LEARNING ADAPTIVE COMMUNICATION RANGES FOR DECENTRALIZED MULTI-AGENT PATHFINDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Communication is essential to decentralized multi-agent pathfinding, where agents must coordinate their movements from limited views of a shared environment. Effective coordination depends on receiving information that is relevant to each agent's situation. These needs vary across locations and planning stages: interacting routes may require broader context, while independent movement can rely on local information. A fixed communication range applies the same scope to both situations, potentially missing useful interactions or introducing redundant exchanges. We propose state-adaptive communication, which adjusts how far each agent communicates as planning unfolds. Agents can use different communication neighborhoods within the same scene and change their scope as surrounding interactions evolve. Experiments across three communication architectures and diverse map and population settings show improved planning success with reduced planner-message payload and overall runtime. The average success-rate improvement reaches 13.5 percentage points on HMAGAT. Behavior analyses further show that communication expands or contracts differently across environments and individual agents. These findings establish adaptive communication scope as a practical way to match information exchange to local coordination needs and improve the efficiency of decentralized planning.

1 INTRODUCTION

Multi-agent pathfinding (MAPF) asks a group of agents to reach their goals without collisions (Stern et al., 2019). It is a basic coordination problem in settings such as warehouse automation and robot fleets, where each agent's movement can affect the routes available to others. Learned decentralized planners address this problem by predicting actions from local observations (Sartoretti et al., 2019; Damani et al., 2021). Communication extends their view beyond what they can directly observe, helping agents anticipate interactions and coordinate their decisions (Li et al., 2020). This becomes especially useful when an agent's immediate surroundings reveal only part of a conflict. Information from other agents can indicate whether an available route is likely to remain clear or become occupied.

Prior work has developed increasingly expressive ways to process information from other agents. Graph attention weighs incoming messages, hypergraph models capture group interactions, and local multi-round communication propagates context through nearby agents (Li et al., 2021c; Jain et al., 2026; Vyaltsev et al., 2026). Adaptive communication methods also learn when to communicate or which agents to address (Singh et al., 2019; Das et al., 2019; Ma et al., 2022). These methods improve how agents use information from their peers. The communication neighborhood determines which agents can supply that information, making its spatial scope another important part of the planning decision.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In the planners considered here, that scope is controlled by a fixed communication range. A single range must serve agents in different locations and at different stages of their routes. This is restrictive because coordination needs are uneven: agents approaching intersecting routes may benefit from broader context, while agents moving independently may need little information beyond their immediate surroundings. A narrow range can omit relevant agents; a broad range can add redundant messages and unnecessary computation. The appropriate scope therefore depends on the agent's current state. Even on the same map, an agent approaching a busy intersection can face a different coordination problem from one moving along an empty route. A setting chosen for the whole episode must accommodate both.

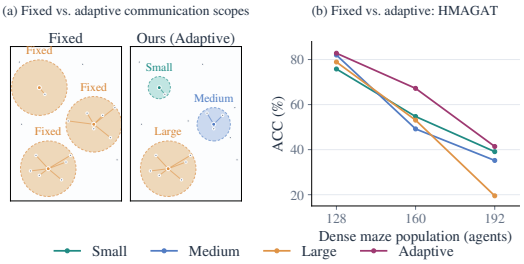


Figure 1: Motivation for adaptive communication. (a) Agent-specific neighborhoods. (b) Fixed versus adaptive ranges on dense mazes.

We propose to learn how far each agent should communicate at each planning step. A state-conditioned selector chooses among small, medium, and large communication neighborhoods. Its predictions jointly define the topology used by an existing planner, allowing agents in the same scene to receive information from different spatial scopes. As their local situations change, the agents can expand or contract their neighborhoods. Figure 1 illustrates this agent-specific adaptation and its benefits on dense mazes.

Our framework learns these decisions from offline evaluations of a frozen planner. By comparing how different communication ranges support expert actions at the same state, we obtain supervision for a lightweight selector. At deployment, the selector adapts neighborhood construction while the planner retains its existing message processing and action policy. This interface supports graph-based, hypergraph-based, and local multi-round communication within a common framework.

Experiments with MAGAT, HMAGAT, and LC-MAPF show that state-adaptive communication improves planning success while reducing communication payload and overall runtime. Comparisons with alternative selection strategies and analyses of deployment behavior further connect these gains to decisions that vary across agents and time.

Our contributions are:

- We formulate communication scope as an agent-specific, state-dependent decision, enabling decentralized MAPF planners to adapt their communication neighborhoods throughout planning.
- We develop an offline-supervised range selector that integrates with three distinct communication architectures while keeping their action policies fixed.
- We demonstrate improvements in planning success and communication efficiency across diverse environments, with controlled comparisons and behavior analyses showing the value of adaptation across agents and planning stages.

2 Related Work

Search-based and reactive MAPF. CBS resolves conflicts through constraint-based search, while EECBS trades optimality for bounded-suboptimal search efficiency (Sharon et al., 2015; Li et al., 2021b). Large-neighborhood methods improve solutions or repair collisions by replanning agent subsets (Li et al., 2021a; 2022a). PIBT emphasizes iterative local coordination (Okumura et al., 2022); LaCAM and its subsequent engineering advances pursue scalable configuration-space search (Okumura, 2023; 2024). Our work addresses the complementary problem of organizing communication within learned decentralized planners.

Learned local planning. DHC combines heuristic guidance with graph-based communication (Ma et al., 2021); SCRIMP uses a transformer-based communication mechanism to support

small local observations (Wang et al., 2023). MAPF-GPT studies imitation learning at scale without an explicit communication module (Andreychuk et al., 2025). MAGAT, HMAGAT, and LC-MAPF offer message-aware graph attention, directed hypergraphs, and local multi-round communication, respectively (Li et al., 2021c; Jain et al., 2026; Vyaltsev et al., 2026). Our framework learns the spatial scope over which these architectures exchange information.

The common interface is neighborhood construction: a graph edge, hypergraph membership, or local communication mask determines which information reaches the action network. This interface provides a shared point of adaptation across architectures with different message-processing mechanisms.

Adaptive multi-agent communication. CommNet and DIAL learn communication jointly with cooperative policies (Sukhbaatar et al., 2016; Foerster et al., 2016). Subsequent work introduces communication gates (Singh et al., 2019), targeted messages (Das et al., 2019), and attentional communication decisions (Jiang & Lu, 2018). Within MAPF, DCC selects decision-relevant neighbors (Ma et al., 2022), while PICO incorporates learned priorities (Li et al., 2022b). We focus on spatial thresholds through the neighborhood-construction interface, using offline expert-action scores to adapt frozen planners across communication architectures without retraining their action policies.

Interaction representations and online adaptation. GAT and GATv2 learn attention weights over graph neighborhoods (Veličković et al., 2018; Brody et al., 2022); HGNN and Hyper-SAGNN model higher-order relations (Feng et al., 2019; Zhang et al., 2020). Adaptive communication scope complements these mechanisms by determining the eligible neighborhoods. We also compare state-conditioned prediction with an online SW-UCB strategy that updates range preferences from recent goal progress, drawing on nonstationary bandit methods (Garivier & Moulines, 2011).

3 Problem Formulation

Consider a traversable graph $G = (V, E)$ and n agents with positions $p_i^t \in V$ and goals $g_i \in V$. At each timestep, an agent moves to an adjacent traversable vertex or waits. The objective is to bring all agents to their goals within the evaluation horizon while avoiding vertex and opposing-edge collisions. The adaptive and fixed-range planners share the same collision handling and goal-arrival conventions.

Let o_i^t denote agent i's local observation and z_i^t its auxiliary features. The agent chooses a communication range r_i^t from a finite candidate set $\mathcal{R}$, defining its eligible neighborhood as

$$\mathcal{N}_i^t(r_i^t) = \{j \neq i : d(p_i^t, p_j^t) \leq r_i^t\}, \qquad r_i^t \in \mathcal{R}, \tag{1}$$

where d is Manhattan distance on the grid. A topology builder $\mathcal{B}$ combines these neighborhoods with the backbone's grouping and eligibility rules to construct a graph, hypergraph, or communication mask. The policy then exchanges information over this structure and produces actions. A fixed-range planner is the special case $r_i^t = r$ for all agents and timesteps.

