GRAPHMAS: A SYSTEMATIC BENCHMARK OF MULTI-AGENT COORDINATION FOR GRAPH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM)-based multi-agent systems coordinate specialized reasoning through aggregation, interaction, and adaptive control, yet their potential for graph learning remains largely unexplored. Graph learning is a natural setting for such systems because useful evidence may arise from heterogeneous local, long-range, global structural, and semantic perspectives whose relevance varies across instances. Existing LLM-based graph learning approaches primarily rely on single-agent reasoning, while multi-agent coordination has been studied mainly in general reasoning settings. Consequently, it remains unclear whether multiple specialized agents can improve graph learning and how different coordination strategies should be designed and evaluated. To address this gap, we introduce GRAPHMAS, a systematic benchmark of multi-agent coordination for graph learning. GRAPHMAS builds a shared pool of graph reasoning specialists and organizes coordination along two dimensions, inter-agent interaction and runtime adaptivity, yielding four paradigms and seven representative coordination methods. Under a unified protocol, we evaluate these methods across seven text-attributed graphs, three domains, and two representative graph learning tasks: node classification and link prediction. Our study shows that heterogeneous graph perspectives are strongly complementary, and that coordinating specialists improves over both individual specialists and single-agent graph reasoning methods, with gains that arise from decomposing reasoning across specialists rather than from broader evidence access alone. However, richer inter-agent interaction does not reliably help, whereas instance-adaptive specialist selection yields the strongest accuracy–efficiency trade-off. We further show that the coordination policy can be learned over a fixed specialist pool and transfers to held-out graphs. GRAPHMAS therefore provides a controlled evaluation framework and empirical principles for understanding when and how multi-agent coordination benefits graph learning. The benchmark code is available at `https://anonymous.4open.science/r/GraphMAS`.

1 INTRODUCTION

LLM-based multi-agent systems coordinate multiple reasoning processes through role specialization, independent judgments, aggregation, and exchanges of intermediate outputs. Such systems have shown promising results in domains including software development and mathematical reasoning (Hong et al., 2024; Du et al., 2024), motivating a growing range of coordination strategies with different communication structures and agent participation rules (Tran et al., 2025). These strategies determine which agents contribute, what information they share, and when reasoning terminates. While multi-agent coordination has been increasingly studied in general reasoning settings, its potential for *graph learning* remains largely unexplored.

Graph learning provides a natural setting for studying multi-agent coordination because useful evidence can arise from heterogeneous graph perspectives whose relevance varies across instances. Predictions on text-attributed graphs may depend on local neighborhoods, longer-range structural relationships, graph-wide relevance, or node semantics. Classical graph learning has long recognized that different receptive fields and structural contexts can be useful for different nodes, for example by adapting neighborhood ranges (Xu et al., 2018) or combining local message passing with global attention (Rampášek et al., 2022). More recently, LLM-based graph learning methods have incorporated

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.

graph structure through instruction tuning, structured prompting, and active graph exploration (Chen et al., 2024; Sun et al., 2025; Liu et al., 2026; Sun et al., 2026). However, these approaches primarily rely on a *single reasoning agent* to integrate heterogeneous structural and semantic evidence within one trajectory. When different graph perspectives provide complementary signals, a single reasoning process may not exploit them equally well.

This observation motivates a different formulation: assigning distinct graph perspectives to specialized reasoning agents and coordinating their judgments for a shared graph learning task. Specialization, however, creates a new coordination problem. An evidence perspective that is informative for one instance may be unhelpful for another, and specialists may therefore exhibit both complementary strengths and conflicting predictions. As illustrated by the oracle gap in Figure 1, specialist success varies across instances, leaving considerable *coordination headroom*: instances missed by the best individual specialist can often be resolved by another graph perspective. The central question is therefore not simply whether more agents improve graph learning, but *when complementary specialists should participate and how their reasoning should be coordinated*.

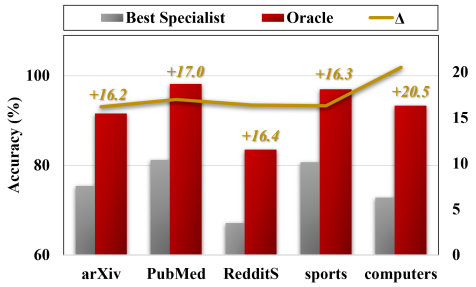

Figure 1: Performance gap between the best specialist and oracle on five LP tasks. The oracle is correct if any specialist is correct; the best specialist is selected per dataset. Exact results are reported in Appendix C.1.

Despite this opportunity, systematically understanding multi-agent coordination for graph learning remains challenging for three reasons. **1) There is no controlled benchmark for comparing coordination strategies in graph learning.** Existing LLM-based graph learning methods differ substantially in how they access and reason over graph information, while general multi-agent methods differ in agent participation, communication, and runtime control. Consequently, directly comparing complete systems can conflate the effects of the underlying graph reasoning capability with those of the coordination mechanism. A meaningful comparison therefore requires a shared specialist pool, consistent graph evidence interfaces, and controlled inference conditions. **2) The source of multi-agent gains is unclear.** Improvement from multiple agents may arise from several factors: access to different graph evidence, decomposition of reasoning across multiple trajectories, specialized role conditioning, or the coordination mechanism. Moreover, comparisons with individual specialists alone cannot establish an advantage over strong single-agent graph learning systems. Controlled analysis is therefore needed to determine whether multi-agent reasoning provides benefits beyond broader evidence access and to understand how graph evidence and specialist roles interact. **3) It remains unclear when additional coordination is actually useful.** Multi-agent systems often increase communication or invoke additional agents uniformly across instances. Yet the need for coordination is inherently instance dependent: specialists may already agree on easy cases, while disagreement may indicate that different graph perspectives support competing predictions. More interaction may therefore increase computation without necessarily improving accuracy. Understanding which specialists should participate and whether they should communicate is essential for both effective and efficient multi-agent graph learning.

To address these challenges, we introduce **GraphMAS**, a systematic benchmark of multi-agent coordination for graph learning. GraphMAS evaluates representative coordination approaches over a shared pool of graph reasoning specialists under consistent experimental conditions. The specialists capture four complementary graph perspectives—proximal neighborhoods, distal neighborhoods, graph-wide structural relevance, and semantic affinity—while sharing the same agentic backbone, retrieval budget, and LLM configuration. We organize coordination along two dimensions: **inter-agent interaction**, which distinguishes independent specialist reasoning from reasoning conditioned on other specialists' intermediate outputs; and **runtime adaptivity**, which distinguishes predefined coordination procedures from instance-dependent participation, communication, or stopping decisions. Crossing these dimensions yields four paradigms: Static Independent, Static Interactive, Dynamic Independent, and Dynamic Interactive. GraphMAS instantiates seven representative coordination methods across these paradigms and evaluates them alongside strong LLM reasoning, agentic graph learning, and individual specialist baselines on node classification and link prediction.

Our study reveals several consistent empirical patterns. First, heterogeneous graph perspectives exhibit substantial instance-dependent complementarity, creating clear headroom for coordination beyond any individual specialist. Second, controlled ablations show that agent specialization is most effective when graph evidence and reasoning roles are differentiated jointly, and distributing complementary perspectives across specialists provides gains beyond single-agent reasoning over the same evidence sources. Third, richer inter-agent interaction does not consistently improve performance or efficiency; instead, instance-adaptive specialist selection achieves a favorable accuracy–efficiency trade-off, with coordination methods differentiating most strongly under specialist disagreement. Finally, we show that the coordination policy itself can be learned while keeping the underlying graph specialists fixed, and that learned coordination can transfer to held-out graphs.

Our main **contributions** are summarized as follows:

- **A systematic benchmark for multi-agent coordination in graph learning.** We introduce GraphMAS, which studies how a shared pool of graph reasoning specialists can be coordinated for graph learning. We characterize coordination along inter-agent interaction and runtime adaptivity, yielding four paradigms instantiated by seven representative coordination methods.
- **A unified and comprehensive evaluation protocol.** GraphMAS evaluates multi-agent coordination methods across seven text-attributed graphs spanning three domains and two representative graph learning tasks. The benchmark measures predictive performance and compares against strong non-agentic, agentic, and individual-specialist baselines.
- **Controlled diagnostic analyses.** We isolate the contributions of evidence allocation, role specialization, and reasoning decomposition through controlled ablations, and analyze coordination behavior through oracle gaps, agreement-stratified evaluation, and cost accounting.
- **Extension to learned coordination.** We demonstrate that coordination can be optimized while keeping graph specialists fixed, establishing learned routing as a strong extension of the benchmark and showing that coordination behavior can generalize on unseen graphs.

2 Related Work

Multi-Agent LLM Systems. Multi-agent LLM systems coordinate agents through task decomposition, role specialization, and intermediate exchanges (Tran et al., 2025). Such coordination has improved software-generation quality in MetaGPT (Hong et al., 2024) and reasoning accuracy and factuality in Multi-Agent Debate (Du et al., 2024). Despite these shared benefits, existing systems adopt different mechanisms for information exchange and execution control. Some aggregate independently generated responses (Jiang et al., 2023; Li et al., 2024), while others incorporate one agent's output into another's reasoning. Participation and communication may follow predefined protocols or depend on model-generated control decisions. For example, DyLAN (Liu et al., 2024b) selects agents through response rankings, while Graph-of-Agents (Yun et al., 2026) uses response evaluations to select relevant agents and construct communication graphs. Inter-agent interaction and runtime adaptivity are thus distinct. Fixed workflows can support intermediate exchanges, while adaptive selection can preserve independent reasoning.

