# Adaptive Communication Range for Scalable Cooperative Multi-Agent Reinforcement Learning

**Huizhong Song[1], Wei Wei[1]*, Lin Li[1], Lijun Zhang[1], Fengjiao Li[1]**
[1]Key Laboratory of Computational Intelligence and Chinese Information Processing
of Ministry of Education, School of Computer and Information Technology, Shanxi University,
{songhuizhong, weiwei, lilynn1116, lijun-zhang, fengjiaoli}@sxu.edu.cn

## Abstract

Cooperative multi-agent reinforcement learning (MARL) has long faced scalability challenges due to the exponential growth of the state-action space as the number of agents increases. Existing methods typically enhance scalability by filtering out communications with low-relevance agents. However, such filtering often relies on global state or fixed graph distribution, limiting adaptability in large-scale, communication-constrained environments. In this paper, we propose a scalable MARL method, called Adaptive Communication Range PPO (ACR-PPO), that decomposes the decision-making under communication budget constraints as a sequential process: a communication policy first selects each agent's communication range within a given budget, followed by a behavior policy that conditions actions on the resulting neighborhood observations. More importantly, we provide a theoretical guarantee of monotonic performance improvement under communication budget constraints. Experiments across diverse scenarios demonstrate that ACR-PPO preserves policy performance while significantly reducing communication costs through adaptive range control.

## 1 Introduction

Cooperative multi-agent reinforcement learning (Cooperative MARL) has achieved notable progress in fields such as game AI [Vinyals et al., 2019], UAV controlling [Cui et al., 2019] and urban transportation [Ghanadbashi and Golpayegani, 2022]. However, scaling MARL to large-scale networked systems is challenging due to the exponential expansion of the joint state-action space, resulting in high communication, computational costs, and low sample efficiency [Oroojlooy and Hajinezhad, 2023]. Overcoming this scalability challenge is crucial for the practical application of cooperative MARL in real-world environments.

To mitigate the scalability challenges, existing methods can be broadly divided into two categories. The first category of methods relies on fully decentralized training paradigms, such as IPPO [De Witt et al., 2020] and DPO [Su and Lu, 2024], which extend single-agent techniques to multi-agent settings through parameter sharing. However, due to the absence of inter-agent communication, these methods often rely on strong assumptions to address the non-stationarity during policy updates. The second category of methods relaxes communication constraints by allowing agents to interact with local neighbors [Zhang et al., 2021], like DGN [Jiang et al., 2020], Scalable AC [Qu et al., 2020], DMPO [Ma et al., 2024], Scal-PPO-L [Zhang et al., 2024b], GSAC [Liang et al., 2025] and BayesG [Duan et al., 2025]. These methods have significantly improved performance, but they generally rely on predefined communication topologies and require complex hyperparameter tuning, which may limit their flexibility. Some methods have been proposed to control communication costs, including role-based [Sheng et al., 2022], gating-based methods [Mao et al., 2020, Wang et al., 2023]. These

*Corresponding author.

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

methods typically require the global state, which may limit their effectiveness in large-scale scenarios. Sampling-based apporaches [Lin et al., 2021, Agorio et al., 2025] are intuitive and relatively easy to implement with a certain degree of flexibility. However, they exhibit limited adaptability and may fail when the underlying distribution changes.

To address the limitations of these methods, we propose a new cooperative MARL method that formulates communication-based decision-making under communication cost budget as a constrained optimization problem[2]. Inspired by multi-agent sequential update[Kuba et al., 2022, Li et al., 2023a], we model this as a sequential update process [Zhong et al., 2024, Gu et al., 2023], in which the communication policy first dynamically determines each agent's communication range, and then the behavior policy selects actions based on the neighbors within that range.

Our main contributions are summarized as follows:

- We propose a scalable cooperative MARL method that formulates communication-based decision-making under communication cost budget as a constrained optimization problem, which enables adaptive adjustment of the communication range under communication budgets.
- We present the formal definition of the time-varying truncate value function and derive policy optimization objectives for each agent by alternately updating the communication and behavior policies, and we also provide a theoretical guarantee that agents can achieve monotonic performance improvement while satisfying communication budgets constraints.
- We evaluate our method on the popular networked multi-agent benchmarks, demonstrating that our method maintains competitive policy performance while effectively reducing communication cost by adaptively minimizing the communication range.

## 2 Preliminaries

### 2.1 Networked Multi-agent MDP

We consider a network of $n$ agents represented as an undirected graph $\mathcal{G} = (\mathcal{N}, \varepsilon)$, where $\mathcal{N} = \{1, ..., n\}$ denotes the set of agents, and $\varepsilon$ represents the set of edges capturing the pairwise relationships among them. The networked multi-agent MDP $(\mathcal{G}, \mathcal{S}, \mathcal{A}, \{\mathcal{M}_{ij}\}_{ij\in\varepsilon}, \{r_i\}_{i\in\mathcal{N}}, P)$ are summarized below:

- **State and action**. Each agent $i \in \mathcal{N}$ has a local state $s_i \in \mathcal{S}_i$ and selects an action $a_i \in \mathcal{A}_i$. The global state is defined as $\mathbf{s} = (s_1, s_2, \ldots, s_n)$, where $\mathbf{s} \in \mathcal{S} := \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_n$. Similarly, the global action is given by $\mathbf{a} = (a_1, a_2, \ldots, a_n)$, with $\mathbf{a} \in \mathcal{A} := \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$.
- **Communication**. $\mathcal{M}$ is the message space. $\mathcal{N}_i$ is the set including node $i$ and all its neighbors on the graph. $\mathcal{N}_i^\kappa$ is the set including node $i$ and all its $\kappa$-hop neighbors on the graph. The set of agents not in $\mathcal{N}_i^\kappa$ is denoted by $\mathcal{N}_{-i}^\kappa := \mathcal{N} \setminus \mathcal{N}_i^\kappa$. Each agent obtains the states of all agents within its $\kappa$-hop neighbors $\mathcal{N}_i^\kappa$ through message $m_{ij} \in \mathcal{M}_{ij}$, denoted by $s_{\mathcal{N}_i^\kappa}$.
- **Reward.** Each agent $i$ has a reward function $r_i(s_i, a_i)$. The global reward is defined as the average of individual rewards: $r(\mathbf{s}, \mathbf{a}) = \frac{1}{n}\sum_{i=1}^n r_i(s_i, a_i)$.
- **Transition decomposition.** The next state $s_i$ is determined by the neighbors state $s_{\mathcal{N}_i}$ and agent $i$'s own action $a_i$ [Qu et al., 2020]:

$$P\left(\mathbf{s}(t+1)|\mathbf{s}(t), \mathbf{a}(t)\right) = \prod_{i=1}^{n} P\left(s_i(t+1)|s_{\mathcal{N}_i}(t), a_i(t)\right). \tag{1}$$

- **Policy factorization.** The joint policy $\psi(\mathbf{a}|\mathbf{s})$ is factorized into individual policies $\psi_i$: $\psi(\mathbf{a}|\mathbf{s}) = \prod_{i\in\mathcal{N}} \psi_i(a_i|s_i)$.
- **Communication cost**. We follow the setting [Joo, 2010] and assume that the communication cost grows exponentially with the communication range. For details on the communication cost, see Appendix B.1.

[2]Our code can be found in `https://github.com/Bonstop/ACRPPO`.

To support the subsequent analysis, we revisit several foundational concepts in MARL. The first key concept is the joint state-action value function $Q$, which is used to estimate the expected cumulative discounted reward from a given state-action pair:

$$Q(\mathbf{s}, \mathbf{a}) := \frac{1}{n} \sum_{i=1}^{n} Q_i^{\theta}(\mathbf{s}, \mathbf{a}). \tag{2}$$

More derivation details can be found in Appendix B.2.

Another key concept is the policy gradient theorem [Schulman et al., 2015](Lemma B.1), which forms the foundation for optimizing the policy objective $J(\theta)$ in many MARL algorithms [Yu et al., 2022, Kuba et al., 2022]. Lemma B.1 shows that the gradient of $J(\theta)$ depends on joint state-action value function $Q$, which typically involves global dependencies and may limit its scalability in large-scale scenarios.

### 2.2 Spatial Correlation Decay

Spatial correlation decay refers to the fact that a certain quantity decays as the distance between two agents increases. Over the past decades, many researchers have leveraged the spatial correlation property, to design scalable, distributed algorithms [Qu et al., 2020, Zhang et al., 2024b] for optimization and control in multi-agent settings, such as traffic flow networks [Li et al., 2020], power networks [Chen et al., 2021], and epidemiological dynamics [Deng et al., 2022].

We start with the definition of the exponential decay property [Qu et al., 2020]. We write state $\mathbf{s}$ as $(s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa})$, which is the states of agents that are in the $\kappa$-hop neighborhood of $i$ and outside of the $\kappa$-hop neighborhood respectively. Similarly, we write $\mathbf{a}$ as $(a_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_{-i}^\kappa})$. The exponential decay property is defined as follows:

**Definition 2.1.** The $(c, \rho)$-exponential decay property holds if, for any $i \in \mathcal{N}$, $s_{\mathcal{N}_i^\kappa} \in \mathcal{S}_{\mathcal{N}_i^\kappa}$, $s_{\mathcal{N}_{-i}^\kappa}, s'_{\mathcal{N}_{-i}^\kappa} \in \mathcal{S}_{\mathcal{N}_{-i}^\kappa}$, $a_{\mathcal{N}_i^\kappa} \in \mathcal{A}_{\mathcal{N}_i^\kappa}$, $a_{\mathcal{N}_{-i}^\kappa}, a'_{\mathcal{N}_{-i}^\kappa} \in \mathcal{A}_{\mathcal{N}_{-i}^\kappa}$, $Q_i$ satisfies:

$$|Q_i(s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_{-i}^\kappa}) - Q_i(s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_i^\kappa}, a'_{\mathcal{N}_{-i}^\kappa})| \leq c\rho^{\kappa+1}. \tag{3}$$

**Lemma 2.2.** *Assume $\forall i$, $r_i$ is upper bounded by $\bar{r}$. $\gamma \in (0, 1)$ is the discount factor. Then the $(c = \frac{\bar{r}}{1-\gamma}, \rho = \gamma)$-exponential decay property holds.*

**Remark.** Lemma 2.2 shows that the exponential decay property holds generally with $\rho = \gamma$, without any assumption on the transition probabilities except for the Equation (1) and the localized policy structure. The proof of Lemma 2.2 can be seen in Appendix B.4.

Based on Definition 2.1, we introduce the truncated $Q$-functions for agent $i$, which reflect that the dependence on distant agents is "truncated":

$$\widetilde{Q}_i(s_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_i^\kappa}) = \sum_{s_{\mathcal{N}_{-i}^\kappa} \in \mathcal{S}_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_{-i}^\kappa} \in \mathcal{A}_{\mathcal{N}_{-i}^\kappa}} \omega_i(s_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_{-i}^\kappa}; s_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_i^\kappa}) Q_i(s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_{-i}^\kappa}), \tag{4}$$

where $\omega_i(\cdot)$ are any non-negative weights satisfying

$$\sum_{s_{\mathcal{N}_{-i}^\kappa} \in \mathcal{S}_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_{-i}^\kappa} \in \mathcal{A}_{\mathcal{N}_{-i}^\kappa}} \omega_i(s_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_{-i}^\kappa}; s_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_i^\kappa}) = 1, \quad \forall (s_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_i^\kappa}) \in \mathcal{S}_{\mathcal{N}_i^\kappa} \times \mathcal{A}_{\mathcal{N}_i^\kappa}.$$

**Lemma 2.3.** *Under the $(c, \rho)$-exponential decay condition, the truncated $Q$-function as defined in Equation (4), satisfies the following relation:*

$$|\widetilde{Q}(s_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_i^\kappa}) - Q(\boldsymbol{s}, \boldsymbol{a})| \leq c\rho^{\kappa+1}, \forall (\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}. \tag{5}$$

Lemma 2.3 shows that, for sufficiently large $\kappa$, spatial correlation decay ensures the truncated $Q$-function is a close approximation to the global $Q$-function. Similarly, the value function $V$ and the advantage function $A$ can also be defined with corresponding properties. The more detail and the proof of Lemma 2.3 is provided in Appendix B.5.

Based on Lemma 2.3 and MAPPO [Yu et al., 2022] loss function (Appendix B.6), we can derive the "Scalable" MAPPO loss function, which can be found in Appendix B.7.

Under the assumption of spatial correlation decay, policy updates can rely on $\kappa$-hop local observations $s_{\mathcal{N}_i^\kappa}$ instead of the global state $\mathbf{s}$. However, in existing methods, the communication range $\kappa$ is typically fixed a priori, as it is either set based on global state information or predefined by a static graph structure, thereby limiting adaptability to dynamic and resource-constrained environments such as UAV networks [Yang et al., 2024].

# 3 Methods

In this section, we introduce time-varying truncation functions to model dynamically communication ranges (Subsection 3.1). Building on this formulation, we address communication-based decision-making under communication cost budget as a constrained optimization problem (Subsection 3.2), and propose a practical algorithm based on parameterized policies (Subsection 3.3).

## 3.1 Time-varying truncate value function

We introduce a time-varying truncated $Q$-value formulation. By leveraging Definition 2.1 and Lemma 2.3, we extend the truncation mechanism to define time-varying truncated state-action value function $Q$, advantage function $A$, and along with their corresponding surrogate objectives $L_{\psi_i}$.

**Definition 3.1.** Let $s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}$ denote the local state of agent $i$ at timestep $t$ within $\kappa_{i,t}$-hop neighbors. The agent policy $\psi_i$ is factorized as $\psi_i(a_{i,t} \mid s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}) = \phi_i(\kappa_{i,t} \mid s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}) \cdot \pi_i(a_{i,t} \mid s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})$, where $\phi_i$ governs the dynamic selection of the communication range $\kappa_{i,t}$, and $\pi_i$ generates the action $a_{i,t}$ given the local state and current communication range. Let $a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}$ denote the actions of neighboring agents (excluding agent $i$) within the $\kappa_{i,t}$-hop neighbors.

For agent $i$'s communication policy $\phi_i$, we can define:

$$V_{\psi_i}^{\phi_i}(s_{\mathcal{N}_i^{\kappa_{i,t-1}}}) = \mathbb{E}_{\kappa_{i,t}\sim\phi_i}\left[\mathbb{E}_{a_{i,t}\sim\pi_i, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}\sim\pi_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}}\left[Q_{\psi_i}^{\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}})\right]\right], \tag{6}$$

$$Q_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) = r_\kappa + \gamma V_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}), \tag{7}$$

$$A_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) = Q_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) - V_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}), \tag{8}$$

$$L_{\psi_i}(\phi_i) = \mathbb{E}_{s_{i,t}\sim P, \kappa_{i,t}\sim\phi_i}[A_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})]. \tag{9}$$

For agent $i$'s behavior policy $\pi_i$, we can define:

$$Q_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) = r_i + \gamma\mathbb{E}_{s_{i,t+1}\sim P}\left[V_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t+1}^{\kappa_{i,t}}})\right], \tag{10}$$

$$A_{\psi_i}^{\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) = Q_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) - Q_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}), \tag{11}$$

$$A_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}) = A_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) + A_{\psi_i}^{\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}), \tag{12}$$

$$L_{\psi_i}(\phi_i, \pi_i) = \mathbb{E}_{s_{i,t}\sim P, \kappa_{i,t}\sim\phi_i, a_{i,t}\sim\pi_i}[A_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}})]. \tag{13}$$

**Remark.** To satisfy the Markov decision process, the $r_\kappa$ in Equation (7) is not the true reward. This is because, in Definition 3.6, we also provide the corresponding cost function.

**Proposition 3.2.** *For any agent $i$, suppose the communication range $\kappa_{i,t}$ follows a policy $\phi_i(s_{i,t})$. Given parameters $\eta = \frac{\bar{r}}{1-\gamma}, \sigma = \gamma$, we can derive:*

$$\left|Q_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) - Q_{\psi_i}(\mathbf{s}, \mathbf{a})\right| \leq \mathbb{E}_{\kappa_{i,t}\sim\phi_i(s_{i,t})}\left[\eta\sigma^{\kappa_{i,t}+1}\right]. \tag{14}$$

Proposition 3.2 establishes that varying the communication range $\kappa_{i,t}$ still preserves the spatial decay property of the truncated function. The proof of Proposition 3.2 is provided in Appendix C.1. Based on this proposition, we can derive the following two corollaries:

**Corollary 3.3.** *For any agent $i$, given parameters $\eta' = 2(C_v + \eta)$, $\sigma = \gamma$, if Proposition 3.2 holds, the following result holds:*

$$\left| A_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) - A_{\psi_i}(\boldsymbol{s}, \boldsymbol{a}) \right| \leq \mathbb{E}_{\kappa_{i,t} \sim \phi(s_{i,t})} \left[ \eta' \sigma^{\min\{\kappa_{i,t-1}, \kappa_{i,t}\}+1} \right]. \tag{15}$$

Corollary 3.3 establishes an upper bound on the error between the time-varying truncated advantage function $A_{\psi_i}^{\phi_i,\pi_i}$ and the fully centralized advantage function $A_{\psi_i}$. This bound shows that the time-varying truncated advantage function can approximate the fully centralized advantage function in policy updates, thereby significantly reducing communication cost. The proof is provided in Appendix C.2.