Because agent positions and selected ranges both evolve, the resulting neighborhoods change throughout an episode. The range decision controls the spatial extent of incoming communication, while the backbone determines how eligible messages are combined. This separation allows communication adaptation to operate at the same decision frequency as the action policy.

Each agent predicts its range from its local observation and auxiliary state information. The information available to the selector and the communication-accounting protocol are specified in Appendix B.

4 Learning Adaptive Communication Ranges

4.1 Counterfactual Range Evaluation

We learn communication scope by comparing the decisions a planner makes with different communication neighborhoods. We collect expert trajectories and evaluate alternative communication

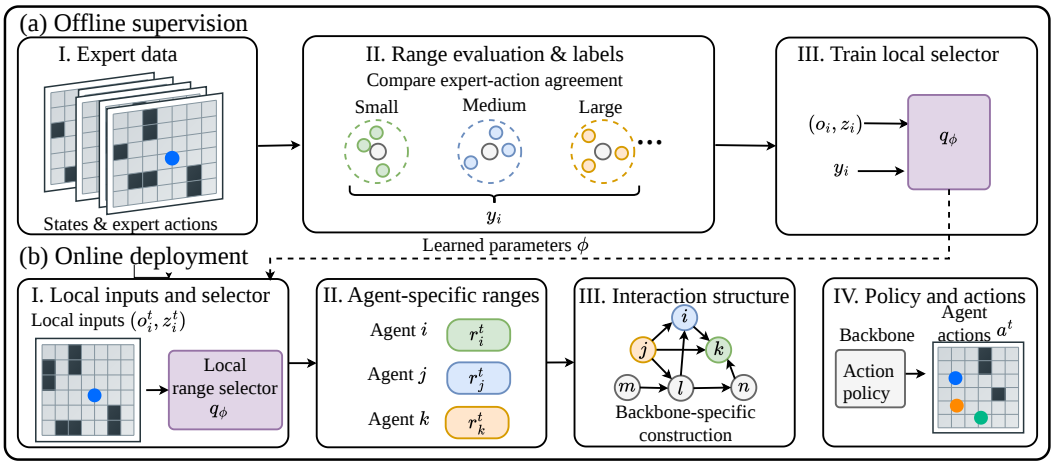


Figure 2: Offline range evaluations supervise a local selector that adapts each agent's communication neighborhood for the frozen planner.

ranges at the same recorded states, keeping the backbone checkpoint fixed. For an agent state s_i with expert action $a_i^\star$, the score of a trial range R is

$$C_i(R) = -\log \pi_R(a_i^\star \mid s_i), \tag{2}$$

where π_R is the policy evaluated after assigning the trial range to all agents and rebuilding the communication topology. A lower score indicates stronger agreement with the expert action. Comparing these scores holds the planning state and action policy fixed while varying the communication context. Each recorded state therefore supplies a direct comparison between candidate neighborhoods. A range that helps at one state may be less useful at another, allowing the supervision to capture differences in communication needs across agents and situations.

We use validation preferences to retain a compact set of representative communication scales, denoted small, medium, and large. Let $\mathcal{R} = \{r_{\text{small}}, r_{\text{medium}}, r_{\text{large}}\}$, ordered by increasing spatial extent. These candidates provide different scopes of incoming information, and the selector learns which scope to use in each state. Candidate selection and offline preference statistics are described in Appendix B.

4.2 Learning from Local State

For each recorded sample, the best-scoring candidate supplies the training target:

$$y_i = \min \arg\min_{R \in \mathcal{R}} C_i(R). \tag{3}$$

The selector predicts this target from a local spatial observation o_i and auxiliary state information z_i. It combines their encoded representations to produce a distribution over communication scales:

$$q_\phi(R \mid o_i, z_i) = \text{softmax}_R \left(f_{\text{head}}\big([f_{\text{CNN}}(o_i), f_{\text{MLP}}(z_i)]\big) \right). \tag{4}$$

The two branches describe the agent's planning context from complementary views. Spatial observations capture the arrangement of nearby agents and obstacles, while auxiliary information describes the agent's broader state. Their combination allows a shared selector to distinguish situations that look similar locally but differ in their coordination needs. The resulting prediction concerns the extent of communication; the backbone continues to determine the movement action.

We train the selector with categorical cross-entropy,

$$\mathcal{L}(\phi) = \mathbb{E}_{i \sim \mathcal{D}_{\text{train}}}[-\log q_\phi(y_i \mid o_i, z_i)], \tag{5}$$

using balanced label sampling and validation-based early stopping. The planner remains frozen, so learning focuses on adapting the context supplied to its existing action policy. Architecture, feature construction, and label-processing details are provided in Appendix B; sampling comparisons appear in Appendix E.1.

4.3 State-Adaptive Communication at Deployment

At each planning step, agent i selects its communication range from its current inputs:

$$r_i^t = \arg\max_{R \in \mathcal{R}} q_\phi(R \mid o_i^t, z_i^t). \tag{6}$$

The selections define the eligible incoming links,

$$A_{ij}^t = \mathbf{1}[j \neq i]\, \mathbf{1}[d(p_i^t, p_j^t) \leq r_i^t], \tag{7}$$

which the backbone's topology builder combines with its existing grouping and message-processing rules. Each row uses the receiving agent's chosen scope, allowing agents at the same timestep to use different neighborhoods. Recomputing these choices throughout planning also lets each agent adjust its scope as local interactions evolve. The eligible links can be asymmetric: an agent choosing a large neighborhood may receive information from another agent whose own neighborhood is small. This gives each agent control over its incoming context without imposing the same communication scope on its neighbors.

For HMAGAT, the range vector $\mathbf{r}^t = (r_1^t, \ldots, r_n^t)$ is incorporated into its hypergraph constructor:

$$\mathcal{H}^t = \mathcal{B}_{\mathrm{HM}}(s^t, \mathbf{r}^t), \qquad \mathbf{a}^t \sim \pi_\theta(\cdot \mid s^t, \mathcal{H}^t). \tag{8}$$

The resulting shared hypergraph supports heterogeneous communication neighborhoods while preserving HMAGAT's group aggregation and action prediction. The same principle supplies graph neighborhoods for MAGAT and local communication masks for LC-MAPF, with a selector trained for each backbone. Each planning step uses one selected topology and one invocation of the action policy.

The selector replaces repeated candidate evaluation with a single prediction at deployment. Range selection and action planning then proceed together at every timestep, so communication can change as agents move through the environment.

Computational overhead. In the optimized implementation, feature extraction and selector inference add slightly over one second per run. Algorithm timing includes both components, topology construction, and policy inference; Appendix F defines the timing boundary.

5 Experiments

We evaluate adaptive communication on MAGAT, HMAGAT, and LC-MAPF. The experiments examine planning quality and efficiency across architectures, the contribution of state-dependent selection, and the evolution of communication during deployment. Detailed ablations use HMAGAT to study the framework's components.

5.1 Experimental Setup

Environments and comparison protocol. Evaluation covers six map families and 21 map and population configurations: dense and sparse mazes, dense and sparse warehouses, an open room, and `ost003d`, with 96 to 256 agents. Dense mazes and warehouses use a 512-step horizon, sparse mazes and warehouses and the open room use 256 steps, and `ost003d` uses 1,024 steps; the wall-clock limit is 90 s for dense mazes, dense warehouses, and `ost003d`, and 60 s otherwise. Each configuration evaluates one map layout over 128 episodes. Main-text results average three selector-training seeds; training and test seeds are disjoint. We use a single NVIDIA A100 GPU. Consistent benchmarking is central to MAPF evaluation (Skrynnik et al., 2025); each adaptive planner is therefore paired with its fixed-range counterpart on identical instances, sharing the backbone checkpoint, collision handling, and other planner settings. Each fixed baseline uses its backbone's default communication range. The adaptive selector chooses small, medium, or large neighborhoods, corresponding to Manhattan radii of 3, 5, and 7 grid cells, respectively. Appendix A lists the environment settings.