LLM-based Graph Learning. LLM-based graph learning requires integrating attribute semantics with relational structure (Jin et al., 2024a). GraphGPT (Tang et al., 2024), LLaGA (Chen et al., 2024), and GraphICL (Sun et al., 2025) address this challenge through graph instruction tuning, structure-aware projection, and structured prompting, respectively. Agentic methods further acquire graph evidence during inference. Graph-CoT (Jin et al., 2024b) interleaves reasoning with graph interaction, while GraphSearch (Liu et al., 2026) and AgentGL (Sun et al., 2026) use graph-aware retrieval and learned navigation, respectively. More recently, GLM (Huan et al., 2025) and MAAGL (Qu et al., 2026) introduced specialized-agent designs for graph question answering and node classification, respectively. Although these studies demonstrate the potential of multi-agent graph reasoning, their focus on specific architectures and task settings leaves systematic comparisons across coordination paradigms and graph learning tasks limited.

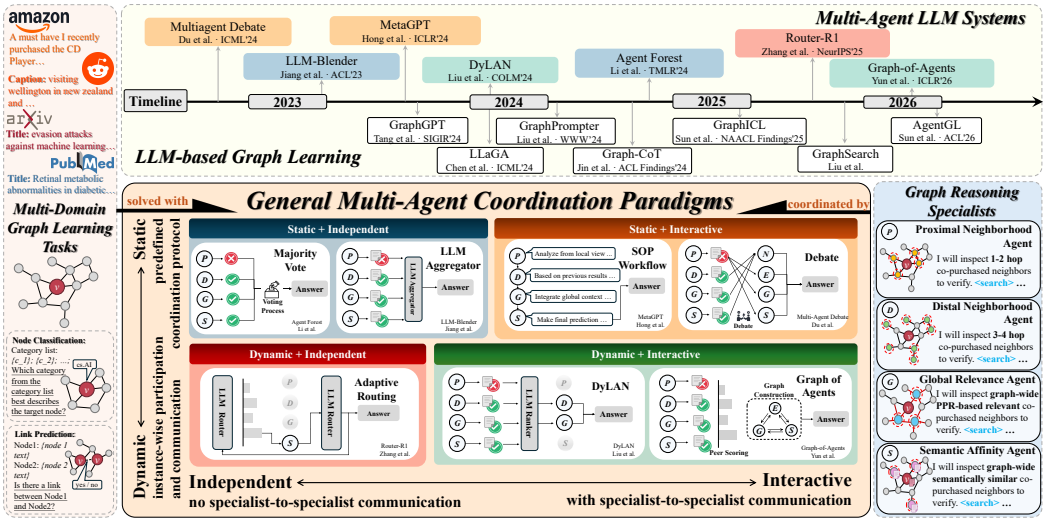

Figure 2: Overview of GraphMAS. A shared pool of graph reasoning specialists is coordinated across representative multi-agent paradigms organized by **inter-agent interaction** and **runtime adaptivity**. The timeline shows prior LLM-based graph learning and general multi-agent systems.

3 GRAPHMAS

GraphMAS provides a systematic framework for studying multi-agent coordination in graph learning. As illustrated in Figure 2, it builds on a shared pool of graph reasoning specialists with heterogeneous graph perspectives, detailed in Section 3.2. Section 3.3 then characterizes the coordination space along two dimensions, and Section 3.4 instantiates representative methods within this space.

3.1 PRELIMINARIES

We consider a text-attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where each node $v \in \mathcal{V}$ carries a textual attribute $t_v \in \mathcal{T}$, and study two representative tasks. In node classification (NC), the goal is to predict the label $y_v \in \mathcal{Y}$ of a target node v. In link prediction (LP), the goal is to predict whether an edge exists between a node pair (u, v). We write q for a generic prediction instance, a target node for NC or a node pair for LP, and y_q for its ground-truth label, which is never observable during inference.

3.2 GRAPH SPECIALISTS

Graph prediction may require local, distant, or semantically related evidence, with the most useful perspective varying by instance. Earlier LLM-based graph methods use fixed-hop subgraphs or predefined neighborhood templates (Liu et al., 2024a; Chen et al., 2024), while agentic methods let a single agent search across structural and semantic perspectives (Liu et al., 2026; Sun et al., 2026). Within one reasoning trajectory, however, complementary perspectives may receive uneven attention, making their contributions difficult to distinguish. GraphMAS assigns these perspectives to specialized agents and studies how to coordinate their predictions.

Shared agentic backbone. All specialists use the same retrieval pipeline, search budget, decoding configuration, and a frozen LLM backbone consistent with prior graph learning works (Liu et al., 2026). Given an instance q, a specialist alternates between reasoning, issuing graph queries, and inspecting the returned evidence through a `<think>` / `<search>` / `<information>` interaction loop until it produces a prediction or reaches its search budget. Each query is handled by a graph-aware retriever that constructs candidates according to the specialist's requested evidence scope and ranks them using structural and semantic relevance signals.

Specialized roles. We select specialists based on **structural relationships** and **semantic similarity**. Structural exploration separates proximal and distal neighborhoods to examine local and longer-range context independently, complemented by graph-wide relevance without a fixed hop constraint. Semantic exploration retrieves textually related nodes regardless of graph proximity. We assign these four views to separate specialists, building on the retrieval modes of prior agentic graph learning (Liu et al., 2026; Sun et al., 2026), and study how to coordinate their predictions:

- **Proximal Neighborhood Agent** focuses on relational evidence close to q. Nearby nodes often provide direct contextual signals through homophily, shared interactions, or local structural patterns. We instantiate this perspective using 1- and 2-hop retrieval.
- **Distal Neighborhood Agent** examines relational evidence beyond the immediate neighborhood, capturing dependencies that require a broader structural receptive field. We instantiate this perspective using 3- and 4-hop retrieval excluding the proximal neighbors.
- **Global Relevance Agent** targets structurally relevant nodes that need not fall within a fixed hop radius. We operationalize this perspective by ranking graph-wide candidates according to Personalized PageRank (Page et al., 1999) with respect to the target, thereby prioritizing nodes with high structural relevance even when they are topologically distant.
- **Semantic Affinity Agent** prioritizes attribute-level relevance over graph proximity, retrieving graph-wide nodes with semantically similar textual attributes, including informative examples that may be weakly connected or structurally distant.

For a node-classification instance $q = v$, each specialist applies its corresponding scope with respect to the target node v. For a link-prediction instance $q = (u, v)$, the same scope is applied to both endpoints, and the retrieved evidence is presented jointly to the specialist. Together, these agents enable broad graph exploration through distinct yet complementary structural and semantic perspectives.

Specialist interface. To support different coordination mechanisms, all specialists expose a common interface. A single invocation of specialist A_i receives a prediction instance q and an optional coordination context z, and returns

$$o = A_i(q, z) = (\hat{y}, r), \tag{1}$$

where $\hat{y}$ is the predicted label and r is a textual response summarizing the evidence and reasoning supporting the prediction. When $z = \varnothing$, the specialist reasons independently. Otherwise, z contains responses made available by the coordination mechanism. The specialist may use these responses to reconsider its prediction, while graph evidence remains restricted to its assigned perspective. This interface allows the same specialist pool to be used across heterogeneous coordination structures.

3.3 The Coordination Design Space

Given a specialist pool $\mathcal{A} = \{A_1, \ldots, A_M\}$, we represent the execution of a multi-agent system on an instance q as a typed coordination graph

$$\mathcal{H}(q) = \big(\mathcal{N}_q, \mathcal{E}_q^{\text{msg}}, \mathcal{E}_q^{\text{ctrl}}\big). \tag{2}$$

The node set contains both specialist and coordination invocations, $\mathcal{N}_q = \mathcal{N}_q^{\text{sp}} \cup \mathcal{N}_q^{\text{coord}}$. A specialist node $n \in \mathcal{N}_q^{\text{sp}}$ invokes a specialist $A_{a(n)}$ and produces $o_n = A_{a(n)}(q, z_n)$. A coordination node $n \in \mathcal{N}_q^{\text{coord}}$ invokes a graph-blind coordination module $C_{c(n)}$, such as a router, aggregator, or ranker, and produces $o_n = C_{c(n)}(q, z_n)$. Coordination modules may condition on the prediction instance, specialist descriptions, and responses made available to them, but cannot invoke retrieval on graphs.

The two edge types distinguish information flow from execution control. A message edge $(n' \to n) \in \mathcal{E}_q^{\text{msg}}$ indicates that $o_{n'}$ is included in the reasoning context z_n. A control edge $(n' \to n) \in \mathcal{E}_q^{\text{ctrl}}$ indicates that the output of n' affects whether or how n is executed, for example through routing, selection, pruning, or stopping. The final prediction is obtained by a fixed readout ρ over the realized execution, $\hat{y}_q = \rho(\mathcal{H}(q), \{o_n\}_{n \in \mathcal{N}_q})$. Any LLM-based aggregation or synthesis is represented explicitly as a coordination node rather than being absorbed into ρ.

Inter-agent interaction. A mechanism is *interactive* if the response of one specialist can enter the reasoning context of another specialist before the latter produces or revises its prediction. It is *independent* otherwise. Coordination modules may therefore aggregate, rank, or route specialist responses without making the specialists themselves interactive.

Runtime adaptivity. A mechanism is *static* when specialist participation and communication follow a predefined protocol shared across instances, possibly with fixed deterministic stopping rules. A mechanism is *dynamic* when model-generated control decisions determine part of the realized execution graph, such as which specialists are invoked or which communication links are formed.

