LEARNING INFORMATIVE AND EFFICIENT MULTI-AGENT COMMUNICATION TOPOLOGIES VIA FINE-GRAINED INFORMATION GAIN REWARDS

Anonymous authors
Paper under double-blind review

ABSTRACT

The communication topology in large language model (LLM)-based multi-agent systems governs how information flows among agents, critically shaping both task performance and communication efficiency. However, existing topology learning methods typically rely on graph-level outcome rewards, which provide limited feedback on individual edge contributions. Consequently, even successful topologies may retain redundant or harmful interactions, introducing unnecessary communication overhead and potentially hindering collaboration. To address this limitation, we propose EdgeIG, an information-theoretically guided framework for fine-grained reward modeling in adaptive communication topology learning. Specifically, EdgeIG guides topology generation policy learning based on the information gain from each transmitted message, which is measured by how that message changes the model's support for the ground-truth answer. The resulting edge-level rewards provide dense supervision, penalizing harmful interactions even within successful topologies. We further combine these rewards with a graph-level advantage to jointly improve task performance and communication efficiency. Extensive experiments on six benchmarks spanning general reasoning, mathematical reasoning, and code generation demonstrate that EdgeIG improves task performance and communication efficiency while maintaining strong robustness under prompt attacks.

1 INTRODUCTION

Agents built on large language models (LLMs) have demonstrated strong performance across tasks like reasoning (Yao et al., 2023), code generation (Zhang et al., 2024a), and complex decision-making (Guo et al., 2024). Building on these capabilities, LLM-based multi-agent systems (MAS) can outperform individual agents on complex tasks through collaboration among agents with different roles (Du et al., 2024; Li et al., 2023; Hong et al., 2024). However, effective collaboration also depends on how agents transmit, exchange, and assimilate information. The communication topology defines the connections through which information flows among agents, playing a central role in the effectiveness and efficiency of their collaboration (Zhang et al., 2025a;b; Li et al., 2026b;a).

Early studies on MAS explored manually designed communication structures, from chains (Zhang et al., 2024b) to fully connected graphs (Du et al., 2024). Recent research has advanced automated topology design through graph pruning and graph generation. Pruning methods start from a predefined communication graph and optimize learnable connection weights to remove redundant links or agents, yielding a sparse communication topology (Zhang et al., 2025a; Wang et al., 2025). Generative methods learn task-conditioned communication structures with graph autoencoders (Zhang et al., 2025b) and autoregressive models (Li et al., 2026b). More recently, diffusion-based generative models have been used to construct task-conditioned communication topologies through iterative denoising (Jiang et al., 2025; Zhang et al., 2026). Compared with fixed, manually designed topologies, these approaches offer greater flexibility in adapting communication structures to different task requirements.

Despite these advances in topology design, existing reward formulations provide limited guidance for evaluating individual communication edges. A common optimization strategy trains graph-

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.

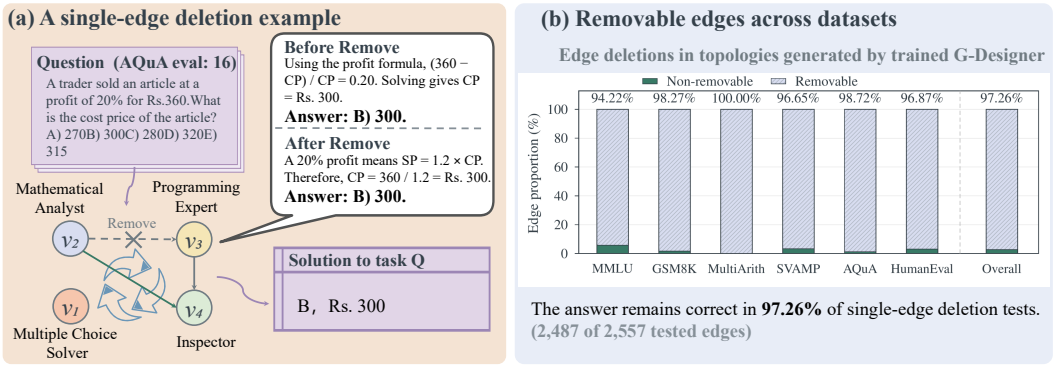


Figure 1: **Removable edges in successful communication topologies.** (a) In an AQuA example, deleting the edge from the Mathematical Analyst to the Programming Expert preserves the correct answer. (b) Single-edge deletion outcomes for initially correct predictions of a trained G-Designer across six datasets. An edge is classified as removable if correctness is preserved after its deletion, and non-removable if the answer becomes incorrect. Overall, 2,487 of 2,557 tested edges (97.26%) are individually removable, pooling deletion tests across datasets.

pruning and graph-generation policies using rewards computed from the final system output (Zhang et al., 2025a;b). A single outcome score evaluates many communication decisions, providing sparse supervision for distinguishing helpful messages from redundant or misleading ones. Our motivation experiment with a trained G-Designer illustrates this limitation: as shown in Figure 1, the answer remains correct in 97.26% of single-edge deletion tests (2,487 of 2,557 tested edges) across six datasets. Thus, a successful topology can receive a high reward even when some of its communication is unnecessary. This limitation motivates fine-grained reward modeling that measures how the information transmitted along each edge contributes to solving the task.

To address this limitation, we propose EdgeIG, a framework for adaptive communication topology learning with fine-grained information-gain rewards. Given a task query, EdgeIG employs a graph neural network (GNN) to construct task-aware agent representations and predict communication links, enabling the communication topology to adapt to task requirements. To guide this topology generation policy, we introduce an edge-level information-gain reward that quantifies the additional information each message provides about the ground-truth answer. Specifically, we obtain the receiver's outputs with and without each incoming message, keeping its other inputs fixed. A frozen decision agent then computes the average log probability of the ground-truth answer conditioned on each output. We define the edge-level information-gain reward as the marginal increase in this log probability when the message is retained compared to when it is removed. This reward provides dense supervision, penalizing harmful interactions even within successful topologies. Finally, we combine discounted edge-level returns with a graph-level advantage incorporating task correctness and prompt-token cost, guiding the topology generation policy to promote beneficial interactions while suppressing harmful or redundant ones.