**Corollary 3.4.** *For any agent $i$, given parameters $\eta' = 2(C_v + \eta), \sigma = \gamma$, let communication policy $\bar{\phi}_i$ denote the updated policy after selecting the communication range at timestep $t$, and let behavior policy $\bar{\pi}_i$ denote the updated policy after executing the action at timestep $t$. Let $\phi_i^{\text{full}}$ denote the fully centralized communication policy, under which agent $i$ communicates globally at every timestep, i.e., if $\kappa_{i,t} \sim \phi_i^{\text{full}}$, then $s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}} = \mathbf{s}$. Combining Equation (13) and Corollary 3.3, we can obtain:*

$$|L_{\psi_i}(\bar{\phi}_i, \bar{\pi}_i) - L_{\psi_i}(\phi_i^{full}, \bar{\pi}_i)| \leq \mathbb{E}_{\kappa_{i,t} \sim \phi_i(s_{i,t})} \left[ \eta' \sigma^{\min\{\kappa_{i,t-1}, \kappa_{i,t}\}+1} \right]. \tag{16}$$

Corollary 3.4 establishes an upper bound on the error between the surrogate objective under the time-varying neighbors communication and global communication. This result shows that agents can achieve similar performance in policy updates only using time-varying local neighbors information, thereby significantly reducing communication cost. The proof is provided in Appendix C.3.

For convenience, let $M_{KL} = \nu_{\phi_i} D_{KL}^{\max}(\phi_i, \bar{\phi}_i) + \nu_{\pi_i} D_{KL}^{\max}(\pi_i, \bar{\pi}_i)$ denote a weighted KL divergence penalty, and $L_{\psi_i}(\bar{\psi}_i) = L_{\psi_i}(\bar{\phi}_i) + L_{\psi_i}(\bar{\phi}_i, \bar{\pi}_i)$. We can get:

**Proposition 3.5.** *Given an agent's policy $\psi_i$, it can be decomposed into two components: the communication policy $\phi_i$ and the behavior policy $\pi_i$. The update rules for these policies are as follows:*

$$\bar{\phi}_i = \arg\max_{\phi} (L_{\psi_i}(\hat{\phi}_i) - \nu_{\phi_i} D_{KL}^{\max}(\phi_i, \hat{\phi}_i)), \tag{17}$$

$$\bar{\pi}_i = \arg\max_{\bar{\pi}_i} \left[ L_{\psi_i}(\bar{\phi}_i, \bar{\pi}_i) - \nu_{\pi_i} D_{KL}^{\max}(\pi_i, \bar{\pi}_i) - \eta' \sigma^{\min\{\kappa_{i,t-1}, \kappa_{i,t}\}+1} \right], \tag{18}$$

*where* $\nu_{\phi_i} = \frac{4\gamma \max_{s_{i,t}, \kappa_{i,t}} |A_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})|}{(1-\gamma)^2}$, $\nu_{\pi_i} = \frac{4\gamma \max_{s_{i,t}, \kappa_{i,t}, a_{i,t}} |A_{\psi_i}^{\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}})|}{(1-\gamma)^2}$.

*Under the constraints imposed by $D_{KL}^{\max}$, the updated policies $\bar{\phi}_i$ and $\bar{\pi}_i$ are encouraged to remain close to $\phi_i$ and $\pi_i$, respectively. The updated joint policy $\bar{\psi}_i$ is thus obtained by solving the constrained optimization problem:*

$$J(\bar{\psi}_i) - J(\psi_i) \geq L_{\psi_i}(\bar{\psi}_i) - \mathbb{E}_{\kappa_{i,t}} \left[ \eta' \sigma^{\min\{\kappa_{i,t-1}, \kappa_{i,t}\}+1} \right] - M_{KL}. \tag{19}$$

Proposition 3.5 shows that the communication policy and the behavior policy can be jointly optimized, enabling agents to make decisions based only on communication within their $\kappa$-hop neighbors. The proof is provided in Appendix C.4.

### 3.2 Scalable MARL with Communication Range

Although the performance of policy $\psi_i$ can be guaranteed to monotonically improve under time-varying conditions, it tends to select larger communication ranges for improving performance. This phenomenon forces us to ensure performance improvement under the upper bound on communication cost.

**Definition 3.6.** Given an agent's policy $\psi_i$, it decomposes into two components: the communication range control policy $\phi_i$ and the behavior policy $\pi_i$. The cost constraint induced by $\phi_i$ is defined as follows:

$$J_c(\psi_i) = \mathbb{E}_{s_{i,t},\kappa_{i,t}}[\gamma^t C(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})] \leq c_i. \tag{20}$$

The surrogate objective under the communication cost constraint is given by:

$$L_{c,\psi_i}(\phi_i) = \mathbb{E}_{s_{i,t},\kappa_{i,t}}[A_{c,\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})]. \tag{21}$$

Similar to Equation (51), we can compute how the communication cost changes as agents update their communication policy:

**Lemma 3.7.** *Let $\psi_i$ and $\bar{\psi}_i$ denote the old and new policies of agent $i$, respectively. Similarly, $\phi_i$ and $\bar{\phi}_i$ denote the old and new communication policies of agent $i$. The following inequality holds:*

$$J_c(\bar{\psi}_i) - J_c(\psi_i) \leq L_{c,\psi_i}(\bar{\phi}_i) + \nu_{c,\phi_i} D_{KL}^{\max}(\phi_i, \bar{\phi}_i), \nu_{c,\phi_i} = \frac{4\gamma \max_{s_{i,t},\kappa_{i,t}} |A_{c,\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})|}{(1-\gamma)^2}. \tag{22}$$

Lemma 3.7 shows that, when the divergence between $\phi_i$ and $\bar{\phi}_i$ is sufficiently small, the change in communication cost before and after the policy update $J_c(\bar{\psi}_i) - J_c(\psi_i)$, is bounded by $L_{c,\psi_i}(\bar{\phi}_i)$ and $\nu_{c,\phi_i} D_{KL}^{\max}(\phi_i, \bar{\phi}_i)$. The proof is provided in Appendix C.5.

From Equation (19), we derive a lower bound on the expected policy improvement for agent $i$, which guarantees monotonic improvement under policy updates. From Equation (22), we derive an upper bound on the communication cost, which constrains adaptive communication range selection. By jointly maximizing the lower bound in Equation (19) and minimizing the upper bound in Equation (22), we arrive at the following theorem. For notational convenience, let $L_{\psi_i}(\hat{\psi}_i) = L_{\psi_i}(\hat{\phi}_i) + L_{\psi_i}(\bar{\phi}_i, \hat{\pi}_i)$:

**Theorem 3.8.** *Let $\bar{\psi}_i = \{\bar{\phi}_i, \bar{\pi}_i\}$. When the updated policy $\bar{\psi}_i$ satisfies both the monotonic improvement condition $J(\bar{\psi}_i) \geq J(\psi_i)$ and the communication cost constraint $J_c(\bar{\psi}_i) \leq c_i$, the policy update for agent $i$ is given by:*

$$\begin{aligned} &\bar{\psi}_i = \arg\max_{\bar{\psi}_i}(L_{\psi_i}(\hat{\psi}_i) - M_{KL}\mathbb{E}_{\kappa_{i,t}\sim\bar{\phi}_i}[\eta' \sigma^{\min\{\kappa_{i,t-1},\kappa_{i,t}\}+1}]), \\ &\textbf{\textit{s.t}}\ \ J_c(\psi_i) + L_{c,\psi_i}(\bar{\phi}_i) + \nu_{c,\phi_i} D_{KL}^{\max}(\phi_i, \bar{\phi}_i) \leq c_i,\ \ M_{KL} \leq \delta, \end{aligned} \tag{23}$$

*where* $M_{KL} = \nu_{\phi_i} D_{KL}^{\max}(\phi_i, \bar{\phi}_i) + \nu_{\pi_i} D_{KL}^{\max}(\pi_i, \bar{\pi}_i)$, $\mu_1 = c_i - J_c(\psi_i) - L_{c,\psi_i}(\bar{\phi}_i)$, $\eta' = 2(C_v + \eta), \sigma = \gamma$, $\nu_{c,\phi_i} = \frac{4\gamma \max_{s_{i,t},\kappa_{i,t}} |A_{c,\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})|}{(1-\gamma)^2}$, $\nu_{\phi_i} = \frac{4\gamma \max_{s_{i,t},\kappa_{i,t}} |A_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})|}{(1-\gamma)^2}$, $\nu_{\pi_i} = \frac{4\gamma \max_{s_{i,t},\kappa_{i,t},a_{i,t},a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}} |A_{\psi_i}^{\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}})|}{(1-\gamma)^2}$, $\delta = \frac{\nu_{\phi_i}}{\nu_{c,\phi_i}}\mu_1 + \nu_{\pi_i} D_{KL}^{\max}(\pi_i, \bar{\pi}_i)$.

Theorem 3.8 guarantees that agent performance can be monotonically improved while keeping the communication range within constraints. The proof is provided in Appendix C.6.

### 3.3 Algorithms

We focus on the practical implementation of the policy update in Theorem 3.8, as outlined in Algorithm 1. Let $\theta_i$ denote the parameters of policy $\psi_i$, with $\theta_i^\phi$ and $\theta_i^\pi$ corresponding to the communication policy $\phi_i$ and the behavior policy $\pi_i$, respectively. For notational convenience. $\lambda_i$ is a scalar variable. We apply Lagrangian relaxation to the communication cost constraint, which yields:

$$\max_{\theta_i^\phi,\theta_i^\pi} \min_{\lambda_i>0} \mathbb{E}_{s,\kappa_{i,t},a_{i,t}} \left[ \min\left( U_i A_{\theta_i}^{\lambda_i,\theta_i^\phi,\theta_i^\pi}, \text{clip}(U_i, 1-\epsilon, 1+\epsilon) A_{\theta_i}^{\lambda_i,\theta_i^\phi,\theta_i^\pi} \right) \right],$$

$$\begin{aligned} A_{\theta_i}^{\lambda_i,\theta_i^\phi,\theta_i^\pi} =& A_{\theta_i}^{\theta_i^\pi}\left(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}\right) \\ &+ A_{\theta_i}^{\theta_i^\phi}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) - \lambda_i \left( A_{c,\theta_i}^{\theta_i^\phi}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) \right), \end{aligned} \tag{24}$$

where $U_i = \frac{\bar{\phi}_i(\theta_i^{\phi}) \cdot \bar{\pi}_i(\theta_i^{\pi})}{\phi_i(\theta_i^{\phi}) \cdot \pi_i(\theta_i^{\pi})}$ is the importance sampling ratio. For more details on the Lagrangian derivation, see Appendix C.7. Each agent $i$ updates $\theta_i^{\phi}$ and $\theta_i^{\pi}$ by maximizing the surrogate objective in Equation (24), allowing the agent to simultaneously maximize the expected return while minimizing its communication range.

# 4 Experiments

Our experiments aim to answer the following two questions:

- How effective is our method in achieving high performance at low cost? (Subsection 4.2)
- Can the communication policy take effect? If so, how does it work? (Subsection 4.3)

## 4.1 Experiments Setup

**Environments.** We evaluate our method on four representative networked multi-agent scenarios. In these scenarios, influences propagate through intermediate nodes, which naturally aligns with spatial decay property. Furthermore, previous baselines [Duan et al., 2025, Liang et al., 2025] that adhere to this property are also evaluated on these scenarios: (1) Cooperative Adaptive Cruise Control (CACC) [Chu et al., 2020] where the objective is to adaptively coordinate a fleet of vehicles to minimize spacing and speed disturbances based on real-time V2V communication, include Catchup (16 Agents) and Slowdown (8 Agents). (2) Autonomous Vehicle Control (Flow) [Vinitsky et al., 2018], which shares similar objectives with CACC. It includes Ring (22 Agents). (3) Adaptive Traffic Signal Control (ATSC) [Chu et al., 2020], where the objective is to dynamically adjust traffic signal phases based on real-time traffic measurements to reduce congestion. It includes Grid (25 Agents), Monaco (28 Agents) and New York(436 Agents). (4) IEEE Power Grid [Chen et al., 2021] improves voltage control in an islanded microgrid (20 Agents).

**Baselines.** We evaluate the following algorithms in our experiments: (1) CPPO: The centralized PPO with a centralized critic trained using global information. This baseline evaluates agent performance when $\kappa$ is set to its maximum. (2) CPPO-Medium: This baseline is used to simulate the scenario where agents adopt a fixed, moderate communication range. (3) DPPO: The decentralized PPO learns with independent actor and critic using only local observations. This baseline analyzes agent performance when $\kappa$ is minimized. (4) CPPO-Sampling: At each timestep, the communication range $\kappa_{i,t}$ is randomly selected. (5) DPPO-SC: We conduct an additional experiments to ensure a fair comparison under the same communication budget as ACR-PPO. (6) BayesG [Duan et al., 2025], which uses Bayesian variational inference on ego-graphs for communication. (7) GSAC [Liang et al., 2025], which achieves fast generalization via causal representation learning and meta Actor-Critic. All experimental results are based on 10 different random seeds.

More environments and baseline details can be found in Appendix D.1 and D.2.

## 4.2 Policy Performance and Communication Cost Control

Figure 1(a)-(d) show the policy performance of different methods, with the horizontal axis representing steps and the vertical axis representing episode reward. Figure 1(e)-(h) show the average communication cost of different algorithms per agent in the corresponding scenarios. The communication cost is primarily based on [Joo, 2010], assuming that the cost increases exponentially with the expansion of the communication range. Since CPPO and DPPO use a fixed communication range, their corresponding communication costs remain constant.

**Experimental results**. (a) Since our method focuses on adjusting the communication range, its performance is typically between CPPO and DPPO. By introducing a communication policy, our method significantly reduces the communication cost. (b) In the Ring environment, all three methods exhibit almost identical performance, as this environment is less sensitive to communication range, resulting in overlapping performance curves. (c) In the Catchup, Monaco, and PowerGrid environments, DPPO method shows unstable policy performance due to non-stationarity issues, while our method stabilizes learning through dynamic communication range adaptation. (d) The communication cost typically exhibits a "rise first, then decline" phenomenon. This suggests that, in the early training stages, agents increase the average communication range to ensure performance; once performance stabilizes, they

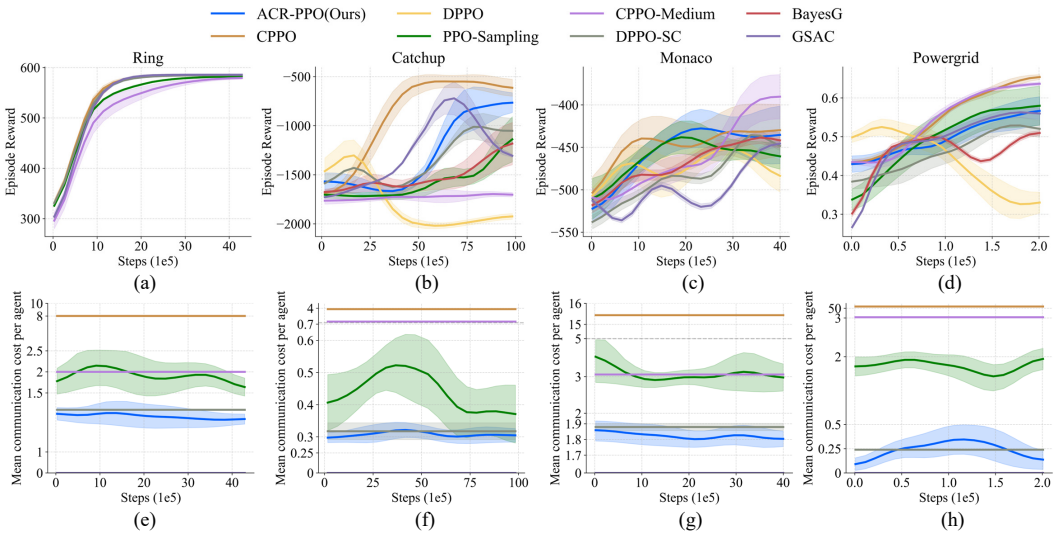



Figure 1: Policy performance and communication cost. (a)-(d) Policy performance of different methods in Ring, Catchup, Monaco, and Powergrid environments. (e)-(h) Average individual communication cost of different methods in the corresponding environments.

subsequently reduce the average communication range to minimize cost. (e) CPPO-Sampling yields decent performance but lacks cost control, primarily because its state-independent sampling cannot dynamically adjust or suppress communication based on situational demands. (f) CPPO-Medium adopts a reduced communication range ($\kappa/2$) by halving the original value. Experimental results show that it still achieves certain performance, but due to an exponential increase in communication cost, the overall communication cost remains high. (g) Both BayesG and GSAC slightly outperform DPPO. BayesG enhances performance by introducing an ego-graph to sample agents within the 1-hop neighborhood, while GSAC leverages meta-learning principles, primarily focusing on policy generalization.