Crossing the two dimensions yields four coordination paradigms: SI **Static Independent**, SX **Static Interactive**, DI **Dynamic Independent**, and DX **Dynamic Interactive**.

Table 1: Coordination methods instantiated in GraphMAS.

Paradigm	Method	Participation	Specialist communication	Runtime adaptation	Final decision
SI **Static Independent**	*Majority Vote*	all M	none	none	plurality vote
	LLM Aggregation	all M	none	none	LLM aggregator
SX **Static Interactive**	*SOP*	all M in fixed order	sequential	none	final specialist
	Debate	all M per round	all-to-all	none	plurality vote
DI **Dynamic Independent**	*Adaptive Routing*	selected sequentially	none	routing and stopping	router
DX **Dynamic Interactive**	*DyLAN*	all M, then top-k	all-to-all	agent pruning	agreement/ranker
	Graph of Agents	all M, then top-k	weighted directed graph	graph construction and pruning	meta-LLM pooling

3.4 Instantiated Coordination Methods

We instantiate seven representative coordination methods spanning the four paradigms over the same specialist pool, as illustrated in Figure 2. They differ in specialist participation, inter-specialist communication, runtime adaptation, and final decision-making. Table 1 summarizes these structures.

Static Independent. All specialists participate according to a fixed protocol and produce their responses without observing the reasoning contexts of one another.

SI **Majority Vote** follows the sampling-and-voting paradigm of Agent Forest (Li et al., 2024). Each specialist independently produces $o_i = A_i(q, \varnothing)$, and the final prediction is $\hat{y}_q = \mathrm{mode}\{\hat{y}_i\}_{i=1}^{M}$.

SI **LLM Aggregation** adapts the generative-fusion component of LLM-Blender (Jiang et al., 2023). Each specialist independently produces a response o_i, after which an aggregator node conditions on the original instance and the response set to generate the final prediction $\hat{y}_q = C_{\mathrm{agg}}\big(q, \{o_i\}_{i=1}^{M}\big)$.

Static Interactive. All specialists participate under a fixed communication structure, and subsequent specialist invocations may condition on responses produced earlier in the execution.

SX **SOP Workflow** is inspired by the standardized operating procedures of MetaGPT (Hong et al., 2024). Specialists act in the fixed order $\pi = (\textsc{Proximal}, \textsc{Distal}, \textsc{Global}, \textsc{Semantic})$. At step t, specialist π_t observes all preceding responses and produces $o_{\pi_t} = A_{\pi_t}(q, \{o_{\pi_s}\}_{s<t})$. The response of the final specialist is returned as the prediction.

SX **Multi-Agent Debate** adapts the iterative debate protocol of multiagent debate (Du et al., 2024). Specialists first respond independently, $o_i^{(0)} = A_i(q, \varnothing)$. At revision round t, each specialist observes its peers' responses and produces $o_i^{(t)} = A_i(q, \{o_j^{(t-1)}\}_{j \neq i})$ with the assigned graph perspective unchanged. Debate terminates upon unanimous agreement or after R revision rounds. If disagreement remains, the final prediction is determined by plurality over the specialists' latest predictions.

Dynamic Independent. Specialist participation is determined at runtime, while each invoked specialist reasons without access to other specialists' responses.

DI **Adaptive Routing** is conceptually inspired by the sequential routing formulation of Router-R1 (Zhang et al., 2025). At step t, the router conditions on the instance, specialist descriptions, and collected responses, and either selects an unqueried specialist a_t or stops with a final prediction. A selected specialist is invoked independently as $o_t = A_{a_t}(q, \varnothing)$. If the budget is exhausted, the same coordination model synthesizes the collected responses into the final answer.

Dynamic Interactive. Specialist participation and communication structure may be determined during execution, and active specialists can revise their predictions using other specialists' responses.

DX **DyLAN** (Liu et al., 2024b) adapts temporal feed-forward interaction and inference-time team reformation to the GraphMAS specialist pool. Specialists first produce independent responses, $o_i^{(0)} = A_i(q, \varnothing)$. If at least three agree, inference terminates. Otherwise, a listwise ranker selects $S = C_{\mathrm{rank}}(q, \{o_i^{(0)}\}_{i=1}^{M})$, with $|S| = k$. The retained specialists then revise their predictions using the other retained responses, $o_i^{(1)} = A_i(q, \{o_j^{(0)}\}_{j \in S \setminus \{i\}})$. The system terminates upon agreement. Otherwise, the ranker selects among the final retained responses.

DX **Graph of Agents** (Yun et al., 2026) is adapted to the GraphMAS specialist pool using response-conditioned graph construction and message passing. Specialists first produce independent responses, $o_i^{(0)} = A_i(q, \varnothing)$, and score the relevance of one another's responses. The top-k specialists form a weighted directed graph $G_q = (S_q, E_q, w_q)$, where $|S_q| = k$. They then refine their responses through forward and reverse message passing over the graph. Finally, a meta-agent uses mean pooling to synthesize the refined responses and relevance scores into the final prediction.

Table 2: Accuracy (%) of LLM reasoning, agentic baselines, graph specialists, and multi-agent methods, with all LLM-based methods using Qwen2.5-32B-Instruct. Avg. averages seven datasets; Avg. Rank averages ranks across 15 methods and 14 dataset–task settings. Top 2 counts settings reaching either of the two highest distinct accuracies. Best and second-best values are **bold** and underlined, respectively. See App. B for implementation details.

Method	Link Prediction								Node Classification								Overall	
	arXiv	products	PubMed	Reddit	sports	arXiv-23	computers	Avg.	arXiv	products	PubMed	Reddit	sports	arXiv-23	computers	Avg.	Avg. Rank	Top 2
LLM Reasoning																		
Chain-of-Thought	76.1	63.8	80.4	55.3	64.8	78.0	63.7	68.9	56.5	66.1	91.6	60.1	53.8	60.1	59.4	63.9	10.8	0
Agentic Methods																		
Search-o1	63.8	79.5	69.8	64.7	79.1	65.3	66.6	69.8	57.3	68.1	89.1	62.0	59.7	60.4	60.9	65.4	11.0	0
Graph-CoT	60.2	78.6	60.9	60.4	70.8	62.0	62.2	65.0	54.5	64.6	70.5	56.1	57.2	59.7	62.6	60.7	13.7	0
GraphSearch	70.3	79.5	75.2	62.7	74.5	70.5	68.3	71.6	57.6	71.7	89.8	67.4	59.8	55.8	**69.9**	67.4	9.4	1
Specialist Agents																		
Proximal Neighborhood	75.4	76.7	74.8	67.1	78.8	71.6	71.6	73.7	56.1	70.4	90.7	**68.3**	60.1	58.3	69.3	67.6	8.4	1
Distal Neighborhood	72.2	83.2	71.0	50.7	75.5	72.1	68.1	70.4	57.7	71.8	90.5	58.1	**62.8**	58.0	63.6	66.1	10.3	1
Global Relevance	68.2	82.7	76.4	58.7	78.6	69.9	72.8	72.5	57.5	72.8	91.1	65.9	59.6	57.5	69.3	67.7	8.8	0
Semantic Affinity	72.6	82.6	**81.2**	60.8	80.7	73.5	68.8	74.3	59.4	70.6	90.5	61.2	58.5	59.3	62.3	66.0	9.2	1
General Multi-Agent Methods																		
Static Independent																		
Majority Vote	74.6	84.3	81.0	61.0	84.8	71.5	74.5	76.0	58.7	73.2	91.6	67.1	61.4	59.9	68.8	68.7	5.8	3
LLM Aggregation	82.9	88.8	75.6	**69.0**	87.2	81.7	**74.9**	80.0	59.3	71.9	**92.2**	67.0	59.2	61.0	69.3	68.6	3.9	**7**
Static Interactive																		
SOP Workflow	73.5	86.5	71.3	62.3	81.8	75.7	71.0	74.6	56.6	**73.8**	90.1	66.1	60.4	57.4	69.2	67.7	8.1	1
Multi-Agent Debate	72.8	87.7	75.8	58.5	86.3	73.9	74.0	75.6	59.4	71.7	91.1	67.5	58.2	59.8	69.0	68.1	7.1	0
Dynamic Independent																		
Adaptive Routing	**83.5**	**89.8**	76.7	67.2	86.9	**83.3**	74.3	**80.2**	**60.1**	72.7	92.0	67.7	59.0	61.0	68.9	68.8	**3.3**	**7**
Dynamic Interactive																		
DyLAN	79.7	86.9	73.7	58.1	86.4	76.2	72.7	76.2	60.0	72.5	91.5	67.3	61.5	**61.3**	69.8	**69.1**	5.1	4
Graph of Agents	82.3	89.4	75.7	63.0	**87.4**	79.8	74.0	78.8	59.5	71.4	92.1	66.5	58.9	61.0	68.3	68.2	5.2	4

4 EXPERIMENTS

In this section, we conduct extensive experiments to systematically investigate multi-agent coordination for graph learning through the following research questions (**RQs**): **RQ1:** How effective is multi-agent graph learning compared with single-agent reasoning? **RQ2:** What drives the gains of multi-agent graph learning? **RQ3:** How do different coordination strategies compare in effectiveness and efficiency? **RQ4:** How does learning affect multi-agent graph learning?

Roadmap. We compare multi-agent and single-agent methods across backbones (§4.2; App. C.2, C.6), identify sources of multi-agent gains across coordination methods (§4.3; App. C.3–C.4), evaluate coordination effectiveness and efficiency under specialist disagreement (§4.4; App. C.5, C.7), and study learned coordination (§4.5). The appendix reports dataset statistics (App. A), implementation details (App. B), oracle gaps (App. C.1), and prompts (App. D).