Our contributions are summarized as follows:

- We identify a granularity mismatch in adaptive multi-agent topology learning: existing methods primarily optimize edges using topology-level performance, without directly measuring the task-dependent contribution of each realized edge.
- We propose an edge-ablation information-gain reward that provides fine-grained reward for individual edges and balances answer-relevant information against communication cost.
- Extensive experiments on six reasoning and code-generation benchmarks, including MMLU and HumanEval, demonstrate the effectiveness of our method in improving task performance and communication efficiency.

2 Related Work

2.1 LLM-Based Multi-Agent Communication Topology Design

Collaboration between multiple LLM-based agents has emerged as an effective approach for solving complex tasks. Existing works have explored diverse ways to organize agent interactions, including role-playing, multi-agent conversation, structured workflows, and debate. For example, CAMEL studies autonomous role-playing between communicative agents (Li et al., 2023), AutoGen provides programmable multi-agent conversation patterns (Wu et al., 2024), MetaGPT organizes specialized roles into structured software-development workflows (Hong et al., 2024), and multi-agent debate enables agents to exchange and refine competing responses (Du et al., 2024). The effectiveness of MAS is closely related to its communication topology, which defines how agents are connected and how they exchange information. Early research on MAS topology design focused on fixed or manually designed communication structures. However, the inherent rigidity of these fixed topologies limits their adaptability across diverse tasks.

To improve adaptability, recent studies focus on learning task-specific communication topologies. AgentPrune and AgentDropout learn sparse communication structures by removing redundant connections and/or agents from predefined topologies (Zhang et al., 2025a; Wang et al., 2025). G-Designer employs a graph auto-encoder to learn task-adaptive communication topologies from task and agent representations (Zhang et al., 2025b). More recently, ARG-Designer generates communication topologies autoregressively by selecting agent roles and establishing communication links (Li et al., 2026b), while GTD and RADAR use graph diffusion to iteratively generate task-specific communication structures (Jiang et al., 2025; Zhang et al., 2026). These methods differ in how communication topologies are constructed and optimized. For approaches that optimize communication topologies using task-specific rewards, the reward is typically defined for the whole sampled graph, providing no direct measure of how each communication edge contributes to the final answer. Our work addresses this limitation by introducing fine-grained edge-level reward modeling for topology optimization.

2.2 Policy Optimization with Fine-Grained Rewards

Optimizing communication topologies with downstream LLM execution is challenging because the task utility is generally non-differentiable. Policy-gradient methods therefore provide a practical approach to optimizing topology generation policies (Zhang et al., 2025b). For advantage estimation in policy-gradient updates, GRPO uses the relative rewards of outputs sampled for the same query, eliminating the need for a separate value model (Shao et al., 2024). When these advantages are computed from a single outcome-based reward for each sampled communication graph, however, the resulting feedback evaluates topologies as a whole without directly measuring individual edge contributions.

Fine-grained reward modeling has been explored to provide denser supervision for intermediate interactions. IGPO introduces an information-gain reward that measures how each interaction changes the model's support for the ground-truth answer (Wang et al., 2026). Building on this principle, we extend information-gain reward modeling from interaction turns to communication edges in MAS, providing edge-level supervision for topology optimization.

3 Preliminaries

3.1 Graph-Structured Multi-Agent Reasoning

Following prior graph-based formulations of MAS (Zhang et al., 2025b; Li et al., 2026b), we represent a communication topology as a directed graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \ldots, v_N\}$ is a set of role-specialized agents and $\mathcal{E}$ contains their communication links. Each agent v_j is an LLM instance with a fixed role instruction r_j. An edge $(v_i, v_j) \in \mathcal{E}$ indicates that the output of v_i is supplied to v_j as part of its input context. The incoming neighbors of v_j are denoted by $\mathcal{N}_{\text{in}}(v_j) = \{v_i : (v_i, v_j) \in \mathcal{E}\}$.

Given a query q, the agents communicate over a fixed acyclic graph for H rounds. Within each round t, agents execute in topological order, so each agent receives its incoming neighbors' outputs

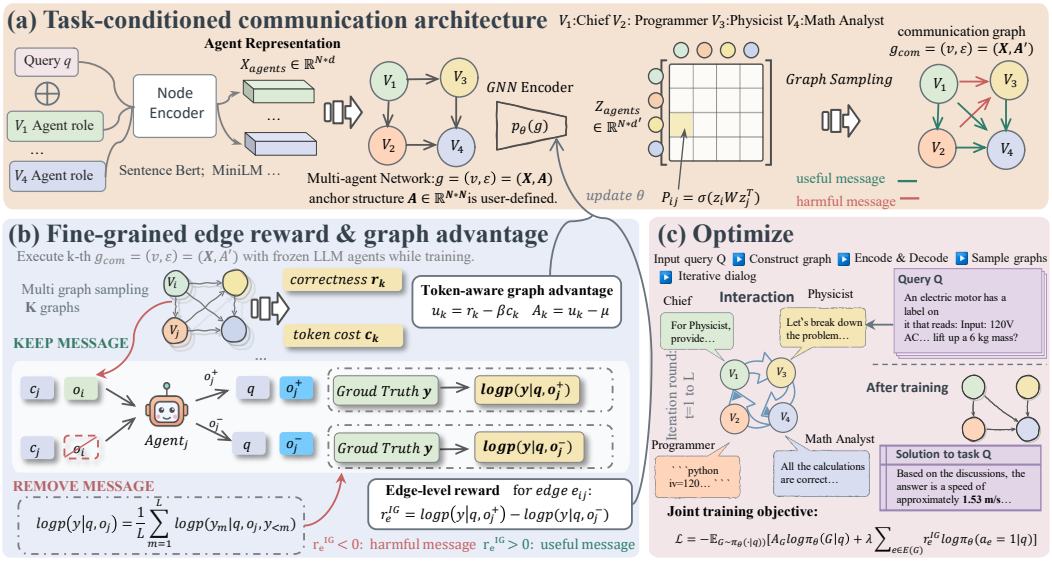


Figure 2: **Overview of the proposed EdgeIG framework.** (a) The topology generation policy encodes the query and agent roles to sample task-conditioned communication graphs. (b) Paired message retention and removal yield edge-level information-gain rewards, while task correctness and prompt-token cost determine the graph-level advantage. (c) The graph-level advantage and discounted edge-level returns jointly optimize the topology generation policy while the LLM agents remain frozen. The learned policy then constructs communication topologies for inference without edge ablation.