The more results are provided in Appendix D.3(Other scenarios), D.4(Large-scale scenarios) and D.5(Fair and comprehensive evalution). Details on our methods(training time and parameters) are provided in Appendix D.6.

### 4.3 Communication Policy Analysis

**Comparing with fixed $\kappa$-hop DPPO.** Figure 2(a) compares the performance of our method with those of DPPO under fixed communication ranges, given the same number of training episodes. We use different fix $\kappa$ values: $\{0, 2, 4, 6, 8, 10, 12, 14, 15\}$. Figure 2(b) illustrates the average communication cost per agent over a single episode under different communication range $\kappa$. The experimental results show that: (a) Overall, as the communication range increases, the performance of DPPO improves; however, the associated communication cost grows exponentially. (b) Although agents achieve the best performance with global communication ($\kappa = 15$), this comes with significantly higher communication cost. In contrast, our method adaptively adjusts the communication range based on the current communication state, significantly reducing communication cost. While achieving nearly the same performance as the full-communication setting, our method reduces the average communication range to only $\kappa = 2.65$ per episode.

**Communication range change.** In order to further reveal the working mechanism of our method, Figure 2(c) presents the evolution of the average communication range $\kappa$ per agent over step during the last training episode (episode=1200). The red line denotes the average $\kappa$ value under the ACR-PPO, as illustrated in Figure 2(a). The results indicate that agents do not require large-scale communication at every timestep. Maintaining a communication range of approximately $\kappa = 2$ to 4 for most of the episode proves sufficient, and temporarily expanding $\kappa$ at critical timesteps ($t = 102, 336, 516$) can effectively achieve performance comparable to that of continuous large-scale communication.

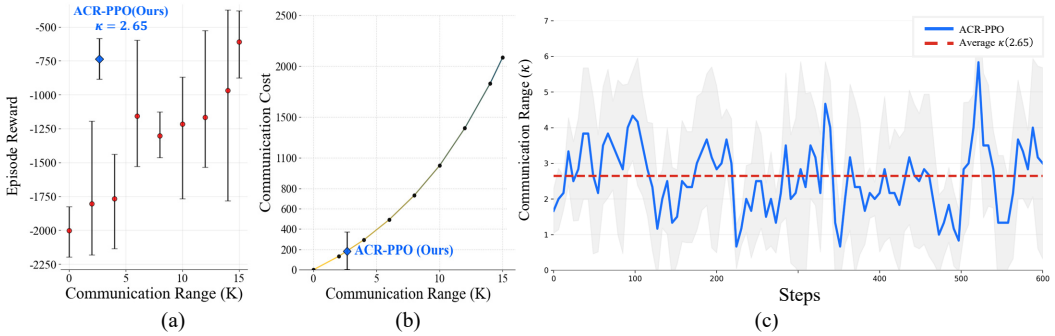



Figure 2: (a) Different fixed $\kappa$-hop DPPO episode reward per agent in Catchup. (b) Different fixed $\kappa$-hop DPPO episode reward per agent in Catchup. (c) Mean Communication range change per agent.

Further analysis on communication range variation can be found in Appendix D.8.

# 5 Related works

## 5.1 Decentralized MARL

As the number of agents increases, existing MARL methods often face scalability challenges. A widely adopted strategy is decentralized MARL. Broadly, decentralized learning methods can be categorized into two types: The first type pursues fully decentralized training, where each agent learns independently without any communication during training [De Witt et al., 2020, Sun et al., 2023, Su and Lu, 2024, Li et al., 2023b]. Although these methods have achieved better performance, even matching centralized methods, they rely on strong assumptions such as knowledge of others' policies or ideal conditions. The second type relaxes full decentralization by allowing agents to communicate with local neighbors, called Networked MARL [Duan et al., 2025, Liang et al., 2025]. Scalable AC [Qu et al., 2020, Zhang et al., 2023] incorporated spatial correlation decay into the learning process, while DMPO [Ma et al., 2024] combined this principle with model-based methods to enhance sample efficiency. In addition, some works in safe MARL [Lu et al., 2021, Ying et al., 2023, Zhang et al., 2024b] also operate under this setting. These studies have significantly advanced the frontier of scalable MARL research. However, most existing methods require manually setting the communication range $\kappa$-hop, which limits their adaptability across varying environments. In contrast, our method follows the same assumptions but uses an adaptive communication mechanism, offering greater flexibility and scalability.

## 5.2 MARL with Communication Constraint

More communication among agents can improve coordination, but it also increases the communication cost [Zhu et al., 2024]. Some MARL methods, including role-based methods [Sheng et al., 2022] and gating-based methods [Mao et al., 2020, Wang et al., 2023], have explored ways to reduce communication cost while maintaining performance. These methods typically rely on global state information or the joint action-value function $Q_{tot}(s, a)$ to regulate communication, making them generally unsuitable for decentralized settings. Sampling-based methods [Lin et al., 2021, Agorio et al., 2025] may offer theoretical guarantees, but they rely on a fixed graph distribution and fail when communication demands shift. Moreover, these sampling methods usually make communication stay around the average of the minimum and maximum communication range.

For a review of related work on sequential update, see Appendix E.1.

# 6 Conclusion

We propose a networked MARL method, called ACR-PPO. The key is that it models communication-aware decision-making under communication budget constraints as a sequential process. More

importantly, our method provides a theoretical guarantee of monotonic performance improvement under communication budget constraints. Experiments across diverse cooperative tasks demonstrate that ACR-PPO maintains policy performance while significantly reducing communication cost through adaptive range control. For future work, we will consider incorporating continual learning into ACR-PPO to enable adaptation to more diverse and dynamic environments.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 62276160, No.62506221), the Natural Science Foundation of Shanxi Province, China (No.202203021211294) and Shanxi Provincial Overseas Study Fund Project (No.20240002).

## References

Leopoldo Agorio, Sean Van Alen, Santiago Paternain, Miguel Calvo-Fullana, and Juan Andres Bazerque. Cooperative multi-agent assignment over stochastic graphs via constrained reinforcement learning. *arXiv preprint arXiv:2502.20462*, 2025.

Dong Chen, Kaian Chen, Zhaojian Li, Tianshu Chu, Rui Yao, Feng Qiu, and Kaixiang Lin. Powernet: multi-agent deep reinforcement learning for scalable powergrid control. *IEEE Transactions on Power Systems*, 37 (2):1007–1017, 2021.

Tianshu Chu, Sandeep Chinchali, and Sachin Katti. Multi-agent reinforcement learning for networked system control. In *ICLR*, 2020.

Jingjing Cui, Yuanwei Liu, and Arumugam Nallanathan. Multi-agent reinforcement learning-based resource allocation for uav networks. *IEEE Transactions on Wireless Communications*, 19(2):729–743, 2019.

Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Dong Deng, Jie Wang, and Liang Zhang. Critical periodic traveling waves for a kermack-mckendrick epidemic model with diffusion and seasonality. *Journal of Differential Equations*, 322:365–395, 2022.

Wei Duan, Jie Lu, En Yu, and Junyu Xuan. Bandwidth-constrained variational message encoding for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2512.11179*, 2025.

Alexei A Fedotov, Peter Harremoës, and Flemming Topsoe. Refinements of pinsker's inequality. *IEEE Transactions on Information Theory*, 49(6):1491–1498, 2003.

Yuanzhe Geng, Erwu Liu, Wei Ni, Rui Wang, Yan Liu, Hao Xu, Chen Cai, and Abbas Jamalipour. Balancing performance and cost for two-hop cooperative communications: Stackelberg game and distributed multi-agent reinforcement learning. *IEEE Transactions on Cognitive Communications and Networking*, 10(6):2193–2208, 2024.

Saeedeh Ghanadbashi and Fatemeh Golpayegani. Using ontology to guide reinforcement learning agents in unseen situations: a traffic signal control system case study. *Applied Intelligence*, 52(2):1808–1824, 2022.

Shangding Gu, Jakub Grudzien Kuba, Yuanpei Chen, Yali Du, Long Yang, Alois Knoll, and Yaodong Yang. Safe multi-agent reinforcement learning for multi-robot control. *Artificial Intelligence*, 319:103905, 2023.

Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. Graph convolutional reinforcement learning. In *ICLR*, 2020.

Er Meng Joo. *New trends in technologies: control, management, computational intelligence and network systems*. BoD–Books on Demand, 2010.

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. In *ICLR*, 2022.

Bin Li, Song Gao, Yunlei Liang, Yuhao Kang, Timothy Prestby, Yuqi Gao, and Runmou Xiao. Estimation of regional economic development indicator from transportation network analytics. *Scientific reports*, 10(1): 2647, 2020.

Chuming Li, Jie Liu, Yinmin Zhang, Yuhong Wei, Yazhe Niu, Yaodong Yang, Yu Liu, and Wanli Ouyang. Ace: cooperative multi-agent q-learning with bidirectional action-dependency. In *AAAI*, 2023a.

Wenhao Li, Bo Jin, Xiangfeng Wang, Junchi Yan, and Hongyuan Zha. F2a2: flexible fully-decentralized approximate actor-critic for cooperative multi-agent reinforcement learning. *Journal of Machine Learning Research*, 24(178):1–75, 2023b.

Hao Liang, Shuqing Shi, Yudi Zhang, Biwei Huang, and Yali Du. Causality meets locality: provably generalizable and scalable policy learning for networked systems. In *NeurIPS*, 2025.

Yiheng Lin, Guannan Qu, Longbo Huang, and Adam Wierman. Multi-agent reinforcement learning in stochastic networked systems. In *NeurIPS*, 2021.

Songtao Lu, Kaiqing Zhang, Tianyi Chen, Tamer Başar, and Lior Horesh. Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning. In *AAAI*, 2021.

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

Hangyu Mao, Zhengchao Zhang, Zhen Xiao, Zhibo Gong, and Yan Ni. Learning agent communication under limited bandwidth by message pruning. In *AAAI*, 2020.

Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.

Guannan Qu, Adam Wierman, and Na Li. Scalable reinforcement learning of localized policies for multi-agent networked systems. In *L4DC*, 2020.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, 2015.

Junjie Sheng, Xiangfeng Wang, Bo Jin, Junchi Yan, Wenhao Li, Tsung-Hui Chang, Jun Wang, and Hongyuan Zha. Learning structured communication for multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 36(2):50, 2022.

Kefan Su and Zongqing Lu. A fully decentralized surrogate for multi-agent policy optimization. *Transactions on Machine Learning Research*, 2024.

Mingfei Sun, Sam Devlin, Jacob Beck, Katja Hofmann, and Shimon Whiteson. Trust region bounds for decentralized ppo under non-stationarity. In *AAMAS*, 2023.

Eugene Vinitsky, Aboudy Kreidieh, Luc Le Flem, Nishant Kheterpal, Kathy Jang, Cathy Wu, Fangyu Wu, Richard Liaw, Eric Liang, and Alexandre M Bayen. Benchmarks for reinforcement learning in mixed-autonomy traffic. In *CoRL*, 2018.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.

Xuefeng Wang, Xinran Li, Jiawei Shao, and Jun Zhang. Ac2c: adaptively controlled two-hop communication for multi-agent reinforcement learning. In *AAMAS*, 2023.

Muning Wen, Jakub Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. In *NeurIPS*, 2022.

Aiwu Yang, Xiaolong Liang, Jiaqiang Zhang, Yueqi Hou, and Ning Wang. Distributed time-varying optimization with coupled constraints: application in uav swarm predefined-time cooperative consensus. *Aerospace Science and Technology*, 147:109034, 2024.

Yuxuan Yi, Ge Li, Yaowei Wang, and Zongqing Lu. Learning to share in networked multi-agent reinforcement learning. In *NeurIPS*, 2022.

Donghao Ying, Yunkai Zhang, Yuhao Ding, Alec Koppel, and Javad Lavaei. Scalable primal-dual actor-critic method for safe multi-agent rl with general utilities. In *NeurIPS*, 2023.

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. In *NeurIPS*, 2022.

Bin Zhang, Hangyu Mao, Lijuan Li, Zhiwei Xu, Dapeng Li, Rui Zhao, and Guoliang Fan. Sequential asynchronous action coordination in multi-agent systems: a stackelberg decision transformer approach. In *ICML*, 2024a.

Haifeng Zhang, Weizhe Chen, Zeren Huang, Minne Li, Yaodong Yang, Weinan Zhang, and Jun Wang. Bi-level actor-critic for multi-agent coordination. In *AAAI*, 2020.

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

Lijun Zhang, Lin Li, Wei Wei, Huizhong Song, Yaodong Yang, and Jiye Liang. Scalable constrained policy optimization for safe multi-agent reinforcement learning. In *NeurIPS*, 2024b.

Yizhou Zhang, Guannan Qu, Pan Xu, Yiheng Lin, Zaiwei Chen, and Adam Wierman. Global convergence of localized policy iteration in networked multi-agent reinforcement learning. In *POMACS*, 2023.

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.

Changxi Zhu, Mehdi Dastani, and Shihan Wang. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38(1):4, 2024.

# A Preliminary lemmas

Before proving the main lemmas, we first establish a series of intermediate results.

## A.1 Bound on the Difference of Expectations

**Lemma A.1.** *For any agent $i \in \mathcal{N}$, let $\boldsymbol{s} = (s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa})$ and $\boldsymbol{s}' = (s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa})$ denote two states that differ only in the states of agents outside the communication neighborhood of agent $i$. Let $\boldsymbol{\mu}, \boldsymbol{\nu}$ be two joint distributions over the state space $\mathcal{S}$, and let $\mu^i, \nu^i$ be the corresponding marginal distributions over $\mathcal{S}^i$ for agent $i$. Define $\delta^i(f) = \sup_{s^i, s^{-i}, s'^{-i}} \left| f(s^i, s^{-i}) - f(s^i, s'^{-i}) \right|$, which measures the maximum variation of function $f(s)$ with respect to changes in the states of agents other than $i$, while keeping agent $i$'s state fixed. Let the L1 norm be defined as $\|f\|_1 = \int |f(s)|\, ds$. Then,*

$$|\mathbb{E}_{s\sim\boldsymbol{\mu}}[f(\boldsymbol{s})] - \mathbb{E}_{s\sim\boldsymbol{\nu}}[f(\boldsymbol{s})]| \leq \sum_{i\in\mathcal{N}} D_{\mathrm{TV}}(\mu^i, \nu^i) \cdot \delta^i(f). \tag{25}$$

*Proof.* We first present the mathematical expression for the total variation distance $D_{TV}$:

$$\begin{aligned} D_{TV}(\boldsymbol{\mu}, \boldsymbol{\nu}) &= \frac{1}{2} \sup_h |\mathbb{E}_{\boldsymbol{\mu}}(h) - \mathbb{E}_{\boldsymbol{\nu}}(h)| \\ &= \frac{1}{2} \int |\boldsymbol{\mu}(s) - \boldsymbol{\nu}(s)| ds. \end{aligned} \tag{26}$$

In multi-agent systems, the number of agents is typically greater than one, i.e., $N = |\mathcal{N}| \geq 2$. Based on this, we can obtain:

$$\begin{aligned} |\mathbb{E}_{\boldsymbol{\mu}} f - \mathbb{E}_{\boldsymbol{\nu}} f| &= \left| \int f(s)(\boldsymbol{\mu}(s) - \boldsymbol{\nu}(s)) ds \right| \\ &\leq \int |f(s)||\boldsymbol{\mu}(s) - \boldsymbol{\nu}(s)| ds \\ &\leq \|f\|_\infty \int |\boldsymbol{\mu}(s) - \boldsymbol{\nu}(s)| ds \\ &= 2\|f\|_\infty D_{TV}(\boldsymbol{\mu}, \boldsymbol{\nu}). \end{aligned} \tag{27}$$

Furthermore, using a coordinate-wise telescoping argument, we obtain

$$|\mathbb{E}_{\boldsymbol{\mu}} f - \mathbb{E}_{\boldsymbol{\nu}} f| \leq \sum_{i=1}^{\mathcal{N}} D_{TV}(\mu^i, \nu^i) \cdot \delta^i(f). \tag{28}$$

The proof is complete.

### A.2 The Lemma of Markov Chain

In this section, we introduce the key lemmas of Markov chain properties.