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate on Node Classification (NC) and Link Prediction (LP) across **7** text-attributed graph (TAG) benchmarks spanning **3** domains: **1) citation networks**: ogbn-Arxiv (2020), PubMed (2008), and Arxiv-2023 (2024); **2) e-commerce networks**: ogbn-Products (2020), Amazon-Sports (2023), and Amazon-Computers (2023); and **3) social networks**: Reddit (2025). Dataset statistics and task construction are provided in Appendix A.

Baselines. We compare the multi-agent coordination methods instantiated in GraphMAS against representative baselines spanning direct LLM reasoning and LLM-based graph learning. **1) LLM reasoning.** *Chain-of-Thought (CoT)* (2022) predicts directly without access to graph information. **2) Agentic methods.** We include *Search-o1* (2025), a general-purpose agentic retrieval method, together with *Graph-CoT* (2024b) and *GraphSearch* (2026), which explicitly interact with graph information. Beyond these baselines, we evaluate all four GraphMAS specialists individually and systematically compare the seven multi-agent coordination methods introduced in Section 3.4: *Majority Vote* (2024), *LLM Aggregation* (2023), *SOP Workflow* (2024), *Multi-Agent Debate* (2024), *Adaptive Routing* (2025), *DyLAN* (2024b), and *Graph of Agents* (2026).

4.2 PERFORMANCE OF MULTI-AGENT GRAPH LEARNING (RQ1)

Table 2 compares direct LLM reasoning, agentic graph reasoning, individual graph specialists, and multi-agent coordination across seven datasets and both graph learning tasks.

⋆ ***Finding 1: Graph specialists are individually competitive and exhibit distinct strengths across datasets and tasks.*** The specialist agents already provide strong standalone graph reasoning. The best specialist-level average reaches **74.3%** on LP and **67.7%** on NC, compared with 71.6% and 67.4% for GraphSearch, respectively. More importantly, the strongest specialist varies across datasets

and tasks, indicating that different graph evidence perspectives capture complementary predictive signals rather than a universally superior graph view.

⋆ ***Finding 2: Coordinating graph specialists consistently improves over individual specialist reasoning, especially on link prediction.*** On LP, all seven multi-agent methods outperform the strongest individual specialist on average, with the best result improving from **74.3% to 80.2%**. On NC, six methods improve, and one matches the strongest specialist, raising the best average from **67.7% to 69.1%**. Multi-agent methods also achieve better overall ranks, ranging from **3.3 to 8.1**, compared with 8.4 for the strongest single-agent method. This shows that coordinating specialist judgments provides gains beyond individual specialists and conventional single-agent reasoning.

4.3 Sources of Multi-Agent Gains (RQ2)

We examine two sources of multi-agent gains: **specialist design**, which varies evidence and reasoning roles, and **reasoning decomposition**, which compares joint versus separate reasoning over complementary graph perspectives. To isolate these effects, we use the non-interactive *LLM Aggregation*. See Appendix C.3–C.4 for details and additional results.

Table 3: Multi-agent gains under *LLM Aggregation*. ✓/✗ denote specialized/shared evidence or roles. Results are average accuracy (%).

Evidence	Role	LP	NC
(a) Specialist Design			
✗	✗	73.0	66.0
✗	✓	69.9	65.4
✓	✗	73.7	66.1
✓	✓	**80.0**	**67.7**
(b) Reasoning decomposition			
Single-agent reasoning		74.6	65.6
Multi-agent reasoning		**80.0**	**67.7**

⋆ ***Finding 3: Agent-specific evidence and specialized roles are most effective when combined.*** We vary whether agents receive distinct graph evidence and specialized roles, while fixing the number of agents, backbone, and coordination mechanism. As shown in Table 3(a), specialized roles alone do not help with shared evidence, reducing LP accuracy from 73.0% to 69.9%, while agent-specific evidence with generic roles yields only marginal gains. Combining both gives the strongest results, reaching **80.0%** on LP and **67.7%** on NC. These results suggest that effective specialization arises from aligning distinct graph evidence with each agent's reasoning role, rather than from either component in isolation.

⋆ ***Finding 4: Reasoning decomposition provides gains beyond access to diverse graph evidence.*** We next examine whether reasoning decomposition provides benefits by comparing a multi-agent system with a single agent that receives the same four evidence sources jointly in one reasoning context. As shown in Table 3(b), single-agent reasoning achieves 74.6% on LP and 65.6% on NC, while multi-agent reasoning reaches **80.0%** and **67.7%**. Thus, the gains extend beyond broader evidence access to decomposing complementary perspectives across separate reasoning processes.

4.4 Effectiveness and Efficiency of Coordination (RQ3)

We compare the seven GraphMAS coordination methods along two questions: which coordination designs provide the strongest predictive and computational trade-offs, and under what conditions their performance differs most.

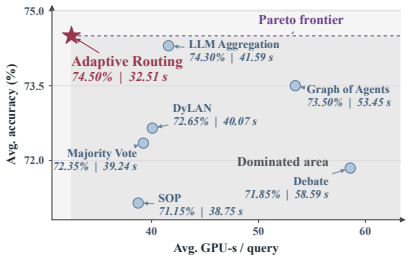

Figure 3: Accuracy–cost trade-offs averaged across NC and LP.

⋆ ***Finding 5: Instance-adaptive specialist selection is more effective than simply increasing inter-agent interaction.*** Despite sharing the same specialist pool, different coordination paradigms exhibit substantial performance differences. Among them, *Adaptive Routing* achieves the best overall average rank of **3.3** and the highest LP accuracy of **80.2%**, while *DyLAN* obtains the highest NC accuracy of **69.1%**. Interestingly, the simpler static-independent *LLM Aggregation* remains highly competitive, reaching **80.0%** on LP with an average rank of 3.9. In contrast, the static-interactive *SOP Workflow* and *Multi-Agent Debate* do not consistently outperform independent coordination.

Figure 3 further compares the accuracy–efficiency trade-off across coordination paradigms. Overall, *Adaptive Routing* is the only method on the Pareto frontier, achieving strong predictive performance with lower inference cost than methods that invoke the full specialist pool. By contrast, interaction-heavy methods such as *SOP Workflow*, *Multi-Agent Debate*, and *Graph of Agents* incur higher computation without corresponding accuracy improvements. Detailed task-specific results and statistics are reported in Appendix C.7.

⋆ ***Finding 6: Coordination strategies differentiate most strongly on instances with specialist disagreement.*** We stratify instances by agreement among the four initial specialist predictions. On LP, the accuracy range across methods grows from 4.4 points under unanimous agreement to 7.1 points in the $3/4$ group and 19.3 points in the $\leq 2/4$ group, where *Adaptive Routing* reaches 69.4% while *Multi-Agent Debate* obtains 50.1%. NC follows the same trend at a smaller scale, with the range increasing from 1.0 to 5.0 points (Appendix C.5). Thus, disagreement does not by itself guarantee coordination gains; rather, it is where the choice of coordination mechanism matters most.



Figure 4: LP accuracy (%) by initial specialist agreement. Parentheses show each group's share.

4.5 Extending GraphMAS to Learned Coordination (RQ4)

We further study whether **learning the coordination policy** improves multi-agent graph reasoning. Since *Adaptive Routing* offers the strongest accuracy–cost trade-off among training-free methods, we optimize its router with PPO (Schulman et al., 2017) while keeping the four graph specialists frozen. We compare the learned router with its training-free counterpart and trained baselines, including GNN, LLM-based, and agentic graph-learning baselines, under a shared train–transfer split. Separate NC and LP routers are trained on four datasets and evaluated on seen and three held-out graphs. Training details and baseline protocols are provided in Appendices B.2 and B.3.

Table 4: Accuracy (%) of learned graph-learning methods under a shared in-domain/transfer split. All LLM-based baselines use Qwen2.5-7B-Instruct; GraphMAS uses a 7B router with four frozen Qwen2.5-32B-Instruct specialists. In-domain/Transfer denote seen/unseen datasets during training. Our method is highlighted. Best and second-best results are **bold** and underlined.

Category	Method	Node Classification								Link Prediction							
		In-domain				*Transfer*			Avg.	*In-domain*				*Transfer*			Avg.
		arXiv	products	PubMed	computers	Reddit	sports	arXiv-23		arXiv	products	PubMed	computers	Reddit	sports	arXiv-23	
GNN	*GCN*	53.2	62.5	73.9	69.4	4.1	4.4	1.8	38.5	74.6	59.9	49.8	72.4	50.1	49.9	54.8	58.8
	RevGAT	47.1	48.4	81.5	68.8	3.9	17.5	0.7	38.3	73.6	57.1	58.6	69.9	49.9	49.3	49.9	58.3
	GraphSAGE	62.0	60.4	51.6	**75.2**	5.9	6.6	1.6	37.6	71.1	77.2	78.1	70.3	55.0	52.1	47.5	64.5
LLM-based GL	*GraphPrompter*	55.2	68.8	90.3	60.6	54.8	30.6	23.6	54.8	91.7	91.4	87.5	**79.9**	61.5	85.2	89.2	83.8
	LLaGA	64.2	56.4	86.5	72.1	7.8	0.3	18.8	43.7	79.5	68.1	81.9	65.6	59.9	60.2	62.5	68.2
Agentic GL	*AgentGL-7B-GRPO*	67.8	**75.3**	92.2	64.8	43.2	55.6	69.8	67.0	87.3	91.1	86.7	74.8	72.8	82.6	87.4	83.2
GraphMAS	*Routing* (Training-free)	59.1	72.9	90.7	68.3	67.5	59.9	60.5	68.4	83.5	89.4	76.2	72.8	64.7	87.4	83.3	79.6
	Routing (PPO)	**68.4**	73.9	**93.1**	72.2	**68.0**	**61.3**	**70.0**	**72.4**	**94.8**	**93.6**	**90.8**	76.2	**73.3**	**89.1**	**93.0**	**87.3**

⋆ ***Finding 7: Learning the coordination policy improves multi-agent graph learning with fixed specialists.*** PPO raises average LP accuracy from **79.6% to 87.3%**. On NC, the learned router achieves **72.4%**, compared with 68.4% for its training-free counterpart. Because the specialists remain frozen, these gains reflect improved coordination rather than changes in specialist capability.