from the same round. Agent v_j generates its response as:

$$P_j^{(t)} = \big(r_j, q, \{o_i^{(t)} : v_i \in \mathcal{N}_{\text{in}}(v_j)\}\big), \qquad o_j^{(t)} = \text{LLM}_j(P_j^{(t)}). \tag{1}$$

Here, $P_j^{(t)}$ and $o_j^{(t)}$ denote the prompt and response at round t, respectively. After H rounds, a separate decision agent aggregates the final-round responses:

$$\hat{y} = \text{Aggregate}\big(q, \{o_j^{(H)}\}_{j=1}^{N}\big). \tag{2}$$

Agent roles and underlying LLMs remain fixed. We omit round indices in the method description; multi-round edge-reward aggregation is specified in Appendix B.

3.2 Task-Conditioned Topology Learning

Given a training dataset $\mathcal{D}$ of query-answer pairs (q, y), a topology generation policy $\pi_\theta(G \mid q)$ generates a communication graph conditioned on the query. A standard learning objective maximizes the expected graph-level outcome reward:

$$\max_\theta \ \mathbb{E}_{(q,y)\sim\mathcal{D},\, G\sim\pi_\theta(\cdot|q)} \left[R(G; q, y)\right], \tag{3}$$

where $R(G; q, y)$ measures final-answer quality, such as correctness, averaged over any stochastic agent generation. This reward evaluates the entire collaboration but does not directly identify the contribution of each participating edge. In particular, a successful graph may still contain redundant or misleading messages. This limitation motivates a finer-grained signal that evaluates what individual messages contribute beyond the receiver's existing information.

4 Method

As depicted in Fig. 2, the distinctive characteristics of EdgeIG can be highlighted in these three aspects:

- EdgeIG employs a task-conditioned topology generation policy to construct adaptive communication graphs based on the task query and agent roles, while enforcing an acyclic communication structure.
- Through the fine-grained information-gain reward, EdgeIG evaluates each realized communication edge by measuring how its transmitted message changes the receiver's support for the correct answer, providing edge-specific supervision for topology learning.
- EdgeIG combines discounted edge-level returns with a graph-level advantage that accounts for task correctness and prompt-token cost to optimize the topology generation policy, while keeping the underlying LLM agents frozen.

In the following sections, we will illustrate the key components of EdgeIG in detail.

4.1 Task-Conditioned Topology Generation Policy Architecture

The topology generation policy consists of a frozen node encoder, a GNN encoder, and a learnable decoder that predicts communication probabilities.

Given a task query Q, we concatenate each agent's role description r_i with the query and use the frozen node encoder to obtain its feature vector x_i:

$$x_i = \mathrm{Encoder}(r_i \oplus Q), \tag{4}$$

where we use the pretrained `all-MiniLM-L6-v2` encoder and keep its parameters frozen.

The GNN encoder applies GATv2 over a fully connected graph, followed by an MLP, to obtain agent representations z_i. The decoder then combines pairs of agent representations through a matrix W to predict communication probabilities. To ensure acyclicity, we allow connections only from lower-indexed agents to higher-indexed agents. The communication probabilities are computed as

$$(P_\theta)_{ij} = \begin{cases} \sigma\left(z_i W z_j^\top\right), & i < j, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

where W is a learnable weight matrix. During training, the communication probabilities parameterize Bernoulli distributions from which communication edges are sampled. During inference, edges with probabilities greater than or equal to 0.5 are retained to construct the communication graph.

4.2 Fine-Grained Edge Information Gain

To quantify the information gain of each realized edge, we consider how its message reduces uncertainty about the correct answer beyond the receiver's existing context. For a labeled training example, we estimate this uncertainty using the negative log likelihood of the ground-truth answer under the frozen decision agent.

Intuitively, if a node output contains answer-relevant information, conditioning on this output should increase the likelihood assigned to the ground-truth answer. Given a question q, the output o_j of node v_j, and the ground-truth answer $y = (y_1, \ldots, y_L)$, we use the length-normalized ground-truth log likelihood

$$\log \pi(y \mid q, o_j) = \frac{1}{L} \sum_{m=1}^{L} \log \pi(y_m \mid q, o_j, y_{<m}), \tag{6}$$

where π denotes the token probability given by the frozen decision agent, and $y_{<m} = (y_1, \ldots, y_{m-1})$ denotes the ground-truth answer tokens preceding y_m. Here, $\log \pi(y \mid q, o_j)$ represents the answer-support score of o_j; a larger value indicates stronger support for the ground-truth answer.

The score above evaluates a node output, whereas topology optimization operates on communication edges. Since an edge affects the final decision by changing the output of its receiver, we quantify its contribution through paired edge ablation. For a realized edge $e = (v_i, v_j)$, retaining its message produces the receiver's output o_j^+. We then remove only this message, keep the remaining inputs of v_j unchanged, and resample its output as o_j^-. The resulting change in ground-truth support defines the fine-grained edge information-gain reward:

$$r_e^{\mathrm{IG}} = \log \pi(y \mid q, o_j^+) - \log \pi(y \mid q, o_j^-). \tag{7}$$

A positive r_e^{IG} indicates that the transmitted message improves the receiver's output and increases its support for the correct decision. In contrast, a negative value indicates that the message introduces distracting or harmful information and that the corresponding edge should be suppressed.

