Asynchronous Cooperative Multi-Agent Reinforcement Learning with Limited Communication

Extended Abstract

Sydney Dolan
Massachusetts Institute of Technology
Cambridge, MA, United States
sydneyd@mit.edu

Siddharth Nayak
Massachusetts Institute of Technology
Cambridge, MA, United States
sidnayak@mit.edu

Jasmine Jerry Aloor
Massachusetts Institute of Technology
Cambridge, MA, United States
jjaloor@mit.edu

Hamsa Balakrishnan
Massachusetts Institute of Technology
Cambridge, MA, United States
hamsa@mit.edu

ABSTRACT

We consider the problem setting in which multiple autonomous agents must cooperatively navigate and perform tasks in an unknown, communication-constrained environment. Traditional multi-agent reinforcement learning (MARL) approaches assume synchronous communications and perform poorly in such environments. We propose AsynCoMARL, an asynchronous MARL approach that uses graph transformers to learn communication protocols from dynamic graphs. AsynCoMARL can accommodate infrequent and asynchronous communications between agents, with edges of the graph only forming when agents communicate with each other. We show that AsynCoMARL achieves similar success and collision rates as leading baselines, despite 26% fewer messages being passed between agents.

KEYWORDS

mult-agent reinforcement learning, satellite navigation

ACM Reference Format:
Sydney Dolan, Siddharth Nayak, Jasmine Jerry Aloor, and Hamsa Balakrishnan. 2025. Asynchronous Cooperative Multi-Agent Reinforcement Learning with Limited Communication: Extended Abstract. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 3 pages.

1 INTRODUCTION

Communication is crucial in cooperative multi-agent systems with partial observability, as it enables a better understanding of the environment and improves coordination. In extreme environments such as those underwater or in space, the frequency of communication between agents is often limited [7, 8]. For example, a satellite may not be able to reliably receive and react to messages from other satellites synchronously due to limited onboard power and communication delays. In these scenarios, agents aim to establish a communication protocol that allows them to operate independently while still receiving sufficient information to effectively coordinate with nearby agents.

Multi-agent reinforcement learning (MARL) has emerged as a popular approach for addressing cooperative navigation challenges involving multiple agents. The classical MARL framework is synchronous, with agents acting and communicating instantly, frequently, and simultaneously, often broadcasting their states to all others. As a result, traditional MARL algorithms are poorly suited to asynchronous settings where agents operate on independent time scales and cannot frequently communicate with one another. We propose AsynCoMARL, a graph transformer-based communication protocol for MARL that relies on dynamic graphs to capture asynchronous and infrequent communications between agents. We empirically evaluate AsynCoMARL on two MARL benchmarks (Cooperative Navigation [2], and Rover-Tower [4]) and show that our method can achieve superior performance while using less communication.

2 METHODOLOGY

Figure 1 provides a high-level summary of our algorithm. We define a time scale τ to account for the specific actions each agent has taken at different time steps. In Figure 1 panel (b), all three agents take their first action at the same t, resulting in the same time reference point for τ_1. We only include those steps where the agents are taking an action. To improve the generalizability of our algorithm to different periods of time between actions τ_1 and τ_2, during training, we randomly generate the period between different actions. This randomly generated period is chosen from a uniform distribution, with different intervals used for training and testing to ensure diversity. For each initiation, the delay is selected randomly from this distribution and remains constant throughout the episode rather than changing after specific actions. This decision avoids creating an incorrect association between certain movement actions and specific delays, as we consider such realism beyond the scope of this work and more applicable to domain-specific problems. Following panel (c) of Figure 1, we rely on a graph transformer to encode messages and characterize relationships between different entities in the environment. Algorithmic details and more experiments can be found in [1].

