Learning Collaborative Information Dissemination with Graph-based Multi-Agent Reinforcement Learning

Raffaele Galliera[†,⋆], **Kristen Brent Venable**[†,⋆], **Matteo Bassani**[†], **Niranjan Suri**[†,⋆,‡]
[†]Institute for Human & Machine Cognition
[⋆]Department of Intelligent Systems & Robotics, The University of West Florida
Pensacola, FL, USA
[‡]US Army Research Laboratory
Adelphi, MD, USA
{rgalliera, bvenable, mbassani, nsuri}@ihmc.org

Abstract

In modern communication systems, efficient and reliable information dissemination is crucial for supporting critical operations across domains like disaster response, autonomous vehicles, and sensor networks. This paper introduces a Multi-Agent Reinforcement Learning (MARL) approach as a significant step forward in achieving more decentralized, efficient, and collaborative solutions. We propose a Partially Observable Stochastic Game (POSG) formulation for information dissemination empowering each agent to decide on message forwarding independently, based on their one-hop neighborhood and the degree of connectivity of each neighbor. This constitutes a significant paradigm shift from traditional heuristics based on Multi-Point Relay (MPR) selection. Our approach harnesses Graph Convolutional Reinforcement Learning, employing Graph Attention Networks (GAT) with dynamic attention to capture essential network features. We propose two approaches, L-DGN and HL-DGN, which differ in the information that is exchanged among agents. We evaluate the performance of our decentralized approaches, by comparing them with a widely-used MPR heuristic, and we show that our trained policies are able to efficiently cover the network while bypassing the MPR set selection process. Our approach promises a first step toward supporting the resilience of real-world broadcast communication infrastructures via learned, collaborative information dissemination.

1 Introduction

Nowadays, group communication, implemented in a broadcast or multicast fashion, finds a natural application in different networking systems, such as Vehicular Ad-hoc Networks (VANETs)(Tonguz et al. [2007], Ibrahim et al. [2020]), with the necessity to disseminate information about the nodes participating, e.g. identity, status, and position, or crucial events happening in the network. In the context of broadcast networking systems, Optimized Link State Routing (OLSR) (Dearlove and Clausen [2014]) is a widely used proactive routing protocol which leverages a technique called **Multi Point Relay (MPR) selection** (Qayyum et al. [2002], Adjih et al. [2005]) to optimize information dissemination. Given a source node sending a message to its neighbors, such a distributed task requires all the nodes to coordinate and disseminate the information across the network while minimizing the number of information forwards needed. Thanks to the exchange of "HELLO messages" present in OLSR, nodes discover information about their two-hop neighborhood and designate specific one-hop neighbors as responsible for forwarding information they transmit, namely their MPR set.

Machine Learning for Systems Workshop at 37th NeurIPS Conference, 2023, New Orleans, LA, USA.

Recently, researchers have considered learning communication protocols (Foerster et al. [2016]) with Multi-Agent Reinforcement Learning (MARL) (Buşoniu et al. [2010]). Nevertheless, learning to communicate with MARL comes with several challenges. In multi-agent systems, actions taken by one agent can significantly impact the rewards and state transitions of other agents, rendering the environment more complex and dynamic, and ensuring that agents develop a shared and consistent communication protocol, is an area of active research. Methods such as CommNet (Sukhbaatar et al. [2016]) and BiCNet (Peng et al. [2017]), focus on the communication of local encodings of agents' observations. Yet another approach, as exemplified by DGN (Jiang et al. [2020]), harnesses the power of Graph Neural Networks (GNNs) to model the interactions and communications between agents. By representing the multi-agent system as a graph, DGN captures the complex relations between agents, facilitating the emergence of effective strategies even when constrained communication may limit the range of cooperation.

While recent work has considered a MARL approach (Kaviani et al. [2023]) addressing the optimization of goodput (bits of useful data delivered at target location per unit of time) in dynamic multicast networks, to the best of our knowledge, no MARL-based method involving active, learned communication by the agents and Graph Convolution has been proposed to address the unique challenges of optimizing the process of information dissemination within a broadcast network.

In this work, we propose **a novel cooperative MARL formulation for information dissemination** and **two different architectures, Local-DGN and Hyperlocal-DGN**, leveraging Graph Convolutional Reinforcement Learning and Graph Attention Network (GAT) with dynamic attention. Our **experimental study** shows that our approach outperforms the widely-used standard heuristic in achieving **network coverage with reduced communication overhead**.