The map families span distinct spatial organizations, from mazes and warehouse layouts to an open room and a pregenerated benchmark map. Varying the population within a family examines how each planner responds to changes in the number of agents sharing the environment. Paired evaluation then compares fixed and adaptive communication at the same map and population, keeping the underlying planning task matched.

Planning and efficiency metrics. We report success rate (ACC), sum of costs (SoC), communication volume, and algorithm wall-clock time. An instance succeeds when all agents reach their goals within both the step and wall-clock limits. For HMAGAT, SoC is averaged over all evaluation episodes, with a failed episode charging every agent the full step limit. To compare relative changes across different cost scales, Figure 3 reports the adaptive-to-fixed SoC ratio within each backbone; values below one indicate lower cost with adaptive communication. Communication counts logical directed message deliveries across all rounds and timesteps; payload accounting is specified in Appendix C. Algorithm time includes feature extraction and selector inference.

SoC summarizes goal-reaching efficiency under the stated failure convention, while cumulative communication reflects both message exchange per step and episode duration. All planners are evaluated under matched step and wall-clock budgets.

Aggregation and uncertainty. We first compute metrics within each configuration. ACC is then averaged equally over the 21 configurations, while each positive cost metric is summarized by the mean of the configuration-wise adaptive-to-fixed ratios. Paired bootstrap resampling over evaluation episodes, with 10,000 resamples, gives 95% intervals of at most ± 1.8 percentage points on configuration-averaged ACC. Repeating selector training under three seeds changes this average by at most ± 0.9 percentage points. Appendix D reports the per-configuration results.

Improvement counts summarize per-configuration point estimates. Evaluation uses disjoint episode seeds on the specified layouts, with uncertainty reported for the aggregate results.

5.2 Main Results across Three Backbones

Relative to the default fixed ranges, adaptive communication improves average success while reducing planner-message payload and overall runtime on all three backbones (Figure 3). HMAGAT increases configuration-averaged ACC from 70.2% to 83.7%, a gain of 13.5 percentage points, with communication reduced to $0.50\times$ and runtime to $0.79\times$ of the fixed baseline. MAGAT improves ACC by 9.1 percentage points with communication at $0.53\times$, while LC-MAPF improves ACC by 5.4 percentage points with communication at $0.70\times$. The gains extend across graph-based, hypergraph-based, and multi-round local communication.

HMAGAT improves ACC in 18 configurations and matches the baseline in the remaining three, while reducing communication in all 21. The gains are especially pronounced in challenging configurations: ACC rises from 18.0% to 46.9% in ER-224 and from 14.1% to 85.2% in SW-224. The full results and path-cost comparisons are reported in Appendix D.

The improvement coverage also extends across the other two architectures. MAGAT improves success in 19 configurations and matches its baseline in two; LC-MAPF improves in 20 and matches in one. The improvements occur across most evaluation settings, with each planner retaining its original action policy.

Runtime gains vary with the communication architecture. LC-MAPF uses padded observations and fixed-size neighbor slots with masking (CognitiveAISystems, 2026). Under this interface, fewer active neighbors reduce message exchange while leaving dense attention dimensions unchanged, limiting the computational savings available from a smaller neighborhood. Appendix F describes the runtime accounting.

5.3 Ablation Studies on HMAGAT

We study candidate screening and selector inputs on three configurations with contrasting communication patterns: dense maze with 192 agents (DM-192), sparse warehouse with 224 agents (SW-224), and empty room with 128 agents (ER-128). Table 1 reports each configuration separately.

The full selector achieves the highest reported success rate in all three configurations. Removing auxiliary features reduces ACC by 3.9 to 7.8 percentage points, while removing the spatial observation reduces it by 8.6 to 15.6 points. Spatial context and scalar descriptors thus provide complementary information, with the spatial observation making the larger contribution in these experiments; the auxiliary-only variant is also the least economical at DM-192 and SW-224, where it attains the lowest success rate while communicating more than the full selector. Retaining the full candidate

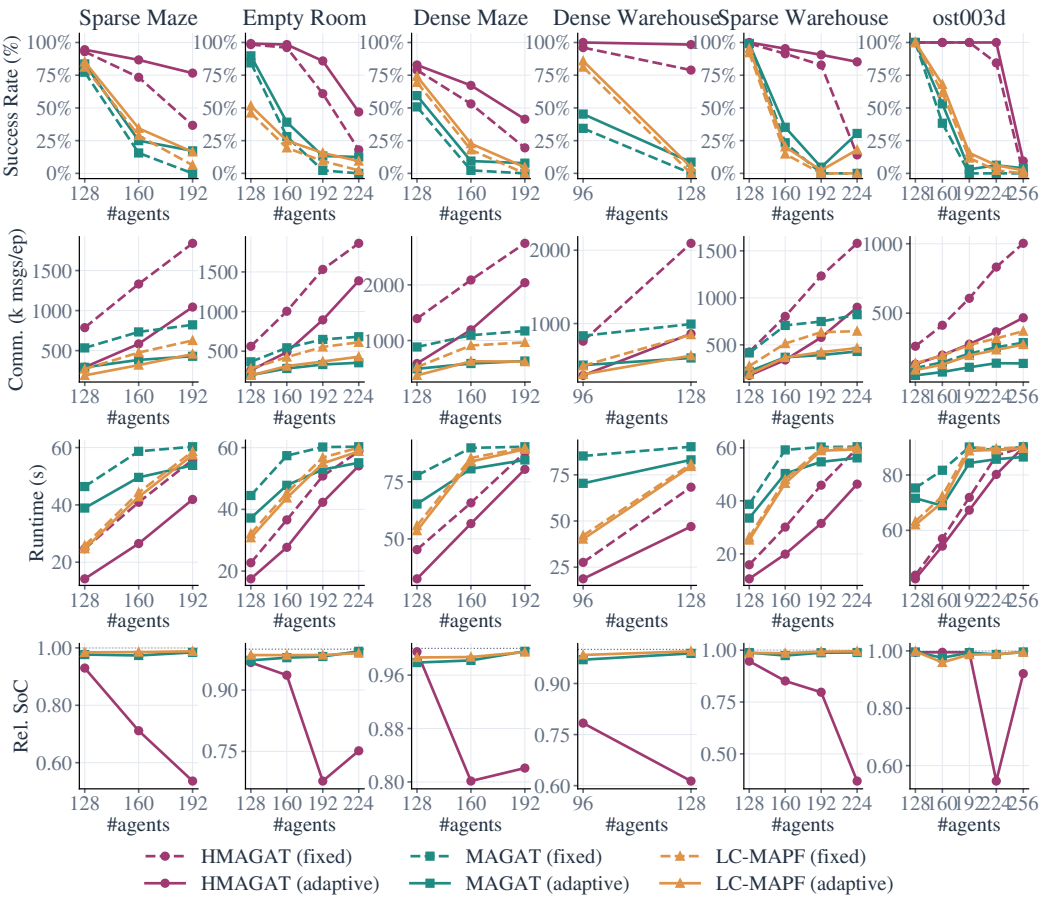


Figure 3: Planning quality and efficiency. Solid: adaptive; dashed: fixed. SoC: within-backbone adaptive-to-fixed ratios, with dotted lines marking parity.

Table 1: HMAGAT ablations. ACC is success rate (%); C is communication normalized to the fixed baseline of Table 7 within each configuration.