In practice, we compute the average log probability of the ground-truth answer under teacher forcing (Wang et al., 2026). Specifically, the answer is appended to the question and receiver's output, and its token-level log probabilities are obtained through a single forward pass of the frozen decision agent. This calculation requires no additional reward model.

4.3 Joint Policy Optimization

4.3.1 Discounted Edge-Level Return

The edge-level information-gain reward captures a message's immediate contribution to the receiver's output but does not include feedback from downstream communication. To incorporate this feedback into topology learning, we compute a discounted edge-level return that combines the immediate reward with discounted information-gain rewards from downstream edges. For an edge e in graph G_k, we define

$$R_{k,e}^{\mathrm{IG}} = r_{k,e}^{\mathrm{IG}} + \sum_{e' \in \mathcal{D}_k(e)} \gamma^{d(e,e')} r_{k,e'}^{\mathrm{IG}}, \tag{8}$$

where $\mathcal{D}_k(e)$ is the set of downstream edges of e, $d(e, e')$ is their directed distance, and $\gamma \in (0, 1]$ is the discount factor. This return assigns larger weights to nearby effects and progressively discounts more distant ones.

4.3.2 Token-Aware Graph-Level Advantage

While edge-level information-gain rewards evaluate individual message contributions, they do not account for communication cost. To encourage efficient communication while maintaining task performance, we introduce a token-aware graph-level advantage.

For each task, we independently sample K graphs and execute the MAS under every realized topology. Let $R_k \in \{0, 1\}$ indicate whether G_k produces the correct final answer, and let T_k denote the total number of prompt tokens consumed by its regular execution, including the final decision agent. To account for communication overhead, we combine task correctness with a normalized prompt-token cost:

$$c_k = \frac{T_k}{\max_{1 \le l \le K} T_l}, \qquad u_k = R_k - \beta c_k, \tag{9}$$

where $\beta \ge 0$ balances task performance and communication efficiency. The token penalty encourages the policy to reduce prompt-token consumption while optimizing task correctness.

Following the group-relative optimization principle (Shao et al., 2024), we define the token-aware graph-level advantage as

$$A_k = u_k - \mu, \tag{10}$$

where μ denotes the mean utility across the K graphs sampled for the same query. This mean-centered advantage evaluates each topology relative to the sampled group while retaining the scale of the token-aware utility. A positive advantage indicates above-average utility, reflecting the trade-off between task correctness and communication cost.

4.3.3 Training Objective

We optimize the topology generation policy by combining the token-aware graph-level advantage with the discounted edge-level returns. The former balances final-answer correctness against prompt-token consumption across sampled graphs, while the latter provides dense edge-level supervision that incorporates both immediate and downstream information-gain rewards:

$$\mathcal{L} = -\frac{1}{K} \sum_{k=1}^{K} \left[A_k \log \pi_\theta(G_k \mid q) + \lambda \sum_{e \in G_k} R_{k,e}^{\mathrm{IG}} \log \pi_\theta(a_{k,e} = 1 \mid q) \right]. \tag{11}$$

where λ balances the graph-level and edge-level signals. Each discounted return is assigned to the topology action that realizes the corresponding edge. During training, gradients are propagated

only through the graph and edge log probabilities, while agent outputs, graph-level advantages, and edge-level returns are treated as constants. The underlying LLM agents and the decision agent remain frozen. At inference time, the topology generation policy requires neither reference answers nor edge ablation or additional reward-scoring calls.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets and Metrics. To evaluate EdgeIG across diverse tasks, we use six established benchmarks spanning general reasoning, mathematical reasoning, and code generation, following prior studies (Zhang et al., 2025a;b; Li et al., 2026b). These include MMLU (Hendrycks et al., 2021) for general reasoning; GSM8K (Cobbe et al., 2021), MultiArith (Roy & Roth, 2015), SVAMP (Patel et al., 2021), and AQuA (Ling et al., 2017) for mathematical reasoning; and HumanEval (Chen et al., 2021) for code generation. We report accuracy on the general and mathematical reasoning benchmarks and pass@1 on HumanEval.

Baselines. We compare EdgeIG against a diverse set of strong baselines covering single-agent prompting, multi-agent collaboration, and automated topology design. Single-agent methods include Vanilla prompting, Chain-of-Thought (CoT) (Wei et al., 2022), and CoT with self-consistency (CoT-SC) (Wang et al., 2023). Multi-agent baselines include GPTSwarm (Zhuge et al., 2024) and DyLAN (Liu et al., 2023). Topology-design baselines include AgentPrune (Zhang et al., 2025a), AgentDropout (Wang et al., 2025), G-Designer (Zhang et al., 2025b), ARG-Designer (Li et al., 2026b), and OFA-MAS (Li et al., 2026c).

Implementation Details. We use Qwen3-8B (Yang et al., 2025) as the LLM backbone for all methods. EdgeIG uses 5 agents for MMLU and HumanEval and 4 for the mathematical reasoning benchmarks. A separate decision agent aggregates their responses after two communication rounds to produce the final answer. We use the pretrained `all-MiniLM-L6-v2` as the node encoder, with an embedding dimension of 384. We train the topology generation policy for 30 updates, using four queries per batch and $K = 8$ sampled graphs per query. For HumanEval, we use test-based scoring for edge rewards. Additional implementation details are provided in Appendix B.

5.2 PERFORMANCE COMPARISON

Table 1: Overall performance comparison on six benchmarks (%). The best results are highlighted in bold. Arrows indicate absolute performance changes relative to Vanilla.

