_t^i$ from the environment. Agents then communicate with neighbors using D-GAT to get global state inference $\hat{\boldsymbol{o}}_t^i$ which is then concatenated with raw local observations to get $\tilde{\boldsymbol{o}}_t^i = [\boldsymbol{o}_t^i, \hat{\boldsymbol{o}}_t^i]$. This combined observation is used by both local policies $\pi_{\theta^i}^i$ to generate actions and by value functions $V_{\phi^i}^i$ to estimate global returns. PPO performs per-agent policy updates using the advantage estimates derived from GAE.

minimizes a consensus loss with respect to its neighbors, given by

$$\mathcal{L}_{\text{consensus}}^i(\boldsymbol{\psi}^i) = \alpha_{\text{consensus}} \frac{1}{|\mathcal{N}^i|} \sum_{j \in \mathcal{N}^i} \text{MSE}\big(\hat{\boldsymbol{h}}^i, \hat{\boldsymbol{h}}^j\big), \tag{11}$$

where $\alpha_{\text{consensus}}$ controls the strength of the regularization, $\text{MSE}(\cdot, \cdot)$ denotes the mean-squared error.

Intuitively, D-GAT integrates D-SGD with a consensus regularization objective, providing a communication framework that enables agents to extract essential global state information while suppressing irrelevant information from less important neighbors. For instance, D-GAT enables agents to prioritize information from collaborators with strong influence on their outcomes, while down-weighting signals from agents whose actions have little impact.

3.2 Distributed Graph-Attention MAPPO

We now introduce our distributed MARL framework, **DG-MAPPO**, illustrated in Figure 2. The central idea is that, given access to global state information and a globally shared (or averaged) reward, each agent $i \in \mathcal{N}$ can independently learn a global state value function $V_{\phi^i}^i : \mathcal{O} \to \mathbb{R}$ defined as

$$V_{\phi^i}^i(\boldsymbol{o}_t) = \mathbb{E}_{a_t \sim \pi_\theta} \left[\sum_{t=0}^{T} \gamma^t R(\boldsymbol{o}_t, \boldsymbol{a}_t) \,\middle|\, \boldsymbol{o}_0 = \boldsymbol{o}_t \right]. \tag{12}$$

In practice, however, agents in our distributed setting cannot directly observe the true global state. To overcome this limitation, we incorporate the **D-GAT communication module** (see Section 3.1), which enables agents to perform multi-hop message passing after each local observation. Through this process, agents obtain an informative approximation of the global state, denoted as $\hat{\boldsymbol{o}}_t^i \in \hat{\mathcal{O}}^i$.

Training the value function solely on $\hat{\boldsymbol{o}}_t^i$ can lead to high variance, particularly in the early stages of learning, which risks destabilizing training. To address this, we provide each agent with a concatenated input combining its own local observation and the global state approximation:

$$\tilde{\boldsymbol{o}}_t^i = [\boldsymbol{o}_t^i \,\|\, \hat{\boldsymbol{o}}_t^i] \in \tilde{\mathcal{O}}^i \tag{13}$$

Table 1: Performance evaluations of win rate and standard deviation on the SMAC benchmark.

Task	Difficulty	MAT-Dec	MAPPO	HAPPO	DG-MAPPO
3m	Easy	**100.0**(1.1)	**100.0**(0.4)	**100.0**(1.2)	**100.0**(1.4)
8m	Easy	97.2(2.5)	96.8(2.9)	97.5(1.1)	**100.0**(1.4)
MMM	Easy	98.1(2.1)	95.6(4.5)	81.2(22.9)	**98.7**(1.7)
5m vs 6m	Hard	83.1(4.6)	88.2(6.2)	73.8(4.4)	**88.7**(4.7)
8m vs 9m	Hard	95.0(4.6)	93.8(3.5)	86.2(4.4)	**95.0**(4.1)
10m vs 11m	Hard	**100.0**(2.0)	96.3(5.8)	77.5(9.7)	**100.0**(1.4)
25m	Hard	86.9(5.6)	**100.0**(2.7)	0.6(0.8)	72.5(10.1)
MMM2	Hard+	91.2(5.3)	81.8(10.1)	0.3(0.4)	**97.5**(1.4)
6h vs 8z	Hard+	93.8(4.7)	88.4(5.7)	0.0(0.0)	**95.0**(2.7)
3s5z vs 3s6z	Hard+	85.3(7.5)	84.3(19.4)	82.8(21.2)	**91.9**(10.7)