**Lemma A.2** (Lemma 14 for [Zhang et al., 2023]). *Consider a Markov Chain with state $z = (z_1, ..., z_n) \in \mathcal{Z}_1 \times ... \times \mathcal{Z}_n$, where each $\mathcal{Z}_i$ is some finite set. Suppose its transition probability factorizes as:*

$$P(z(t+1)|z(t)) = \prod_{i=1}^{n} P_i(z_i(t+1)|z_{\mathcal{N}_i}(t)),$$

*and define $C^z$ to be the following matrix:*

$$C^z_{ij} = \sup_{z_{-j}} \sup_{z_j, z'_j} TV(P_i(\cdot|z_j, z_{-j}), P_i(\cdot|z'_j, z_{-j})). \tag{29}$$

*Then, fixing a pair of $(i, j)$ for any $z = (z_j, z_{-j})$, $z' = (z'_j, z_{-j})$, the following holds:*

*(1) We have $TV(\upsilon_{i,t}, \upsilon'_{i,t}) \leq [(C^z)^t e_j]_i$, where $\upsilon_{i,t}$ is the distribution of $z_{i,t}$ given $z(0) = z$, and $\upsilon'_{i,t}$ is the distribution of $z_{i,t}$ given $z(0) = z'$, and $e_j$ is the indicator vector in which the $j$'th entry is 1 and all other entries are 0.*

*(2) For any function $f : \mathbb{R}^{\mathcal{Z}} \to \mathbb{R}$, we have $|\mathbb{E}_{z\sim\upsilon_t} f(z) - \mathbb{E}_{z\sim\upsilon'_t} f(z)| \leq \sum_{l=1}^{n} [(C^z)^t e_j]_l \delta(f)$, where $\delta_l(f) = \sup_{z_{-l}} \sup_{z_l, z'_l} |f(z_l, z_{-l}) - f(z'_l, z_{-l})|$ and $\upsilon_t$ is the distribution of $z(t)$ given $z(0) = z$, and $\upsilon'_t$ is the distribution of $z(t)$ given $z(0) = z'$.*

Lemma A.2 quantifies how perturbations in the initial state of agent $j$ j propagate through the network over time, bounding the resulting deviation in both marginal distributions and expected function values via the powers of the interaction matrix $C^z$.

### A.3 The Definition of Exponential Decay Matrix

**Definition A.3** (Exponential Decay Matrix). Let $\rho \in (0, 1)$ and $c > 0$. A matrix $A \in \mathbb{R}^{n\times n}$ is said to be $(c, \rho)$-exponentially decaying with respect to the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ if for all $i, j \in \mathcal{N}$:

$$A_{ij} \leq c \cdot \rho^{\text{dist}(i,j)},$$

where dist$(i, j)$ denotes the shortest path distance between nodes $i$ and $j$ in $\mathcal{G}$.

This definition formalizes the notion of spatial locality, ensuring that the interaction strength between any two agents diminishes exponentially as their graph distance increases. Then, we introduce the following assumptions regarding the graph structure, policy, and environment dynamics.

### A.4 The Assumption of Graph Structure, Policy and Dynamics

**Assumption A.4** (Graph Structure and System Stability). Under our framework, the underlying graph structure is required to satisfy the following assumptions:

- The interaction graph $\mathcal{G}$ has a bounded maximum degree $\Delta$.
- The policy interaction matrix $Z^\psi$ is $(c_Z, \rho)$-exponentially decaying.
- The closed-loop state transition interaction matrix $C^z$ (capturing both environment dynamics and policy dependencies) is $(c_{C^z}, \rho)$-exponentially decaying.
- The system satisfies the stability condition $\gamma\lambda(C^z) < 1$, where $\lambda(C^z)$ is the spectral radius of $C^z$.

**Assumption A.5** (Exponential Decay of Policy and Dynamics). We assume that the policy interactions and state transition couplings decay exponentially with graph distance. The policy interaction matrix $Z^\psi \in \mathbb{R}^{n\times n}$, where entry $Z^\psi_{ij}$ measures the sensitivity of agent $i$'s action distribution to changes in agent $j$'s state (i.e., $Z^\psi_{ij} = \sup_{s_{-j}, s_j, s'_j} TV(\psi_i(\cdot|s_j, s_{-j}), \psi_i(\cdot|s'_j, s_{-j}))$), is $(c_Z, \rho_Z)$-exponentially decaying. The induced state transition interaction matrix $C^z \in \mathbb{R}^{n\times n}$, where entry $C^z_{ij}$ bounds the influence of agent $j$'s current state on agent $i$'s next state distribution under policy $\psi$, is $(c_{C^z}, \rho_{C^z})$-exponentially decaying. Let $\rho = \max(\rho_Z, \rho_{C^z}) < 1$.

Assumption A.4 collectively ensure that the system remains stable and that both policy and dynamics interactions are sufficiently localized, which is a prerequisite for establishing the exponential decay of value functions. Assumption A.5 posits that both the policy's sensitivity to state changes and the environment's dynamic couplings are spatially localized, decaying exponentially with the distance between agents.

### A.5 The bound of interaction matrix of local value function

We aim to bound the interaction matrix of the local value function, $Z^{V^\psi}$, defined as:

$$Z_{ij}^{V^\psi} = \sup_{s_{-j}} \sup_{s_j, s_j'} \left| V_i^\psi(s_j, s_{-j}) - V_i^\psi(s_j', s_{-j}) \right|.$$

**Lemma A.6** (Exponential Decay of $V^\psi$)**.** *Under Assumption A.5, the interaction matrix $Z^{V^\psi}$ is $(c_V, \rho)$-exponentially decaying, where $c_V$ is a constant depending on $\bar{r}, \gamma, c_Z, c_{C^z}$, and the spectral properties of $C^z$. Specifically:*

$$Z_{ij}^{V^\psi} \leq c_V \cdot \rho^{dist(i,j)}.$$

Lemma A.6 confirms that the spatial locality inherent in the policy and dynamics propagates to the value function, ensuring that long-term rewards are primarily determined by nearby agents. The relevant proof is as follows:

*Proof.* Recall the definition of the local value function:

$$V_i^\psi(s) = \mathbb{E}\left[ \sum_{t=0}^{\infty} \gamma^t r_i(s_{i,t}, a_{i,t}) \mid s(0) = s \right]. \tag{30}$$

Let $f_i(s) = \mathbb{E}_{a \sim \psi(\cdot|s)}[r_i(s_i, a_i)]$ be the expected immediate reward for agent $i$. Consider two global states $\mathbf{s} = (s_j, s_{-j})$ and $\mathbf{s}' = (s_j', s_{-j})$ that differ only at agent $j$'s state. Let $b_t$ and $b_t'$ denote the distributions of the global state $s(t)$ at time $t$ given initial states $s(0) = s$ and $s(0) = s'$, respectively. By the triangle inequality and linearity of expectation:

$$\left| V_i^\psi(s) - V_i^\psi(s') \right| \leq \sum_{t=0}^{\infty} \gamma^t \left| \mathbb{E}_{s(t) \sim b_t}[f_i(s(t))] - \mathbb{E}_{s(t) \sim b_t'}[f_i(s(t))] \right|. \tag{31}$$

We apply Lemma A.2(b), for any function $f : \mathcal{S} \to \mathbb{R}$, we have:

$$\left| \mathbb{E}_{\pi_t}[f] - \mathbb{E}_{\pi_t'}[f] \right| \leq \sum_{l=1}^{n} [(C^z)^t]_{jl} \delta_l(f), \tag{32}$$

where $\delta_l(f) = \sup_{s_{-l}} \sup_{s_l, s_l'} |f(s_l, s_{-l}) - f(s_l', s_{-l})|$ measures the sensitivity of $f$ to changes in agent $l$'s state. According to Lemma A.1, for $f = f_i$, the sensitivity is bounded by the policy's dependence on state $l$ scaled by the reward bound:

$$\delta_l(f_i) \leq \bar{r} \cdot \sup_{s_{-l}} \sup_{s_l, s_l'} TV(\psi_i(\cdot|s_l, s_{-l}), \psi_i(\cdot|s_l', s_{-l})) = \bar{r} Z_{il}^\psi. \tag{33}$$

Combining with Equ (31), (32), (33), we can get:

$$\left| V_i^\psi(s) - V_i^\psi(s') \right| \leq \sum_{t=0}^{\infty} \gamma^t \sum_{l=1}^{n} [(C^z)^t]_{jl} \bar{r} Z_{il}^\psi = \bar{r} \sum_{t=0}^{\infty} \gamma^t \left[ Z^\psi (C^z)^t \right]_{ij}. \tag{34}$$

Thus, the interaction matrix satisfies the element-wise inequality:

$$Z^{V^\psi} \preceq \bar{r} \sum_{t=0}^{\infty} \gamma^t Z^\psi (C^z)^t. \tag{35}$$

To establish the exponential decay of $Z^{V^\psi}$, we analyze the term $M_t = Z^\psi (C^z)^t$. Under Assumption A.5, $Z^\psi$ is $(c_Z, \rho)$-decaying and $C^z$ is $(c_{C^z}, \rho)$-decaying. It is a standard result in the analysis of spatially decaying systems that the product of two exponentially decaying matrices is also exponentially

decaying with the same rate $\rho$ (up to a constant factor dependent on the graph degree). Specifically, there exists a constant $K_{prod}$ such that:

$$[Z^{\psi}(C^z)^t]_{ij} \leq K_{prod} \cdot \lambda(C^z)^t \cdot \rho^{\text{dist}(i,j)}, \tag{36}$$

where $\lambda(C^z)$ is the spectral radius of $C^z$. Note that the spatial decay rate $\rho$ is preserved because the convolution of exponential kernels $\sum_k \rho^{d(i,k)}\rho^{d(k,j)}$ is dominated by $\rho^{d(i,j)}$ multiplied by a polynomial factor in the number of paths, which is absorbed into the constant $K_{prod}$ for graphs with bounded growth. Substituting the bound for $M_t$ into Equation (35):

$$Z_{ij}^{V^{\psi}} \leq \bar{r} \sum_{t=0}^{\infty} \gamma^t \left( K_{prod} \lambda(C^z)^t \rho^{\text{dist}(i,j)} \right) = \bar{r} K_{prod} \rho^{\text{dist}(i,j)} \sum_{t=0}^{\infty} (\gamma \lambda(C^z))^t. \tag{37}$$

$\sum_{t=0}^{\infty}(\gamma\lambda(C^z))^t$ converges to $\frac{1}{1-\gamma\lambda(C^z)}$ provided that $\gamma\lambda(C^z) < 1$. This condition is satisfied if the system is stable and the discount factor $\gamma$ is sufficiently small relative to the mixing rate of the induced chain. Let $c_V = \frac{\bar{r}K_{prod}}{1-\gamma\lambda(C^z)}$. We can obtain:

$$Z_{ij}^{V^{\psi}} \leq c_V \cdot \rho^{\text{dist}(i,j)}.$$

This completes the proof that $Z^{V^{\psi}}$ is $(c_V, \rho)$-exponentially decaying.

# B Preliminaries

## B.1 Communication cost

In networked cooperative multi-agent systems, multi-hop communication enables the connection of more agents. However, the communication cost also increases exponentially. In this paper, we follow the setting of [Joo, 2010] and model the communication cost as an exponentially increasing function of the communication range:

$$c_{i,t} = e^{\alpha \kappa_{i,t}} - 1, \tag{38}$$

where $\alpha$ is a hyperparameter, setting to 0.1.

## B.2 Joint State-Action Value Function

The detail of the joint state-action value function is as follows:

$$\begin{aligned} Q(\mathbf{s}, \mathbf{a}) &= \mathbb{E}_{\mathbf{a}\sim\psi}[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}(t), \mathbf{a}(t)) | \mathbf{s}(0) = \mathbf{s}, \mathbf{a}(0) = \mathbf{a}] \\ &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\mathbf{a}\sim\psi}[\sum_{t=0}^{\infty} \gamma^t r_i(s_{i,t}, a_{i,t}) | \mathbf{s}(0) = \mathbf{s}, \mathbf{a}(0) = \mathbf{a}] \\ &:= \frac{1}{n} \sum_{i=1}^{n} Q_i^{\theta}(\mathbf{s}, \mathbf{a}). \end{aligned} \tag{39}$$

## B.3 Multi-agent Policy Gradient

**Lemma B.1.** *Let $\nu^{\theta}$ be a distribution over the state space defined as $\nu^{\theta}(\boldsymbol{s}) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \nu_t^{\theta}(\boldsymbol{s})$, where $\pi^{\theta}(s)$ represents the distribution of $\boldsymbol{s}(t)$ under a fixed joint policy parameter $\theta = (\theta_1, \theta_2, ..., \theta_n)$, given that $\boldsymbol{s}(0)$ is sampled from $\pi_0$. $\psi$ is the joint policy. Then we have,*

$$\nabla J(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{\boldsymbol{s}\sim\nu^{\theta}, \boldsymbol{a}\sim\psi(\cdot|\boldsymbol{s}(t))} \left[ \nabla \log \psi(\boldsymbol{a}|\boldsymbol{s}) Q^{\theta}(\boldsymbol{s}, \boldsymbol{a}) \right]. \tag{40}$$

### B.4 The proof of Lemma 2.2

*Proof.* For notational simplicity, we define $\mathbf{s} = (s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa})$, $\mathbf{a} = (a_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_{-i}^\kappa})$, $\mathbf{s}' = (s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa})$, $\mathbf{a}' = (a_{\mathcal{N}_i^\kappa}, a'_{\mathcal{N}_{-i}^\kappa})$. Let $\iota_{i,t}$ denote the distribution $(s_{i,t}, a_{i,t})$ at timestep $t$ under policy $\psi$ starting from $(\mathbf{s}(0), \mathbf{a}(0)) = (s, a)$, and let $\iota'_{i,t}$ denote the corresponding distribution starting from $(\mathbf{s}(0), \mathbf{a}(0)) = (\mathbf{s}', \mathbf{a}')$. Then, we have $\iota_{i,t} = \iota'_{i,t}$ for all $t \leq \kappa$. The reason is that, due to the local dependence structure in Equation (1) and the localized policy structure, $\iota_{i,t}$ only depends on $(s_{\mathcal{N}_i^t}, a_{\mathcal{N}_i^t})$ which is the same as $(s'_{\mathcal{N}_i^t}, a'_{\mathcal{N}_i^t})$ when $t \leq \kappa$ per the way the initial state $(\mathbf{s}, \mathbf{a})$, $(\mathbf{s}', \mathbf{a}')$ are chosen. With these definitions, we introduce the truncated $Q$-function:

$$
\begin{aligned}
&|Q_i^\psi(\mathbf{s}, \mathbf{a}) - Q_i^\psi(\mathbf{s}', \mathbf{a}')| \\
&= \sum_{t=0}^{\infty} |\gamma^t \mathbb{E}_{(s_i, a_{i,t}) \sim \iota_{i,t}} r_i(s_{i,t}, a_{i,t}) - \gamma^t \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \iota'_{i,t}} r_i(s'_i, a'_i)| \\
&= \sum_{t=0}^{\kappa} |\gamma^t \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \iota_{i,t}} r_i(s_{i,t}, a_{i,t}) - \gamma^t \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \iota'_{i,t}} r_i(s_{i,t}, a_{i,t})| \\
&\quad + \sum_{t=\kappa+1}^{\infty} |\gamma^t \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \iota_{i,t}} r_i(s_{i,t}, a_{i,t}) - \gamma^t \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \iota'_{i,t}} r_i(s_{i,t}, a_{i,t})| \qquad (41) \\
&= \sum_{t=\kappa+1}^{\infty} \gamma^t |\mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \iota_{i,t}} r_i(s_{i,t}, a_{i,t}) - \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \iota'_{i,t}} r_i(s_{i,t}, a_{i,t})| \\
&\leq \sum_{t=\kappa+1}^{\infty} \gamma^t \bar{r} D_{TV}(\iota_{i,t}, \iota'_{i,t}) \leq \frac{\bar{r}}{1-\gamma} \gamma^{\kappa+1}. \qquad (42)
\end{aligned}
$$

In Equation (41), the first term equals 0 because $\iota_{i,t} = \iota'_{i,t}$ for all $t \leq \kappa$. The derivation of the final step is provided in Appendix A.1. Equation (42) holds because the TV distance is bounded by 1. Detailed descriptions of this part can be found in [Qu et al., 2020].