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

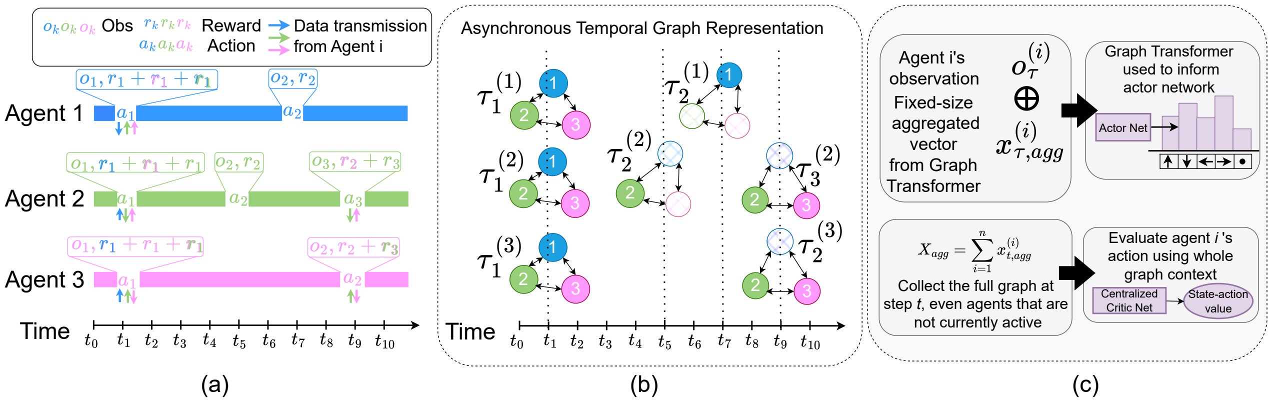

Figure 1: Overview of AsynCoMARL: (a) Agents acting simultaneously at t receive a shared reward. Arrows show recent graph observations. (b) Active agents map to nodes on a dynamic graph, communicating with nearby agents. (c) Agent i's observation merges with graph transformer outputs for the actor network, while the critic evaluates its action using the full graph.

Algorithm	$f_{\text{comm}} \downarrow$	$T \downarrow$	# col $\downarrow$	$S\% \uparrow$
GCS [10]	1.0	0.36	0.34	100
asyncMAPPO [11]	0.21	0.10	0.86	100
Actor-Attention-Critic [4]	0.21	0.42	0.30	100
TransfQmix [3]	0.13	0.83	0.02	42
CACOM [6]	0.26	0.99	0.17	0
DGN [5]	0.20	0.96	0.12	0
AsynCoMARL	**0.10**	**0.24**	**0.45**	**97**

Table 1: Comparison of AsynCoMARL against other methods in the Cooperative Navigation environment.

3 RESULTS

Cooperative Navigation: Table 1 compares the performance of AsynCoMARL against the other baselines. The collision rate (# col) should be judged in the context of success rate ($S\%$), as some policies do not move the agents from their initial positions. When evaluating AsynCoMARL's performance in the context of these other baselines, our method is able to achieve high success rates ($S\%$), efficient episode completion rates (T), and relatively low collision rates, despite 26% fewer messages being passed between agents (f_{comm}). The temporal graph formulation of our model, which inherently allows communications to be masked to reduce communication overhead during training, leads to a method capable of handling trade-offs between communication frequency, success, and collision avoidance. Both asyncMAPPO and Actor-Attention-Critic demonstrate comparable performance in success and collision rates. Similar to the design of AsynCoMARL, Actor-Attention-Critic is designed to dynamically select which agents to focus on. This reduces f_{comm} and leads to improved success and collision rates. However, this attention mechanism overlooks relationships between agents captured by the graph representation used in AsynCoMARL, leading Actor-Attention-Critic to have a higher communication frequency and episode completion rates.

Rover-Tower: Table 2 shows AsynCoMARL against the best-performing baselines from the prior experiment. The reward function associated with this environment does not include any collision penalty, so we do not include the # col metric. In this environment, rovers must rely on encoded messages from their corresponding tower to determine their action selection, whereas towers have more advanced observation abilities. To account for these two classes, Actor-Attention-Critic creates a separate network for the rover class and the tower class, whereas AsynCoMARL does not. Despite the fact that AsynCoMARL is using a singular network to represent both the rovers and the towers, it still achieves a comparable success rate to the Actor-Attention-Critic. Additionally, AsynCoMARLrelies on less communication and produces faster episode completion rates than other baselines, suggesting that AsynCoMARL is a more efficient, generalizable communication protocol for this environment.