⋆ ***Finding 8: Learned coordination generalizes beyond router-training graphs.*** The learned router improves on both training and held-out datasets. On LP, it raises average accuracy on Reddit, Sports, and arXiv-23 from **78.5% to 85.1%** and outperforms AgentGL-7B-GRPO on all seven datasets. These results show that the learned specialist-selection policy extends beyond the training graphs and provides a strong learned-coordination baseline for GraphMAS.

5 Conclusion

We presented GraphMAS, a controlled framework for studying multi-agent coordination in graph learning. Our results suggest that the value of multi-agent reasoning lies less in simply adding more agents or communication, and more in how complementary graph perspectives are decomposed, specialized, and selectively coordinated. Across tasks and coordination paradigms, carefully structured specialist reasoning can outperform both individual specialists and strong single-agent systems, while adaptive coordination can achieve competitive performance with substantially less specialist computation. Moreover, further gains from learning suggest that coordination itself can be a learnable component of graph reasoning systems. These findings position multi-agent graph learning as a problem of coordination design rather than agent scaling alone. We hope GraphMAS provides a common foundation for developing coordination mechanisms that more effectively exploit complementary graph evidence across diverse graph reasoning settings.

AI USE STATEMENT

In this work, we used generative AI tools solely for proofreading and improving the readability of the manuscript. We did not use generative AI to clean or reformat datasets, conduct qualitative or thematic analyses, or interpret results. The remaining disclosure categories do not apply to this work. We independently reviewed the final manuscript to ensure that it accurately reflects our intended meaning. We take full responsibility for all content in this work, including any text, claims, or artifacts produced with the assistance of generative AI.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. The benchmark code is available at the anonymous repository linked in the abstract. Section 3.2 describes the graph reasoning specialists and their shared interface, and Section 3.4 and Table 1 specify all seven coordination methods. Appendix A describes the datasets, preprocessing, evaluation sampling, and link prediction construction. Appendix B.1 reports the backbone models, decoding settings, retrieval configurations, and coordination hyperparameters; Appendix B.2 details learned-router training, with hyperparameters in Table 6; and Appendix B.3 describes the baseline protocols. Appendix D provides the core prompt templates for all specialists and coordination methods. All models are publicly available Hugging Face checkpoints, explicitly mentioned in Appendix B.

REFERENCES

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large language and graph assistant. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 7809–7823. PMLR, 2024. URL https://proceedings.mlr.press/v235/chen24bh.html.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Combining neural networks with personalized pagerank for classification on graphs. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1gL-2A9Ym.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: LLM-to-LM interpreter for enhanced text-attributed graph representation learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=RXFVcynVe1.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin

Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22118–22133, 2020.

Chengying Huan, Ziheng Meng, Yongchao Liu, Zhengyi Yang, Yun Zhu, Yue Yun, Shipeng Li, Rong Gu, Xiabao Wu, Haitao Zhang, Chuntao Hong, Shaonan Ma, Guihai Chen, and Chen Tian. Scaling graph chain-of-thought reasoning: A multi-agent framework with efficient LLM serving. *arXiv preprint arXiv:2511.01633*, 2025. URL https://arxiv.org/abs/2511.01633.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.792. URL https://aclanthology.org/2023.acl-long.792/.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 36 (12):8622–8642, 2024a. doi: 10.1109/TKDE.2024.3469578.

Bowen Jin, Chulin Xie, Jiawei Zhang, Kashob Kumar Roy, Yu Zhang, Zheng Li, Ruirui Li, Xianfeng Tang, Suhang Wang, Yu Meng, and Jiawei Han. Graph chain-of-thought: Augmenting large language models by reasoning on graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 163–184, Bangkok, Thailand, 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.11. URL https://aclanthology.org/2024.findings-acl.11/.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Guohao Li, Matthias Müller, Bernard Ghanem, and Vladlen Koltun. Training graph neural networks with 1000 layers. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6437–6449. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/li21o.html.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=bgzUSZ8aeg.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.276. URL https://aclanthology.org/2025.emnlp-main.276/.

Jiajin Liu, Yuanfu Sun, Dongzhe Fan, and Qiaoyu Tan. GraphSearch: Agentic search-augmented reasoning for zero-shot graph learning. *arXiv preprint arXiv:2601.08621*, 2026. URL https://arxiv.org/abs/2601.08621.

Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V. Chawla. Can we soft prompt llms for graph learning tasks? In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 481–484, New York, NY, USA, 2024a. Association for Computing Machinery. ISBN 9798400701726. doi: 10.1145/3589335.3651476. URL https://doi.org/10.1145/3589335.3651476.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic LLM-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*, 2024b. URL https://openreview.net/forum?id=XII0Wp1XA9.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, 1999.

Liang Qu, Jianxin Li, and Hua Wang. Multi-agent agentic graph learning via structural signatures. *arXiv preprint arXiv:2609.09565*, 2026. URL https://arxiv.org/abs/2609.09565.

Ladislav Rampášek, Mikhail Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://arxiv.org/abs/2205.12454.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL https://arxiv.org/abs/1506.02438.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008. doi: 10.1609/aimag.v29i3.2157.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16857–16867, 2020.

Yuanfu Sun, Zhengnan Ma, Yi Fang, Jing Ma, and Qiaoyu Tan. GraphICL: Unlocking graph learning potential in LLMs through structured prompt design. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 2440–2459, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-naacl.131. URL https://aclanthology.org/2025.findings-naacl.131/.

Yuanfu Sun, Kang Li, Dongzhe Fan, Jiajin Liu, and Qiaoyu Tan. AgentGL: Towards agentic graph learning with LLMs via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25313–25335, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1161. URL https://aclanthology.org/2026.acl-long.1161/.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. GraphGPT: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 491–500. Association for Computing Machinery, 2024. doi: 10.1145/3626772.3657775. URL https://doi.org/10.1145/3626772.3657775.

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. Multi-agent collaboration mechanisms: A survey of LLMs. *arXiv preprint arXiv:2501.06322*, 2025. URL https://arxiv.org/abs/2501.06322.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, 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*, volume 35, 2022. doi: 10.52202/068431-1800. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5453–5462. PMLR, 2018. URL https://proceedings.mlr.press/v80/xu18c.html.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, Weiwei Deng, Qi Zhang, Lichao Sun, Xing Xie, and Senzhang Wang. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264, 2023.

Hao Yan, Chaozhuo Li, Jun Yin, Zhigang Yu, Weihao Han, Mingzheng Li, Zhengxin Zeng, Hao Sun, and Senzhang Wang. When graph meets multimodal: Benchmarking and meditating on multimodal attributed graph learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 5842–5853. Association for Computing Machinery, 2025. doi: 10.1145/3711896.3737404.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Sukwon Yun, Jie Peng, Pingzhi Li, Wendong Fan, Jie Chen, James Zou, Guohao Li, and Tianlong Chen. Graph-of-Agents: A graph-based framework for multi-agent LLM collaboration. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=34cANdsHKV.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-R1: Teaching LLMs multi-round routing and aggregation via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4719. URL https://proceedings.nips.cc/paper_files/paper/2025/hash/ceaa137fce916aba5c65fceb1309088b-Abstract-Conference.html.

A Dataset Details

We evaluate on seven text-attributed graph (TAG) benchmarks spanning three domains: citation networks, e-commerce graphs, and social networks. Each node is associated with natural-language attributes, such as paper titles and abstracts, product descriptions, or social-media text, while edges encode the native relations of the corresponding graph. These benchmarks therefore require reasoning jointly over node semantics and graph structure. We consider both node classification (NC), which predicts the original multi-class node label, and link prediction (LP), which predicts whether an edge exists between a pair of nodes. We follow the dataset splits aligned with past research (Sun et al., 2025; 2026; Liu et al., 2026) and use the official test split when available. For Reddit, which originates from a multimodal graph benchmark, we remove image attributes and retain only textual node attributes, yielding a TAG setting. We provide the same processed graph and textual attributes to every method. Table 5 summarizes the statistics of TAG datasets used in our experiments.

Following prior agentic graph learning research (Liu et al., 2026; Sun et al., 2026), we subsample the evaluation sets to control inference cost. For each dataset–task pair, we randomly sample 1,000 instances from its designated test set, where an instance is a target node for NC and a node pair for LP. For LP, evaluation sets are balanced within each dataset, with 500 positive pairs, sampled from held-out edges, and 500 negative pairs, sampled uniformly from unconnected node pairs. All evaluation edges are removed from the graph exposed to specialists and baselines, so retrieval cannot directly reveal the queried link. When fewer than 1,000 eligible test instances are available, we evaluate on the full available set. The resulting sampled instances are fixed across all compared methods to ensure that performance and inference cost are measured on identical queries.