Our approach is a first step towards collaborative autonomous agents capable of optimizing information dissemination while exploiting communication mechanisms present in broadcast protocols. In this way, our work underscores the versatility of MARL in present and future, real-world applications such as information dissemination in social networks (Guille et al. [2013]), space networks (Ye and Zhou [2021]), and vehicle-safety-related communication services (Ma et al. [2012]).

2 Method

In the proposed decentralized approach, nodes in a network represent agents in a Partially Observable Stochastic Game (POSG), with the actions available being to forward a message or not. Agents discover one-hop neighbors and gather information about their neighbors through protocols such as OLSR (Clausen and Jacquet [2003], Dearlove and Clausen [2014]). It is worth mentioning that, in such protocols, agents are required to know the unique identifiers of each of their two-hop neighbors, while in our setting they are only aware of the number of connections their one-hop neighbors have. The agents receive a reward signal based on their 2-hop coverage and penalties depending on their behavior and the neighborhood's activity.

Agents are categorized into active, done, and idle, transitioning between these sets based on message receipt. Upon receiving the message, agents start their individual experience lasting a fixed number of steps, named *local horizon*, representing the portion of the entire dissemination process where the agent actively interacts with its neighborhood. The dissemination process is discretized into time steps where active agents act simultaneously.

Observations $\mathcal{O}^i_{i\in I}$. Each node in the graph has a set of six features, including its number of neighbors, the number of messages transmitted, and its action history. The latter identifies in which, if any, of its active turns, the agent forwarded the message. Assuming the local horizon is equal to k, this last feature can be represented as a binary array of size k. An agent's observation includes its own features and those of its one-hop neighbors.

Actions $\mathcal{A}^i_{i\in I}$. For any time step t, every active agent has two possible actions: to forward or not the information to their neighbors. Despite being not desirable, active agents are allowed to forward the information multiple times. If an agent forwards a message, each of its neighbors will receive it.

Rewards $\mathcal{R}^i{}_{i\in I}$. At the end of each step every agent is issued with a reward signal. Such signals are made of two main components, a positive and a negative reward. The positive term rewards the agent based on its two-hop coverage, i.e. how many one- and two-hop neighbors have been covered, capturing the ability of observable (the agent's neighborhood) and unobservable agents to

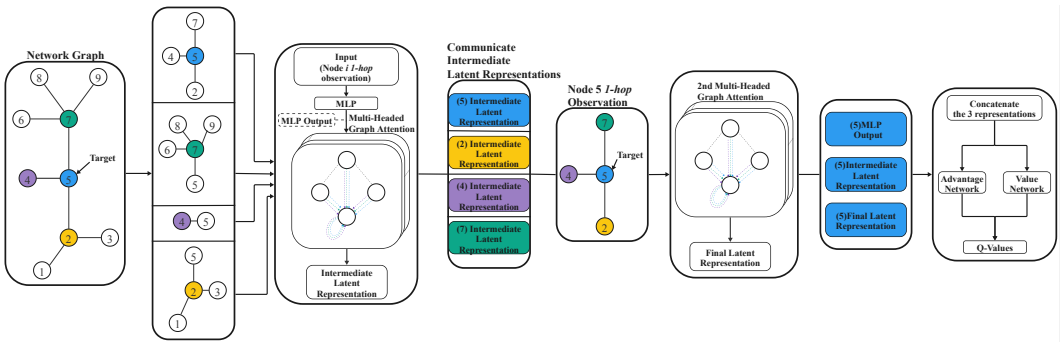

Figure 1: The Local-DGN (L-DGN) architecture.

contribute to the dissemination task. One of two different penalties might be issued, based on the agent's behavior. If the agent has ever forwarded the message, it will participate in a *"neighborhood shared transmission cost"* punishing the agent for the number of forwards sensed in its neighborhood. Otherwise, the agent will incur a penalty based on the *"coverage potential"* of its neighborhood, determined by the size of its most connected neighbor's neighborhood.

2.1 Local and Hyperlocal Relation Kernels

Our methods, namely L-DGN and HL-DGN, are based on the concept of Relation Kernels, essential in Graph Convolutional Reinforcement Learning as implemented in the DGN model (Jiang et al. [2020]). The L-DGN architecture is illustrated in Figure 1, comprising an encoder module with multiple stages with two GATs layers. The HL-DGN, is presented in Figure 2 and comprises a simpler architecture, with a single GAT layer and a global max pool operation.