This representation ensures that agents retain a reliable self-signal while progressively benefiting from improved global context. Each agent's critic is then trained by minimizing the Bellman error:

$$\mathcal{L}^i_{\text{critic}}(\phi^i) = \mathbb{E}_{a_t \sim \pi_\theta}\left[\sum_{t=0}^{T-1} R(\boldsymbol{o}_t, \boldsymbol{a}_t) + \gamma V^i_{\phi^i}(\tilde{\boldsymbol{o}}^i_{t+1}) - V^i_{\phi^i}(\tilde{\boldsymbol{o}}^i_t)\right]^2. \tag{14}$$

Although agents cannot directly access the joint global policy π_θ, they can still learn state-value functions consistent with it. This is possible because agents experience a common reward signal—either provided by a global reward mechanism or obtained through averaging local rewards via consensus, and they condition the state-value function on $\tilde{\boldsymbol{o}}^i_t$, consisting of the global state representation. Agents can now estimate the global advantage function $A^i_\theta : \mathcal{O} \times \mathcal{A} \to \mathbb{R}$ leveraging the shared/average reward $R(\boldsymbol{o}_t, \boldsymbol{a}_t)$ and the local estimate of the global value function V^i. In practice, the agents use **generalized advantage estimation** (GAE) (Schulman et al., 2015) to independently approximate a low-bias, low-variance global advantage estimate leveraging the local stored trajectories.

Following Theorem 1, the policy parameters of each agent θ^i are updated using a clipped policy gradient objective, as in PPO (Schulman et al., 2017):

$$\mathcal{L}^i_{\text{PPO}}(\theta^i) = \mathbb{E}_t\left[\min\left(\frac{\pi^i_{\theta^i}(a^i_t \mid \tilde{o}^i_t))}{\pi^i_{\theta^i_{\text{old}}}(a^i_t \mid \tilde{o}^i_t))} A^i_\theta(\boldsymbol{o}_t, \boldsymbol{a}_t),\ \text{clip}\left(\frac{\pi^i_{\theta^i}(a^i_t \mid \tilde{o}^i_t))}{\pi^i_{\theta^i_{\text{old}}}(a^i_t \mid \tilde{o}^i_t))}, 1 \pm \epsilon\right) A^i_\theta(\boldsymbol{o}_t, \boldsymbol{a}_t)\right)\right],$$

where ϵ is the clip parameter. Overall, using DG-MAPPO, each agent performs local actor–critic updates using only its own observation and local approximation of global state acquired through multi-hop communication, while coordination emerges organically from the shared reward structure and multi-hop message passing. The pseudocode is provided in the Appendix 1.

4 Results

Our distributed MARL framework offers a principled alternative to the widely adopted CTDE paradigm for cooperative MARL. Instead of relying on centralized critics and global information, our approach enables agents to collaborate using only local observations and peer-to-peer communication. By leveraging the dynamic communication graph, agents can mimic—and in many cases surpass—the benefits of CTDE methods. A distinctive advantage is that communication is actively used during both training and execution, allowing agents to maintain awareness of their neighbors' states and adapt their coordination in real time. This design not only ensures robustness in fully distributed settings without a central controller but also allows agents to selectively attend to informative messages while filtering out irrelevant noise. Importantly, the flexibility of our framework extends naturally to heterogeneous teams, where agents with different roles can effectively exchange information and align their strategies toward maximizing the overall team return.

We evaluate DG-MAPPO on the StarCraftII Multi-Agent Challenge (SMAC) benchmark (Samvelyan et al., 2019), a widely used testbed for cooperative MARL. SMAC consists of micromanagement tasks of varying difficulty, ranging from small homogeneous battles (e.g., 3m, 8m) to heterogeneous or large-scale scenarios (e.g., 3s5z vs 3s6z, MMM2). Following prior works such as MAPPO and MAT, we report the median win rate across multiple random seeds together with standard deviations, and we compare against CTDE baselines: MAPPO, HAPPO, and MAT-Dec.

4.1 Experiment Setup

While CTDE baselines exploit global observations available in SMAC, our DG-MAPPO algorithm relies solely on local observations. Agents **communicate at every timestep** via the D-GAT communication module and select actions leveraging the inferred global state, receiving a shared reward from the environment. To ensure fully distributed learning, each agent maintains its own local dataset and performs optimization independently. Parameter averaging with local neighbors is applied only to the D-GAT networks, as detailed in Section 3.1. Additionally, we track the average node degree at the end of each episode (see Appendix A.4) to provide insight into graph connectivity.

