Learning Cooperative Policies Under Delayed Communication

Anonymous CVPR submission

Paper ID *****

Abstract

Multi-agent systems rely on inter-agent communication to achieve coordinated behavior, yet most learning-based approaches assume idealized communication that is instantaneous, synchronized, and noise-free. In real-world settings, however, communication is inherently delayed, asynchronous, and unreliable, leading to significant degradation in learned policies when deployed outside of simulation. In this work, we study how to enable robust cooperative intelligence under realistic communication constraints. We propose a delay-aware multi-agent reinforcement learning (MARL) framework that explicitly models both fixed and time-varying communication latency during training, using delay distributions grounded in real-world measurements. To ensure reliable execution, we integrate a safety-aware control layer based on Control Barrier Functions (CBFs), which filters unsafe actions while preserving the learned policy behavior. We evaluate our approach in both simulation and real-world multi-agent systems, demonstrating that policies trained with time-varying delay modeling achieve improved coordination stability and zero-shot transfer under realistic communication conditions. Compared to approaches that assume perfect communication or rely on domain randomization, our method achieves a superior safety–efficiency trade-off and significantly reduces collision rates in hardware deployment. These results highlight the importance of incorporating structured communication constraints into the learning process and provide a practical pathway toward deploying cooperative multi-agent systems in real-world environments.

1. Introduction

Multi-agent systems are a cornerstone of modern intelligent systems, enabling coordinated behavior across applications such as autonomous driving, robot teams, and distributed sensing. A fundamental assumption underlying many learning-based approaches to multi-agent coordination is the availability of reliable inter-agent communication. In practice, however, communication is neither instantaneous nor perfectly synchronized. Real-world systems operate under latency, packet loss, and asynchronous updates, resulting in temporally misaligned information that can significantly degrade coordination performance. This mismatch between idealized training assumptions and real-world communication constraints remains a critical barrier to deploying cooperative multi-agent intelligence outside of simulation.

Recent advances in multi-agent reinforcement learning (MARL) have demonstrated the ability to learn complex cooperative behaviors in dynamic environments [4]. However, the majority of these methods assume perfect or near-perfect communication, where shared observations are immediately available and fully consistent across agents.While techniques such as domain randomization and noise injection improve robustness [8], they do not explicitly model the structured temporal effects induced by communication latency. Prior work has shown that timing mismatch and stochastic delays can significantly degrade sim-to-real performance [7]. In addition to robustness, safety remains a critical requirement for real-world multi-agent systems, particularly in safety-critical domains [1]. Prior work in safe multi-agent reinforcement learning has explored shielding-based mechanisms for constraining unsafe behavior during learning and execution [3].

In this work, we address the problem of learning cooperative policies under realistic communication constraints. We propose a delay-aware multi-agent learning framework that explicitly incorporates communication latency into both the problem formulation and the training process. Rather than assuming synchronized observations, agents are trained on delayed and potentially inconsistent information sampled from both fixed and time-varying delay distributions grounded in real-world measurements. This design enables policies to reason over temporally misaligned inputs and develop coordination strategies that remain robust under imperfect communication.

In addition to robustness, safety remains a critical requirement for real-world multi-agent systems, particularly in safety-critical domains. To address this, we integrate a safety-aware control mechanism based on Control Bar-

rier Functions (CBFs), which enforces provable safety constraints during execution. This safety layer operates alongside the learned policy, minimally modifying actions only when necessary to prevent unsafe behavior. By combining delay-aware learning with formal safety guarantees, the proposed framework ensures that coordination remains both effective and reliable under realistic operating conditions.

We validate our approach in both simulation and real-world multi-agent systems, demonstrating consistent zero-shot transfer from simulation to hardware under delayed communication. Our results show that explicitly modeling communication latency during training leads to improved coordination stability, reduced safety violations, and stronger robustness compared to approaches that assume ideal communication or rely solely on domain randomization. Notably, policies trained under time-varying delay distributions exhibit the best overall performance, highlighting the importance of capturing realistic communication dynamics during learning.