B Implementation Details and Hyperparameters

B.1 General and Training-Free Methods

Following prior work (Chen et al., 2024; Sun et al., 2026; 2025; Jin et al., 2024b; Liu et al., 2026), we use **accuracy** as the primary metric for all graph reasoning tasks. All results are reported from single

Table 5: Statistics of the seven text-attributed graphs used in GraphMAS. #Classes denotes the node-classification label space; link prediction is formulated as binary prediction.

Domain	Dataset	#Nodes	#Edges	#Classes
Citation Network	ogbn-Arxiv	169,343	1,166,245	40
	PubMed	19,717	44,338	3
	Arxiv-2023	46,198	78,548	40
E-commerce	ogbn-Products (subset)	54,025	74,420	47
	Amazon-Sports	173,055	1,946,555	13
	Amazon-Computers	87,229	808,310	10
Social Network	Reddit	13,037	566,160	20

runs conducted under the same experimental setup on 2 * NVIDIA A100 80 GB GPUs. We follow standard training and evaluation protocols for all baselines (Chen et al., 2024; Liu et al., 2024a).

For a controlled comparison across LLM-based graph learning methods and multi-agent systems, we use the same official Hugging Face checkpoint, `Qwen/Qwen2.5-32B-Instruct` (Yang et al., 2024), for training-free methods. We use a decoding temperature of 0.7 for all Qwen2.5 inference calls, including those made by baselines, graph specialists, and coordination modules.

For Proximal and Distal Neighborhood retrieval, we cap the candidate set at 200 nodes, augment each search query with the target node's title for candidate ranking, and return the top 3 nodes as evidence. Semantic affinity retrieval uses the SentenceTransformer model `all-mpnet-base-v2` (Reimers & Gurevych, 2019; Song et al., 2020). We precompute 768-dimensional corpus embeddings and rank candidates by cosine similarity between ℓ_2-normalized representations. For each query node, the semantic neighbor list contains the top 10 nodes after excluding the query node itself, of which the first 3 are provided to the agent. For the Global Relevance specialist, we approximate personalized PageRank (PPR) using APPNP (Gasteiger et al., 2019) with $\alpha = 0.15$ and $K = 30$. Here, α denotes the teleport probability to the personalization seed at each propagation step, and K is the number of propagation iterations. The resulting scores determine the PPR-ranked node lists.

Each specialist may issue at most 4 `<search>` calls per instance, after which it must commit to a prediction. Adaptive Routing may invoke at most M=4 specialist calls before final synthesis, matching the learned router. Multi-Agent Debate runs at most R=2 revision rounds. DyLAN and Graph of Agents both retain the top k=3 specialists after ranking and relevance scoring, respectively.

B.2 Training Details of the Learned Router

Routing formulation. We train only the router agent of *Adaptive Routing*, while keeping the four graph specialists frozen. Each episode corresponds to one prediction instance (a node for NC or a node pair for LP). At each turn, the router observes the target information, label space, previously collected specialist reports, and remaining call budget, and chooses either to invoke one specialist or to produce the final prediction. The router can make at most four specialist calls and must invoke at least one specialist before producing the final answer.

Reward. We optimize a terminal reward that jointly encourages prediction accuracy and efficient specialist usage:

$$r = \begin{cases} r_{\text{fmt}}, & \text{if the output format is invalid,} \\ (1-\alpha)R_{\text{out}} + \alpha R_{\text{cost}}, & \text{otherwise,} \end{cases} \tag{3}$$

where $R_{\text{out}} \in \{0, 1\}$ denotes answer correctness and

$$R_{\text{cost}} = R_{\text{out}} \cdot \frac{4 - n_{\text{calls}}}{3}.$$

We reward call efficiency only on correct trajectories, preventing the router from reducing cost through premature stopping at the expense of accuracy.

Training data. For each task, we sample 3,000 training instances per graph from the four in-domain graphs (arXiv, products, PubMed, computers), and hold out a further 256 instances per graph for validation; both are disjoint from the test instances. Dataset construction is detailed in Appendix A. To make RL training tractable, each specialist's response to each training instance is precomputed once and replayed; at evaluation time, the frozen specialists are invoked online.

Optimization. The router is initialized from `Qwen/Qwen2.5-7B-Instruct`, which also serves as the KL-reference model, and is optimized with PPO (Schulman et al., 2017) using a learned value head and GAE (Schulman et al., 2018). All specialists use Qwen2.5-32B-Instruct. NC and LP share the configuration in Table 6. Checkpoints are selected by accuracy on the in-domain validation split; the held-out transfer graphs are never used for training or model selection.

Table 6: Main hyperparameters for learned-router training.

Hyperparameter	**Value**
Router initialization / reference	Qwen2.5-7B-Instruct
Frozen specialists	Qwen2.5-32B-Instruct
Training iterations	150
Trajectories per iteration	128
Maximum specialist calls	4
Actor / critic learning rate	5×10^{-7} / 1×10^{-5}
PPO epochs	2
Mini-batch size	32
Policy / value clip ratio	0.2 / 0.2
KL coefficient	0.002
GAE γ / λ	1.0 / 0.95
Format reward r_{fmt}	-1
Cost weight α	0.1

B.3 Implementation of Learned Baselines

Shared protocol. Table 4 compares learned GraphMAS with conventional GNN, LLM-based graph-learning, and agentic baselines. All learned methods are trained on the four in-domain graphs (arXiv, products, PubMed, computers). *In-domain* results are measured on test instances of these graphs that are disjoint from training and validation; *transfer* results are measured on Reddit, sports, and arXiv-23, which are never seen during training or model selection and receive no fine-tuning or adaptation. All methods are evaluated on the same 1,000 test instances per dataset.

Backbone and compute. All LLM-based learned methods use Qwen2.5-7B-Instruct as the trainable backbone. GraphMAS additionally queries four frozen Qwen2.5-32B-Instruct specialists at inference time, so its trainable parameters match those of the baselines but its inference-time model capacity does not. We therefore treat the comparison against external baselines as a system-level reference, and isolate the effect of router training with the controlled comparison below.

Baseline Details. We include GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), and RevGAT (Li et al., 2021) as conventional graph-learning baselines. GraphPrompter (Liu et al., 2024a) and LLaGA (Chen et al., 2024) are trained and evaluated following their original procedures under the shared split, on the same processed graph structures and node texts used by GraphMAS. Since both predict labels in text, target label names are provided in the prompt and no label-space mapping is needed. We further compare with AgentGL-7B-GRPO (Sun et al., 2026) as a learned agentic graph-reasoning baseline. We retain its original training and inference procedure while using the same in-domain training graphs, held-out transfer graphs, and evaluation instances.

Controlled comparison: learned vs. training-free routing. Learned GraphMAS is compared with its training-free counterpart under identical conditions: the same Qwen2.5-7B-Instruct router backbone, the same four frozen specialists, the same retrieval settings, the same decoding temperature, and the same test instances.

Table 7: Accuracy (%) of the best individual specialist and the specialist oracle. Best Specialist selects the highest-performing specialist separately for each dataset, and Oracle counts an instance as correct if any specialist is correct. Δ denotes the Oracle–Best Specialist gap in percentage points. Avg. denotes the mean across the seven datasets.

	Link Prediction			**Node Classification**		
Dataset	Best Specialist	Oracle	Δ	Best Specialist	Oracle	Δ
arXiv	75.4	**91.6**	+16.2	59.4	**74.4**	+15.0
products	83.2	**96.0**	+12.8	72.8	**80.6**	+7.8
PubMed	81.2	**98.2**	+17.0	91.1	**94.8**	+3.7
Reddit	67.1	**83.5**	+16.4	68.3	**74.8**	+6.5
sports	80.7	**97.0**	+16.3	62.8	**75.3**	+12.5
arXiv-23	73.5	**92.4**	+18.9	59.3	**72.1**	+12.8
computers	72.8	**93.3**	+20.5	69.3	**81.0**	+11.7
Avg.	76.3	**93.1**	+16.9	69.0	**79.0**	+10.0

Table 8: Accuracy (%) of baseline variants. Avg. denotes the mean across the seven datasets.

Method	arXiv	products	PubMed	Reddit	sports	arXiv-23	computers	Avg.
Link Prediction								
GraphSearch-R	63.8	82.2	70.6	67.8	76.7	63.5	67.2	70.3
GoA-max	83.7	90.3	74.2	67.9	85.6	82.3	73.7	79.7
Node Classification								
GraphSearch-R	57.6	70.8	90.4	63.5	61.9	52.8	68.1	66.4
GoA-max	59.3	71.4	92.2	66.2	59.2	60.9	68.3	68.2

C Detailed Experiment Results

C.1 Specialist Complementarity and Oracle Gap

Table 7 details the performance of the best individual specialist and the specialist oracle across all datasets. The average gap of **16.9 percentage points** on link prediction and **10.0 percentage points** on node classification reveals substantial complementary strengths and motivates effective coordination among specialists.

C.2 Additional Results for Baseline Variants

GraphSearch (Liu et al., 2026) and Graph of Agents (Yun et al., 2026) each provide two variants. GraphSearch-R expands neighborhoods recursively, whereas GraphSearch-F flexibly retrieves across local and global neighborhoods. GoA-mean aggregates refined responses, whereas GoA-max selects the refined response of the most influential agent. Table 2 uses GraphSearch-F and GoA-mean. We ran these variants on the same datasets, and the results are detailed in Table 8.