4.2 Performance on SMAC

Table 1 compares DG-MAPPO with strong CTDE baselines (MAPPO, HAPPO, and MAT-Dec) on the SMAC benchmark. DG-MAPPO achieves consistently strong results across diverse tasks, ranging from small homogeneous battles to challenging heterogeneous and large-scale scenarios. In easier homogeneous settings, all methods reach near-perfect win rates, indicating that local coordination suffices for simple tasks. As difficulty increases, however, the benefits of distributed communication and independent learning become clear. For example, in hard+ scenarios such as *MMM2* and *6h vs 8z*, HAPPO collapses, MAPPO succeeds with centralized critics, yet DG-MAPPO still attains a strong 92.2% win rate in a fully distributed setting. Likewise, in heterogeneous scenarios like *3s5z vs 3s6z*, DG-MAPPO outperforms CTDE baselines by a notable margin, demonstrating its capacity to handle both heterogeneity and scaling challenges.

Beyond average performance, DG-MAPPO also exhibits greater robustness across runs, with lower variance compared to centralized baselines. This suggests that distributed communication via D-GAT provides more stable learning dynamics, even under partial observability. We also observe that performance scales favorably as the number of agents increases, though at the cost of higher communication overhead due to multi-hop message passing. Nevertheless, DG-MAPPO maintains competitive results without relying on privileged information or centralized critics, highlighting its scalability and practical relevance for large-scale MARL. These findings confirm the effectiveness of D-GAT for distributed communication and global state inference: unlike CTDE methods, DG-MAPPO maintains high performance without centralized training, while remaining robust and scalable in complex cooperative MARL environments.

5 Conclusion

In this work, we presented DG-MAPPO, a principled distributed MARL framework that leverages D-GAT for global state inference and enables collaborative policy learning without relying on a centralized controller or global information. Our experiments on the challenging SMAC benchmark demonstrate that DG-MAPPO consistently matches or surpasses state-of-the-art CTDE approaches, highlighting its scalability and effectiveness in cooperative tasks under partial observability and dynamic environments. Beyond strong empirical performance, our work underscores the potential of communication-driven distributed frameworks to overcome the limitations of centralized training, providing a foundation for more resilient, scalable, and realistic multi-agent learning systems. While DG-MAPPO performs well across a wide range of tasks, we observe that scalability becomes more challenging in large-scale teams, where long-range communication becomes ineffective due to information squashing. In particular, the 25m scenario reveals room for improvement, suggesting that future work should focus on designing more efficient deep-communication strategies to further enhance scalability. We believe DG-MAPPO provides a promising step toward advancing distributed multi-agent learning at scale, opening opportunities for more robust and efficient frameworks in complex real-world environments.

REFERENCES

Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pp. 344–353. PMLR, 2019.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. Graph convolutional reinforcement learning. *arXiv preprint arXiv:1810.09202*, 2018.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. *Advances in neural information processing systems*, 30, 2017.

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

Angelia Nedić and Asuman Ozdaglar. Subgradient methods for saddle-point problems. *Journal of optimization theory and applications*, 142(1):205–228, 2009.

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

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhuojun Tian, Zhaoyang Zhang, Zhaohui Yang, Richeng Jin, and Huaiyu Dai. Distributed learning over networks with graph-attention-based personalization. *IEEE Transactions on Signal Processing*, 71:2071–2086, 2023.

John N Tsitsiklis. Problems in decentralized decision making and computation. Technical report, 1984.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022.

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

Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Basar. Fully decentralized multi-agent reinforcement learning with networked agents. In *International conference on machine learning*, pp. 5872–5881. PMLR, 2018.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

A Appendix

A.1 Use of LLMs

We used a large language model (OpenAI's ChatGPT) as a writing assistant to help polish the clarity, grammar, and readability of certain sections of the paper (e.g., abstract, introduction, and conclusion). The model was not used for generating research ideas, designing experiments, or analyzing results. All technical content, experiments, and conclusions were conceived and validated solely by the authors.