The contributions of this work are threefold:

- We introduce a delay-aware learning formulation that treats inter-agent communication as a structured, time-varying process rather than unstructured noise, enabling policies to reason over temporally misaligned information.
- We propose a unified learning-and-execution framework that integrates delay-aware policy learning with safety-constrained control, allowing robust coordination under imperfect communication without modifying the underlying policy structure.
- We demonstrate that explicitly modeling communication delay during training is sufficient to enable zero-shot sim-to-real transfer under real-world latency, outperforming domain randomization and fixed-delay baselines.

Our findings highlight the importance of incorporating structured communication constraints into the learning process and provide a practical step toward deploying cooperative multi-agent intelligence in real-world environments.

2. Method

At a high level, our framework modifies the standard multi-agent learning pipeline by introducing delay injection during training and safety filtering during execution, enabling consistent training and deployment under communication constraints. We present a delay-aware learning framework for multi-agent coordination that departs from standard assumptions of instantaneous and synchronized information exchange by explicitly modeling communication delay throughout both training and deployment. By combining delay-aware policy learning with a safety-constrained execution layer, the proposed approach enables robust and reliable coordination in real-world multi-agent systems operating under imperfect communication.

2.1. Problem Setup: Learning Under Delayed Communication

We consider a multi-agent system in which each agent must make decisions based on both local observations and information received from neighboring agents. Unlike conventional formulations that assume immediate access to shared information, we model inter-agent communication as delayed and potentially asynchronous.

At each time step, agent i receives a local observation o_i and a set of shared observations from neighboring agents. Due to communication latency, these shared observations are not current but instead correspond to past states. We denote the effective input to the policy as a delay-aware observation:

$$s_i^{\text{del}} = \{o_i, \tilde{o}_j \mid j \in \mathcal{N}_i\}, \tag{1}$$

where $\tilde{o}_j$ represents a delayed observation from agent j.

The objective is to learn decentralized policies $\pi_\theta(a_i \mid s_i^{\text{del}})$ that enable coordinated behavior despite temporal misalignment in shared information. This setting introduces additional uncertainty and partial observability, requiring agents to reason over stale and inconsistent inputs during decision-making.

2.2. Delay-Aware Multi-Agent Learning

The key idea of our approach is to explicitly incorporate communication delay into the training process. Rather than treating communication imperfections as unstructured noise, we model delay as a structured temporal process and expose agents to these conditions during learning, in line with broader efforts toward robust reinforcement learning under uncertainty [5]. This formulation enables agents to reason over temporally misaligned information and learn coordination strategies that remain robust under asynchronous communication.

We consider two types of delay models:

- **Fixed Delay:** A constant delay of k time steps is applied to all shared observations.
- **Time-Varying Delay:** Delay is sampled from a distribution D_{lat}, representing realistic communication latency observed in real-world systems.

During training, shared observations are passed through a delay buffer that simulates communication latency. For each agent i, the received observation from agent j at time t is given by:

$$\tilde{o}_j^t = o_j^{t-d_j^t}, \quad d_j^t \sim D_{\text{lat}}. \tag{2}$$

By training under both fixed and time-varying delay conditions, agents learn to anticipate and compensate for stale information. This results in policies that are inherently robust to asynchronous communication and variability in latency.

Importantly, the delay distributions used during training are grounded in real-world measurements, ensuring that the learned policies are exposed to realistic communication conditions. This design enables improved generalization and supports reliable zero-shot transfer to real-world systems.

In practice, the learning and execution process follows a simple loop:

1. Collect local and delayed neighbor observations.
2. Construct delay-aware state s_i^{del}.
3. Compute action $a_i \sim \pi_\theta(\cdot \mid s_i^{\text{del}})$.
4. Apply safety filter to obtain safe action a_i^{safe}.
5. Execute action and store transition for learning.