	DM-192		SW-224		ER-128	
Variant	ACC ↑	C ↓	ACC ↑	C ↓	ACC ↑	C ↓
Full selector	41.4	0.74	85.2	0.57	99.2	0.46
Full candidate pool	39.1	0.59	84.4	0.41	97.7	0.28
Observation only	33.6	0.70	78.1	0.53	95.3	0.34
Auxiliary features only	25.8	0.76	71.9	0.61	90.6	0.40

pool reduces communication at the cost of success rate (0.59 against 0.74 at DM-192, 0.41 against 0.57 at SW-224, and 0.28 against 0.46 at ER-128, with 2.3, 0.8, and 1.5 points of ACC lost), showing a trade-off between lower communication and higher success rate. Appendix E.1 compares the sampling strategies and reports the same metrics for each variant.

All input ablations retrain the selector under matched settings. The observation-only variant retains local spatial information, and the auxiliary-only variant retains the scalar state descriptors. Their comparison with the full selector isolates the contribution of combining these sources. Candidate screening makes a different intervention: it changes the available range labels while retaining the full input. The two sets of ablations therefore address both what the selector observes and which communication scales it predicts.

Table 2: Communication adaptation strategies on HMAGAT. ACC is success rate (%); C is communication relative to the fixed reference in each configuration.

	DM-160		ER-224		SW-192	
Strategy	ACC ↑	C ↓	ACC ↑	C ↓	ACC ↑	C ↓
Fixed (default)	53.1	1 (ref.)	18.0	1 (ref.)	82.8	1 (ref.)
Frequency-matched random	51.6	0.86	21.8	0.73	73.4	0.83
Density-based	57.8	0.77	40.6	0.65	78.9	0.77
SW-UCB	61.7	0.69	43.0	**0.59**	83.6	0.77
Ours	**67.2**	**0.57**	**46.9**	0.75	**90.6**	**0.47**

5.4 Comparing Communication Adaptation Strategies

We compare five strategies on the same HMAGAT backbone and candidate set using DM-160, ER-224, and SW-192 (Table 2). Fixed uses the backbone's default range. Frequency-matched random samples ranges independently from a validation-estimated distribution, retaining marginal range preferences while removing state dependence. Density-based maps local agent density to a range through validation-selected bins and assignments. SW-UCB updates each agent's range preferences from recent goal progress. Ours predicts a range from the current local observation and auxiliary features. All adaptive strategy settings are selected on validation data before test evaluation, and runtime includes feature extraction and selection.

These strategies use different information to select communication ranges. Random selection uses a learned marginal frequency but no current-state information. The density rule conditions on one local statistic, with validation choosing its bins and assignments. SW-UCB adapts from recent outcomes, treating candidate ranges as arms and using goal progress as feedback. The proposed selector combines current spatial and scalar features in a learned prediction. Comparing these strategies examines whether the current local configuration provides useful information beyond aggregate preferences, density, or recent progress.

State-conditioned selection achieves the highest ACC in each configuration. Frequency-matched random reduces communication but scores below the fixed reference at DM-160 (1.5 points) and SW-192 (9.4 points), and 3.8 points above it at ER-224, where the fixed reference is weak. Density-based selection improves ACC at DM-160 and ER-224, and SW-UCB further benefits from online feedback. Our selector exceeds SW-UCB by 5.5, 3.9, and 7.0 percentage points on DM-160, ER-224, and SW-192, respectively. It also uses less communication than SW-UCB at DM-160 (0.57 against 0.69) and SW-192 (0.47 against 0.77); at ER-224 the advantage is in success rate only (0.75 against 0.59). These comparisons support learning communication scope from the agent's current state. Appendix E provides the individual fixed-range results.

Comparison with individual fixed ranges. The fixed-range comparisons in Appendix Table 11 show that the strongest uniform choice changes across configurations. Medium performs best on DM-128, small on DM-160, DM-192, and SW-192, and large on ER-160. Adaptive communication achieves the highest success-rate point estimate in each case. On DM-160, it reaches 67.2% compared with 54.7% for the strongest fixed candidate; on SW-192, the corresponding values are 90.6% and 85.9%. These comparisons extend the default-baseline results by showing how adaptation performs against alternative uniform choices. The variation among fixed candidates also supports selecting communication scope at the agent level, where local conditions can differ within a single configuration.

5.5 Communication across Time and Agents

Different layouts lead to different communication trajectories. Narrow passages in mazes and aisle intersections in warehouses channel agents into shared routes. As agents advance through these structures, an initially local interaction can become connected to traffic farther along a corridor or aisle. The observed communication patterns are consistent with this progression. In DM-192, agents initially choose small neighborhoods, while medium neighborhoods become dominant later in the episode. SW-224 shows a similar expansion, but retains a more varied final mixture: approximately

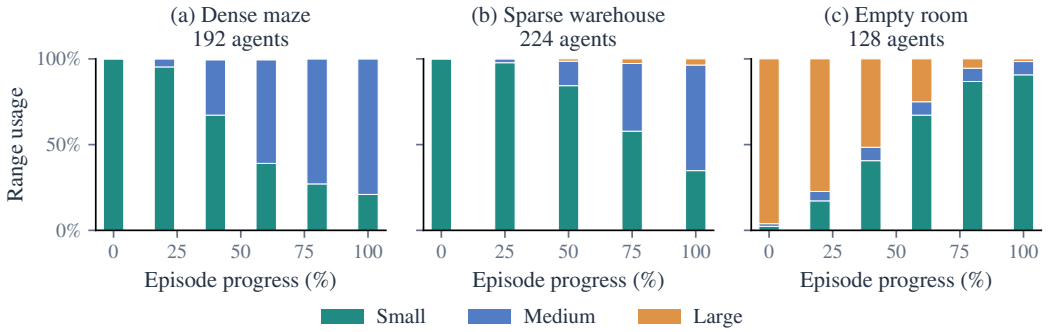


Figure 4: Small, medium, and large range usage over episode progress. Bars normalize multi-trajectory count summaries; Appendix G lists counts and intervals.

78 small, 138 medium, and eight large selections. Expansion is selective, with different agents using different scopes within the same layout.

The open room presents a different coordination setting. Without obstacles funneling movement into corridors, routes can cross throughout the space, making broader context useful early in planning. As agents spread out and complete their routes, these interactions can subside. ER-128 exhibits a corresponding contraction: large neighborhoods dominate the initial interval, whereas small neighborhoods dominate the final interval. The contrast suggests that communication follows the interactions encountered during planning: it expands in corridor-based layouts and contracts in the open room.

Communication follows each agent's local situation. The population-level trends also contain substantial variation between agents. At timestep 22 of a 76-step ER-128 episode, 29 agents choose small neighborhoods, 12 choose medium neighborhoods, and 87 choose large neighborhoods. Two agents at this same instant illustrate why a shared global range would be restrictive. Agent a6 occupies a crowded area, with 25% agent occupancy in its observation window, and selects a large neighborhood. Expanding its scope from small to large makes 28 additional neighbors eligible, providing substantially more context about surrounding traffic.

Agent a30 is in a much quieter part of the same room, with only 2.5% observation-window agent occupancy, and selects a small neighborhood. The same expansion would add only two neighbors. Thus, even while broad communication dominates the population, an individual agent can retain a compact scope where its local situation differs. The same selector therefore accommodates both changes during an episode and differences between agents at a single timestep. Appendix G provides the corresponding spatial, goal-related, and prediction data.

6 Conclusion

We present a framework that learns to adapt communication ranges to the local states of individual agents in decentralized MAPF. Offline range evaluations supervise communication decisions that collectively shape a dynamic topology during planning. Integrations with MAGAT, HMAGAT, and LC-MAPF show improvements in planning success alongside lower communication and overall runtime across diverse environments. Strategy comparisons demonstrate the value of state-conditioned adaptation, while deployment analysis reveals differentiated communication across agents and planning stages. The observed gains hold across the evaluated map families and communication architectures.

Communication scope can be adjusted throughout planning while preserving the message-processing mechanisms of an existing planner. The strategy comparisons show the benefit of using current state to make these adjustments. Future work could extend the approach to richer communication choices and refine the selector through online feedback.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and editing, searches for literature and publicly available code, figure and table design, interpretation of experimental results, and software development, including the implementation of training, evaluation, and plotting code. The research ideas were developed by the authors without the use of generative AI. All AI-assisted code was reviewed and tested by the authors. The authors take responsibility for the final manuscript and associated research artifacts.