### B.5 The proof of Lemma 2.3

*Proof.* We show that the truncated $Q$-function is a good approximation of the global $Q$-function. Let $(c = \frac{\bar{r}}{1-\gamma}, \rho = \gamma)$. From the above bound, it follows that:

$$
\begin{aligned}
&|Q_i(s_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_i^\kappa}) - Q_i(\mathbf{s}, \mathbf{a})| \\
&= |\sum_{s'_{\mathcal{N}_{-i}^\kappa}, a'_{\mathcal{N}_{-i}^\kappa}} \omega_i(s'_{\mathcal{N}_{-i}^\kappa}, a'_{\mathcal{N}_{-i}^\kappa}; s_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_i^\kappa}) Q_i(s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_i^\kappa}, a'_{\mathcal{N}_{-i}^\kappa}) - Q_i(s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_{-i}^\kappa})| \\
&\leq \sum_{s'_{\mathcal{N}_{-i}^\kappa}, a'_{\mathcal{N}_{-i}^\kappa}} \omega_i(s'_{\mathcal{N}_{-i}^\kappa}, a'_{\mathcal{N}_{-i}^\kappa}; s_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_i^\kappa}) |Q_i(s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_i^\kappa}, a'_{\mathcal{N}_{-i}^\kappa}) - Q_i(s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa}, a_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_{-i}^\kappa})| \\
&\leq c\rho^{\kappa+1}. \qquad (43)
\end{aligned}
$$

Truncated $Q$-function proof completed.

Next, we begin to prove Truncated $V$- function. As a direct consequence of Lemma A.6, the error incurred by truncating the global state to the $\kappa$-hop neighborhood decays exponentially with $\kappa$. For any agent $i$, let $s_{\mathcal{N}_i^\kappa}$ denote the states of agents within $\kappa$ hops, and $s_{\mathcal{N}_{-i}^\kappa}$ denote the states of agents outside this neighborhood. Then:

$$
\sup_{s_{\mathcal{N}_i^\kappa}} \sup_{s_{\mathcal{N}_{-i}^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}} \left| V_i^\psi(s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa}) - V_i^\psi(s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}) \right| \leq \sum_{j \notin \mathcal{N}_i^\kappa} Z_{ij}^{V^\psi} \leq C' \rho^\kappa, \qquad (44)
$$

where $C'$ is a constant depending on $c_V$ and the graph structure. This implies that for any fixed local state $s_{\mathcal{N}_i^\kappa}$ and any pair of distant states $s_{\mathcal{N}_{-i}^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}$, the following holds:

$$
\left| V_i^\psi(s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa}) - V_i^\psi(s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}) \right| \leq C' \rho^\kappa. \qquad (45)
$$

Similar to Equation (4), we can get:

$$\widetilde{V}_i^{\psi}(s_{\mathcal{N}_i^\kappa}) = \sum_{s'_{\mathcal{N}_{-i}^\kappa}} \chi_i(s'_{\mathcal{N}_{-i}^\kappa}; s_{\mathcal{N}_i^\kappa}) V_i^{\psi}(s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}), \tag{46}$$

where $\chi_i(\cdot)$ are any non-negative weights satisfying $\sum_{s'_{\mathcal{N}_{-i}^\kappa}} \chi_i(s'_{\mathcal{N}_{-i}^\kappa}; s_{\mathcal{N}_i^\kappa}) = 1, \quad \forall s_{\mathcal{N}_i^\kappa} \in \mathcal{S}_{\mathcal{N}_i^\kappa}$, and:

$$\begin{aligned}
|\widetilde{V}_i^{\psi}(s_{\mathcal{N}_i^\kappa}) - V_i^{\psi}(s)| &= \left| \sum_{s'_{\mathcal{N}_{-i}^\kappa}} \chi_i(s'_{\mathcal{N}_{-i}^\kappa}; s_{\mathcal{N}_i^\kappa}) V_i^{\psi}(s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}) - V_i^{\psi}(s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa}) \right| \\
&\leq \sum_{s'_{\mathcal{N}_{-i}^\kappa}} \chi_i(s'_{\mathcal{N}_{-i}^\kappa}; s_{\mathcal{N}_i^\kappa}) \left| V_i^{\psi}(s_{\mathcal{N}_i^\kappa}, s'_{\mathcal{N}_{-i}^\kappa}) - V_i^{\psi}(s_{\mathcal{N}_i^\kappa}, s_{\mathcal{N}_{-i}^\kappa}) \right| \\
&\leq C_V \cdot \rho^{\kappa+1}.
\end{aligned} \tag{47}$$

Truncated $V$-function proof completed.

Next, by combining Equations (43) and (47), we can obtain:

$$\begin{aligned}
|\widetilde{A}_i^{\psi}(s_{\mathcal{N}_\kappa^i}, a_{\mathcal{N}_\kappa^i}) - A_i^{\psi}(\mathbf{s}, \mathbf{a})| &= |(\widetilde{Q}_i^{\psi}(s_{\mathcal{N}_\kappa^i}, a_{\mathcal{N}_\kappa^i}) - \widetilde{V}_i^{\psi}(s_{\mathcal{N}_\kappa^i})) - (Q_i^{\psi}(\mathbf{s}, \mathbf{a}) - V_i^{\psi}(\mathbf{s}))| \\
&= |(\widetilde{Q}_i^{\psi}(s_{\mathcal{N}_\kappa^i}, a_{\mathcal{N}_\kappa^i}) - Q_i^{\psi}(\mathbf{s}, \mathbf{a})) + (V_i^{\psi}(\mathbf{s}) - \widetilde{V}_i^{\psi}(s_{\mathcal{N}_\kappa^i}))| \\
&\leq |\widetilde{Q}_i^{\psi}(s_{\mathcal{N}_\kappa^i}, a_{\mathcal{N}_\kappa^i}) - Q_i^{\psi}(\mathbf{s}, \mathbf{a})| + |\widetilde{V}_i^{\psi}(s_{\mathcal{N}_\kappa^i}) - V_i^{\psi}(\mathbf{s})| \\
&\leq (c + C_V)\rho^{\kappa+1} = C_A \cdot \rho^{\kappa+1}.
\end{aligned} \tag{48}$$

Advantage function proof completed.

### B.6 MAPPO loss function

The loss function of MAPPO [Yu et al., 2022] is defined as:

$$\begin{aligned}
L_{actor}(\theta_i) &= J(\theta_i) - J(\theta_i^{old}) \\
&= \frac{1}{B} \sum_{b=1}^{B} \min \left[ \frac{\psi_{\theta_i}(a_i|o_i)}{\psi_{\theta_i^{old}}(a_i|o_i)}, \text{clip} \left( \frac{\psi_{\theta_i}(a_i|o_i)}{\psi_{\theta_i^{old}}(a_i|o_i)}, 1-\epsilon, 1+\epsilon \right) \right] A(\mathbf{s}, \mathbf{a}).
\end{aligned} \tag{49}$$

$$L_{critic}(\theta_i) = \frac{1}{B} \sum_{b=1}^{B} \left( \max \left[ (V_{\theta_i} - R)^2, (\text{clip}(V_{\theta_i}, V_{\theta_i^{old}} - \epsilon, V_{\theta_i^{old}} + \epsilon) - R)^2 \right] \right). \tag{50}$$

The advantage function is defined as $A(\mathbf{s}, \mathbf{a}) = r_t(\mathbf{s}, \mathbf{a}) + V(\mathbf{s}_{t+1}) - V(\mathbf{s}_t)$, with $B$ denoting the batch size. Agents share a common actor and critic network to enhance training efficiency.

### B.7 Scalable MAPPO loss function

$L_a$ is Actor loss, and $L_c$ is Critic loss:

$$\begin{aligned}
L_a(\theta_i) &= J(\theta_i) - J(\theta_i^{old}) \\
&= \frac{1}{B} \sum_{b=1}^{B} \min \left[ \frac{\psi_{\theta_i}(a_i|o_i)}{\psi_{\theta_i^{old}}(a_i|o_i)}, \text{clip} \left( \frac{\psi_{\theta_i}(a_i|o_i)}{\psi_{\theta_i^{old}}(a_i|o_i)}, 1-\epsilon, 1+\epsilon \right) \right] \widetilde{A}(s_{\mathcal{N}_i^\kappa}, a_{\mathcal{N}_i^\kappa}).
\end{aligned} \tag{51}$$

$$\begin{aligned}
&L_c(\theta_i) \\
&= \frac{1}{B} \sum_{i=1}^{B} (\max[(\widetilde{V}_{\theta_i}(s_{\mathcal{N}_i^\kappa}) - R)^2, (\text{clip}(\widetilde{V}_{\theta_i}(s_{\mathcal{N}_i^\kappa}), \widetilde{V}_{\theta_i^{old}}(s_{\mathcal{N}_i^\kappa}) - \epsilon, \widetilde{V}_{\theta_i^{old}}(s_{\mathcal{N}_i^\kappa}) + \epsilon) - R)^2]).
\end{aligned} \tag{52}$$

# C Methods

## C.1 The proof of Proposition 3.2

*Proof.* Let $s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}$ and $a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}$ represent the states and actions of agent $i$ and its $\kappa_{i,t}$-hop neighbors at timestep $t$. Conversely, $s_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}$ and $a_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}$ denote the states and actions of all agents outside this neighbor. Let the truncated joint state-action tuples be denoted as $\mathbf{s} = (s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, s_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}), \quad \mathbf{a} = (a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}), \quad \mathbf{s}' = (s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, s'_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}), \quad \mathbf{a}' = (a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a'_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}})$. Let $\zeta_{i,u}$ denote the state-action visitation distribution at timestep $t$ under policy $\psi_i$ starting from $(\mathbf{s}, \mathbf{a})$, and let $\zeta'_{i,u}$ denote the corresponding distribution starting from $(\mathbf{s}', \mathbf{a}')$. Similarly, we assume $\zeta_{i,u} = \zeta'_{i,u}$ for all $u \leq \kappa_{i,t}$. With definitions, we now define the truncated $Q$-function at timestep $t$:

$$
\begin{aligned}
&|Q_i(\mathbf{s}, \mathbf{a}) - Q_i(\mathbf{s}', \mathbf{a}')| \\
&= \mathbb{E}_{\kappa_{i,t} \sim \phi_i(s_{i,t})} \left[ \sum_{u=0}^{\infty} |\gamma^u \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \zeta_{i,u}} r_i(s_{i,t}, a_{i,t}) - \gamma^u \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \zeta'_{i,u}} r_i(s'_i, a'_i)| \right] \\
&\leq \mathbb{E}_{\kappa_{i,t} \sim \phi_i(s_{i,t})} \left[ \sum_{u=0}^{\infty} \gamma^u \left| \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \zeta_{i,u}} r_i(s_{i,t}, a_{i,t}) - \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \zeta'_{i,u}} r_i(s_{i,t}, a_{i,t}) \right| \right] \\
&= \mathbb{E}_{\kappa_{i,t} \sim \phi_i(s_{i,t})} \left[ \sum_{u=\kappa_{i,t}+1}^{\infty} \gamma^u \left| \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \zeta_{i,u}} r_i(s_{i,t}, a_{i,t}) - \mathbb{E}_{(s_{i,t}, a_{i,t}) \sim \zeta'_{i,u}} r_i(s_{i,t}, a_{i,t}) \right| \right] \\
&\leq \mathbb{E}_{\kappa_{i,t} \sim \phi_i(s_{i,t})} \left[ \sum_{u=\kappa_{i,t}+1}^{\infty} \gamma^u \bar{r} TV(\zeta_{i,u}, \zeta'_{i,u}) \right] \\
&\leq \mathbb{E}_{\kappa_{i,t} \sim \phi_i(s_{i,t})} \left[ \frac{\bar{r}}{1-\gamma} \gamma^{\kappa_{i,t}+1} \right].
\end{aligned} \tag{53}
$$

Let $\eta = \frac{\bar{r}}{1-\gamma}, \sigma = \gamma$. Then, we obtain:

$$
|Q_i(\mathbf{s}, \mathbf{a}) - Q_i(\mathbf{s}', \mathbf{a}')| \leq \mathbb{E}_{\kappa_{i,t} \sim \phi_i(s_{i,t})} \eta \sigma^{\kappa_{i,t}+1}. \tag{54}
$$

By definition, the truncated Q-function $Q_{\psi_i}^{\phi_i, \pi_i}$ approximates the global Q-value using only the local neighborhood information determined by the sampled radius $\kappa_{i,t}$. This is equivalent to evaluating the global Q-function at a perturbed state $\mathbf{s}'$ where distant components are ignored.

Applying the spatial decay bound derived in Lemma 2.3 for a fixed $\kappa$, and taking the expectation over the sampling distribution $\kappa_{i,t} \sim \phi_i(s_{i,t})$, we obtain:

$$
\left| Q_{\psi_i}^{\phi_i, \pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) - Q_{\psi_i}(\mathbf{s}, \mathbf{a}) \right| \leq \mathbb{E}_{\kappa_{i,t} \sim \phi_i(s_{i,t})} \left[ \eta \sigma^{\kappa_{i,t}+1} \right]. \tag{55}
$$

This completes the proof.

## C.2 The proof of Corollary 3.3

*Proof.* Let $s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}$ and $a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}$ represent the states and actions of agent $i$ and its $\kappa_{i,t}$-hop neighbors at timestep $t$. Conversely, $s_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}$ and $a_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}$ denote the states and actions of all agents outside this neighbor. For notational simplicity, we drop the timestep index and define the truncated joint state-action tuples as $\mathbf{s} = (s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, s_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}), \quad \mathbf{a} = (a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}), \quad \mathbf{s}' = (s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, s'_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}}), \quad \mathbf{a}' = (a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a'_{\mathcal{N}_{i,t}^{-\kappa_{i,t}}})$. Let $\rho_{t,i,\kappa}$ denote the state-action visitation distribution at timestep $t$ under policy $\psi$, starting from $(s(0), a(0)) = (s_\kappa, a_\kappa)$, and let $\rho'_{t,i,\kappa}$ be the corresponding distribution starting from $(s(0), a(0)) = (s'_\kappa, a'_\kappa)$. We assume that $\rho_{t,i,\kappa} = \rho'_{t,i,\kappa}$ for all $t \leq \kappa$, meaning that the influence of distant agents outside $\mathcal{N}_i^\kappa$ vanishes within $\kappa$ steps. Then, for any agent $i$, and under the assumption

that the states and actions of distant agents differ, we obtain the following bound:

$$\begin{aligned}
&\left|A_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) - A_{\psi_i}(\mathbf{s},\mathbf{a})\right| \\
&= \left|(Q_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) - V_{\psi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}})) - (Q_{\psi_i}(\mathbf{s},\mathbf{a}) - V_{\psi_i}(\mathbf{s}))\right| \\
&= \left|Q_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) - Q_{\psi_i}(\mathbf{s},\mathbf{a})) + (V_{\psi_i}(\mathbf{s}) - V_{\psi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}))\right| \\
&\leq \underbrace{\left|Q_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) - Q_{\psi_i}(\mathbf{s},\mathbf{a})\right|}_{(1)} + \underbrace{\left|V_{\psi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}) - V_{\psi_i}(\mathbf{s})\right|}_{(2)}.
\end{aligned} \tag{56}$$

We analyze these two terms separately. For term (1), the proof can be found in Appendix C.1. For term (2), since $\kappa_{i,t-1}$ is already a deterministic action here, this term is consistent with Equation (47):

$$\left|V_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}) - V_{\psi_i}(\mathbf{s})\right| \leq C_V \cdot \rho^{\kappa_{i,t-1}+1}. \tag{57}$$

Combining Equation (55), (56), and (57), we obtain:

$$\left|A_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}) - A_{\psi_i}(\mathbf{s},\mathbf{a})\right| \leq \mathbb{E}_{\kappa_{i,t}\sim\phi_i(s_{i,t})}\left[2(C_v+\eta)\rho^{\min\{\kappa_{i,t-1},\kappa_{i,t}\}+1}\right]. \tag{58}$$

Let $\eta' = 2(C_v+\eta), \sigma = \rho = \gamma$. This completes the proof.

### C.3 The proof of Corollary 3.4

*Proof.* The surrogate objective for agent $i$ under policy $\psi_i = (\phi_i, \pi_i)$ is given by

$$\begin{aligned}
&L_{\psi_i}(\phi_i,\pi_i) \\
&= \mathbb{E}_{s_{i,t}\sim P,\, \kappa_{i,t}\sim\phi_i(\cdot|s_{i,t}),\, a_{i,t}\sim\pi_i(\cdot|s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}},\kappa_{i,t})}\left[A_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}})\right],
\end{aligned} \tag{59}$$

and the surrogate objective with global communication under policy $\psi_i$ is defined as

$$L_{\psi_i}(\pi_i) = \mathbb{E}_{s_{i,t}\sim P,\, \kappa_{i,t}\sim\phi_i(\cdot|s_{i,t}),\, a_{i,t}\sim\pi_i(\cdot|s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}},\kappa_{i,t})}\left[A_{\psi_i}(\mathbf{s},\mathbf{a})\right]. \tag{60}$$

To facilitate gradient estimation under the global communication setting, we consider the case where the communication range is fixed to its maximum value. Let $\phi_i^{full}$ denote this deterministic policy where $\kappa_{i,t}$ is always set to the predefined maximum. The objective can thus be rewritten as:

$$L_{\psi_i}(\phi_i^{full},\pi_i) = \mathbb{E}_{s_{i,t}\sim P,\, a_{i,t}\sim\pi_i(\cdot|s_{\mathcal{N}_{i,t}^{\kappa_{max}}},\kappa_{max})}\left[A_{\psi_i}^{\phi_i^{full},\pi_i}(\mathbf{s},\mathbf{a})\right], \tag{61}$$

where the expectation is now taken over the state and action distributions conditioned on the full communication graph.