GraphSearch-R achieves lower average accuracy than GraphSearch-F on both tasks and remains **below all evaluated multi-agent methods** in average accuracy. GoA-max improves average LP accuracy over GoA-mean from 78.8% to 79.7%, approaching LLM Aggregation (80.0%) and Adaptive Routing (80.2%), while both GoA variants achieve 68.2% on NC. However, **neither alternative** exceeds the strongest multi-agent method in average accuracy on either task.

C.3 Detailed Results for Specialist Design

We study specialist design through a controlled 2×2 ablation over **evidence allocation** (shared vs. agent-specific) and **role conditioning** (generic vs. specialized). Under shared evidence, all four agents receive the same graph evidence, collected from the four available perspectives within a fixed overall evidence budget. Under agent-specific evidence, the four agents receive different graph views,

with each agent restricted to one of the proximal neighborhood, distal neighborhood, graph-wide relevance, and semantic-affinity perspectives. Under generic roles, all agents follow the same general graph-reasoning instruction, whereas under specialized roles, each agent is explicitly conditioned to reason according to its assigned graph perspective. The number of agents, backbone, evidence budget, and coordination mechanism are held fixed across settings. Table 9 reports the specialist-design analysis across all seven coordination methods, using 100 randomly sampled evaluation instances per dataset and task.

Table 9: Per-dataset accuracy (%) under different specialist designs. Shared/Agent-specific denote common/specialist-specific evidence; Generic/Specialized denote general/specialist-specific role instructions. **Bold** and underlined values indicate the highest and second-highest accuracies within each method and column.

Method	Evidence	Role	Link Prediction								Node Classification							
			arXiv	products	PubMed	Reddit	sports	arXiv-23	computers	Avg.	arXiv	products	PubMed	Reddit	sports	arXiv-23	computers	Avg.
Majority Vote	Shared	Generic	82	75	74	58	78	74	71	73.1	63	68	**92**	70	55	51	63	66.0
	Shared	Specialized	**83**	77	70	**62**	76	80	66	73.4	64	66	91	**72**	55	51	62	65.9
	Agent-specific	Generic	82	80	78	56	83	**82**	76	76.7	68	69	89	70	**59**	54	64	67.6
	Agent-specific	Specialized	80	**85**	**80**	**62**	**85**	**82**	**80**	**79.1**	**69**	**70**	87	70	57	**57**	**65**	**67.9**
LLM Aggregation	Shared	Generic	77	76	72	67	69	82	68	73.0	61	68	88	70	57	**56**	62	66.0
	Shared	Specialized	69	76	60	66	73	72	73	69.9	62	67	**91**	67	**58**	53	60	65.4
	Agent-specific	Generic	**87**	77	70	62	67	**86**	67	73.7	65	68	89	68	55	**56**	62	66.1
	Agent-specific	Specialized	**87**	**85**	**77**	**68**	**89**	77	**77**	**80.0**	**66**	**69**	**91**	**71**	**58**	54	**65**	**67.7**
SOP Workflow	Shared	Generic	75	73	66	50	70	78	67	68.4	57	66	**89**	53	49	**49**	55	59.7
	Shared	Specialized	58	63	48	47	59	64	55	56.3	51	64	88	52	48	46	53	57.4
	Agent-specific	Generic	**82**	75	63	55	61	73	67	68.0	56	67	**89**	35	52	**49**	54	57.4
	Agent-specific	Specialized	72	**86**	**73**	**59**	**74**	**83**	**68**	**73.6**	**62**	**68**	88	**67**	**58**	47	**57**	**63.9**
Multi-Agent Debate	Shared	Generic	88	77	79	54	**85**	79	68	75.7	65	61	91	**73**	51	53	63	65.3
	Shared	Specialized	83	80	76	54	78	83	68	74.6	64	62	92	68	51	52	68	65.3
	Agent-specific	Generic	83	**82**	**82**	56	76	86	72	76.7	**70**	64	91	60	53	54	63	65.0
	Agent-specific	Specialized	**89**	81	78	**57**	**85**	**88**	**78**	**79.4**	**70**	**67**	**93**	**73**	**54**	**55**	**70**	**68.9**
Adaptive Routing	Shared	Generic	87	77	77	62	85	81	70	77.0	66	65	**91**	**71**	55	**55**	60	66.1
	Shared	Specialized	**89**	80	76	**70**	80	**89**	66	78.6	66	64	**91**	**71**	52	52	**66**	66.0
	Agent-specific	Generic	83	82	**83**	60	83	86	77	79.1	71	**69**	90	69	58	54	63	67.7
	Agent-specific	Specialized	86	**84**	76	66	**87**	80	**78**	**79.6**	**73**	68	90	**71**	**60**	52	61	**67.9**
DyLAN	Shared	Generic	**82**	75	73	57	78	77	68	72.9	64	65	**93**	**72**	57	**53**	63	66.7
	Shared	Specialized	**82**	77	69	**61**	75	77	67	72.6	66	67	90	71	54	**53**	66	66.7
	Agent-specific	Generic	81	78	**83**	54	80	**82**	**77**	**76.4**	**69**	**70**	89	**72**	60	**53**	64	68.1
	Agent-specific	Specialized	79	**82**	**83**	52	**86**	77	75	76.3	**69**	67	91	70	**61**	**53**	**67**	**68.3**
Graph of Agents	Shared	Generic	82	73	76	58	81	77	71	74.0	67	62	**92**	71	56	53	63	66.3
	Shared	Specialized	87	77	74	61	77	**85**	69	75.7	65	62	91	**72**	54	**55**	62	65.9
	Agent-specific	Generic	82	79	**85**	53	79	81	75	76.3	70	64	90	68	56	**55**	**64**	66.7
	Agent-specific	Specialized	**89**	**84**	78	**63**	**88**	77	**76**	**79.3**	**71**	**67**	90	69	**57**	53	62	**67.0**

C.4 Detailed Results for Reasoning Decomposition

Table 10 reports per-dataset accuracy of single-agent reasoning, where one agent receives all four evidence sources jointly, and multi-agent reasoning with LLM Aggregation, using the same 100 sampled instances per dataset and task.

C.5 Coordination under Specialist Disagreement

Table 11 details the performance of each coordination method across levels of initial specialist agreement on both link prediction and node classification. Agreement is measured by the largest number of specialists producing the same initial prediction. The results show that coordination has little effect under unanimous agreement, while differences grow as agreement decreases.

C.6 Additional Backbone Ablation

We further examine whether the observed multi-agent gains persist under different LLM backbones. In addition to the main Qwen2.5-32B setting, we rerun all inference calls with `Qwen/Qwen2.5-7B-Instruct` (Yang et al., 2024), a smaller model from the same family, and `zai-org/GLM-4-32B-0414` (GLM et al., 2024), a similarly sized model from a different family. Due to computational resource constraints, we use a compact subset of three datasets, including Reddit, Sports, and Computers, and randomly sample 100 instances per dataset and task.

Table 12 shows that the multi-agent advantage persists under both backbones. Within this ablation, *Adaptive Routing* achieves the **highest three-dataset average** for both NC and LP under both backbones. Compared with the strongest single-agent baseline, its gains are **6.4 and 7.3 points** with

Table 10: Per-dataset accuracy (%) of single-agent and multi-agent reasoning over the same four evidence sources. Single-agent reasoning receives all evidence sources jointly in one reasoning context; multi-agent reasoning uses LLM Aggregation. Avg. denotes the mean across the seven datasets. The best results within each column are shown in **bold**.

Method	arXiv	products	PubMed	Reddit	sports	arXiv-23	computers	Avg.
Link Prediction								
Single-agent reasoning	78	83	76	57	75	**83**	70	74.6
Multi-agent reasoning	**87**	**85**	**77**	**68**	**89**	77	**77**	**80.0**
Node Classification								
Single-agent reasoning	65	68	88	64	57	**54**	63	65.6
Multi-agent reasoning	**66**	**69**	**91**	**71**	**58**	**54**	**65**	**67.7**

Table 11: Accuracy (%) conditioned on agreement among the four graph specialists. Percentages in parentheses denote the share of instances in each agreement group. The best and second-best results within each group are shown in **bold** and underlined respectively.

	Link Prediction			**Node Classification**		
Method	4/4 (54.5%)	3/4 (31.4%)	≤ 2/4 (14.1%)	4/4 (67.5%)	3/4 (19.0%)	≤ 2/4 (13.5%)
Majority	**88.7**	68.9	53.9	**79.7**	49.7	39.5
LLM Agg.	**88.7**	**71.4**	66.5	**79.7**	49.4	39.1
SOP	84.3	64.6	56.6	78.7	49.5	36.1
Debate	**88.7**	64.3	50.1	**79.7**	47.7	38.5
Routing	**88.7**	71.1	**69.4**	**79.7**	50.0	39.7
DyLAN	87.9	65.9	55.3	79.5	**51.5**	**41.1**
GoA	88.6	69.7	63.7	79.5	51.3	38.9

Qwen2.5-7B, and **8.0 and 12.0 points** with GLM-4-32B on NC and LP, respectively. These results suggest that the benefits of coordinating graph specialists are not restricted to the backbone used.

Table 12: Backbone ablation on three datasets using 100 randomly sampled instances per dataset and task. Results report accuracy (%). Avg. is the mean across the three datasets. Best and second-best distinct values within each backbone and column are **bold** and underlined respectively.

