BG Pulse Journal USA
ISSN:1947-2536 http://www.bgpulseusa.com/index.php/bg

GRAPH NEURAL NETWORK (GNN) BASED TOPOLOGY CONTROL IN SELF-ORGANIZING WHEELED ROBOT SWARMS

Axmedov Jasur Rajab o‘g‘li
Master’s Student, ISMA University, Riga, Latvia
doi: https://doi.org/10.5281/zenodo.22824390

Abstract: This study examines a Graph Neural Network (GNN)-based approach for controlling communication topology and coordinated motion in self-organizing wheeled robot swarms under dynamically changing spatial and network conditions. The proposed framework represents robots as graph nodes and wireless communication links as graph edges, enabling the neural model to evaluate connectivity patterns and generate topology-aware motion decisions that preserve network coherence during collective navigation.
Keywords: Graph Neural Networks, robot swarms, topology control, self-organization, wheeled robots, wireless communication, network connectivity, multi-robot systems, artificial intelligence, cooperative navigation.

Introduction: The development of autonomous multi-robot systems has increased the importance of decentralized coordination mechanisms capable of maintaining reliable interaction among multiple mobile agents. A wheeled robot swarm can be regarded as a distributed cyber-physical system in which each robot performs sensing, computation, communication, and locomotion while simultaneously interacting with neighboring agents.[1,94 p]

In such systems, communication topology is not a fixed structural property because robot positions continuously change during navigation. Consequently, a movement that is optimal from an individual robot's perspective may reduce communication quality or disconnect a portion of the swarm from the remaining agents. This relationship establishes a direct dependency between physical movement and network structure.

Graph theory provides an appropriate mathematical representation for this problem. Let the swarm communication structure be represented by a graph (G=(V,E)), where (V) denotes the set of robots and (E) represents communication links established between neighboring agents. The graph changes when robots move, when communication distances vary, or when environmental conditions affect wireless connectivity.[3,136 p]

Traditional topology-control methods commonly rely on predefined distance thresholds, local heuristics, or manually designed optimization rules. Although such mechanisms can provide acceptable performance in relatively stable environments, their effectiveness may decrease when the swarm encounters dynamic obstacles, irregular spatial distributions, communication interference, or rapidly changing network conditions.

Graph Neural Networks provide a suitable computational framework because they process relational information directly. Unlike conventional neural architectures that primarily operate on fixed-dimensional vectors or regular grid structures, GNNs can learn representations from graph-structured data. Each robot can therefore be characterized not only by its own state but also by information received from neighboring robots.

Theoretical basis of gnn-based topology control. The fundamental principle of GNN-based topology control is the transformation of the robot swarm into a dynamic graph. Each vertex corresponds to an individual robot, while an edge indicates that two robots can exchange information within the defined communication conditions.

For robot (i), the feature vector may include position, velocity, orientation, residual energy, communication quality, and local neighbor information. The edge representation can contain distance, received signal strength, packet-loss rate, or estimated link reliability. Such representation enables the neural model to learn relationships between physical configuration and communication structure.

A general message-passing mechanism can be expressed as:

[h_i^{(l+1)}=\sigma\left(W_1h_i^{(l)}+\sum_{j\in N(i)}W_2h_j^{(l)}\right),] where (h_i^{(l)}) represents the latent state of robot (i) at layer (l), (N(i)) denotes its neighboring robots, (W_1) and (W_2) are trainable parameter matrices, and (\sigma) represents a nonlinear activation function.[4,87 p]

The equation demonstrates that the updated representation of an individual robot depends on both its own state and information aggregated from neighboring agents. Therefore, the learned representation contains local relational information that can be used for topology-aware decision-making.

The central advantage of this mechanism is that the model does not treat robots as independent entities. Instead, it evaluates their structural relationships within the swarm. This characteristic is particularly important when the objective is to maintain global connectivity through decentralized local actions.

Dynamic communication topolog. During collective movement, the topology of a robot swarm may undergo continuous changes. If two robots move beyond the effective communication range, their corresponding edge can disappear from the graph. Conversely, a new communication relationship can emerge when previously separated robots approach one another.

The dynamic graph can consequently be represented as:

[G_t=(V,E_t),] where (t) denotes the current time and (E_t) represents the communication relationships existing at that moment.

Topology control therefore becomes a sequential decision problem. At each time step, the system observes the current graph, evaluates its structural properties, predicts the consequences of possible movements, and selects actions that support both navigation and communication stability.[5,66 p]

A GNN can assist this process by learning the relationship between graph configurations and suitable motion decisions. The resulting controller may assign each robot a desired direction, velocity, or local trajectory while considering the connectivity requirements of the entire swarm.

Connectivity preservation. Network connectivity is one of the fundamental requirements of a self-organizing robot swarm. A connected communication graph ensures that information can propagate between different parts of the swarm through direct or multi-hop communication paths.