Combining Equation (59), (61) and Lemma A.1, we can bound the difference between the surrogate objectives as follows:

$$\begin{aligned}
&|L_{\psi_i}(\bar{\phi}_i,\bar{\pi}_i) - L_{\psi_i}(\phi_i^{full},\bar{\pi}_i)| \\
\leq &\left|\mathbb{E}_{s_{i,t},\,\kappa_{i,t},\,a_{i,t},a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}}\left[A_{\psi_i}^{\bar{\phi}_i,\bar{\pi}_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}})\right] - \mathbb{E}_{s_{i,t},\,a_{i,t},a_{\mathcal{N}_{-i,t}^{\kappa_{\max}}}}\left[A_{\psi_i}^{\phi_i^{full},\bar{\pi}_i}(\mathbf{s},\mathbf{a})\right]\right| \\
\leq &\mathbb{E}_{\kappa_{i,t}\sim\bar{\phi}_i(s_{i,t})}\left[\mathbb{E}_{s_{i,t},a_{i,t},a_{\mathcal{N}_{-i,t}}}\left|A_{\psi_i}^{\bar{\phi}_i,\bar{\pi}_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}) - A_{\psi_i}^{\phi_i^{full},\bar{\pi}_i}(\mathbf{s},\mathbf{a})\right|\right] \\
\leq &\mathbb{E}_{\kappa_{i,t}\sim\bar{\phi}_i(s_{i,t})}\left[2(C_v+\eta)\rho^{\min\{\kappa_{i,t-1},\kappa_{i,t}\}+1}\right].
\end{aligned} \tag{62}$$

Finally, by denoting $\eta' = 2(C_v+\eta)$ and setting $\sigma = \rho = \gamma$, we obtain Corollary 3.4.

### C.4 The proof of Proposition 3.5

*Proof.* Equation (17) is derived from the TRPO framework [Schulman et al., 2015]. Substituting Corollary 1 into the trust region method gives Equation (18). Combining Equation (17), (18) and applying the multi-agent trust region theorem from HATRPO [Kuba et al., 2022], it follows that:

$$
\begin{aligned}
&J(\bar{\psi}_i) - J(\psi_i) \\
&\geq \mathbb{E}_{s_{i,t},\, \kappa_{i,t}} \left[ A_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) \right] + \mathbb{E}_{s_{i,t},\, \kappa_{i,t},\, a_{i,t},\, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}} \left[ A_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}) \right] \\
&- \mathbb{E}_{\kappa_{i,t}} \left[ \eta' \sigma^{\min\{\kappa_{i,t-1}, \kappa_{i,t}\}+1} \right] - \left[ \nu_{\phi_i} D_{KL}^{\max}(\phi_i, \bar{\phi}_i) + \nu_{\pi_i} D_{KL}^{\max}(\pi_i, \bar{\pi}_i) \right].
\end{aligned} \tag{63}
$$

Let

$$
\begin{aligned}
&L_{\psi_i}(\bar{\psi}_i) \\
&= \mathbb{E}_{s_{i,t},\, \kappa_{i,t}} \left[ A_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) \right] + \mathbb{E}_{s_{i,t},\, \kappa_{i,t},\, a_{i,t},\, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}} \left[ A_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}) \right], \\
&M_{\mathrm{KL}} = \nu_{\phi_i} D_{\mathrm{KL}}^{\max}(\phi_i, \bar{\phi}_i) + \nu_{\pi_i} D_{\mathrm{KL}}^{\max}(\pi_i, \bar{\pi}_i),
\end{aligned}
$$

we arrive at Proposition 3.5.

### C.5 The proof of Lemma 3.7

*Proof.* From Theorem 1 in TRPO, we have the following result for the policy $\psi_i$ of agent $i$:

$$
\begin{aligned}
&J_c(\bar{\psi}_i) - J_c(\psi_i) \leq L_{\psi_i}(\bar{\psi}_i) + \frac{4\alpha^2 \gamma \max_{s,\kappa_{i,t}} |A_{c,\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})|}{(1-\gamma)^2} \\
&\alpha = D_{TV}^{\max}(\psi_i, \bar{\psi}_i) = \max_s D_{TV}(\psi_i, \bar{\psi}_i).
\end{aligned} \tag{64}
$$

Subsequently, by applying Pinsker's inequality [Fedotov et al., 2003] $D_{\mathrm{TV}}(p,q)^2 \leq D_{\mathrm{KL}}(p,q)$, we can obtain:

$$
\begin{aligned}
&J_c(\bar{\psi}_i) - J_c(\psi_i) \leq L_{\psi_i}(\bar{\psi}_i) + \frac{4\gamma \max_{s,\kappa_{i,t}} |A_{c,\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})|}{(1-\gamma)^2} D_{KL}^{\max}(\psi_i, \bar{\psi}_i), \\
&D_{KL}^{\max}(\psi_i, \bar{\psi}_i) = \max_s D_{KL}(\psi_i, \bar{\psi}_i).
\end{aligned} \tag{65}
$$

Note that the surrogate objective $L_{\psi_i}(\bar{\psi}_i)$ consists of two components: one related to communication and the other to decision-making. Since the communication cost depends only on the communication policy $\phi_i$ and not on the decision policy $\pi_i$, its gradient with respect to $\pi_i$ is zero. Consequently, when optimizing $\pi_i$ independently, the communication cost contributes nothing to the policy update. Since the communication cost is independent of $\pi_i$, its contribution to the change in the objective is zero.

Similarly,

$$
D_{KL}^{\max}(\psi_i, \bar{\psi}_i) = \max_s D_{KL}(\psi_i, \bar{\psi}_i) = \max_s D_{KL}(\phi_i, \bar{\phi}_i). \tag{66}
$$

Let $\nu_{c,\phi_i} = \frac{4\gamma, \max_{s_i,\kappa_{i,t}} \left| A_{c,\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) \right|}{(1-\gamma)^2}$, we can obtain:

$$
J_c(\bar{\psi}_i) \leq J_c(\psi_i) + L_{c,\psi_i}(\bar{\phi}_i) + \nu_{c,\phi_i} D_{KL}^{\max}(\phi_i, \bar{\phi}_i). \tag{67}
$$

We arrive at Lemma 3.7.

### C.6 The proof of Theorem 3.8

*Proof.* Based on Proposition 3.5 and Lemma 3.7, we can derive that, to improve reward performance while satisfying communication cost constraints, each agent must maximize its surrogate return, ensure that communication cost remains below a given threshold, and restrict policy updates within a local neighborhood—measured in terms of the maximum KL divergence.

First, for the communication cost upper bound constraint on each agent, we have:

$$J_c(\psi_i) + L_{c,\psi_i}(\bar{\phi}_i) + \nu_{c,\phi_i} D_{KL}^{\max}(\phi_i, \bar{\phi}_i) \leq c_i, \tag{68}$$

we can obtain:

$$D_{KL}^{\max}(\phi_i, \bar{\phi}_i) \leq \frac{c_i - J_c(\psi_i) - L_{c,\psi_i}(\bar{\phi}_i)}{\nu_{c,\phi_i}}. \tag{69}$$

Let $\mu_1 = c_i - J_{\mathrm{c}}(\psi_i) - L_{\mathrm{c},\psi_i}(\bar{\phi}_i)$.

Substituting Equation (69) into $M_{\mathrm{KL}}$, we obtain that the KL constraint between the old and new policies should be defined as:

$$\delta = \frac{\nu_{\phi_i}}{\nu_{c,\phi_i}} \mu_1 + \nu_{\pi_i} D_{KL}^{\max}(\pi_i, \bar{\pi}_i), \tag{70}$$

where $M_{KL} = \nu_{\phi_i} D_{KL}^{\max}(\phi_i, \bar{\phi}_i) + \nu_{\pi_i} D_{KL}^{\max}(\pi_i, \bar{\pi}_i)$,

$$\eta' = 2(C_v + \eta), \sigma = \gamma,$$

$$\nu_{c,\phi_i} = \frac{4\gamma \max_{s_{i,t},\kappa_{i,t}} |A_{c,\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})|}{(1-\gamma)^2},$$

$$\nu_{\phi_i} = \frac{4\gamma \max_{s_{i,t},\kappa_{i,t}} |A_{\psi_i}^{\phi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})|}{(1-\gamma)^2},$$

$$\nu_{\pi_i} = \frac{4\gamma \max_{s_{i,t},\kappa_{i,t},a_{i,t},a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}} |A_{\psi_i}^{\phi_i,\pi_i}(s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}})|}{(1-\gamma)^2},$$

$$\mu_1 = c_i - J_c(\psi_i) - L_{c,\psi_i}(\bar{\phi}_i),$$

$$\delta = \frac{\nu_{\phi_i}}{\nu_{c,\phi_i}} \mu_1 + \nu_{\pi_i} D_{KL}^{\max}(\pi_i, \bar{\pi}_i).$$

We arrive at Theorem 3.8.

### C.7 The detail of Algorithm 1

We focus on the practical implementation of the policy update in Theorem 3.8. Let $\theta_i$ denote the parameters of policy $\psi_i$, with $\theta_i^{\phi}$ and $\theta_i^{\pi}$ corresponding to the communication policy $\phi_i$ and the behavior policy $\pi_i$, respectively. $\lambda_i$ is a scalar variable. We apply Lagrangian relaxation to the communication cost constraint, yielding:

$$\begin{aligned}\max_{\theta_i^{\phi},\theta_i^{\pi}} \min_{\lambda_i>0} \Bigg[ & \mathbb{E}_{s_{i,t},\kappa_{i,t},a_{i,t},a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}}} \left[ A_{\theta_i}^{\theta_i^{\pi}} \left( s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}} \right) \right] \\ & + \mathbb{E}_{s_{i,t},\kappa_{i,t}} A_{\theta_i}^{\theta_i^{\phi}} \left( s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t} \right) - \lambda_i \left( \mathbb{E}_{s_{i,t},\kappa_{i,t}} \left[ A_{c,\theta_i}^{\theta_i^{\phi}} \left( s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t} \right) \right] \right) \Bigg], \end{aligned} \tag{71}$$

$$\textbf{s.t} \quad M_{KL} \leq \delta,$$

$\lambda_i$ is a scalar variable. In the implementation, we employed primal-dual optimization to achieve automated parameter $\lambda_i$ selection. Let

$$\begin{aligned} A_{\theta_i}^{\lambda_i,\theta_i^{\phi},\theta_i^{\pi}} = & A_{\theta_i}^{\theta_i^{\pi}} \left( s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}, a_{i,t}, a_{\mathcal{N}_{-i,t}^{\kappa_{i,t}}} \right) + A_{\theta_i}^{\theta_i^{\phi}} (s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) \\ & - \lambda_i \left( A_{c,\theta_i}^{\theta_i^{\phi}} (s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t}) \right), \end{aligned} \tag{72}$$

**Algorithm 1** ACR-PPO

**Input**: Learning rates $\alpha_\theta, \alpha_\lambda$, mini-batch size $B$, number of episodes $Z$, horizon $T$, discount factor $\gamma$.

**Initialize**: Policy parameters $\theta_i^\phi, \theta_i^\pi$ and Lagrangian multipliers $\lambda_i$ for all $i \in \mathcal{N}$, replay buffer $\mathcal{B}$.

**for** $z = 1$ **to** $Z$ **do**
  Reset environment.
  **for** $t = 0$ **to** $T - 1$ **do**
    **for** $i = 1$ **to** $n$ **do**
      Sample communication range $\kappa_{i,t} \sim \phi_i(\cdot|s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}})$.
      Sample action $a_{i,t} \sim \pi_i(\cdot|s_{\mathcal{N}_{i,t}^{\kappa_{i,t-1}}}, \kappa_{i,t})$.
    **end for**
    Execute joint action, receive rewards $\{r_{i,t}\}$ and next states $\{s_{\mathcal{N}_{i,t+1}^{\kappa_{i,t}}}\}$.
    Store transition $\tau = (s_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, a_{\mathcal{N}_{i,t}^{\kappa_{i,t}}}, r_{i,t}, s_{\mathcal{N}_{i,t+1}^{\kappa_{i,t}}})$ into $\mathcal{B}$.
  **end for**
  Compute advantage estimates and cost advantages over the collected batch $\mathcal{B}$.
  **for** $i = 1$ **to** $n$ **do**
    Perform $K$ epochs of mini-batch (size $B$) to optimize $\theta_i^\phi, \theta_i^\pi$ using Eq. (24).
    Update Lagrangian multiplier $\lambda_i$ via primal-dual optimization.
  **end for**
  Clear buffer $\mathcal{B}$.
**end for**

we can obtain:

$$\max_{\theta_i^\phi, \theta_i^\pi} \min_{\lambda_i > 0} \left[ \mathbb{E}_{s,\kappa_{i,t},a_{i,t}} A_{\theta_i}^{\lambda_i, \theta_i^\phi, \theta_i^\pi} \right], \quad \textbf{s.t } M_{KL} \leq \delta. \tag{73}$$

After rearranging the terms, we can obtain:

$$\max_{\theta_i^\phi, \theta_i^\pi} \min_{\lambda_i > 0} \mathbb{E}_{s,\kappa_{i,t},a_{i,t}} \left[ \min \left( U_i A_{\theta_i}^{\lambda_i, \theta_i^\phi, \theta_i^\pi}, \text{clip}(U_i, 1-\epsilon, 1+\epsilon) A_{\theta_i}^{\lambda_i, \theta_i^\phi, \theta_i^\pi} \right) \right], \tag{74}$$

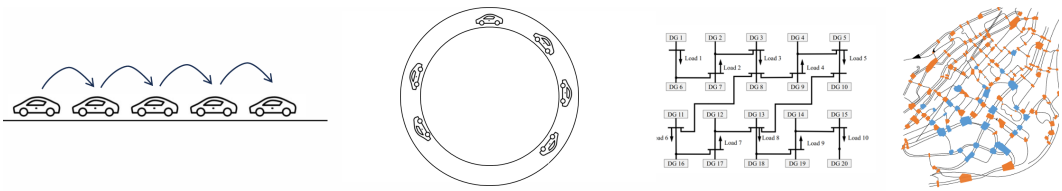
where $U_i = \frac{\bar{\phi}_i(\theta_i^\phi) \cdot \bar{\pi}_i(\theta_i^\pi)}{\phi_i(\theta_i^\phi) \cdot \pi_i(\theta_i^\pi)}$ is the importance sampling ratio, measuring the deviation of agent $i$'s updated individual policy (comprising both communication and behavior components) from its current policy. The function $\text{clip}(U_i, 1-\epsilon, 1+\epsilon)$ constrains $U_i$ within the interval $[1-\epsilon, 1+\epsilon]$. This formulation ensures that the policy update remains conservative by preventing large deviations in the probability ratio, regardless of the sign of the advantage function $A_{\theta_i}$. Each agent $i$ updates $\theta_i^\phi$ and $\theta_i^\pi$ by maximizing this surrogate objective, allowing the agent to simultaneously maximize the expected return while minimizing its communication range.

# D Experiments

## D.1 Environments

The simulation environment setup is shown in Figure 3.

**CACC** [Chu et al., 2020]. For both CACC tasks, we simulate a platoon of 8 vehicles over a 60-second period with a 0.1-second control interval. Each vehicle observes and shares its headway ($h$), velocity ($v$), and acceleration ($a$) with neighbors within a two-step communication range. The safety constraints are defined as: $h \geq 1$m, $v \leq 30$m/s, and $|a| \leq 25$m/s$^2$. While safe RL is relevant in this context, it represents a substantial topic that falls outside the scope of this paper. Therefore, we employ a simple heuristic optimal velocity model (OVM) to perform longitudinal vehicle control under the aforementioned constraints. The OVM behavior is influenced by hyper-parameters: headway gain ($\alpha$), relative velocity gain ($\beta$), stop headway ($h_{\text{st}} = 5$m), and full-speed headway ($h_{\text{go}} = 35$m). Typically, $\alpha$ and $\beta$ represent human driver behavior; however, in this work, we train NMARL to



(a) (b) (c) (d)

Figure 3: The environments. (a) CACC (Catchup), where the objective is to adaptively coordinate a fleet of vehicles to minimize spacing and speed disturbances based on real-time V2V communication. (b) Flow (Ring), which shares similar objectives with CACC—maintaining target speed and avoiding collisions. (c) IEEE PowerGrid (PowerGrid), where the objective is to improve voltage control performance in an islanded microgrid system. (d) ATSC (Monaco), where the objective is to dynamically adjust traffic signal phases based on real-time traffic measurements to reduce congestion.

recommend appropriate $(\alpha, \beta)$ pairs for each OVM controller. These recommendations are selected from four predefined levels: $(0, 0)$, $(0.5, 0)$, $(0, 0.5)$, and $(0.5, 0.5)$. Assuming the target headway and velocity profile are $\bar{h} = 20$m and $v_t$, respectively, the cost function for each agent is defined as $(h_{it} - \bar{h})^2 + (v_{it} - v_t)^2 + 0.1u_{it}^2$. When a collision occurs ($h_{it} < 1$m), a substantial penalty of 1000 is imposed on each agent, and the state becomes absorbing. Additionally, a collision avoidance cost of $5(2h_{\text{st}} - h_{it})^2$ is incorporated during training to penalize potential collision scenarios.