Method	MMLU	GSM8K	AQuA	MultiArith	SVAMP	HumanEval	Avg.
Vanilla	72.78	88.15	82.68	96.67	93.40	81.37	85.84
CoT	76.81 ↑ 4.03	91.81 ↑ 3.66	83.07 ↑ 0.39	97.50 ↑ 0.83	93.60 ↑ 0.20	87.58 ↑ 6.21	88.39 ↑ 2.55
CoT-SC	77.34 ↑ 4.56	92.48 ↑ 4.33	84.13 ↑ 1.45	97.67 ↑ 1.00	93.83 ↑ 0.43	87.08 ↑ 5.71	88.75 ↑ 2.91
GPTSwarm (AP impl.)	79.73 ↑ 6.95	93.84 ↑ 5.69	79.36 ↓ 3.32	97.83 ↑ 1.16	94.60 ↑ 1.20	88.12 ↑ 6.75	88.91 ↑ 3.07
DyLAN	78.43 ↑ 5.65	89.92 ↑ 1.77	86.22 ↑ 3.54	97.67 ↑ 1.00	93.90 ↑ 0.50	85.98 ↑ 4.61	88.69 ↑ 2.84
AgentPrune	78.43 ↑ 5.65	92.78 ↑ 4.63	85.04 ↑ 2.36	**98.33** ↑ 1.66	94.90 ↑ 1.50	87.58 ↑ 6.21	89.51 ↑ 3.67
AgentDropout	79.31 ↑ 6.53	93.21 ↑ 5.06	84.66 ↑ 1.98	97.72 ↑ 1.05	94.85 ↑ 1.45	88.06 ↑ 6.69	89.63 ↑ 3.79
G-Designer	78.65 ↑ 5.87	93.05 ↑ 4.90	84.52 ↑ 1.84	98.13 ↑ 1.46	95.00 ↑ 1.60	88.40 ↑ 7.03	89.63 ↑ 3.78
ARG-Designer	76.68 ↑ 3.90	89.18 ↑ 1.03	83.95 ↑ 1.27	97.81 ↑ 1.14	93.57 ↑ 0.17	86.78 ↑ 5.41	88.00 ↑ 2.15
OFA-MAS	79.91 ↑ 7.13	91.01 ↑ 2.86	84.11 ↑ 1.43	98.21 ↑ 1.54	93.97 ↑ 0.57	**89.26** ↑ 7.89	89.41 ↑ 3.57
EdgeIG (Ours)	**81.05** ↑ 8.27	**94.75** ↑ 6.60	**87.26** ↑ 4.58	98.04 ↑ 1.37	**95.42** ↑ 2.02	89.17 ↑ 7.80	**90.95** ↑ 5.11

To evaluate the effectiveness of EdgeIG in learning communication topologies, we compare its task performance with single-agent and multi-agent baselines across six benchmarks. As shown in Table 1, EdgeIG achieves the highest average score of 90.95%, outperforming the strongest competing methods, AgentDropout and G-Designer, by 1.32 percentage points on average. In particular, EdgeIG obtains the best performance on four of the six benchmarks, including MMLU, GSM8K, AQuA, and SVAMP. Compared with the strongest baseline on each of these datasets, EdgeIG improves accuracy by 1.14, 0.91, 1.04, and 0.42 points, respectively. On the remaining benchmarks,

EdgeIG remains highly competitive, trailing the best result by only 0.29 points on MultiArith and 0.09 points on HumanEval. These results demonstrate the effectiveness of EdgeIG across general reasoning, mathematical reasoning, and code generation. Notably, these gains are observed across benchmarks with different reasoning characteristics rather than being concentrated in a single task family. This consistency suggests that the learned topology can adapt its communication pattern to heterogeneous tasks while preserving strong performance across domains.

5.3 Ablation Study

Table 2: Results of ablation study. Δ denotes the change relative to Vanilla.

	AQuA		MMLU	
Method	Acc.	Δ	Acc.	Δ
Vanilla	82.68	–	72.78	–
GraphIG	85.38	+2.70	77.78	+5.00
EdgeIG	**87.26**	+4.58	**81.05**	+8.27
w/o token cost	84.43	+1.75	77.78	+5.00
w/o edge-level reward	85.38	+2.70	75.16	+2.38

To evaluate the effectiveness of the information-gain reward and the token-cost term, we conduct ablation studies on AQuA and MMLU, with results reported in Table 2. We first compare EdgeIG with GraphIG to assess the benefit of fine-grained reward allocation. GraphIG sums the edge-level information-gain rewards and assigns the resulting reward to the entire graph, whereas EdgeIG assigns each reward to its corresponding communication edge. EdgeIG outperforms GraphIG on both benchmarks, with a larger gain of 3.27 percentage points on MMLU, supporting the benefit of edge-level reward allocation. We then remove the information-gain reward while retaining the graph-level advantage to examine the contribution of the reward itself. Removing this reward reduces accuracy on both benchmarks. For example, MMLU accuracy drops from 81.05% to 75.16%, highlighting the importance of information-gain rewards for learning effective communication topologies. Finally, we remove the token-cost term from the graph-level advantage while retaining the correctness reward and information-gain rewards. Accuracy decreases by 2.83 percentage points on AQuA and 3.27 points on MMLU, indicating that accounting for communication cost also benefits task performance.