Training is performed in a Double Deep Q Networks (DDQNs) fashion (Hasselt et al. [2016]) while using n-step Temporal Difference (TD) estimation with n equal to the agent's local horizon. Both methods employ full parameter sharing across the agents and a circular replay buffer is utilized to store tuples of observations, actions, and rewards. Final latent representations are then processed by a Dueling Q-Network (Wang et al. [2016]) to compute the predicted Q values.

In L-DGN, feature vectors from nodes in local regions are processed with two GATs layers, expanding the agent's receptive field to its two-hop neighborhood and enhancing cooperation. This model can be naturally implemented in broadcast network protocols, allowing agents to share their learned neighborhood-wise latent representations, instead of MPR sets as seen for OLSR. Such embedding communication enables better collaboration between the agents while preserving information privacy, as they do not require explicit communication of two-hop neighbor identifiers. The architecture utilizes a combination of Multi Layer Perceptron (MLP), multi-headed GATs (Veličković et al. [2018]), and dynamic attention (Brody et al. [2022]) to produce a final latent representation.

We hypothesize a more communication-efficient approach by restricting information exchange. To this end, we remove the second GAT layer present in L-DGN, which involves a message exchange process corresponding to the communication of the MPR sets employed in OLSR. This leads to the development of Hyperlocal-DGN (HL-DGN) (Figure 2), which aims to reduce communication overhead. While following the same learning methods, HL-DGN adopts a simplified policy parameterization, with a single GAT layer followed by a global max-pooling layer, inspired by traditional approaches seen in Convolutional Neural Networks (CNNs) (Cireşan et al. [2011]).

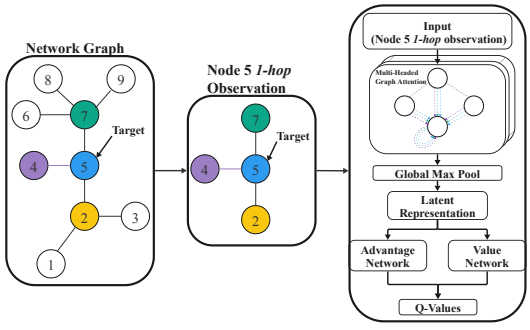

Figure 2: The Hyperlocal-DGN (HL-DGN).

The rationale for using a pooling layer is that agents can make informed decisions by simply observing their immediate neighbors and processing the neighborhood dynamics from each neighbor's perspective, eliminating the need to share their latent representations.

3 Experiments

A first set of 50K connected static graph topologies is generated, with 20 nodes per graph, and no constraints on the number of one-hop neighbors. In addition, two separate sets of 100 topologies are used for testing, respectively with 20 and 50 nodes per graph. When training, the environment selects a random graph, as well as a random node to be the source of the information to disseminate. During testing a precise node is systematically chosen to be the source in order to encourage reproducibility and coherence of the results. In our experiments, we set the value of the local horizon to 4.

We conduct a comprehensive comparative analysis involving our two novel methodologies, L-DGN and HL-DGN, alongside the MPR heuristic, and DGN-R, the variant of DGN (Jiang et al. [2020]) that does not include Temporal Relation Regularization, which is not required in our setting where agent interaction is temporally bounded by a short local-horizon. To ensure an equitable evaluation, we maintain consistent hyperparameters and layer dimensions for each layer type across all three models. In both our proposed methodologies and DGN-R, we employ GAT with dynamic attention.

We note that, in our setting, all the nodes begin the process simultaneously and, due to assumed perfect synchronization among nodes, every node diffuses information at precisely the same time. To this end, a "bootstrap phase" is defined, during which nodes engage in two successive rounds of HELLO messages, each serving a distinct purpose. The first round establishes the presence of nodes and forms initial network connectivity. In the second round, nodes exchange the acquired information within their one-hop neighborhood, leading to each node gaining knowledge of their two-hop neighborhood. We note that the information exchanged between the nodes in this phase depends on the approach being used (i.e., neighbors' ids for MPR, and neighborhood size for all other methods). After this bootstrap phase, a third round is dedicated to broadcasting pre-calculated MPR sets or latent representations for, respectively, MPR selection and L-DGN or DGN-R. Summarizing, the MPR heuristic, DGN-R, and L-DGN, all demonstrate a control message overhead proportional to three times the number of nodes, while HL-DGN demonstrates an overhead scaled down to two times the node count, thanks to the absence of the third round of HELLO messages.