**Flow** [Vinitsky et al., 2018]. We consider a Ring network scenario in which a fleet of vehicles operates to maintain a target speed while avoiding collisions, a control objective closely aligned with CACC. For each vehicle, the state is defined by its longitudinal position and velocity, and the control action corresponds to its acceleration. From a networked systems perspective, each vehicle exchanges information exclusively with its immediate predecessor and successor, thereby forming a closed-loop communication topology.

**IEEE PowerGrid** [Chen et al., 2021]. The IEEE PowerGrid simulation platform is constructed based on the line and load specifications and there are 20 agents in this environment. To better represent real-world power systems, random load variations are introduced across the entire microgrid with $20\%$ perturbations from the nominal values specified. At each simulation step, random disturbances within $5\%$ of the nominal values for each load are also added to simulate the stochastic disruptions present in real-world power grids. The DGs are controlled with a sampling time of 0.05s, and each DG can communicate with its neighbors via local communication edges.

**ATSC** [Chu et al., 2020]. For both scenarios, each episode simulates peak-hour traffic, and a 5s control interval is applied to prevent traffic lights from switching too frequently, based on RL control latency and driver response delay. Thus, one MDP step corresponds to 5s simulation and the horizon is 720 steps. Further, a 2s yellow time is inserted before switching to red light for safety purposes. In ATSC, the real-time traffic flow, that is, the total number of approaching vehicles along each incoming lane, is measured by near-intersection induction-loop detectors (ILDs). The cost of each agent is the sum of queue lengths along all incoming lanes. The Grid scenario has 25 agents, while the Monaco scenario has 28 agents. To validate the performance of our method in larger-scale scenarios, we additionally adopt an ATSC-NewYork environment, which simulates traffic signal control in New York City and involves 436 agents.

**Communication Cost**. For consistency in evaluation and ease of computation, all experiments adopt this setting. See Appendix B.1.

### D.2 Baselines

In addition to CPPO and DPPO, we introduce some additional baseline methods for comparison.

**CPPO-Sampling**. To verify that the performance gains of ACR-PPO stem from adaptive policy learning rather than mere stochasticity, we introduce this baseline. In this setup, the communication range $\kappa_{i,t}$ for each agent is randomly sampled from a uniform distribution at every timestep, independent of the state. This baseline serves as a control to demonstrate that our method's superior

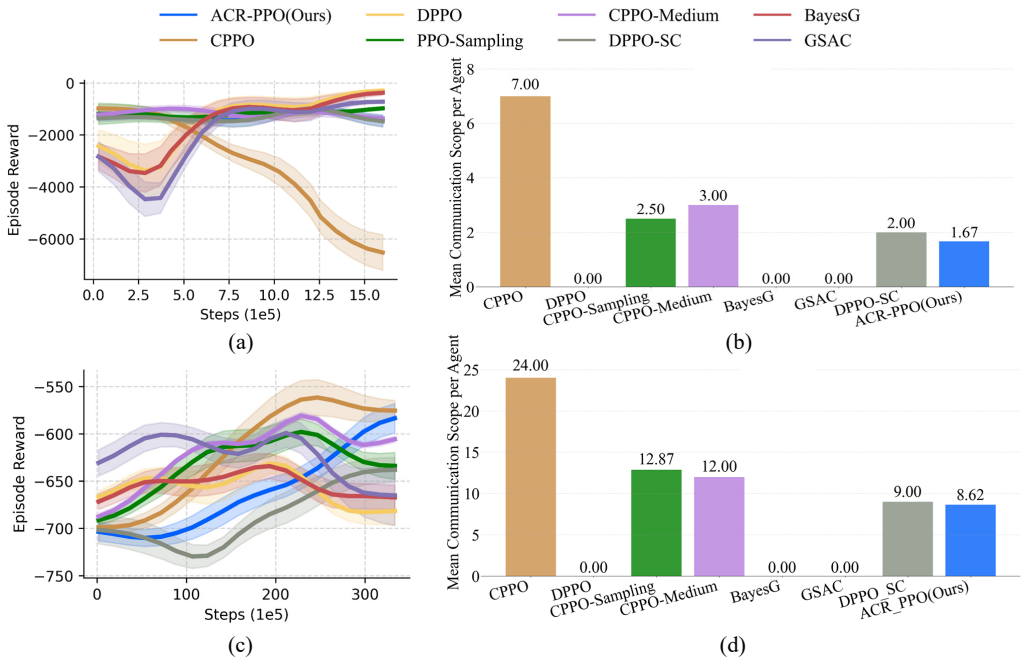



Figure 4: The policy performance and communication range per agent. (a) The policy performance in Slowdown. (b) The average communication range per agent in Slowdown. (c) The policy performance in Grid. (d) The average communication range per agent in Grid. Experimental results show that our method can maintain performance as much as possible while adaptively reducing the communication range, thereby lowering costs.

efficiency is due to the intelligent, state-dependent selection of $\kappa$, as opposed to random exploration of the communication space.

**DPPO-SC**. To ensure a fair comparison under equivalent communication constraints, we introduce DPPO-SC (DPPO with Same Constraint) as a controlled baseline. Specifically, we align the fixed communication range of DPPO-SC with the average effective range achieved by our method. For example, in the Catchup scenario (Fig. 1(b)), our method dynamically adjusts the communication range to an average of $\kappa \approx 2.65$. Consequently, we set the static range for DPPO-SC to $\kappa = 3$ (the nearest integer upper bound) to match this communication budget.

**BayesG** [Duan et al., 2025]. BayesG employs Bayesian variational inference to let each agent sample a latent communication mask over its local neighborhood, enabling end-to-end joint learning of sparse interaction topologies and decision-making policies. Due to the sampling space, this work only considers 1-hop neighbors.

**GSAC** [Liang et al., 2025]. GSAC is a scalable and generalizable Actor-Critic framework for large-scale networked systems that couples causal representation learning with meta-learning; it enables each agent to learn sparse local causal masks yielding compact state representations, which bound value function truncation errors for efficient graph-based learning, while a meta actor-critic trains a shared policy across domains conditioned on these compact factors, allowing rapid adaptation to new environments with minimal data and providing theoretical guarantees on causal recovery and convergence.

### D.3 More Results

More experimental results are shown in Figure 4. We have added additional experimental results for the Slowdown and Grid environments. The results indicate: (1) Our method can minimize costs as much as possible by adaptively adjusting the communication range while maintaining performance. (2) The performance of CPPO under the Slowdown environment is not as good as that of DPPO. We

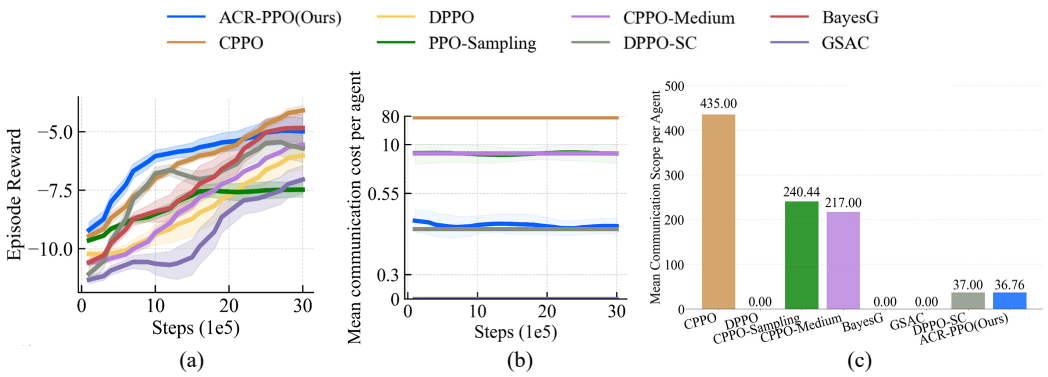



Figure 5: Experiments conducted in a larger-scale traffic scenario, "ATSC-NewYork (Agent number=436)". (a) Performance of the experiment. (b) Average communication cost per agent. (c) Average communication range per agent. The results show that our method remains applicable to larger-scale problems and effectively reduces communication costs.

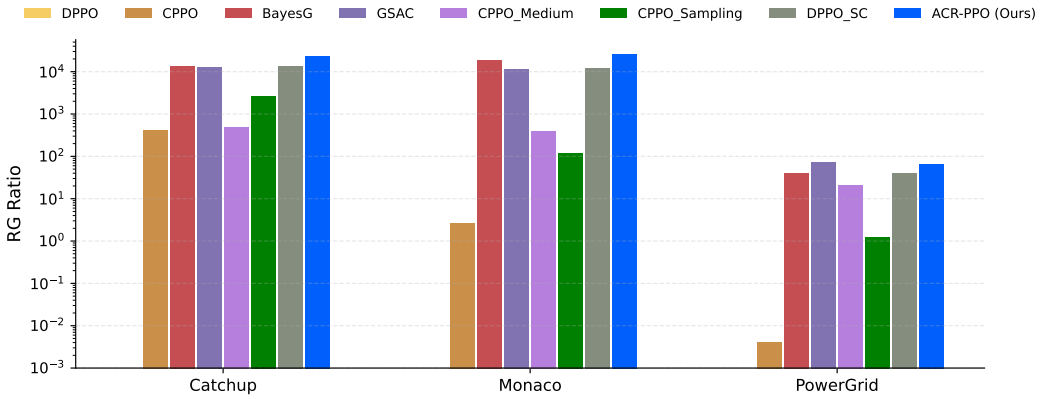



Figure 6: The fair and Comprehensive Evaluation. We propose a novel metric, called "RG ratio", for the comprehensive evaluation of method performance and cost.

speculate that this may be due to excessive communication leading to the generation of redundant information. (3) In Figure 4(a), our method closely methods the performance of DPPO, whereas in Figure 4(c), our method closely methods the performance of CPPO. This demonstrates that our method can effectively adapt to different environments.

### D.4 Scalability Evaluation

To further demonstrate the scalability of our method, we evaluate on a "New York" environment comprising 436 agents, which simulates traffic light control in a real-world urban setting. The results are presented in the Figure 5. The results show that our algorithm scales effectively to larger scenarios, achieving lower communication costs while maintaining near-optimal performance.

### D.5 The fair and Comprehensive Evaluation

To further demonstrate the performance of our method fairly, we additionally designed a metric: Relative Gain Ratio (RG ratio). This metric comprehensively considers the advantages of various baselines in terms of cost (compared with CPPO) and performance (compared with DPPO), where a

higher value indicates superior efficiency:

$$RG\ ratio = \frac{R_{method} - R_{DPPO}}{C_{method}/C_{CPPO}}. \tag{75}$$

The experimental results are illustrated in Figure 6, where the horizontal axis represents the three scenarios and the vertical axis denotes the RG ratio of each method. Since DPPO serves as the baseline for performance, its metric value is set to 0. The results demonstrate that: (1) Our method achieves superior performance per unit cost while adhering to the same cost upper bound; and (2) BayesG and GSAC exhibit suboptimal performance because they both employ a 1-hop formulation, yielding results on Monaco that are comparable to those of DPPO.

### D.6 The Implement Detail of Our methods

**Framework**. The overall architecture of the algorithm is illustrated in the figure. **Each agent** maintains two sub-policies that are updated sequentially: the communication policy first dynamically determines the agent's communication range, and the behavior policy then selects actions based on neighboring agents within that range.

**Model Size.** Due to varying observation dimensions across environments, we take the Catchup environment as an example(agent number $= 16$, obs_dim $= 30$, action_dim $= 5$):

- Policy net per agent: $30 \times 64 + 64 \times 64 + 64 \times 5 \approx 6{,}000$ params
- Value net per agent : $30 \times 64 + 64 \times 64 + 64 \times 1 \approx 6{,}000$ params
- Communication Policy net per agent: $30 \times 64 + 64 \times 64 + 64 \times 16 \approx 7{,}000$ param
- Communication Value net per agent : $30 \times 64 + 64 \times 64 + 64 \times 1 \approx 6{,}000$ params
- **Total:** $\approx$**25,000** params per agent.

**HyperParameters Setting**. Table 1 presents the parameters associated with ACR. Additional pertinent parameters follow the conventions of CACC, Flow, ATSC and IEEE PowerGrid.

**Training Time**. Table 2 shows the training time of the overall algorithm. We observe that training time increases primarily because our method employs a parameter-sharing mechanism, which requires a uniform batch size and leads to longer training time.

**Computing Setting**. We conducted our experiments on a system equipped with an Intel(R) Core i9-13900K processor (8 performance cores and 16 efficiency cores, totaling 32 threads) with a base frequency of 3.0 GHz. The system is integrated with one NVIDIA GeForce RTX 4090 GPU to accelerate the training process. The operating system is Ubuntu 20.04 LTS.

**Computational Complexity**. Based on the description in Algorithm 1, the time complexity of ACR is $\mathcal{O}(T \cdot N \cdot H \cdot (M + P))$, where: $T$ denotes the total number of timesteps, $N$ is the number of agents, $H$ represents the number of PPO epochs per update, $M$ is the number of parameters in the communication policy, $P$ is the number of parameters in the behavior policy. In terms of communication cost, our method achieves adaptive control by dynamically restricting the communication range through the communication policy. Specific experimental results are shown in the second row of Figure 1.

**The discussion of temporal dependencies**. Adding a communication policy incurs some training-time overhead; however, since this network is small, the training time of ACR-PPO does not increase significantly compared to DPPO, which operates without communication but theoretically lacks convergence guarantees and performance advantages. Additionally, since we do not simulate the actual communication process between agents or between agents and a central server, CPPO achieves the shortest runtime. However, in real-world scenarios, the demand for extensive communication, along with issues such as time delay and packet loss caused by dense communication, imposes significant pressure on centralized training. Our work aims to mitigate these substantial overheads by learning communication policies, which is often a worthwhile endeavor.

### D.7 Hyperparameter Analysis

**Hyperparameter** $\lambda_i$. This section presents the evolution of $\lambda_i$. By employing primal-dual optimization [Zhang et al., 2024b], our method enables the adaptive adjustment of $\lambda_i$. Table 3 illustrates the variation of $\lambda_i$ across different episodes in two scenarios.

Table 1: Hyperparameters Setting of ACR.

| Hyperarameter | Ring | Catchup | Monaco | PowerGrid |
|---|---|---|---|---|
| clip | 0.15 | 0.2 | 0.2 | 0.2 |
| target KL | 7.5e-3 | 0.01 | 7.5e-3 | 7.5e-3 |
| batch size | 256 | 256 | 256 | 256 |
| buffer size | 15 | 15 | 15 | 15 |
| gamma | 0.99 | 0.99 | 0.99 | 0.99 |
| learning rate | 5e-4 | 5e-5 | 5e-4 | 5e-5 |
| learning rate V | 5e-4 | 7e-4 | 5e-4 | 5e-4 |
| activation | ReLU | ReLU | ReLU | ReLU |
| nn hidden size | 64 | 64 | 64 | 64 |
| cost bound | 3000 | 1200 | 3000 | 3000 |

Table 2: Training Time in our experiments.(min)

| Algorithm | CPPO | CPPO-Medium | DPPO | PPO-Reg | CPPO-Sampling | ACR-PPO |
|---|---|---|---|---|---|---|
| Slowdown | 22±1 | 22±3 | 31±0 | 41±1 | 15±2 | 39±2 |
| Grid | 297±2 | 294±1 | 412±3 | 370±0 | 301±0 | 307±5 |
| Ring | 277±1 | 284±3 | 266±2 | 301±6 | 277±2 | 299±10 |
| Catchup | 20±5 | 25±2 | 21±1 | 41±1 | 20±0 | 25±4 |
| Monaco | 155±5 | 149±5 | 144±12 | 147±11 | 147±4 | 151±0 |
| PowerGrid | 94±0 | 109±0 | 94±1 | 105±1 | 108±1 | 105±2 |
| Eight | 105±2 | 92±5 | 78±1 | 120±16 | 104±1 | 106±3 |
| New York | 431±32 | 412±7 | 388±12 | 392±8 | 374±10 | 525±2 |

Table 3: Hyperparameter sensitivity. The vertical axis represents different scenarios, and the horizontal axis indicates episodes during training.