5.4 Token Efficiency

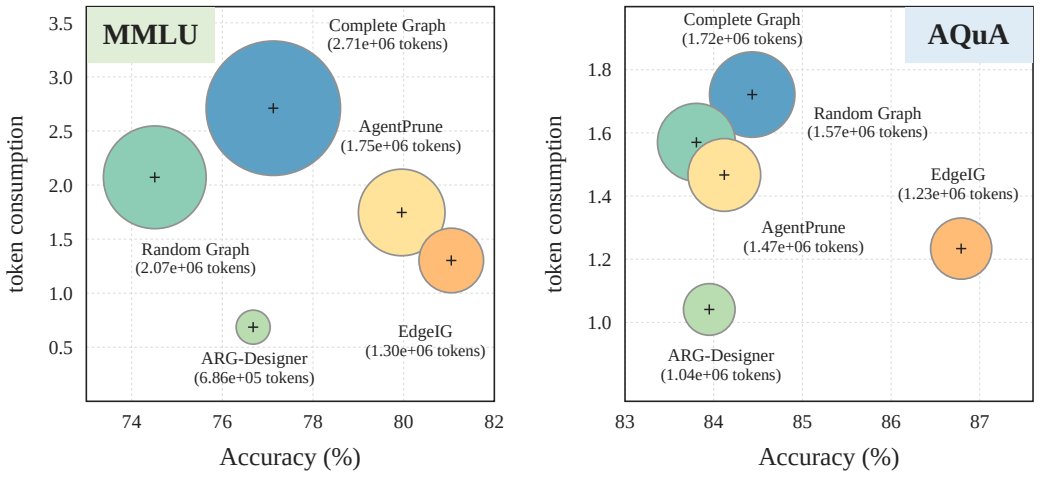


Figure 3: Visualization of accuracy and total LLM inference token consumption (input and output) for different communication topologies on MMLU (left) and AQuA (right). Crosses indicate means over three seeds. Bubble diameter is proportional to token consumption.

To assess whether EdgeIG reduces communication overhead while maintaining task performance, we compare accuracy and total LLM inference token consumption on MMLU and AQuA. Figure 3 reports results averaged over three seeds. Token counts include both input and output tokens and exclude training costs.

EdgeIG achieves higher accuracy with lower token consumption than Complete Graph, Random Graph, and AgentPrune on both datasets. Compared with Complete Graph, it reduces token consumption from 2.71M to 1.30M on MMLU and from 1.72M to 1.23M on AQuA, saving approximately 52.0% and 28.5%, respectively. Relative to AgentPrune, the corresponding savings are 25.7% and 16.3%, with accuracy gains of 1.09 and 2.67 percentage points.

ARG-Designer uses fewer tokens, whereas EdgeIG improves accuracy by 4.37 points on MMLU and 2.84 points on AQuA. Together, these results show that EdgeIG achieves the highest accuracy among the compared methods while reducing communication costs relative to fixed topologies and AgentPrune.

5.5 Robustness to Prompt Attacks

To assess the robustness of the learned communication topologies when an individual agent receives adversarial instructions, we inject adversarial prompts into a single agent on MMLU following prior work (Zhuge et al., 2024). As shown in Figure 4, EdgeIG achieves the highest post-attack accuracy of 77.12%, with a decrease of only 3.88 percentage points from 81.0%, substantially smaller than the drops observed for Complete and Random topologies. It also outperforms the strongest attacked baseline, AgentPrune, by 6.52 points. These results indicate that the communication topologies learned by EdgeIG sustain effective collaboration under localized prompt attacks.

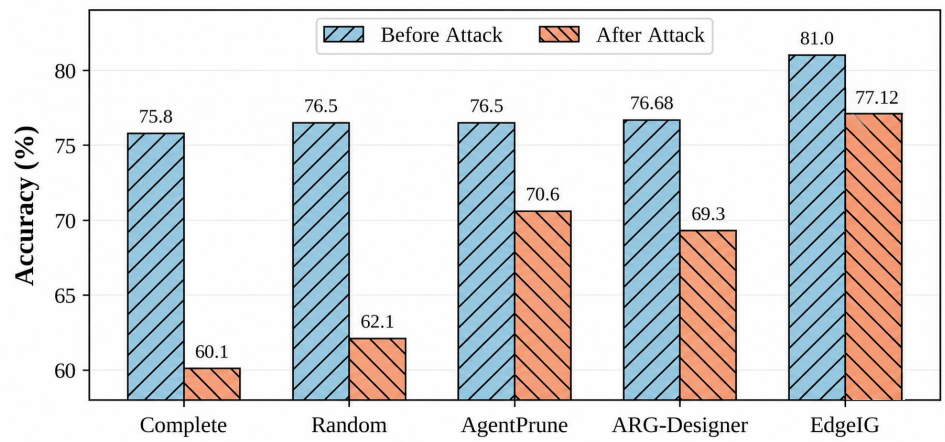


Figure 4: Accuracy (%) of different multi-agent communication-topology methods before and after prompt attacks on MMLU.

6 Conclusion

In this paper, we present EdgeIG, a framework for adaptive communication topology learning with fine-grained information-gain rewards. By measuring how each realized communication edge changes the model's support for the ground-truth answer and combining this edge-level supervision with a token-aware graph-level advantage, EdgeIG learns task-adaptive communication topologies that promote useful information flow while reducing unnecessary communication. Extensive experiments across six benchmarks show that EdgeIG improves overall task performance and communication efficiency while maintaining strong robustness under prompt attacks. These results highlight the benefit of providing supervision at the same granularity as the communication decisions being optimized. More broadly, EdgeIG demonstrates the importance of distinguishing the utility of individual communication edges for adaptive topology learning.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, drafting and revising limited portions of the manuscript, and LaTeX formatting. The research methodology, experimental design, implementation, data analysis, and conclusions were developed and verified by the authors. All AI-assisted content was reviewed and revised by the authors, who take full responsibility for the final manuscript.

REFERENCES

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Rei Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Xudong Guo, Kaixuan Huang, Jiale Liu, Wenhui Fan, Natalia Vélez, Qingyun Wu, Huazheng Wang, Thomas L. Griffiths, and Mengdi Wang. Embodied LLM agents learn to cooperate in organized teams. In *Language Gamification – NeurIPS 2024 Workshop*, 2024. URL https://neurips.cc/virtual/2024/106577.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024.

Eric Hanchen Jiang, Guancheng Wan, Sophia Yin, Mengting Li, Yuchen Wu, Xiao Liang, Xinfeng Li, Yizhou Sun, Wei Wang, Kai-Wei Chang, and Ying Nian Wu. Dynamic generation of multi-LLM agents communication topologies with graph diffusion models. *arXiv preprint arXiv:2510.07799*, 2025.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large scale language model society. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Junzhi Li, Peng He, Qirui Ji, Wei Wang, Lixiang Liu, and Chuxiong Sun. Discovering efficient and explainable communication topologies for llm-based multi-agent systems via causal inference. *arXiv preprint arXiv:2608.12921*, 2026a.