Let (L=D-A) denote the graph Laplacian, where (D) is the degree matrix and (A) is the adjacency matrix. The second-smallest eigenvalue of (L), commonly denoted by (\lambda_2), provides an important measure of algebraic connectivity. A positive value of (\lambda_2) indicates that the graph is connected.

Accordingly, topology-aware navigation can incorporate connectivity into the decision process. A controller may penalize actions that substantially decrease (\lambda_2) while allowing movements that preserve sufficient structural redundancy.

This principle changes the objective of robot navigation. Instead of minimizing only travel distance or maximizing individual motion efficiency, the controller considers a multi-objective problem involving spatial progress, communication reliability, collision avoidance, and network connectivity.

Gnn-based motion decision mechanism. The proposed approach can be organized into four computational stages: graph construction, feature aggregation, topology assessment, and motion generation.

During graph construction, each robot identifies neighboring agents according to communication range or measured link quality. The resulting adjacency structure is transmitted to the GNN-based controller.

During feature aggregation, the neural network combines information from neighboring robots. This operation allows an agent to estimate whether its current movement may contribute to network fragmentation or improve local connectivity.

During topology assessment, the learned representation is used to estimate the structural condition of the surrounding communication graph. The model can identify weak links, isolated regions, highly dependent relay robots, and redundant communication paths.

During motion generation, the controller produces a movement command that balances the navigation objective with the topology-preservation requirement. For a wheeled robot, the output may correspond to linear velocity (v_i) and angular velocity (\omega_i).[7,39 p]

The resulting control command can be formulated conceptually as:

[u_i=\pi_{\theta}(h_i,G_t),] where (u_i) represents the motion command, (\pi_{\theta}) is the learned policy, (h_i) is the robot's latent representation, and (G_t) is the current communication graph.

Self-organization and decentralized coordination. Self-organization implies that global swarm behavior emerges from interactions among individual agents without requiring continuous centralized control. This property reduces dependence on a single coordination node and can improve scalability in large robotic systems.

A decentralized GNN architecture is particularly appropriate for this purpose because message passing can be performed using local neighborhood information. Each robot can obtain information from nearby agents and independently calculate its control action.

Such an architecture also provides a degree of structural flexibility. If the number of robots changes, the graph representation can accommodate a different number of nodes without requiring the same fixed input dimension used by conventional fully connected neural networks.

The resulting system can therefore support heterogeneous spatial configurations while preserving a common computational principle.

Collision avoidance and topological stability. Maintaining communication connectivity alone is insufficient for safe swarm operation. Robots must also prevent physical collisions while preserving their collective objective. Therefore, topology control should operate jointly with motion-safety constraints.

For two robots (i) and (j), a minimum separation condition can be represented as:

[d_{ij}\geq d_{\min},] where (d_{ij}) is the Euclidean distance between the robots and (d_{\min}) is the required safety distance.

The controller must consequently identify feasible actions that satisfy both connectivity and collision-avoidance requirements. This creates a constrained coordination problem in which movement decisions influence multiple system-level properties simultaneously.[6,72 p]

A GNN can facilitate this process by incorporating neighboring positions and velocities into the learned representation. The resulting policy can distinguish between a communication link that should be preserved and a spatial configuration that may create a collision risk.

Training strategy. The effectiveness of the proposed framework depends strongly on the quality and diversity of training data. Training scenarios should represent different robot densities, communication ranges, obstacle configurations, initial formations, and navigation objectives.

Simulation-based training is particularly useful because it allows researchers to generate large numbers of graph configurations without exposing physical robots to unnecessary operational risks. Each training episode can contain a sequence of robot states, communication graphs, control actions, and corresponding performance measurements.

The optimization objective may combine several components:

[J=\alpha J_{\mathrm{nav}}+\beta J_{\mathrm{conn}}+\gamma J_{\mathrm{safe}}+\delta J_{\mathrm{energy}},] where (J_{\mathrm{nav}}) represents navigation error, (J_{\mathrm{conn}}) denotes connectivity degradation, (J_{\mathrm{safe}}) describes collision-related penalties, and (J_{\mathrm{energy}}) represents energy consumption. The coefficients (\alpha,\beta,\gamma,\delta) determine the relative importance of each objective.

Such a formulation allows the learning process to move beyond simple trajectory imitation. The neural controller is instead optimized according to measurable system-level criteria.[8,56 p]

Performance evaluation. Evaluation of the proposed topology-control framework should consider both networking and robotics indicators. Communication connectivity can be assessed through the number of disconnected components, link survival rate, packet-delivery performance, and algebraic connectivity.

Motion performance can be evaluated using trajectory error, navigation time, path length, average velocity, and collision frequency. Computational efficiency should also be examined because decentralized robotic systems often operate under limited processing and communication resources.