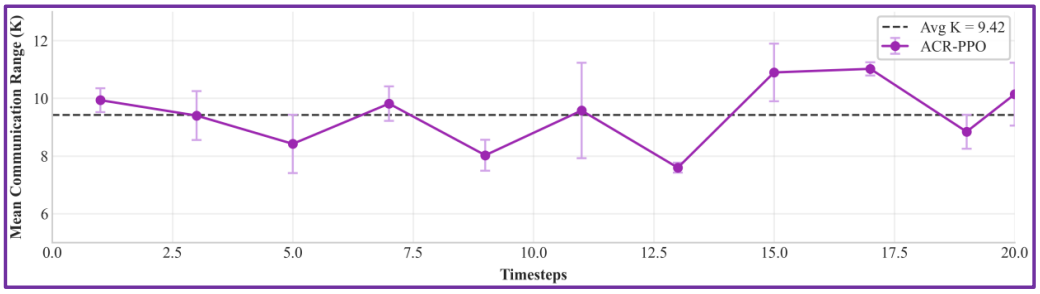
| Scenario | Episode 100 | Episode 300 | Episode 500 | Episode 700 |
|---|---|---|---|---|
| Catchup | 0.00 | 0.11 | 0.00 | 0.07 |
| Monaco | 0.01 | 0.02 | 0.00 | 0.01 |

Table 4: Results for Catchup and Powergrid under different $\alpha$ values with specific cost bounds.

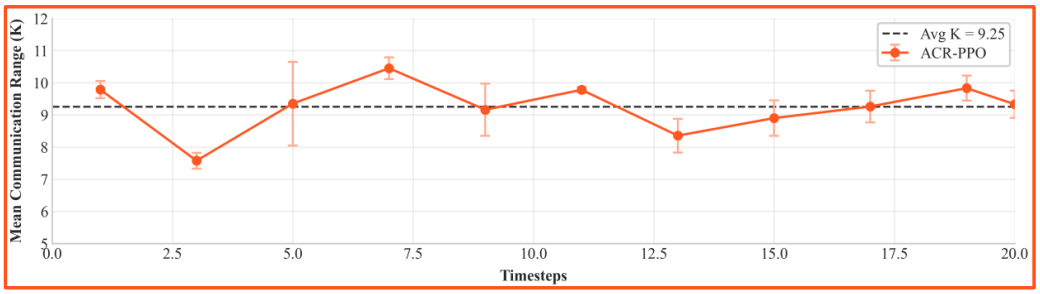
| Hyperparater $\alpha$ | Catchup (Cost_bound=1200) | Powergrid (Cost_bound=3000) |
|---|---|---|
| 0.5 | -759.25[-812.56, -705.93] | 0.53[0.45, 0.60] |
| 0.3 | -768.72[-799.63, -737.81] | 0.54[0.52, 0.55] |
| 0.1 | -739.13[-785.87, -692.38] | 0.54[0.53, 0.55] |
| 0.05 | -739.80[-766.11, -713.48] | 0.53[0.51, 0.54] |

Table 5: Performance under Different Cost Bounds.

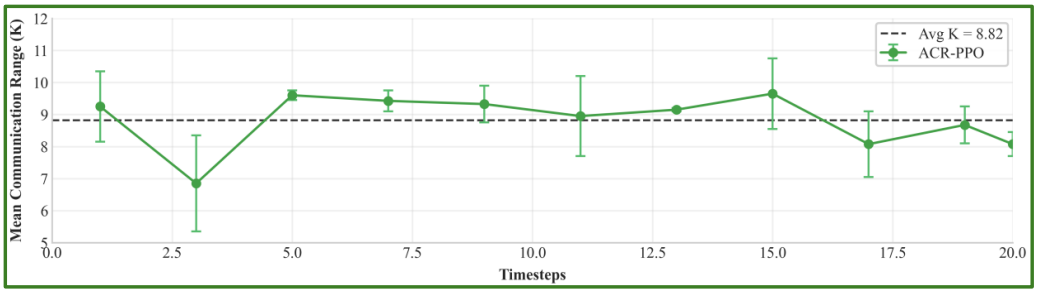
| Cost Bound | Catchup Performance | Powergrid Performance |
|---|---|---|
| 3000 | — | 0.54 [0.53, 0.55] |
| 1200 | -739.13 [-785.87, -692.38] | — |
| 1000 | -846.03 [-965.72, -746.34] | 0.47 [0.45, 0.49] |
| 500 | -925.16 [-958.22, -832.09] | 0.45 [0.41, 0.49] |
| 300 | -1125.13 [-1256.41, -993.85] | 0.43 [0.40, 0.46] |
| 0 | -1953.02 [-2031.38, -1874.65] | 0.42 [0.38, 0.46] |




(a) Powergird (20 agents version, Episode No.100)




(b) Powergird (20 agents version, Episode No.400)




(c) Powergird (20 agents version, Episode No.800)

Figure 7: Performance of the average communication range of agents across episodes. We extract the training performance from three fixed episodes throughout the entire training process: (a) early training (Episode No.100), (b) mid-training (Episode No.400), and (c) late training (Episode No.800). The results show that, over time, the average communication range gradually decreases due to the constraints imposed by our method's policy. Although these reductions are small, considering that this represents the average across all agents and given the exponential nature of communication costs, even minor reductions in communication range lead to significant decreases in overall communication cost.

**Communication cost hyperparameter** $\alpha$. $\alpha$ in Equation (38) controls the rate of exponential growth of communication cost with respect to communication range. Experiments on the Catchup and PowerGrid maps demonstrate that our method is robust to variations in $\alpha$, as shown in Table 4.

**Communication cost bound**. As shown in Table 5, the performance of ACE-PPO on both Catchup and Powergrid improves as the cost bound increases. This is because a looser cost constraint enhances the algorithm's tolerance, allowing the policy to expand its communication range and thereby optimize task performance.

Table 6: The communication range of each agent in PowerGrid (Episode 800, 20-agent setting). Each row corresponds to an agent (indexed by agent ID), and each column corresponds to one of the 20 time steps. Agents are ordered according to their average communication range over the episode.

| ID | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 | 20 | Avg $\kappa$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No.9 | 13.5 | 9.5 | 15 | 16 | 15.5 | 11.5 | 11.5 | 14.5 | 10.5 | 14 | 2.5 | 12.19 |
| No.17 | 12 | 9 | 12.5 | 12.5 | 8.5 | 10 | 16.5 | 13 | 12.5 | 4.5 | 11.5 | 11.14 |
| No.2 | 17.5 | 10.5 | 10.5 | 9 | 10 | 9.5 | 4 | 10.5 | 12 | 7 | 11.5 | 10.18 |
| No.10 | 11.5 | 6 | 9 | 6.5 | 12 | 11 | 14 | 16 | 4 | 11 | 5 | 9.64 |
| No.14 | 17.5 | 8 | 12 | 8 | 8.5 | 16 | 5.5 | 7 | 3 | 13.5 | 6 | 9.55 |
| No.11 | 8.5 | 8.5 | 10 | 11 | 10.5 | 7.5 | 12.5 | 5 | 5 | 10.5 | 14.5 | 9.41 |
| No.15 | 6.5 | 9.5 | 13 | 9.5 | 7 | 10 | 6.5 | 14 | 7 | 8.5 | 11.5 | 9.36 |
| No.1 | 17.5 | 8 | 6.5 | 9.5 | 8.5 | 4.5 | 3.5 | 14 | 11.5 | 9.5 | 9 | 9.27 |
| No.4 | 1 | 8 | 2 | 16.5 | 10 | 15.5 | 17.5 | 4 | 7.5 | 9.5 | 5.5 | 8.82 |
| No.18 | 9.5 | 0 | 12 | 6 | 11 | 5.5 | 9.5 | 4.5 | 9.5 | 12.5 | 15.5 | 8.68 |
| No.5 | 8 | 7 | 8 | 8.5 | 18 | 14 | 11 | 1.5 | 8.5 | 6 | 3 | 8.50 |
| No.16 | 8 | 5 | 6 | 6 | 9 | 12.5 | 11 | 16 | 5 | 4.5 | 8.5 | 8.31 |
| No.7 | 7 | 1 | 10.5 | 12 | 15.5 | 7.5 | 8 | 6 | 7.5 | 6 | 10 | 8.27 |
| No.20 | 3.5 | 10.5 | 7 | 12.5 | 1.5 | 1.5 | 7.5 | 11 | 12 | 12 | 11.5 | 8.23 |
| No.8 | 8 | 12 | 7.5 | 3.5 | 7 | 8.5 | 7.5 | 15 | 5.5 | 4.5 | 8 | 7.91 |
| No.3 | 8.5 | 4.5 | 4 | 9 | 10.5 | 14.5 | 12 | 4 | 7 | 3.5 | 8 | 7.77 |
| No.13 | 1.5 | 4 | 15 | 11.5 | 9 | 2 | 8 | 8 | 10.5 | 10.5 | 2.5 | 7.50 |
| No.6 | 2 | 4.5 | 15 | 1.5 | 5.5 | 8 | 8.5 | 9 | 6 | 11 | 10.5 | 7.41 |
| No.12 | 14 | 10 | 6 | 6.5 | 6.5 | 1.5 | 3 | 11.5 | 14.5 | 3 | 2 | 7.14 |
| No.19 | 9.5 | 1.5 | 10.5 | 13 | 2.5 | 8 | 5.5 | 8.5 | 2.5 | 12 | 5 | 7.14 |

### D.8 Further Analysis on Communication Range Variation

To further investigate how the communication range evolves during training, we conduct three additional experiments.

**Communication Range Selection Across Episodes**. As shown in Figure 7, we present results from Episode No.100, No.400, and No.800 in the PowerGrid environment (20-agent version). The results indicate that, during early training (Episode No.100), the algorithm is still exploring, and the average communication range is relatively high (approximately $\kappa = 9.42$), resembling random sampling behavior. As training progresses (Episode No.400), agents gradually reduce their communication range ($\kappa = 9.25$), and by the end of training (Episode No.800), the average range further decreases to $\kappa = 8.82$. Although this average reduction appears small, it is important to note that communication cost typically grows exponentially with the communication range. Therefore, even minor reductions in $\kappa$ can lead to significant savings in overall communication cost.

**Individual communication patterns at the end of training**. As shown in Table 6, we present the per-step communication range for each agent during Episode No.800. The results reveal that while the overall average communication range is $\kappa = 8.82$, individual agents exhibit considerable variation. Some agents (e.g., Agent 9, Agent 17, and Agent 2) maintain a communication range larger than the average, whereas others (e.g., Agent 12 and Agent 19) operate with significantly smaller ranges. We hypothesize that agents requiring larger communication ranges are often located at critical junctions or high-load nodes in the power grid, where coordination with neighboring agents is essential. In contrast, agents with smaller communication ranges are likely situated in peripheral areas, where local coordination suffices. This spatial variation in communication needs reflects the heterogeneous structure of the environment and highlights the adaptability of our algorithm in tailoring communication strategies to local requirements.

**Visualization.** We analyze the communication range during high-reward moments in PowerGrid through visualization. As shown in the topology in Figure 8(a) in late training timesteps (Figures 8(b) and (c)), node communication ranges exhibit clear topological dependence: high-degree nodes (node 2, 9 and 12) maintain larger communication ranges, while low-degree edge nodes (e.g., node 19) significantly reduce theirs. This is because high-degree nodes are critical to global voltage stability and require extensive communication to secure system-level rewards; low-degree nodes have limited

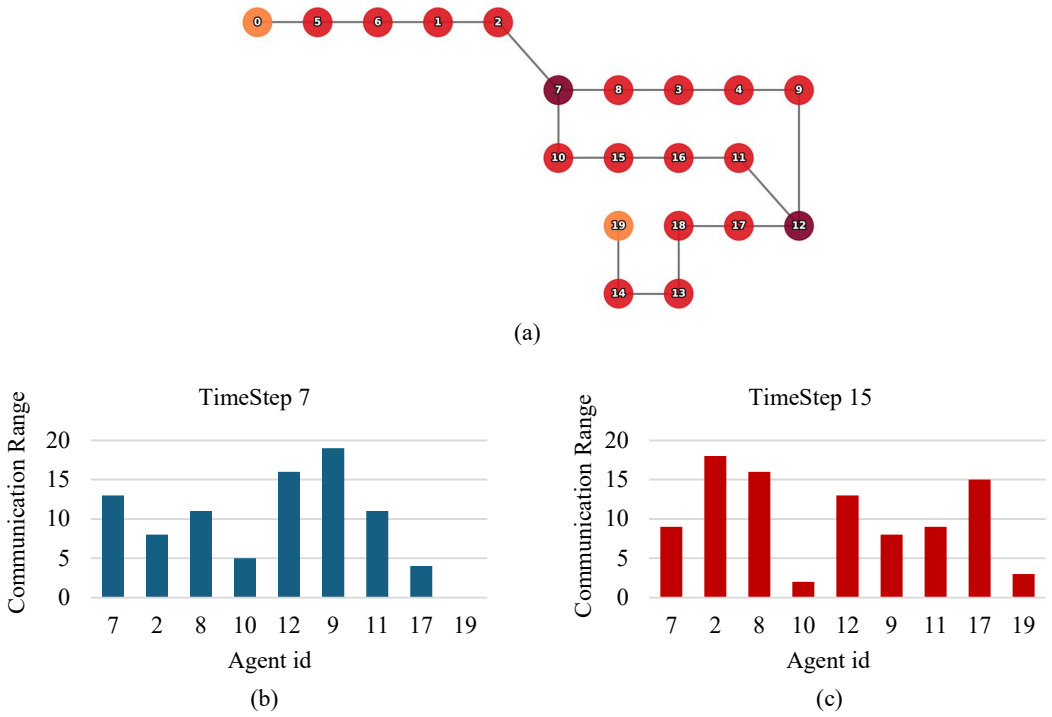



Figure 8: Visualization of Communication Strategies in PowerGrid. (a) The graph topology of the 20-agent environment. (b)-(c) Communication ranges for representative agents at TimeStep 7 and TimeStep 15, respectively.

local influence, so compressing their communication ranges does not significantly compromise overall performance.

# E Related works

## E.1 Sequential Updating

Sequential updating is a widely adopted strategy for policy iteration in MARL. Methods such as HATRPO [Kuba et al., 2022], MAT [Wen et al., 2022], HARL [Zhong et al., 2024] and ACE [Li et al., 2023a] employ a sequential update method by iteratively optimizing policies across different agents. However, these methods focus on inter-agent sequential optimization, whereas our work targets the intra-agent sequential iteration of distinct sub-policies. Other methods, such as STEER [Zhang et al., 2024a], LtoS [Yi et al., 2022], and Bi-AC [Zhang et al., 2020], adopt a hierarchical structure to coordinate the policy updates among agents. However, these methods primarily focus on asynchronous action coordination, reward sharing, and Nash equilibrium computation. A related method is [Geng et al., 2024], which considers relay nodes as leaders and sources as followers. In contrast, our work does not rely on explicit leader-follower grouping; instead, it emphasizes the sequential update of sub-policies within each individual agent based on its multi-hop communication context.

# F Impact Statement

This paper presents work whose goal is to advance the field of multi-agent reinforcement learning. Our contribution is purely methodological and does not have any direct societal impact.

## G The Usage of LLMs

This paper only employs LLM for polishing the language and descriptions.

## H Limitation

Our work may have several limitations as follows. First, ACR requires more time for the search process. Compared to fixed $\kappa$-hop or graph structure-based sampling methods, our method typically demands additional computational resources and longer running times to find a suitable communication range in different timesteps. Second, ACR relies on the assumption that the distribution of communication policies before and after updates remains approximately unchanged. This limits the effectiveness of our method in scenarios where the communication range changes abruptly or frequently. Moreover, this paper focuses on adaptive communication range selection and cost optimization among agents, and does not model the potential internal delays introduced by sequential update mechanisms. Although such delays have not been systematically studied in the MARL field and are not the core focus of our work, we have added a discussion of the potential impacts of such temporal dependencies in Appendix D.6, aiming to provide insights for future work.

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The abstract and introduction accurately reflect the paper's contributions by clearly defining the proposed ACR-PPO method, its two-stage sequential decision-making process, and its theoretical guarantee of monotonic performance improvement.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of this paper in Appendix H.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: We provide detailed theoretical proofs in the Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide implementation details for reproducing our results in Appendix D.6.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The corresponding code is available in the supplementary material.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We provide the training and test details in the Appendix D.6.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: All experimental results in this paper include information about statistical significance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide information on the resources required to reproduce the experiments in the Appendix D.6.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We conform to the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: Our method has no societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: Our paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All third-party assets used in the paper, including simulation environments, baseline implementations, and datasets, are properly cited in the main text.

    Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We provide relevant codes in the supplementary materials.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: Our paper is not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A].

    Justification: Our paper is not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The statement regarding the use of LLMs is provided in Appendix G.

Guidelines:

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