ETHICS STATEMENT

This work studies multi-agent navigation in simulation. Physical deployment requires validation under sensing uncertainty, robot dynamics, and communication failures.

REPRODUCIBILITY STATEMENT

The supplement specifies map configurations, candidate evaluation, label construction, implementation details, and metric conventions. The experiment tables and deployment summaries record the numerical comparisons discussed in the main text. Appendix B provides the selector architecture and training settings.

REFERENCES

Anton Andreychuk, Konstantin Yakovlev, Aleksandr Panov, and Alexey Skrynnik. MAPF-GPT: Imitation learning for multi-agent pathfinding at scale. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pp. 23126–23134, 2025. doi: 10.1609/aaai.v39i22.34477.

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

CognitiveAISystems. LC-MAPF: Official implementation. https://github.com/CognitiveAISystems/LC-MAPF, 2026.

Mehul Damani, Zhiyao Luo, Emerson Wenzel, and Guillaume Sartoretti. PRIMAL2: Pathfinding via reinforcement and imitation multi-agent learning—lifelong. IEEE Robotics and Automation Letters, 6(2):2666–2673, 2021. doi: 10.1109/LRA.2021.3062803.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Michael Rabbat, and Joelle Pineau. TarMAC: Targeted multi-agent communication. In Proceedings of the International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pp. 1538–1546, 2019. URL https://proceedings.mlr.press/v97/das19a.html.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pp. 3558–3565, 2019. doi: 10.1609/aaai.v33i01.33013558.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In Advances in Neural Information Processing Systems, volume 29, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/c7635bfd99248a2cdef8249ef7bfbef4-Abstract.html.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In Algorithmic Learning Theory, volume 6925 of Lecture Notes in Computer Science, pp. 174–188. Springer, 2011. doi: 10.1007/978-3-642-24412-4_16.

Rishabh Jain, Keisuke Okumura, Michael Amir, Pietro Liò, and Amanda Prorok. Pairwise is not enough: Hypergraph neural networks for multi-agent pathfinding. In International Conference on Learning Representations, 2026. URL https://openreview.net/forum?id=KF7sO7S4kG.

Jiechuan Jiang and Zongqing Lu. Learning attentional communication for multi-agent cooperation. In Advances in Neural Information Processing Systems, volume 31, 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/hash/6a8018b3a00b69c008601b8becae392b-Abstract.html.

Jiaoyang Li, Zhe Chen, Daniel Harabor, Peter J. Stuckey, and Sven Koenig. Anytime multi-agent path finding via large neighborhood search. In Proceedings of the International Joint Conference on Artificial Intelligence, pp. 4127–4135, 2021a. doi: 10.24963/ijcai.2021/568.

Jiaoyang Li, Wheeler Ruml, and Sven Koenig. EECBS: A bounded-suboptimal search for multi-agent path finding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pp. 12353–12362, 2021b. doi: 10.1609/aaai.v35i14.17466.

Jiaoyang Li, Zhe Chen, Daniel Harabor, Peter J. Stuckey, and Sven Koenig. MAPF-LNS2: Fast repairing for multi-agent path finding via large neighborhood search. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pp. 10256–10265, 2022a. doi: 10.1609/aaai.v36i9.21266.

Qingbiao Li, Fernando Gama, Alejandro Ribeiro, and Amanda Prorok. Graph neural networks for decentralized multi-robot path planning. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 11785–11792, 2020. doi: 10.1109/IROS45743.2020.9341668.

Qingbiao Li, Weizhe Lin, Zhe Liu, and Amanda Prorok. Message-aware graph attention networks for large-scale multi-robot path planning. IEEE Robotics and Automation Letters, 6(3):5533–5540, 2021c. doi: 10.1109/LRA.2021.3077863.

Wenhao Li, Hongjun Chen, Bo Jin, Wenzhe Tan, Hongyuan Zha, and Xiangfeng Wang. Multi-agent path finding with prioritized communication learning. In IEEE International Conference on Robotics and Automation, pp. 10695–10701, 2022b. doi: 10.1109/ICRA46639.2022.9811643.

Ziyuan Ma, Yudong Luo, and Hang Ma. Distributed heuristic multi-agent path finding with communication. In IEEE International Conference on Robotics and Automation, pp. 8699–8705, 2021. doi: 10.1109/ICRA48506.2021.9560748.

Ziyuan Ma, Yudong Luo, and Jia Pan. Learning selective communication for multi-agent path finding. IEEE Robotics and Automation Letters, 7(2):1455–1462, 2022. doi: 10.1109/LRA.2021.3139145.

Keisuke Okumura. LaCAM: Search-based algorithm for quick multi-agent pathfinding. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pp. 11655–11662, 2023. doi: 10.1609/aaai.v37i10.26377.

Keisuke Okumura. Engineering LaCAM*: Towards real-time, large-scale, and near-optimal multi-agent pathfinding. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, pp. 1501–1509, 2024. doi: 10.65109/yqid4918.

Keisuke Okumura, Manao Machida, Xavier Défago, and Yasumasa Tamura. Priority inheritance with backtracking for iterative multi-agent path finding. Artificial Intelligence, 310:103752, 2022. doi: 10.1016/j.artint.2022.103752.

Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset. PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning. IEEE Robotics and Automation Letters, 4(3):2378–2385, 2019. doi: 10.1109/LRA.2019.2903261.

Guni Sharon, Roni Stern, Ariel Felner, and Nathan R. Sturtevant. Conflict-based search for optimal multi-agent pathfinding. Artificial Intelligence, 219:40–66, 2015. doi: 10.1016/j.artint.2014.11.006.

Amanpreet Singh, Tushar Jain, and Sainbayar Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=rye7knCqK7.

Alexey Skrynnik, Anton Andreychuk, Anatolii Borzilov, Alexander Chernyavskiy, Konstantin Yakovlev, and Aleksandr Panov. POGEMA: A benchmark platform for cooperative multi-agent pathfinding. In International Conference on Learning Representations, 2025. URL https://openreview.net/forum?id=6VgwE2tCRm.

Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Eli Boyarski, and Roman Barták. Multi-agent pathfinding: Definitions, variants, and benchmarks. In Proceedings of the International Symposium on Combinatorial Search, volume 10, pp. 151–158, 2019. doi: 10.1609/socs.v10i1.18510.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In Advances in Neural Information Processing Systems, volume 29, 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/hash/55b1927fdafef39c48e5b73b5d61ea60-Abstract.html.

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.

Valeriy Vyaltsev, Alsu Sagirova, Anton Andreychuk, Oleg Bulichev, Yuri Kuratov, Konstantin Yakovlev, Aleksandr Panov, and Alexey Skrynnik. Learning to communicate locally for large-scale multi-agent pathfinding. arXiv preprint arXiv:2605.07637, 2026. URL https://arxiv.org/abs/2605.07637.

Yutong Wang, Bairan Xiang, Shinan Huang, and Guillaume Sartoretti. SCRIMP: Scalable communication for reinforcement- and imitation-learning-based multi-agent pathfinding. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 9301–9308, 2023. doi: 10.1109/IROS55552.2023.10342305.

Ruochi Zhang, Yuesong Zou, and Jian Ma. Hyper-SAGNN: A self-attention based graph neural network for hypergraphs. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=ryeHuJBtPH.

A Environment Configuration

Tables 3 and 4 list the grid sizes, structure parameters, episode horizons, time limits, and agent populations. Grid sizes are reported as the traversable area and the area including the boundary padding. The case-study room is an open 39×39 area with no obstacles; the dense-maze snapshot has a traversable area of 21×21. Warehouse obstacle parameters are generator settings.

Table 3: Map configurations. Grid sizes are the traversable area and the area including the boundary padding. Structure parameters are generator settings. Time limits are the wall-clock budgets applied to the algorithm (Section F).