Table 1 presents a summary of our results. We note that with 20 nodes (resp. 50 nodes), HL-DGN successfully attains full coverage while employing 13.17 (resp. 35.1) data messages requiring one less round of HELLO messages. Moreover, L-DGN model maintains a high coverage rate of 99.95% (resp. 93.3%), with the overall lowest message count of 11.84 (resp. 25.42). Both the MPR heuristic and the DGN-R achieve full coverage, the first employing 12.05 (resp. 30.8), while the second 21.06 (resp. 60.65).

Nodes	Method	Coverage	Data Messages	Control Overhead
20				
	MPR	100%	12.05	60
	DGN-R	100%	21.06	60
	L-DGN	99.95%	11.84	60
	HL-DGN	100%	13.17	40
50				
	MPR	100%	30.8	150
	DGN-R	99.98%	60.65	150
	L-DGN	93.3%	25.42	150
	HL-DGN	100%	35.1	100

Table 1: Performance comparison.

4 Conclusion

Our results underscore the efficacy of our proposed MARL-based approaches to learning effective forwarding strategies compatible with communication mechanisms present in widely-used broadcast protocols, such as OLSR Dearlove and Clausen [2014].

This work paves the way for the integration of learned strategies to optimize forwarding decisions in real group communication protocols, and to an investigation of more complex and dynamic systems. Orthogonally, the application of our approach can be extended beyond broadcast networks to the dissemination of information in domains with higher levels of abstraction, such as social networks.

References

Cédric Adjih, Philippe Jacquet, and Laurent Viennot. Computing connected dominated sets with multipoint relays. *Ad Hoc & Sensor Wireless Networks*, 1(1-2):27–39, 2005. URL https://inria.hal.science/inria-00471715.

Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *International Conference on Learning Representations (Poster)*, 2022. URL https://openreview.net/forum?id=F72ximsx7C1.

Lucian Buşoniu, Robert Babuška, and Bart De Schutter. *Multi-agent Reinforcement Learning: An Overview*, pages 183–221. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-14435-6. doi: 10.1007/978-3-642-14435-6_7. URL https://doi.org/10.1007/978-3-642-14435-6_7.

Dan C. Cireşan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, page 1237–1242. AAAI Press, 2011. ISBN 9781577355144.

Thomas H. Clausen and Philippe Jacquet. Optimized Link State Routing Protocol (OLSR). RFC 3626, October 2003. URL https://www.rfc-editor.org/info/rfc3626.

Christopher Dearlove and Thomas H. Clausen. Optimized Link State Routing Protocol Version 2 (OLSRv2) and MANET Neighborhood Discovery Protocol (NHDP) Extension TLVs. RFC 7188, April 2014. URL https://www.rfc-editor.org/info/rfc7188.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2145–2153, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A. Zighed. Information diffusion in online social networks: A survey. *SIGMOD Rec.*, 42(2):17–28, jul 2013. ISSN 0163-5808. doi: 10.1145/2503792.2503797. URL https://doi.org/10.1145/2503792.2503797.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2094–2100. AAAI Press, 2016.

Banar Fareed Ibrahim, Mehmet Toycan, and Hiwa Abdulkarim Mawlood. A comprehensive survey on vanet broadcast protocols. In *2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM)*, pages 298–302, 2020. doi: 10.1109/ICCAKM46823.2020.9051462.

Jiechuan Jiang, Chen Dun, Tiejun Huang, and Zongqing Lu. Graph convolutional reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkxdQkSYDB.

Saeed Kaviani, Bo Ryu, Ejaz Ahmed, Deokseong Kim, Jae Kim, Carrie Spiker, and Blake Harnden. Deepmpr: Enhancing opportunistic routing in wireless networks through multi-agent deep reinforcement learning, 2023.

Xiaomin Ma, Jinsong Zhang, Xiaoyan Yin, and Kishor S. Trivedi. Design and analysis of a robust broadcast scheme for vanet safety-related services. *IEEE Transactions on Vehicular Technology*, 61(1):46–61, 2012. doi: 10.1109/TVT.2011.2177675.

Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games, 2017.

A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pages 3866–3875, 2002. doi: 10.1109/HICSS.2002.994521.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2252–2260, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Ozan Tonguz, Nawapom Wisitpongphan, Fan Bai, Priyantha Mudalige, and Varsha Sadekar. Broadcasting in vanet. In *2007 Mobile Networking for Vehicular Environments*, pages 7–12, 2007. doi: 10.1109/MOVE.2007.4300825.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1995–2003. JMLR.org, 2016.

Zipeng Ye and Qingrui Zhou. Performance evaluation indicators of space dynamic networks under broadcast mechanism. *Space: Science & Technology*, 2021, 2021. doi: 10.34133/2021/9826517. URL https://spj.science.org/doi/abs/10.34133/2021/9826517.

