OPENMAS-GCOM. A DIAGNOSTIC BENCHMARK FOR GRAPH-ENHANCED MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph-enhanced multi-agent systems (G-MAS) coordinate large language model agents through structured communication and role assignment, which determine information exchange and collaboration patterns. As G-MAS develops toward complex collaborative reasoning, comprehensive evaluation becomes increasingly important. However, existing benchmarks provide limited diagnostic evaluation of collaboration. They mainly evaluate final task performance, providing limited insights into how G-MAS approaches handle information integration, erroneous information propagation, and execution recovery during collaboration. To address this limitation, we propose OpenMAS-GCom, a comprehensive diagnostic benchmark for G-MAS evaluation. Specifically, (1) From the evaluation perspective, OpenMAS-GCom introduces a controlled intervention framework that represents G-MAS through collaboration units, communication links, shared intermediate information, and execution strategies. It enables systematic analysis of organizational components while keeping tasks, models, and resource settings fixed. (2) From the task perspective, OpenMAS-GCom introduces G-MAS-Complex, a new benchmark containing 400 multi-document tasks that require information integration, conflict resolution, and structured answer generation. OpenMAS-GCom covers 29 datasets across six domains and evaluates 17 representative configurations, including single-agent, ordinary multi-agent, and graph-enhanced multi-agent systems. Extensive experiments reveal that graph enhancement does not uniformly improve G-MAS performance. On G-MAS-Complex, the advantage of graph-enhanced approaches is mainly achieved by a small subset of configurations, with graph-based methods showing large performance gaps on increasingly difficult collaboration tasks. Moreover, methods with comparable initial accuracy exhibit substantially different robustness under corrupted information, revealing hidden differences in collaboration reliability beyond final task scores. Our code is available at https://anonymous.4open.science/r/OpenMAS-GCom-123456/.

1 INTRODUCTION

Large language models (LLMs) have enabled multi-agent systems (MAS) to solve complex tasks through collaboration among specialized agents. Recently, graph-enhanced multi-agent systems (G-MAS) have introduced communication graphs to explicitly model interactions among agents, where communication structures and role assignments determine information exchange and collaboration patterns. With the rapid development of G-MAS, a comprehensive benchmark is needed to evaluate and understand different collaboration designs.

However, existing G-MAS evaluations still leave two questions insufficiently resolved. First, identifying individual organizational contributions requires matched comparisons. Prior studies examine communication sparsity and information propagation (Li et al., 2024, Shen et al., 2025), while MAST analyzes failures in system design, agent coordination, and task verification (Cemri et al., 2026). When models, agent numbers, communication structures, and role assignments change together, final task scores cannot isolate their individual contributions. These analyses motivate a common protocol that measures contributions by changing one organizational component at a time.

Second, collaboration tasks should expose how organizations handle information dependencies and conflicting records. G-MAS methods are often evaluated on mathematical reasoning, question an-

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.

swering, and code generation (Zhang et al., 2025). MultiAgentBench evaluates collaboration and competition across interactive scenarios, while Collab-Overcooked examines joint task execution using process metrics (Zhu et al., 2025, Sun et al., 2025). To assess information integration, tasks should require agents to combine facts from multiple documents, resolve conflicting records, and return jointly correct output fields. Combining such tasks with controlled interventions enables evaluation of both final performance and the effects of individual organizational components.

To address these limitations, we propose OpenMAS-GCom, a comprehensive diagnostic benchmark for G-MAS evaluation. From the evaluation perspective, OpenMAS-GCom introduces a controlled intervention framework that represents G-MAS through four components, including collaboration units, communication links, shared intermediate information, and execution strategies. It enables systematic analysis of different components while keeping tasks, models, and resource settings fixed. From the task perspective, OpenMAS-GCom introduces G-MAS-Complex, a benchmark containing 400 multi-document tasks that evaluate collaboration through information integration, conflict resolution, and structured answer generation.

OpenMAS-GCom covers 29 datasets across six domains and evaluates 17 representative configurations, including single-agent systems, ordinary multi-agent systems, and graph-enhanced multi-agent systems. Experiments analyze organizational contributions and show how G-MAS-Complex distinguishes the performance of existing approaches.