2.3. Safety-Constrained Execution

To ensure reliable deployment in safety-critical environments, we incorporate a safety-aware control layer that operates alongside the learned policy. In the presence of delayed and asynchronous communication, agents may act on stale or inconsistent information, increasing the risk of unsafe behavior. The safety layer mitigates this risk by filtering the policy's action before execution.

We implement this safety layer using Control Barrier Functions (CBFs), which enforce forward invariance of a predefined safe set. Given a reference control input $\mathbf{u}_{\text{ref}}$ from the policy, the safety filter solves the following optimization problem:

$$\begin{aligned} \min_{\mathbf{u}} \quad & \|\mathbf{u} - \mathbf{u}_{\text{ref}}\|^2 \\ \text{s.t.} \quad & \dot{h}(\mathbf{x}) + \alpha h(\mathbf{x}) \geq 0, \end{aligned} \tag{3}$$

where $h(\mathbf{x})$ defines the safe set and $\alpha > 0$ is a tunable parameter.

This formulation minimally modifies the proposed action while ensuring that safety constraints are satisfied. The safety layer operates in a decentralized manner, using locally available state information and delayed observations of neighboring agents. To account for uncertainty introduced by communication delays, safety margins are conservatively adjusted to maintain robustness under imperfect information.

Together with delay-aware learning, this safety-constrained execution mechanism enables multi-agent systems to achieve coordinated behavior that is both robust to communication imperfections and provably safe in real-world deployment.

3. Experiments

We evaluate our approach with the goal of answering three key questions: (1) Does delay-aware learning improve coordination under imperfect communication? (2) Does the learned policy transfer reliably to real-world systems? and (3) How does communication impact safety and coordination?

Table 1. Real-world performance under increasing scenario complexity. Each entry reports (collisions, completion time).

Method	Easy	Medium	Hard
Delay-Aware (TV)	0, 33.1	0, 34.0	0, 34.5
Delay-Aware (Fixed)	0, 33.6	0, 34.8	0, 35.2
No Communication	0, 36.7	1, 37.9	1, 39.5
No Safety	1, 36.8	2, 37.5	3, 38.0
Domain Randomization	2, 36.5	3, 37.8	5, 39.2

3.1. Experimental Setup

We conduct experiments in both simulation and real-world multi-agent systems. In simulation, we use the CARLA driving simulator [2] to train and evaluate policies under controlled communication delay settings. For real-world validation, we deploy trained policies on a fleet of 1/10th-scale autonomous vehicles built on the F1TENTH platform [6].

Communication delay is modeled during training using both fixed and time-varying latency. Fixed delay applies a constant lag to shared observations, while time-varying delay samples latency from a distribution that reflects real-world communication variability. During deployment, agents operate under naturally occurring communication delays, without any additional adaptation.

We compare against several baselines, including policies trained with ideal (no-delay) communication, policies trained with domain randomization [8], and variants without safety constraints or inter-agent communication. We also compare against a prior robust-safe MARL formulation that does not explicitly model communication delay during training [9].

3.2. Real-World Coordination Under Delayed Communication

We first evaluate performance in real-world multi-agent scenarios requiring coordination under delayed communication. Table 1 summarizes results across multiple environments and levels of complexity.

Delay-aware policies achieve zero collisions across all scenarios while maintaining competitive completion times. In contrast, removing safety constraints leads to frequent collisions, and domain randomization alone fails to ensure safe behavior. These results demonstrate that explicitly modeling communication delay during training is critical for reliable real-world coordination.

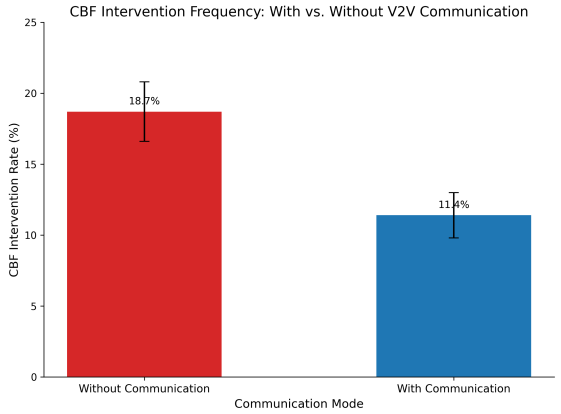

Figure 1. Safety intervention frequency with and without inter-agent communication. Communication reduces unsafe behaviors and improves coordination stability.