Family	Structure parameters	Horizon	Time	Agents
Dense maze	21×21 / 31×31; obstacle density 0.30 to 0.40; straight-corridor rate 0.75 to 0.85	512	90 s	128/160/192
Dense warehouse	23×23 / 33×33; wall thickness 8, height 2; horizontal gap 1; vertical gap 1	512	90 s	96/128
Empty room	39×39 / 49×49; open area, no obstacles	256	60 s	128/160/192/224
`ost003d`	194×194 / 204×204; pregenerated real map; minimum start-to-goal distance 10	1024	90 s	128/160/192/224/256
Sparse maze	21×21 / 31×31; obstacle density 0.08 to 0.24; straight-corridor rate 0.75 to 0.85	256	60 s	128/160/192
Sparse warehouse	23×23 / 33×33; wall thickness 8, height 2; horizontal gap 1; vertical gap 2	256	60 s	128/160/192/224

Table 4: All 21 map and population configurations. Fixed-range baseline evaluation covers the entire list.

Family	Agent populations	Number of configurations
Dense maze	128, 160, 192	3
Dense warehouse	96, 128	2
Empty room	128, 160, 192, 224	4
`ost003d`	128, 160, 192, 224, 256	5
Sparse maze	128, 160, 192	3
Sparse warehouse	128, 160, 192, 224	4
Total		21

B Implementation and Training Details

The selector combines a CNN encoding of a $2 \times 11 \times 11$ occupancy observation with an MLP encoding of auxiliary state features. The observation window is fixed while the predicted communication range determines the eligible neighborhood. Feature construction follows each backbone's information model. Table 5 summarizes the architecture and training settings.

Table 5: Implementation specification.

Item	Specification
Trial and candidate ranges	A broader trial pool is screened to obtain small, medium, and large ranges for each backbone. Their spatial thresholds are specified in Section 5.1.
Expert and collection	Expert trajectories indexed by collection range, map, and population; produced by a centralized search-based solver with a fixed expansion budget, filtering instances the solver fails or exceeds the horizon.
Score evaluation	Negative log probability of the expert action with all agents set to the trial range; frozen checkpoint; single precision, no clipping, batched evaluation.
Candidate screening	Validation-label frequency $\geq 5\%$ plus minimizing-set coverage; small, medium, and large ranges retained for all backbones.
Tie processing	Exact bit equality at single precision; smallest minimizing range as label; tied samples retained.
Observation input	Channels: agent occupancy and obstacle occupancy; zero padding at map boundaries; no clipping.
Auxiliary inputs	Normalized distance-to-goal, BFS cost-to-go, per-candidate neighbor counts; z-scored with training statistics.
Network and optimization	CNN 32/64/64 (3×3); MLP 2×128; fused head 2×128; Adam, learning rate 10^{-4}; batch 256; dropout 0.1.
Sampling and stopping	Equal-quota over range labels; early stopping on validation loss with patience 5.
Splits and seeds	Map/episode-level partitions; three training seeds; each configuration evaluates a single map layout over 128 episodes.
Mixed graph construction	Manhattan distance; row-wise agent thresholds; backbone grouping, cap, and self-edge rules unchanged.
Baseline protocol	Backbone checkpoints fixed; each backbone's default fixed communication range; identical instances, seeds, and collision handling.
Runtime environment	Single NVIDIA A100; PyTorch 2.x; batched per-step inference; CUDA-synchronized timing for the components in Equation 11.

B.1 Offline Supervision and Input Construction

Data organization and preference distribution. Map- or episode-level partitions are established before extracting states; training and test seeds are disjoint. Candidate screening, label processing, sampling configuration, and early stopping use the training and validation partitions. The offline preference distribution summarizes approximately 10.9 million samples over the selected candidates. Small and medium neighborhoods account for 41.4% of samples, while 32.9% have equal scores across the selected candidates (Figure 5). The distribution includes states encountered at different points along expert trajectories, so each communication scale is evaluated over a variety of local configurations and planning stages.

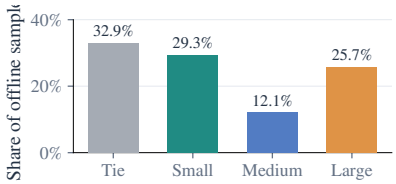


Figure 5: Offline preferences for small, medium, and large ranges, including equal-score ties.

Each sample compares the candidate ranges at a common recorded state. This organization makes the preference distribution a summary of how communication scope affects the planner's support for expert actions. A preference for small indicates that a compact neighborhood attains the best score at that state; preferences for medium or large identify states where a wider neighborhood does so. The equal-score category records states for which all retained candidates provide the same expert-action score. These categories describe the supervision available to the selector before closed-loop deployment.

Label construction and sampling. Scores are stored in single precision; exact equality defines a tie, and the smallest minimizing range supplies the label. All tied samples are retained. Collection ranges index trajectory-generation configurations, while trial ranges index evaluations at each recorded state. Separating these roles allows a state collected under one communication setting to supply comparisons among several candidate settings. Training targets use expert-action agreement. Equal-quota sampling then balances the range labels presented during optimization, giving the selector repeated exposure to each communication scale. Appendix E.1 compares this procedure with natural and map-stratified sampling.

Information available to the selector. The observation and auxiliary branches provide complementary descriptions of the same agent state. Occupancy channels describe the arrangement of nearby agents and obstacles. Goal-related features describe progress toward the destination, and per-candidate neighbor counts summarize the number of agents eligible at each communication scale. BFS cost-to-go uses the known static map, and neighbor counts use dynamic agent positions supplied by the simulator. Auxiliary inputs are standardized using training statistics. Communication results measure planner-message payload under this supplied-information protocol; position acquisition, neighbor discovery, and range signaling are outside the measured payload, with $B_{\text{discovery}} = 0$ in the reported experiments.

Offline supervision and deployment. Offline scores compare uniform-range topologies at fixed recorded states. At deployment, the individual predictions jointly construct a mixed-range topology. The shared selector applies the same decision rule to every agent, while its state-dependent inputs allow different predictions within the same scene. The backbone processes the resulting graph, hypergraph, or communication mask through its existing action policy. The complete adaptive planner is evaluated through closed-loop episodes.

C Communication and Metric Accounting

Communication is measured through logical directed payload deliveries, accounting for the computation and message reuse of each backbone. Let $\mathcal{M}_{b,t,k}$ be the multiset of deliveries for backbone b, timestep t, and communication round k. If message m contains d_m scalar values represented using w_m bytes each, then

$$B_b = \sum_t \sum_k \sum_{m \in \mathcal{M}_{b,t,k}} d_m w_m + B_{\text{discovery}}. \tag{9}$$

Self-information is excluded. Multiple rounds and actual repeated deliveries are included. Padding represents local computation rather than a transmitted payload unless the implementation sends it. $B_{\text{discovery}}$ includes extra exchanges needed to construct selector features or identify neighbors, if such exchanges are required. This measure assumes logical point-to-point delivery; physical broadcast airtime and network protocol overhead are separate quantities.

Table 6: Backbone-specific accounting under the common logical-payload metric.

Backbone	Accounting
MAGAT	Delivered feature vectors on directed inter-agent links are counted in each communication round using the actual payload dimensions and precision. Attention coefficients are local calculations unless transmitted.
HMAGAT	Feature deliveries needed to realize the chosen hypergraph computation are counted. Reuse of one received feature across several groups counts as one delivery, while repeated transmissions are counted separately.
LC-MAPF	Valid neighbor-message deliveries are counted across all communication rounds using the actual message dimensions. Fixed empty slots and masked padding are not transmissions unless explicitly sent.
Selector	Position acquisition, neighbor discovery, and range signaling are not modeled; supplied simulator features incur no recorded message payload.

Payload widths, precision, rounds, deduplication, active-link logging, and discovery assumptions follow the evaluated code. The metric describes application-level payload under logical point-to-point delivery.

Episode length and communication cost. Tables report cumulative episode payload, combining communication per step and episode duration. Runtime uses the same episode and timeout conventions.