Algorithm	$f_{\text{comm}} \downarrow$	$T \downarrow$	$S\% \uparrow$
Actor-Attention-Critic [4]	0.21	0.84	56%
AsyncMAPPO [11]	0.24	0.98	0%
TransfQmix [3]	0.40	0.98	0%
AsynCoMARL	**0.14**	**0.55**	**50%**

Table 2: Comparison against other methods in the Rover-Tower environment.

Discussion: In future research, we aim to explore more advanced communication protocol architectures that can model different action-communication constraints common in real-world settings. Additionally, we want to investigate the feasibility of integrating additional mechanisms like control barrier functions to reduce the overall number of collisions. *

*The authors would like to thank the MIT SuperCloud [9] and the Lincoln Laboratory Supercomputing Center for providing high-performance computing resources that have contributed to the research results reported in this paper. This work was supported in part by NASA under grant #80NSSC23M0220 and the University Leadership Initiative (grant #80NSSC20M0163), but this article solely reflects the opinions and conclusions of its authors and not any NASA entity. The research was sponsored by the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notice herein. Sydney Dolan was supported in part by the National Science Foundation Graduate Research Fellowship under Grant No. 1650114. J. Aloor was also supported in part by a Mathworks Fellowship.

REFERENCES

[1] Sydney Dolan, Siddharth Nayak, Jasmine Jerry Aloor, and Hamsa Balakrishnan. 2025. Asynchronous Cooperative Multi-Agent Reinforcement Learning with Limited Communication. arXiv:2502.00558 [cs.MA] https://arxiv.org/abs/2502.00558

[2] Sydney Dolan, Siddharth Nayak, and Hamsa Balakrishnan. 2023. Satellite Navigation and Coordination with Limited Information Sharing. arXiv:2211.03658 [cs.MA]

[3] Matteo Gallici, Mario Martin, and Ivan Masmitja. 2023. TransfQMix: Transformers for leveraging the graph structure of multi-agent reinforcement learning problems. *arXiv preprint arXiv:2301.05334* (2023).

[4] Shariq Iqbal and Fei Sha. 2019. Actor-Attention-Critic for Multi-Agent Reinforcement Learning. arXiv:1810.02912 [cs.LG] https://arxiv.org/abs/1810.02912

[5] Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. 2020. Graph Convolutional Reinforcement Learning. In *ICLR*.

[6] Xinran Li and Jun Zhang. 2024. Context-aware Communication for Multi-agent Reinforcement Learning. arXiv:2312.15600 [cs.LG] https://arxiv.org/abs/2312.15600

[7] Antoni Martorell-Torres, José Guerrero-Sastre, and Gabriel Oliver-Codina. 2024. Coordination of marine multi robot systems with communication constraints. *Applied Ocean Research* 142 (2024), 103848. https://doi.org/10.1016/j.apor.2023.103848

[8] Issa A.D. Nesnas, Lorraine M. Fesq, and Richard A. Volpe. 2021. Autonomy for Space Robots: Past, present, and future. *Current Robotics Reports* 2, 3 (Jun 2021), 251–263. https://doi.org/10.1007/s43154-021-00057-2

[9] Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, Michael Jones, Anna Klein, Lauren Milechin, Julia Mullen, Andrew Prout, Antonio Rosa, Charles Yee, and Peter Michaleas. 2018. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*. IEEE, 1–6.

[10] Jingqing Ruan, Yali Du, Xuantang Xiong, Dengpeng Xing, Xiyun Li, Linghui Meng, Haifeng Zhang, Jun Wang, and Bo Xu. 2022. GCS: Graph-based Coordination Strategy for Multi-Agent Reinforcement Learning. arXiv:2201.06257 [cs.MA] https://arxiv.org/abs/2201.06257

[11] Chao Yu, Xinyi Yang, Jiaxuan Gao, Jiayu Chen, Yunfei Li, Jijia Liu, Yunfei Xiang, Ruixin Huang, Huazhong Yang, Yi Wu, and Yu Wang. 2023. Asynchronous Multi-Agent Reinforcement Learning for Efficient Real-Time Multi-Robot Cooperative Exploration. arXiv:2301.03398 [cs.RO] https://arxiv.org/abs/2301.03398