3.3. Impact of Delay Modeling

We analyze the effect of different delay models on policy robustness. Policies trained under fixed delay perform well when the deployment latency matches training conditions but degrade under variability. In contrast, policies trained with time-varying delay distributions maintain stable performance across a range of communication conditions.

This highlights that modeling communication as a stochastic temporal process during training leads to more robust coordination strategies than optimizing for a single fixed latency. Time-varying delay encourages policies to reason over uncertainty in shared information, improving generalization to real-world environments.

3.4. Role of Communication in Safety and Coordination

We further examine how inter-agent communication impacts safety by measuring the frequency of safety interventions required during execution. Figure 1 shows that incorporating communication reduces the need for reactive safety corrections.

When communication is disabled, agents rely solely on local observations, leading to more frequent safety violations and increased reliance on the safety layer. In contrast, access to shared (even delayed) information enables more informed decision-making, reducing unsafe actions and improving overall coordination.

4. Discussion and Conclusion

We presented a delay-aware multi-agent learning framework that enables robust coordination under imperfect communication. Our results show that explicitly modeling communication delay as a structured, time-varying process is critical for bridging the gap between simulation and real-world deployment.

In particular, policies trained with time-varying delay distributions demonstrate improved robustness compared to fixed-delay and domain randomization approaches, highlighting the importance of capturing realistic communication dynamics during training.

We further show that integrating a safety-constrained execution layer ensures reliable deployment by mitigating risks arising from delayed and asynchronous information. Together, these results suggest that incorporating communication constraints directly into the learning process is essential for scalable cooperative intelligence.

Future work will explore more complex communication settings, including packet loss and bandwidth limitations.

References

[1] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022. 1

[2] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. pages 1–16, 2017. 3

[3] Ingy ElSayed-Aly, Suda Bharadwaj, Christopher Amato, Rüdiger Ehlers, Ufuk Topcu, and Lu Feng. Safe multi-agent reinforcement learning via shielding. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, page 483–491, Richland, SC, 2021. International Foundation for Autonomous Agents and Multiagent Systems. 1

[4] Songyang Han, Shanglin Zhou, Jiangwei Wang, Lynn Pepin, Caiwen Ding, Jie Fu, and Fei Miao. A multi-agent reinforcement learning approach for safe and efficient behavior planning of connected autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 25(5):3654–3670, 2024. 1

[5] Yongyuan Liang, Yanchao Sun, Ruijie Zheng, and Furong Huang. Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. *Advances in Neural Information Processing Systems*, 35:22547–22561, 2022. 2

[6] Matthew O'kelly, Hongrui Zheng, Dhruv Karthik, Rahul Mangharam, Hugo Jair Escalante, and Raia Hadsell. F1tenth: An open-source evaluation environment for continuous control and reinforcement learning. *Proceedings of Machine Learning Research*, 123:77–89, 2020. 3

[7] Sandeep Singh Sandha, Luis Garcia, Bharathan Balaji, Fatima Anwar, and Mani Srivastava. Sim2real transfer for deep reinforcement learning with stochastic state transition delays. In *Proceedings of the 2020 Conference on Robot Learning*, pages 1066–1083. PMLR, 2021. 1

[8] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for

transferring deep neural networks from simulation to the real world. *IEEE International Conference on Intelligent Robots and Systems*, 2017-September:23–30, 2017. 1, 3

[9] Zhili Zhang, H M Sabbir Ahmad, Ehsan Sabouni, Yanchao Sun, Furong Huang, Wenchao Li, and Fei Miao. Safety guaranteed robust multi-agent reinforcement learning with hierarchical control for connected and automated vehicles. 2023. 3