Aggregation formulas. For configuration c, let $\bar{X}_{b,c}^{\text{fixed}}$ and $\bar{X}_{b,c}^{\text{adapt}}$ be paired-instance means of a positive cost metric. The main cost summary is

$$\overline{\rho}_{b,X} = \frac{1}{21}\sum_{c=1}^{21} \frac{\bar{X}_{b,c}^{\text{adapt}}}{\bar{X}_{b,c}^{\text{fixed}}}, \qquad \overline{\text{ACC}}_b = \frac{1}{21}\sum_{c=1}^{21} \text{ACC}_{b,c}. \tag{10}$$

The reported cost reduction is $100(1-\overline{\rho}_{b,X})\%$. Each configuration has equal weight, and individual configuration results are reported separately.

D Complete Results and Improvement Coverage

Table 7 reports all HMAGAT metrics; Tables 8 and 9 report success rates for MAGAT and LC-MAPF. ACC is in percent, SoC follows the stated path-cost convention, communication follows the logical-payload metric, and time is in seconds. Statistical uncertainty is computed from paired evaluation episodes.

E Additional HMAGAT Controls

The strategy comparison uses DM-160, ER-224, and SW-192. Candidate sets, backbone checkpoints, evaluation instances, and collision handling are matched across strategies. Each strategy is evaluated against HMAGAT's default fixed reference for its configuration (Table 7); additional fixed-range comparisons appear in Table 11.

Fixed and random controls. Table 11 compares small, medium, and large fixed ranges on dense mazes, an open room, and a sparse warehouse, alongside adaptive communication. The main fixed reference uses HMAGAT's default range. Adaptive communication achieves the highest success-rate point estimate in every listed configuration. In dense mazes, the strongest fixed candidate varies with population: medium performs best at the lowest population, while small performs best at the

Table 7: HMAGAT, all 21 configurations. SoC is the raw sum of costs under the all-instance convention: a failed episode charges every agent the full step limit, and the value is the mean over all 128 evaluation episodes, failures included. Comm. is the logical payload volume per episode; Time is algorithm wall-clock seconds.

	Fixed				Adaptive			
Configuration	ACC	SoC	Comm.	Time	ACC	SoC	Comm.	Time
DM-128	78.9	18,822	1,394,000	45.3	**82.8**	**18,728**	**586,000**	**32.5**
DM-160	53.1	42,923	2,089,000	65.8	**67.2**	**34,396**	**1,192,000**	**56.7**
DM-192	19.5	81,252	2,748,000	87.9	**41.4**	**66,690**	**2,040,000**	**80.5**
DW-96	96.1	8,005	760,000	27.6	**100.0**	**6,273**	**293,000**	**18.7**
DW-128	78.9	25,457	2,094,000	68.3	**98.4**	**15,619**	**861,000**	**47.0**
ER-128	98.4	7,355	562,000	22.7	**99.2**	**7,115**	**258,000**	**17.5**
ER-160	96.1	10,850	1,004,000	36.6	**98.4**	**10,159**	**488,000**	**27.7**
ER-192	60.9	26,126	1,532,000	50.8	**85.9**	**17,701**	**895,000**	**42.3**
ER-224	18.0	49,197	1,861,000	58.8	**46.9**	**36,969**	**1,388,000**	**54.1**
OST-128	100.0	20,566	262,000	43.8	100.0	**20,463**	**133,000**	**42.5**
OST-160	100.0	25,762	413,000	57.0	100.0	**25,633**	**198,000**	**54.3**
OST-192	100.0	31,015	607,000	71.9	100.0	**30,860**	**277,000**	**67.3**
OST-224	84.4	66,169	832,000	87.0	**100.0**	**36,148**	**366,000**	**80.2**
OST-256	0.0	262,144	1,003,000	90.1	**9.4**	**241,327**	**467,000**	**90.0**
SM-128	93.0	6,220	787,000	24.7	**94.5**	**5,783**	**293,000**	**14.2**
SM-160	73.4	15,314	1,332,000	40.9	**86.7**	**10,894**	**584,000**	**26.5**
SM-192	36.7	33,906	1,840,000	55.7	**76.6**	**18,176**	**1,044,000**	**41.9**
SW-128	99.2	4,386	424,000	15.9	**100.0**	**4,154**	**175,000**	**10.6**
SW-160	91.4	9,055	803,000	30.1	**95.3**	**7,712**	**339,000**	**19.9**
SW-192	82.8	15,543	1,233,000	45.9	**90.6**	**12,393**	**579,000**	**31.5**
SW-224	14.1	50,884	1,580,000	59.8	**85.2**	**18,768**	**900,000**	**46.3**

Table 8: MAGAT, ACC per configuration (%).

Config	Fixed	Adaptive	Δ	Config	Fixed	Adaptive	Δ
DM-128	50.8	**59.4**	+8.6	SM-128	77.3	**83.6**	+6.3
DM-160	2.3	**9.4**	+7.1	SM-160	15.6	**25.0**	+9.4
DM-192	0.0	**7.8**	+7.8	SM-192	0.0	**17.2**	+17.2
DW-96	34.4	**45.3**	+10.9	SW-128	99.2	99.2	0.0
DW-128	0.0	**8.6**	+8.6	SW-160	23.4	**35.2**	+11.8
ER-128	84.4	**89.8**	+5.4	SW-192	0.0	**4.7**	+4.7
ER-160	28.1	**39.1**	+11.0	SW-224	0.0	**30.5**	+30.5
ER-192	2.3	**13.3**	+11.0	OST-128	100.0	100.0	0.0
ER-224	0.0	**12.5**	+12.5	OST-160	38.3	**53.1**	+14.8
OST-192	0.0	**3.1**	+3.1	OST-224	0.0	**6.2**	+6.2
OST-256	0.0	**3.9**	+3.9				

Table 9: LC-MAPF, ACC per configuration (%).

Config	Fixed	Adaptive	Δ	Config	Fixed	Adaptive	Δ
DM-128	69.5	**74.2**	+4.7	SM-128	81.2	**85.2**	+4.0
DM-160	18.0	**22.7**	+4.7	SM-160	28.9	**34.4**	+5.5
DM-192	0.0	**4.7**	+4.7	SM-192	6.2	**16.4**	+10.2
DW-96	81.2	**85.9**	+4.7	SW-128	92.2	**95.3**	+3.1
DW-128	0.0	**4.7**	+4.7	SW-160	14.8	**20.3**	+5.5
ER-128	46.1	**51.6**	+5.5	SW-192	0.0	**2.3**	+2.3
ER-160	19.5	**25.0**	+5.5	SW-224	0.0	**18.0**	+18.0
ER-192	9.4	**15.6**	+6.2	OST-128	100.0	100.0	0.0
ER-224	2.3	**9.4**	+7.1	OST-160	61.7	**68.0**	+6.3
OST-192	11.7	**15.6**	+3.9	OST-224	2.3	**6.2**	+3.9
OST-256	0.0	**2.3**	+2.3				

Table 10: Point-estimate changes across all 21 configurations (improved / unchanged / degraded). ACC is episode success rate; communication is per-episode payload; time is algorithm wall-clock time. SoC changes compare adaptive and fixed communication within each backbone under its corresponding accounting convention.

Backbone	ACC	SoC	Communication	Time
HMAGAT	18/3/0	21/0/0	21/0/0	21/0/0
MAGAT	19/2/0	21/0/0	21/0/0	21/0/0
LC-MAPF	20/1/0	21/0/0	21/0/0	21/0/0

two higher populations. The large range is strongest in ER-160, whereas small is strongest in SW-192. The main-text comparison in Section 5.4 discusses how these choices vary across configurations. The random strategy samples independently from a validation-estimated range distribution, preserving marginal preferences while removing state dependence.

Density-based selection. The density-based selector uses a descriptor from the permitted local information. Validation selects the density bins and their range assignments, which may be nonmonotonic. The evaluated variant uses the 11×11 agent-occupancy statistic with three validation-selected bins mapped to small/medium/large, with a nonmonotonic middle-bin assignment.