Shiyuan Li, Yixin Liu, Qingsong Wen, Chengqi Zhang, and Shirui Pan. Assemble your crew: Automatic multi-agent communication topology design via autoregressive graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23142–23150, 2026b.

Shiyuan Li, Yixin Liu, Yu Zheng, Mei Li, Quoc Viet Hung Nguyen, and Shirui Pan. OFA-MAS: One-for-all multi-agent system topology design based on mixture-of-experts graph generative models. In *Proceedings of the ACM Web Conference 2026*, pp. 1333–1344, 2026c. doi: 10.1145/3774904.3792537. URL https://doi.org/10.1145/3774904.3792537.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. Dynamic LLM-agent network: An LLM-agent collaboration framework with agent team optimization. *arXiv preprint arXiv:2310.02170*, 2023.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*, 2021.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn search agents. In *International Conference on Learning Representations*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. AgentDropout: Dynamic agent elimination for token-efficient and high-performance LLM-based multi-agent collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 24013–24035. Association for Computational Linguistics, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. In *International Conference on Learning Representations*, 2025a.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-Designer: Architecting multi-agent communication topologies via graph neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b.

Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. CodeAgent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13643–13658. Association for Computational Linguistics, 2024a. doi: 10.18653/v1/2024.acl-long.737. URL https://aclanthology.org/2024.acl-long.737/.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö. Arik. Chain of agents: Large language models collaborating on long-context tasks. In *Advances in Neural Information Processing Systems*, volume 37, pp. 132208–132237, 2024b. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ee71a4b14ec26710b39ee6be113d7750-Abstract-Conference.html.

Zhen Zhang, Wanjing Zhou, Juncheng Li, Hao Fei, Jun Wen, and Wei Ji. RADAR: Redundancy-aware diffusion for multi-agent communication structure generation. *arXiv preprint arXiv:2605.09907*, 2026.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. GPTSwarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

A Algorithm Workflow

Algorithm 1 summarizes the complete training workflow of EdgeIG. Following the presentation style of graph-based topology learners, the procedure separates task-conditioned topology construction, multi-agent execution, fine-grained edge evaluation, and policy optimization. Only the trainable parameters of the topology generation policy are updated; the underlying LLM agents, the pretrained node encoder, and the decision agent remain frozen.

Algorithm 1: Training workflow of EdgeIG

Input: Training set $\mathcal{D}$; topology generation policy π_θ composed of a frozen node encoder, a GNN encoder (GATv2 followed by an MLP), and a learnable decoder parameterized by W; fixed LLM agents $\{v_i\}_{i=1}^N$; frozen decision agent π; number of sampled graphs K; coefficients β, γ, and λ; learning rate α.

/ Construct task-conditioned topology distribution */*
for each query-answer pair $(q, y) \in \mathcal{D}$ **do**
  for agent $i \in \{1, \dots, N\}$ **do**
    $x_i \leftarrow \mathrm{Encoder}(r_i \oplus q)$
  end for
  $\{z_i\}_{i=1}^N \leftarrow \mathrm{MLP}(\mathrm{GATv2}(\{x_i\}_{i=1}^N, A_{\mathrm{full}}))$
  $(P_\theta)_{ij} \leftarrow \begin{cases} \sigma(z_i W z_j^\top), & i < j, \\ 0, & \text{otherwise}, \end{cases} \quad \forall i, j.$
  Sample K graphs $G_k \sim \pi_\theta(\cdot \mid q)$ from the Bernoulli edge distributions.

  / Execute sampled graphs and evaluate realized edges */*
  for graph $k \in \{1, \dots, K\}$ **do**
    Execute the MAS under G_k in topological order for the configured communication rounds; obtain regular node outputs $\{o_{k,j}^+\}$ and final answer $\hat{y}_k$.
    $R_k \leftarrow \mathbb{I}[\hat{y}_k = y]$; record the regular-execution prompt-token cost T_k.
    for realized edge $e = (v_i, v_j) \in G_k$ **do**
      Remove only the message from v_i to v_j, keep all other inputs of v_j unchanged, and resample $o_{k,j}^-$.
      $s_{k,e}^+ \leftarrow \frac{1}{L} \sum_{m=1}^L \log \pi(y_m \mid q, o_{k,j}^+, y_{<m}), \quad s_{k,e}^- \leftarrow \frac{1}{L} \sum_{m=1}^L \log \pi(y_m \mid q, o_{k,j}^-, y_{<m})$.
      $r_{k,e}^{\mathrm{IG}} \leftarrow s_{k,e}^+ - s_{k,e}^-$.
    end for
    for realized edge $e \in G_k$ **do**
      $R_{k,e}^{\mathrm{IG}} \leftarrow r_{k,e}^{\mathrm{IG}} + \sum_{e' \in \mathcal{D}_k(e)} \gamma^{d(e,e')} r_{k,e'}^{\mathrm{IG}}$.
    end for
  end for

  / Compute token-aware graph-level advantage */*
  for graph $k \in \{1, \dots, K\}$ **do**
    $c_k \leftarrow T_k / \max_{1 \le l \le K} T_l, \qquad u_k \leftarrow R_k - \beta c_k$.
  end for
  $\mu \leftarrow \frac{1}{K} \sum_{l=1}^K u_l, \qquad A_k \leftarrow u_k - \mu, \forall k$.