5 Appendix

5.1 Hyperparameters

	Hyperparameter	Value
Training		
	Training steps	1×10^6
	Learning rate	1×10^{-3}
	Buffer size	1×10^5
	Gamma	0.99
	Batch size	32
	Exploration Decay	Exponential
	Local Horizon	4
	N-Step Estimation	4
	Training Frequency	1 every 10 steps
	Gradient Steps	1
	Parallel Training Envs	40
	Experience Replay	Uniform
	Seed	9
Policy Parameters		
	MLP Hidden Size	512
	GAT Attention Heads	4
	GAT Hidden Size	128 (each head)
	A-Network Hidden Sizes	[128, 128]
	V-Network Hidden Sizes	[128, 128]

Table 2: Hyperparameters used across our experiments.

5.2 Ablation Study and Convergence

Along with L-DGN, HL-DGN, and DGN-R, we investigate the training performance of three ablations of our proposed architectures, namely DGN-R-Duel, L-DGN-MP, and L-DGN-MPNC. Such performance is measured in terms of the summation of the returns achieved by each agent that has participated in the dissemination task, named "graph return" (Figure 3). Given that our environment is highly dynamic in terms of the entities contributing to the dissemination task at each timestep, such a metric allows us to understand if the local rewards assigned to each agent correlate with a desired overall collaboration across the entire graph, measured in terms of summations of the rewards achieved.

DGN-R-Duel. The implementation of this method lies between L-DGN and DGN-R. Starting from the latter, we added the dueling network instead of a single MLP stream as the action decoder. Figure 3 shows the positive impact of the dueling network in the final strategy, which significantly outperforms DGN-R after 600K steps. From such a learning trajectory, we can also deduce the impact of another main component of our L-DGN, the n-step return estimation proportional to the local horizon. With the addition of such n-step returns, we obtain our L-DGN architecture, and we can notice how such a component helps the learned strategy to converge earlier and less abruptly.

L-DGN-MP. This method removes the second GAT layer of L-DGN and replaces it with the global max pool operator (later adopted by HL-DGN). The concatenation of the output of every encoding stage is still present here. We can notice a slight drop in performance when compared to L-DGN.

L-DGN-MPNC. This method removes both the second GAT layer of L-DGN, as well as the concatenation of the output of every encoding stage. We notice a decrease in performance when compared to L-DGN. It can also be seen that HL-DGN can be derived from L-DGN-MPNC after the ablation of the MLP encoding stage and that HL-DGN does not suffer from such performance reduction.

In summary, these ablation studies centered around L-DGN allow us to both understand the strengths of this approach when compared to DGN-R, as well as motivate the design of the HL-DGN architecture, which exhibits a simplified structure, less communication overhead, and only slightly underperforms in terms of graph return during training.

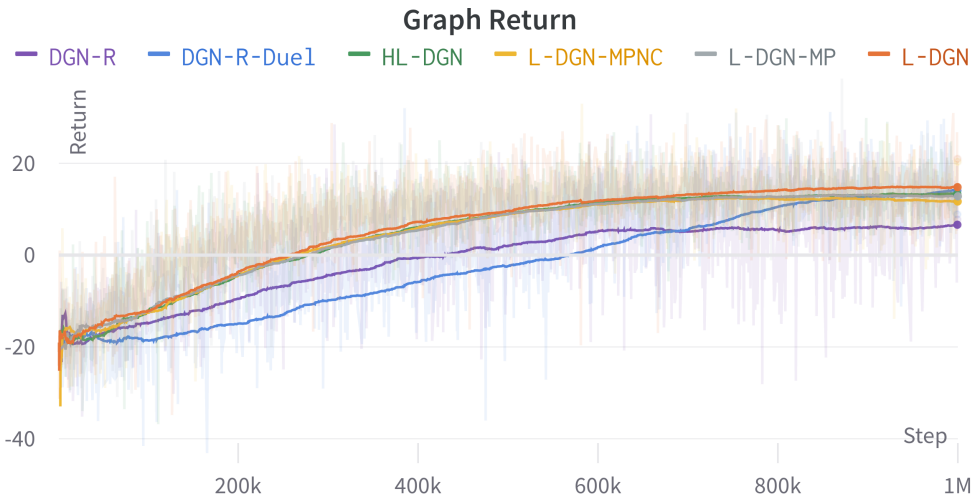

Figure 3: Graph return and convergence of the various methods used for the ablation study.