A.2 Pseudo Code for DG-MAPPO

Algorithm 1 Distributed Graph-Transformer MAPPO

Input: Number of agents and hops n, learning rate α, episodes K, steps per episode T
Initialize: D-GAT $\{\psi^i\}_{i\in\mathcal{N}}$, Critic $\{\phi^i\}_{i\in\mathcal{N}}$, Policy $\{\theta^i\}_{i\in\mathcal{N}}$, Replay Buffer $\{\boldsymbol{\xi}^i\}_{i\in\mathcal{N}}$
for $k = 0, 1, \ldots, K-1$ **do**
  for $t = 0, 1, \ldots, T-1$ **do**
    Receive local observations $\{\boldsymbol{o}_t^i\}_{i\in\mathcal{N}}$ from environment.
    Perform multi-hop communication using D-GAT to infer global state $\{\hat{\boldsymbol{o}}_t^i\}_{i\in\mathcal{N}}$.
    Sample actions using local policies $a_t^i \sim \pi_{\theta^i}^i \quad \forall i \in \mathcal{N}$.
    Perform the joint action $\boldsymbol{a}_t$ in the environment and observe joint reward $R(\boldsymbol{o}_t, \boldsymbol{a}_t)$.
    Store $(\boldsymbol{o}_t^i, \hat{\boldsymbol{o}}_t^i, \boldsymbol{a}_t^i, R(\boldsymbol{o}_t, \boldsymbol{a}_t)$ in the buffer $\{\boldsymbol{\xi}^i\}$
  end for
  Sample random minibatch ξ^i from $\boldsymbol{\xi}^i$
  Infer state-values for all agents $\{V_{\phi^i}^i(\tilde{\boldsymbol{o}}^i)\}_{i\in\mathcal{N}}$, where $\tilde{\boldsymbol{o}}^i = (\boldsymbol{o}^i \| \hat{\boldsymbol{o}}^i)$.
  Calculate $\{\mathcal{L}_{\text{critic}}^i(\phi^i)\}_{i\in\mathcal{N}}$ using Equation (14).
  Estimate global advantage A^i based on $V_{\phi^i}^i(\tilde{\boldsymbol{o}}^i)$ and $R(\boldsymbol{o}_t, \boldsymbol{a}_t) \quad \forall i \in \mathcal{N}$ using GAE.
  Compute policy loss $\{\mathcal{L}_{\text{PPO}}^i(\theta^i)\}_{i\in\mathcal{N}(\phi^i)}$ using Equation (15).
  Compute D-GAT consensus regularizer loss $\{\mathcal{L}_{\text{consensus}}^i(\boldsymbol{\psi}^i)\}_{i\in\mathcal{N}}$ using Equation (11).
  Update D-GAT, value critic, and policy networks by minimizing $\mathcal{L}_{\text{critic}}^i(\phi^i) + \mathcal{L}_{\text{PPO}}^i(\theta^i) + \mathcal{L}_{\text{consensus}}^i(\boldsymbol{\psi}^i) \quad \forall i \in \mathcal{N}$ using gradient decent.
end for