  / Update EdgeIG topology generation policy parameters */*
  $\mathcal{L} \leftarrow -\frac{1}{K} \sum_{k=1}^K \left[A_k \log \pi_\theta(G_k \mid q) + \lambda \sum_{e \in G_k} R_{k,e}^{\mathrm{IG}} \log \pi_\theta(a_{k,e} = 1 \mid q) \right]$.
  $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}$.
end for

B Implementation Details

We set the generation temperature to 0.2 and limit each response to $1{,}000$ tokens. We use two GATv2 layers with four attention heads and a dropout rate of 0.1. We optimize the topology generation policy using Adam with a learning rate of 10^{-3}. The prompt-token cost coefficient, downstream discount factor, and edge-level reward weight are set to $\beta = 0.5$, $\gamma = 0.2$, and $\lambda = 1$, respectively. We fix the random seed to 42 and keep the LLM agents and decision agent frozen throughout training. During the first two warm-up updates, the edge-level term is disabled. We then activate the edge-level rewards and apply a $\tanh$ transformation with temperature 1 to the discounted returns before policy optimization. At inference time, edges with predicted probabilities of at least 0.5 are retained, and no reference answer, edge ablation, or additional reward-scoring call is required.

For HumanEval, multiple code implementations can satisfy the same task specification. We therefore use execution-based feedback instead of the likelihood of a single reference solution. Specifically, we evaluate the receiver's outputs obtained with and without each incoming message using the same set of test assertions. Let $s(o)$ denote the fraction of assertions passed by output o. The

edge reward is computed as

$$r_e = s(o_j^+) - s(o_j^-).$$

This reward measures the improvement in test performance when the message is retained and replaces the likelihood-based reward in the discounted edge-level return. For multi-round communication, the fine-grained edge reward is averaged across rounds before computing discounted returns.

C Case Study

Figure 5 compares representative cases of EdgeIG and AgentPrune on GSM8K and AQuA.

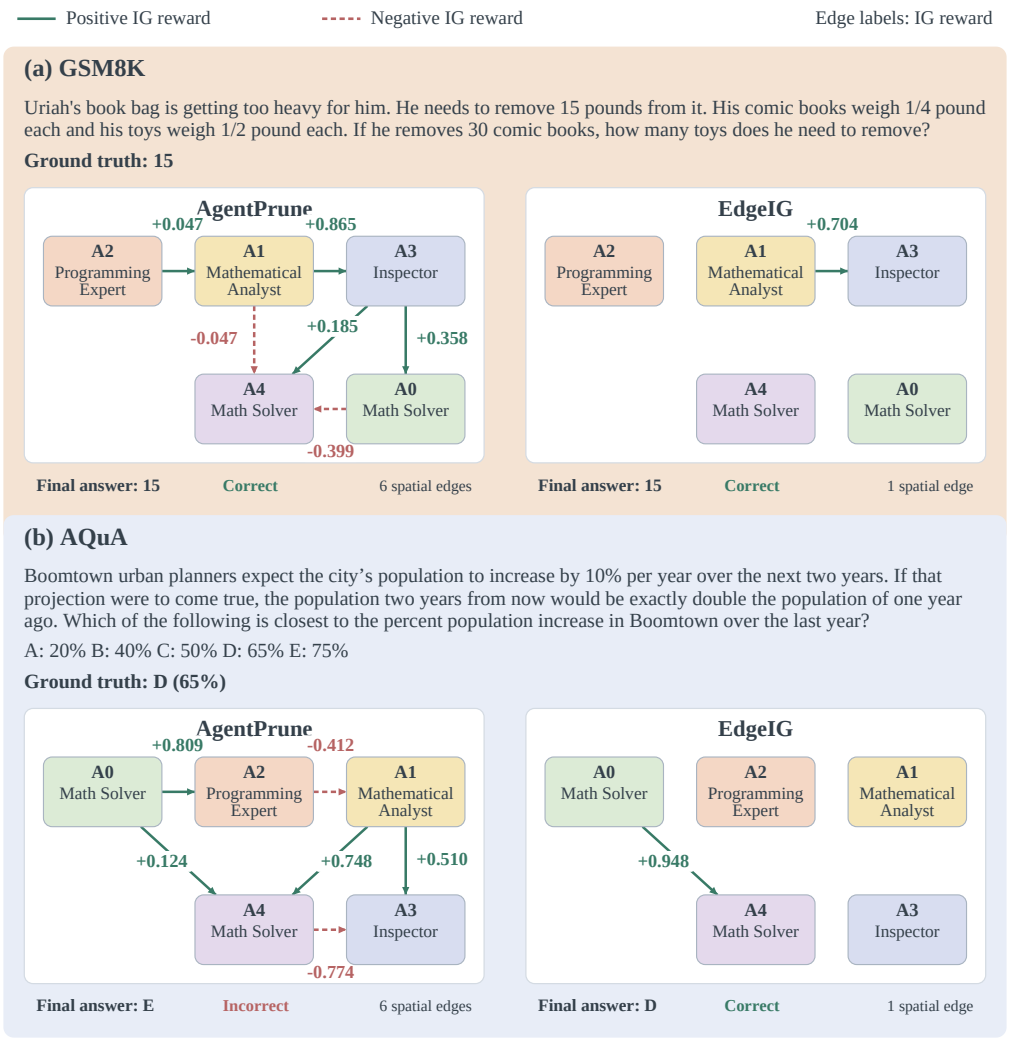


Figure 5: Case studies comparing the communication topologies of EdgeIG and AgentPrune on GSM8K and AQuA. Edge labels denote IG rewards; the decision agent is omitted.

Across both cases, EdgeIG retains only a single positively rewarded communication edge while achieving the correct answer, whereas AgentPrune preserves substantially more connections, including negatively rewarded interactions. These examples show that EdgeIG can selectively identify and remove redundant or potentially harmful communication, preserving task-relevant information flow with a much more compact topology.

D DATASET STATISTICS

We present the dataset statistics in Table 3, following the same experimental setup as G-Designer (Zhang et al., 2025b).

Table 3: Dataset descriptions and statistics.

Category	Dataset	Answer Type	Metric	#Test	License
General reasoning	MMLU	Multi-choice	Acc.	153	MIT License
Math reasoning	GSM8K	Number	Acc.	1,319	MIT License
	MultiArith	Number	Acc.	600	Unspecified
	SVAMP	Number	Acc.	1,000	MIT License
	AQuA	Multi-choice	Acc.	254	Apache-2.0
Code generation	HumanEval	Code	Pass@1	164	MIT License