Our Contributions. (1) Problem Formalization. We formalize G-MAS evaluation by representing systems through collaboration units, communication links, shared intermediate information, and execution strategies, providing a unified framework for analyzing different organizational components. (2) Comprehensive Benchmark. We propose OpenMAS-GCom, which integrates 29 datasets across six domains, 17 representative configurations, and G-MAS-Complex with 400 multi-document tasks for collaboration evaluation. (3) Analytical Insights. Through extensive experiments, we analyze the influence of different organizational components and summarize insights into G-MAS design. (4) Open-Source Library and Detailed Repository. We release an open-source benchmark library that integrates datasets, evaluation protocols, intervention modules, and experimental tools to facilitate reproducible research on G-MAS.

2 Problem Statement

OpenMAS-GCom records predictions and execution traces through a common pipeline, then measures task-score changes through paired interventions on agent organizations.

2.1 End-to-End Collaboration Pipeline

Step 1. Organization Construction. Given a task domain T with evaluation set $D_T = \{(x_i, y_i)\}_{i=1}^{N}$, method F constructs and executes an organization. Under configuration λ, construction produces $\mathcal{O}_i = F_{\text{build}}(x_i, \lambda)$. The organization may be fixed across inputs or constructed per task. We represent it as

$$\mathcal{O} = (V, E, S, \pi), \tag{1}$$

where V contains agents $V_A \subseteq V$, tools, and controllers, E specifies directed communication, S holds shared information and intermediate states, and π defines activation, routing, state updates, and termination. Each unit declares a role, local state, and input/output schemas.

Step 2. Collaborative Execution and Trace Recording.

For input x_i, the policy activates units sequentially, in parallel, or through repeated debate. Graph-based execution constrains transfers and activations by E. The runner records the trace

$$\tau_i = ((v_t, m_t, a_t))_{t=1}^{T_i}, \tag{2}$$

where v_t, m_t, and a_t denote the executed unit, available message state, and output or control action. The completed trace yields final output o_i and records token usage, calls, latency, executed units and relations, retries, and termination status.

Each method specifies prompts, parameters, tools, controllers, and shared-state access.

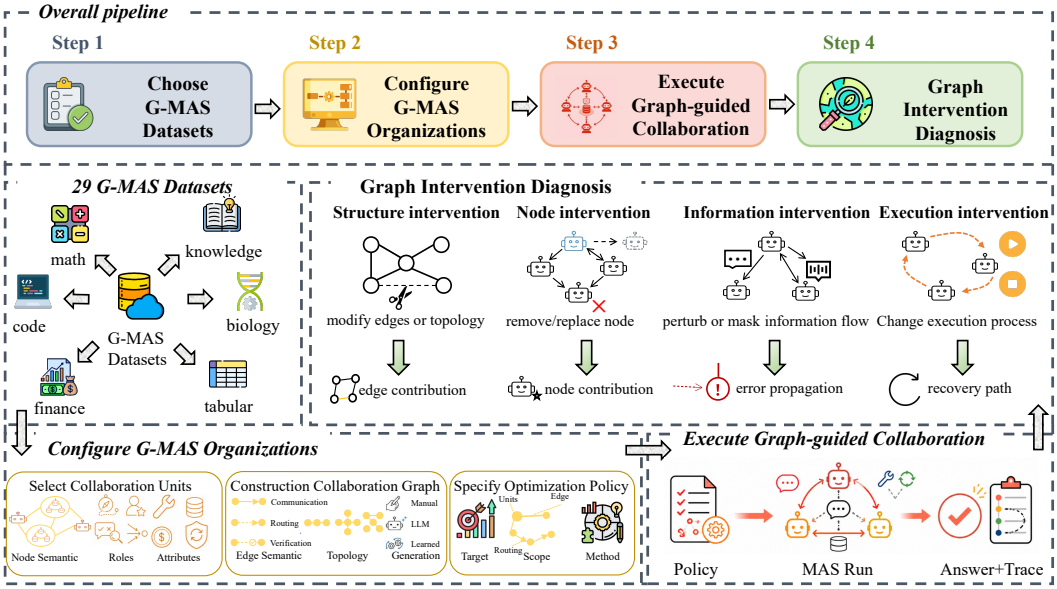

Figure 1. OpenMAS-GCom evaluation pipeline. The benchmark selects datasets, configures agent organizations, executes collaboration, and applies four interventions.