A comparative experimental design should include at least three categories of controllers: conventional rule-based topology control, classical optimization-based coordination, and the proposed GNN-based mechanism. Such comparison would allow the contribution of relational learning to be distinguished from improvements caused by general controller complexity.

The evaluation should additionally test scalability by gradually increasing the number of robots. A successful topology-control method should maintain stable performance as swarm size increases rather than relying on a fixed number of agents.

Discussion. The theoretical significance of GNN-based topology control lies in its ability to connect network structure with physical robot behavior within a unified learning framework. In conventional approaches, communication management and navigation are frequently treated as separate computational tasks. The proposed methodology instead considers them as mutually dependent components of a single multi-agent control problem.

When a robot changes its position, it can modify the communication graph. The modified graph subsequently changes the information available to neighboring robots, which can influence their future movements. This feedback mechanism produces a coupled cyber-physical process.

GNN-based learning is suitable for this environment because graph representations preserve the relational structure of the swarm. The model can learn which neighboring interactions are significant for maintaining connectivity and which movements may create structural vulnerabilities.

Nevertheless, several limitations must be considered. Neural controllers may demonstrate sensitivity to training distributions, sensor uncertainty, communication delays, and previously unseen environmental configurations. Consequently, simulation performance cannot by itself establish practical reliability.

For deployment on physical wheeled robots, the learned controller should therefore be evaluated under communication noise, packet loss, localization errors, dynamic obstacles, and changes in robot density. Robustness testing is essential for determining whether the learned topology-control policy can transfer from simulated conditions to real-world environments.

Conclusion. A Graph Neural Network provides a theoretically appropriate mechanism for topology control in self-organizing wheeled robot swarms because the communication relationships among robots naturally form a dynamic graph. By integrating robot states, neighboring interactions, communication conditions, and graph-level information, a GNN-based controller can generate motion decisions that simultaneously support navigation and preservation of network connectivity.

The proposed framework treats topology maintenance as an integral component of collective robotic behavior rather than as an independent communication-management task. Its principal scientific value lies in establishing a relational learning mechanism through which local robot actions can be informed by the structural condition of the surrounding swarm.[9,31 p]

Future research should concentrate on robust GNN architectures, decentralized training, uncertainty-aware topology prediction, sim-to-real transfer, energy-efficient communication, and experimental validation using physical wheeled robot platforms. These directions can contribute to the development of scalable self-organizing robotic networks capable of maintaining reliable communication while performing autonomous collective missions.

REFERENCES

1. Corso, G., Stark, H., Jegelka, S., Jaakkola, T., & Barzilay, R. (2024). Graph neural networks. Nature Reviews Methods Primers, 4, 17.
2. Tolstaya, E., Gama, F., Paulos, J., Pappas, G., Kumar, V., & Ribeiro, A. (2020). Learning decentralized controllers for robot swarms with graph neural networks. Proceedings of the Conference on Robot Learning, 100, 671–682.
3. Jiang, C. J., Huang, X., & Guo, Y. (2023). End-to-end decentralized formation control using a graph neural network-based learning method. Frontiers in Robotics and AI.
4. Ong, P., Capelli, B., Sabattini, L., & Cortés, J. (2023). Nonsmooth control barrier function design of continuous constraints for network connectivity maintenance. Automatica, 156, 111209.
5. Simonetto, A., Keviczky, T., & Babuška, R. (2013). Constrained distributed algebraic connectivity maximization in robotic networks. Automatica.
6. Ghedini, C., Ribeiro, C. H. C., & Sabattini, L. (2018). Toward efficient adaptive ad-hoc multi-robot network topologies. Ad Hoc Networks, 74, 57–70.
7. Eshaghi, K., Najafizadeh Sari, N., Haigh, C., Roman, D., Nejat, G., et al. (2024). Restoring connectivity in robotic swarms – A probabilistic approach. Journal of Intelligent & Robotic Systems, 110, 90.
8. Shi, G., Ishat-E-Rabban, M., Bonner, G., et al. (2025). Fast k-connectivity restoration in multi-robot systems for robust communication maintenance: algorithmic and learning-based solutions. Autonomous Robots, 49, 34.
9. Zhang, S., & Du, X. (2026). GNN enhanced reinforcement learning for robot navigation in complex topological networks. Scientific Reports, 16, 22888.
10. Chen, M., Zeng, M., Ma, W., He, X., Al-Dulaimi, A., & Mumtaz, S. (2026). Task-driven semantic collaborative communication helps multi-robot systems with embodied intelligence. IEEE Communications Magazine, 64(4), 42–49.

11. Gama, F., Tolstaya, E., & Ribeiro, A. (2020). Graph neural networks for decentralized controllers. arXiv preprint.
12. Agarwal, S., Ribeiro, A., & Kumar, V. (2023). Asynchronous perception-action-communication with graph neural networks. arXiv preprint.