Method	**Qwen2.5-7B**								**GLM-4-32B**							
	Link Prediction				**Node Classification**				**Link Prediction**				**Node Classification**			
	Reddit	sports	computers	Avg.	Reddit	sports	computers	Avg.	Reddit	sports	computers	Avg.	Reddit	sports	computers	Avg.
LLM Reasoning																
Chain-of-Thought	52	60	60	57.3	44	32	52	42.7	49	65	54	56.0	63	68	54	61.7
Agentic Graph Learning																
Graph-CoT	48	48	45	47.0	40	26	46	37.3	48	51	57	52.0	50	59	54	54.3
Search-o1	59	71	64	64.7	57	45	48	50.0	59	71	59	63.0	62	69	62	64.3
GraphSearch-F	65	70	63	66.0	52	48	51	50.3	52	56	58	55.3	68	65	67	66.7
General Multi-Agent Methods																
Majority Vote	55	76	**69**	66.7	56	52	58	55.3	54	**84**	65	67.7	68	73	71	70.7
LLM Aggregation	**67**	75	56	66.0	53	49	55	52.3	60	83	67	70.0	70	74	73	72.3
SOP Workflow	61	60	53	58.0	52	41	57	50.0	62	78	67	69.0	69	75	64	69.3
Multi-Agent Debate	54	70	56	60.0	57	41	58	52.0	54	82	56	64.0	70	72	72	71.3
DyLAN	53	73	60	62.0	57	**54**	58	56.3	60	73	66	66.3	65	70	67	67.3
Graph of Agents	62	78	66	68.7	54	51	**61**	55.3	54	71	62	62.3	**71**	73	69	71.0
Adaptive Routing	**67**	**84**	**69**	**73.3**	**58**	51	**61**	**56.7**	**68**	83	**74**	**75.0**	**71**	**78**	**75**	**74.7**

C.7 Computational Cost Results

We randomly sampled 100 evaluation instances and ran all seven coordination methods on this shared sample. Table 13 reports per-question costs, measured under consistent experimental settings across methods and datasets.

Specialist operations count specialist rollouts and standalone revision, message, or peer-judgment attempts. Each rollout counts as one operation, which may include multiple model requests.

LLM calls count individual prompts dispatched to the model, including all specialist and coordination requests; prompts within a batch are counted separately.

Input and output tokens are summed from the actual token IDs recorded by the inference engine. These include repeated input histories, intermediate generations, and discarded responses, but exclude retrieval encoder tokens.

GPU-seconds measure allocated GPU time: twice the elapsed wall time of the measured workload, divided by the number of questions. This includes inference, retrieval, internal queueing, and accounting overhead, but excludes startup, warmup, and waiting before GPU allocation. Precomputed corpus embedding costs are excluded.

Table 13: Per-question computational costs on node classification (NC) and link prediction (LP). GPU-s denotes allocated A100 GPU-seconds. The minimum within each task is shown in **bold**.

Method	Specialist ops	LLM calls	Input tokens	Output tokens	GPU-s
Node Classification					
Adaptive Routing	**1.80**	**7.12**	**6,836**	**787**	**21.28**
Majority Vote	4.00	8.93	9,395	1,364	33.53
LLM Aggregation	4.00	9.94	10,536	1,433	36.13
SOP Workflow	4.00	9.18	11,662	1,324	33.53
Multi-Agent Debate	5.32	11.93	13,823	1,842	42.57
DyLAN	4.18	9.24	9,625	1,415	34.89
Graph of Agents	12.00	15.92	16,780	2,189	48.32
Link Prediction					
Adaptive Routing	**3.35**	14.15	15,263	1,559	**43.73**
Majority Vote	4.00	11.84	14,121	1,648	44.94
LLM Aggregation	4.00	12.53	14,767	1,689	47.04
SOP Workflow	4.00	**11.14**	15,395	**1,533**	43.97
Multi-Agent Debate	6.96	19.49	26,140	2,809	74.60
DyLAN	4.14	11.70	**13,666**	1,649	45.25
Graph of Agents	12.00	17.89	19,944	2,383	58.57

D Prompt Templates

We provide the core prompt instructions used by GraphMAS for reproducibility. To improve readability, we omit repeated task inputs, candidate-label lists, and structured output schemas that are inserted mechanically at runtime. Blue text denotes method-specific content instantiated at inference time, such as specialist roles or peer responses. Majority Vote requires no additional LLM prompt and is therefore omitted.

All graph specialists share the same reasoning template and differ only in their assigned retrieval scope. Coordination methods either append method-specific instructions to this template or invoke separate coordination-only LLM calls.

D.1 Graph Reasoning Specialists

All four specialists share the same reasoning protocol and differ only in their retrieval scope. Table 14 summarizes the shared prompt and role-specific search instructions.

D.2 Static Coordination

Table 15 reports the prompts for LLM Aggregation, SOP Workflow, and Multi-Agent Debate. LLM Aggregation uses a separate coordinator, whereas SOP and Debate append peer information to the specialist prompt.

Table 14: Core prompt instructions for GraphMAS graph reasoning specialists. All specialists share the same reasoning protocol and differ in their admissible retrieval scope.

Component	Stage	Core Prompt Instruction
Graph Specialist	Reasoning	You are the specialist role, a reasoning assistant for graph prediction. You have access to ONE graph search scope: assigned scope. Reason inside `<think>...</think>`; use `<search>...</search>` when additional graph evidence is needed; after receiving `<information>...</information>`, reason over the new evidence before making the final prediction.
Proximal Neighborhood	Retrieval	Search the target node's 1-hop and 2-hop neighbors using `mode=local` and `hop=1|2`.
Distal Neighborhood	Retrieval	Search the target node's 3-hop and 4-hop neighbors using `mode=local` and `hop=3|4`.
Global Relevance	Retrieval	Search globally structure-relevant nodes selected by Personalized PageRank using `mode=global`.
Semantic Affinity	Retrieval	Search nodes whose attributes are semantically similar to the target node using `mode=attribute`.

Table 15: Core prompt instructions for static coordination methods. LLM Aggregation combines independent responses, while SOP and Debate expose specialists to intermediate peer judgments.

Method	Stage	Core Prompt Instruction
LLM Aggregation	Aggregation	You are the coordinator of a multi-agent graph prediction system and have no search tool. Aggregate the specialist judgments and target information. Prefer agreement supported by relevant evidence; weight specialists by confidence and evidence quality.
SOP Workflow	Handoff	Earlier specialists reported: preceding specialist reports. Use their findings as context, then add evidence from YOUR own scope and correct earlier conclusions when your evidence disagrees. The final specialist additionally weighs all previous findings and produces the pipeline's final prediction.
Multi-Agent Debate	Revision	You previously predicted previous prediction. Other specialists reported: peer responses. Re-examine the case from YOUR scope. You may search again within your scope; revise your answer if persuaded by another specialist, otherwise defend it with evidence from your scope.

D.3 Adaptive Routing

Table 16 shows the routing and final-synthesis prompts. The router observes previously collected specialist reports, while newly invoked specialists remain independent of the routing history.

D.4 DyLAN

Table 17 reports the ranking, revision, and final-selection prompts used in our DyLAN instantiation. Revision reuses the specialists' initially retrieved evidence and performs no additional graph retrieval.

D.5 Graph of Agents

Table 18 reports the prompts for peer scoring, bidirectional message passing, and final pooling. Message-passing calls reuse the specialists' initially retrieved graph evidence.

Table 16: Core prompts for Adaptive Routing. Specialists are consulted sequentially, while the router adapts participation and stopping to the current instance.

Method	Stage	Core Prompt Instruction
Adaptive Routing	Routing	You are the adaptive router of a multi-agent graph prediction system. Available specialists are: specialist roles and retrieval scopes. After reading the specialist reports collected so far, either CALL another specialist to gather or corroborate evidence, or ANSWER if the evidence is already sufficient. At most remaining calls additional specialist calls are allowed.
Adaptive Routing	Final synthesis	The routing budget is exhausted. Commit to a final prediction using the collected specialist reports: consulted specialist reports. Do not invoke graph retrieval.

Table 17: Core prompts for the DyLAN instantiation. Ranking and revision are triggered only when the initial specialist responses do not satisfy the early-agreement criterion.

Method	Stage	Core Prompt Instruction
DyLAN	Ranking	Compare the candidate specialist responses according to evidence relevance, specificity, logical consistency, output validity, robustness to noisy graph context, and absence of unsupported assumptions. Do not solve the task or produce a new prediction. Rank the provided responses and retain the top-k specialists.
DyLAN	Revision	You are the retained specialist role. Reconsider your previous response using your own specialist evidence and other retained responses. Preserve your assigned role and do not claim access to graph evidence outside your scope. Incorporate useful peer evidence and explicitly resolve disagreements.
DyLAN	Final selection	Compare the retained revised responses and select the single response best supported by its stated graph evidence. Do not create a new answer or combine responses.

Table 18: Core prompts for Graph of Agents, covering peer scoring, bidirectional message passing, and final pooling.

Method	Stage	Core Prompt Instruction
Graph of Agents	Peer scoring	As the judge specialist, evaluate the other selected specialists according to evidence relevance, reasoning consistency, specificity, robustness, and output validity. Assign non-negative relevance scores that sum to 1. Do not score yourself or produce a new prediction.
Graph of Agents	Source-to-target	As the target specialist, refine your response using messages from higher-relevance specialists. Treat peer messages as communicated conclusions rather than graph evidence you directly retrieved. Incorporate useful information and reject unsupported claims.
Graph of Agents	Target-to-source	As the source specialist, use the revised responses of lower-relevance specialists to finalize your own response. Preserve your original evidence scope and critically incorporate useful corrections or consensus.
Graph of Agents	Pooling	Produce the final prediction from refined specialist responses and relevance scores. Use relevance scores as guidance rather than proof of correctness, and compare the stated evidence and reasoning. Do not invoke graph tools or introduce new graph evidence.