SW-UCB. SW-UCB maintains per-agent recent returns for each candidate range, with exploration of unobserved candidates. The evaluated variant uses one-step change in the agent's normalized goal-distance estimate as feedback, a 50-step sliding window, exploration coefficient 0.2, and optimistic initialization; feedback updates stop once the agent reaches its goal.

The all-instance path-cost and runtime ratios of the reported selector, taken from Table 7, are 0.80 and 0.86 on DM-160, 0.75 and 0.92 on ER-224, and 0.80 and 0.69 on SW-192.

Table 11: Fixed-range and adaptive HMAGAT success rates (%) on five configurations. Small, medium, and large denote the candidate communication scales; adaptive results match Table 7.

Configuration	Small	Medium	Large	Adaptive
DM-128	75.8	82.0	78.9	**82.8**
DM-160	54.7	49.2	53.1	**67.2**
DM-192	39.1	35.2	19.5	**41.4**
ER-160	89.8	93.0	96.1	**98.4**
SW-192	85.9	84.4	82.8	**90.6**

E.1 Sampling and Ablation Metrics

The natural sampler preserves empirical training frequencies. The evaluated equal-quota variant balances over radius labels only, and the stratified variant balances jointly over radius labels and map families. Table 12 compares the sampling strategies through closed-loop planning quality and resource use.

Table 12: Sampling comparison and supplementary ablation results (ACC (%) / communication normalized to the fixed baseline of Table 7).

Variant	DM-192	SW-224	ER-128
Natural sampling	39.8 / 0.65	83.6 / 0.44	97.7 / 0.71
Equal-quota sampling	41.4 / 0.74	85.2 / 0.57	99.2 / 0.46
Stratified sampling	40.6 / 0.63	84.4 / 0.42	98.4 / 0.70
Full candidate pool	39.1 / 0.59	84.4 / 0.41	97.7 / 0.28
Observation only	33.6 / 0.70	78.1 / 0.53	95.3 / 0.34
Auxiliary features only	25.8 / 0.76	71.9 / 0.61	90.6 / 0.40

F Runtime Analysis

Following implementation optimization, feature extraction and selector inference add slightly over one second per run. Both components are included in the algorithm timing boundary.

The timing boundary includes episode preprocessing and each planning step:

$$T_{\text{algo}} = T_{\text{pre}} + \sum_{t=0}^{T-1} \left(T^t_{\text{features}} + T^t_{\text{selector}} + T^t_{\text{topology}} + T^t_{\text{policy}} \right). \tag{11}$$

Preprocessing includes cost-to-go construction; feature extraction includes the state descriptors used by the selector; topology construction and policy inference follow the selected backbone. GPU timings use synchronization at component boundaries. Environment simulation, rendering, file I/O, and initialization outside this boundary are tracked separately.

LC-MAPF uses tokenized observations padded to 256 entries and masked, fixed-size neighbor slots (CognitiveAISystems, 2026). This interface retains dense attention dimensions as the active communication neighborhood changes, accounting for the limited computational savings available from reducing active neighbors. The relevant implementation components are the observation generator and policy model.

G Deployment Statistics and Case Data

Communication range usage. For a progress bin b and a stated set of eligible agent-timestep observations $\mathcal{I}_b$, range usage is

$$u_{b,R} = \frac{\sum_{(e,t,i)\in\mathcal{I}_b} \mathbf{1}[r^t_{e,i} = R]}{|\mathcal{I}_b|}. \tag{12}$$

Usage is weighted by agent-timesteps. Figure 4 normalizes each reported, rounded count vector by its own total. Table 13 gives the underlying summaries for the three representative environments. Reading down a configuration shows how its communication mix changes over planning progress; reading across a row shows the coexistence of different scopes within that interval. This connects the episode-level trends to the agent-specific decisions that produce them.

Table 13: Multi-trajectory range-use summaries. Counts are approximate and may differ slightly from nominal populations after rounding. Unreported intervals are not interpolated in Figure 4.

Configuration	Progress interval (%)	Small	Medium	Large
DM-192	0 to 10	192	0	0
	20 to 30	183	9	0
	40 to 50	129	62	1
	60 to 70	75	116	1
	80 to 90	52	140	0
	90 to 100	40	151	0
SW-224	0 to 10	224	0	0
	30 to 40	219	5	0
	50 to 60	189	32	3
	70 to 80	129	88	6
	90 to 100	78	138	8
ER-128	0 to 10	3	2	124
	20 to 30	22	7	99
	40 to 50	52	10	66
	60 to 70	86	10	32
	80 to 90	112	10	7
	90 to 100	117	10	2

How the population-level mix evolves. DM-192 shifts from exclusively small neighborhoods in the initial interval to predominantly medium neighborhoods near the end. Large neighborhoods remain rare throughout the reported intervals. SW-224 also shifts toward medium neighborhoods, with

a small group using large neighborhoods later in planning. In both settings, agents move through corridors or aisles that connect local encounters along shared routes. The observed expansion is consistent with agents drawing on a wider part of that traffic as their routes interact.

ER-128 follows a different progression. Large neighborhoods dominate the early interval, while small neighborhoods become increasingly common as the episode advances. In an open room, crossing routes can involve agents approaching from several directions. As those routes separate and agents reach their destinations, compact neighborhoods can become sufficient for more of the population. The continued coexistence of small, medium, and large choices shows that the transition is distributed across individual agents.

A shared scene with different local decisions. Table 14 examines two agents at the same timestep of the same ER-128 episode. At this instant, the population contains 29 small, 12 medium, and 87 large selections. The two agents share the map and planning time but encounter different nearby traffic. Positions and goals use row and column coordinates; distance and BFS cost-to-go entries report the logged input features. The table combines spatial context, goal-related state, and predicted range probabilities for each agent.

Table 14: Agent-specific communication for a6 and a30 in ER-128, episode `ep0000`, at $t = 22/76$.

Descriptor	a6	a30
Selected range	Large	Small
Position	$(31, 10)$	$(16, 4)$
Goal	$(21, 11)$	$(14, 1)$
5×5 agent occupancy	40%	4%
Distance feature	0.14	0.06
BFS cost-to-go feature	0.20	0.09
Neighbors at small/medium/large ranges	10 / 24 / 38	3 / 3 / 5
Observation-window agent occupancy	25%	2.5%
Selector probabilities, small/medium/large	22.2 / 26.0 / 51.6%	35.7 / 29.9 / 34.3%

Spatial context at two scales. Agent a6 has 40% agent occupancy in its immediate 5×5 area and 25% across the observation window. Agent a30 has corresponding occupancies of 4% and 2.5%. The difference is therefore visible both close to each agent and across its wider observation. Expanding from small to large adds 28 eligible neighbors for a6 and two for a30. For a6, the additional scope reaches a substantially larger group of surrounding agents; for a30, the same spatial expansion changes the available neighborhood much less. The selected large and small ranges align with these contrasting communication contexts.

Range predictions within the same population. The probability vectors show how a shared model produces the two choices. Large receives the highest probability for a6, while small receives the highest probability for a30. These predictions are made alongside the other agents' choices, so the population's preference for large neighborhoods at this timestep does not impose a large range on every agent. Goal-related features also differ between the two agents and enter the selector together with occupancy information. The case therefore illustrates how local spatial and planning context can produce different communication scopes within one shared scene.

Additional snapshots in structured maps. The dense-maze snapshot at $t = 194/389$ has 61 small, 131 medium, and zero large selections. Medium communication is common at this instant, while a substantial subset of agents still uses small neighborhoods. The sparse-warehouse snapshot at $t = 103/148$ has 118 small, 101 medium, and five large selections, showing all three scopes in use at once. These snapshots complement the progress summaries by displaying their agent-level mixtures at specific timesteps. Across the three environments, the communication topology is assembled from agents making different choices as they encounter different parts of the same planning problem.