A.3 Hyper-parameters For DG-MAPPO

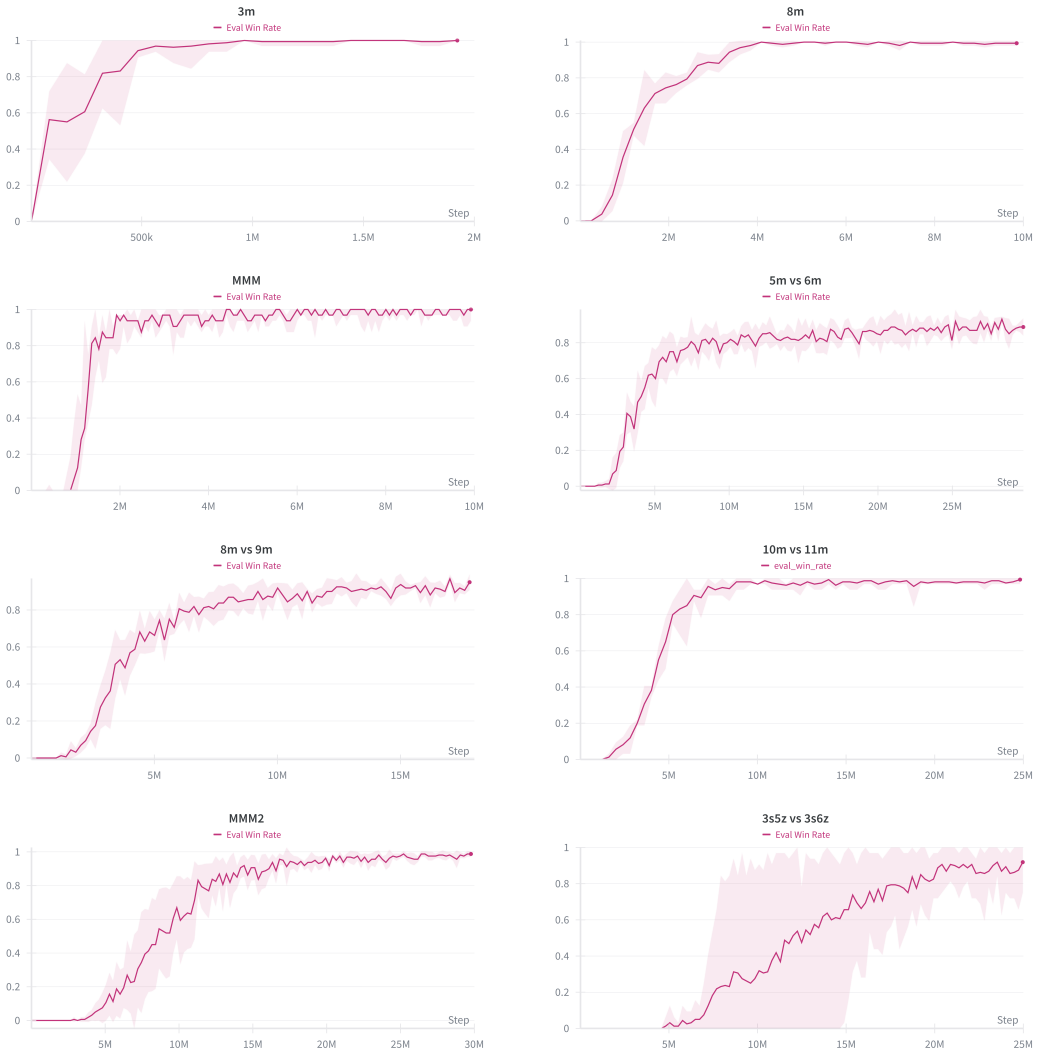
Table 2: Common hyper-parameters used for DG-MAPPO

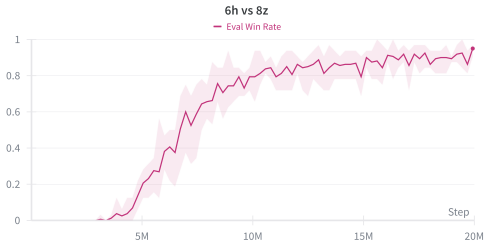
Parameter	Value	Parameter	Value	Parameter	Value
critic lr	5e-4	actor lr	5e-4	use GAE	True
gain	0.01	optim eps lr	1e-5	training threads	32
entropy coeff	0.001	max grad norm	10	optimizer	Adam
hidden layer dim	128	use huber loss	True	gae lambda	0.95
D-GAT lr	5e-4	num heads	1	$\alpha_{\text{consensus}}$	20

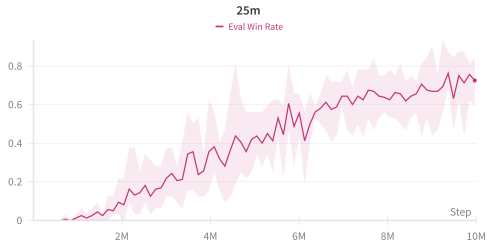
Table 3: Different hyper-parameters used for DG-MAPPO

Maps	ppo epochs	ppo clip	batch size	rollout threads	episode length	gamma	steps
3m	10	0.05	2048	32	100	0.98	2e6
8m	10	0.05	2048	32	300	0.98	1e7
MMM	10	0.05	2048	32	300	0.98	1e7
5m vs 6m	5	0.05	3200	32	300	0.98	3e7
8m vs 9m	10	0.05	2048	16	500	0.98	2e7
10m vs 11m	5	0.05	3200	32	500	0.98	2.5e7
25m	5	0.05	4800	16	300	0.95	1e7
MMM2	5	0.05	3200	32	300	0.95	3e7
6h vs 8z	10	0.05	2048	32	300	0.98	2e7
3s5z vs 3s6z	5	0.2	3200	32	300	0.98	3e7

A.3.1 DG-MAPPO Evaluation Plots







A.4 Average Node Degree (At end of Episode)