Step 3. Task Scoring.

The task score is

$$s(F, \mathcal{O}_T, T) = \frac{1}{N} \sum_{i=1}^{N} M_T(P(o_i), y_i), \tag{3}$$

where P parses the output, M_T is the task-specific metric, and $\mathcal{O}_T = \{\mathcal{O}_i\}_{i=1}^{N}$ collects the organizations used for the task.

2.2 Two-Stage Diagnostic Pipeline

The reference stage records predictions, traces, and costs. The intervention stage changes one organizational component and reruns the same tasks under fixed controls.

$$\mathcal{C} = \{D_T, \Theta, M_T, P\}, \tag{4}$$

Here, D_T, M_T, and P fix tasks, scoring, and parsing. Θ records the backend, prompts, decoding and routing settings, stopping rules, and per-task execution and communication budgets. For intervention r, operator I_r applies edit $\Delta\mathcal{O}_{r,i}$ to each reference organization.

$$\mathcal{O}'_{r,i} = I_r(\mathcal{O}_i, \Delta\mathcal{O}_{r,i}). \tag{5}$$

Scoring each reference and modified organization on matched tasks under $\mathcal{C}$ yields the intervention effect.

$$\Delta_r(F, T) = s(F, \mathcal{O}'_{r,T}, T) - s(F, \mathcal{O}_T, T). \tag{6}$$

Here $\mathcal{O}'_{r,T} = \{\mathcal{O}'_{r,i}\}_{i=1}^{N}$ collects modified organizations. A negative value indicates a score decrease. The analysis covers the full execution, including subsequent messages and actions.

3 Benchmark Design

OpenMAS-GCom integrates task selection, organization configuration, collaborative execution, and paired diagnosis (Figure 1).

3.1 Task and Dataset Ecosystem

Broad Task Suite. The benchmark covers 29 datasets across mathematics, knowledge, code generation, biomedical QA, finance, and tabular reasoning. Dataset loaders provide task inputs, reference answers, and metric metadata to the common runner.

G-MAS-Complex Construction. G-MAS-Complex combines the Test-A and Test-B sets of version v6.3.2 into 400 frozen multi-document tasks. Medium, Hard, Ultra, and Extreme each contain 100 tasks. Tasks require combining information across documents, resolving dependencies, and excluding conflicting records before returning a structured answer.

Difficulty increases with input length, records to reject, and the size of the task's Oracle graph. The reported tier statistics range from 11,036 to 37,061 input characters, 6 to 27 rejected records, 27 to 59 Oracle nodes, and 38 to 98 Oracle edges. Longer inputs increase the material to inspect, while additional records and dependencies increase selection and integration work. All tiers share six checked outputs comprising final-layer values, a mission result, a global checksum, a final-layer ranking, source-document identifiers, and critical checkpoints. Strict exact match requires all six to match the reference. Appendix A.1 gives tier statistics and Appendix B.2 explains scoring.

3.2 Algorithm Taxonomy

Definition of G-MAS. A graph-enhanced configuration has at least two agent units and an explicit directed graph that constrains routing, activation, verification, or execution order. Classification follows the graph exposed by the adapter and its runtime use. The graph may be fixed, generated per task, or learned. Equation 1 defines the representation.

Protocol-Specified Coordination. These methods follow prescribed interaction procedures. DeepSeek and Qwen are single-agent references (DeepSeek-AI, 2025, Qwen Team, 2025). AutoGen and MADebate use configured conversation or debate procedures (Wu et al., 2023, Du et al., 2023). This category also includes fixed graph organizations with predefined communication edges.

Task-Adaptive Orchestration. These methods adapt agent selection, model routing, or other task-level choices. DyLAN studies dynamic team selection (Liu et al., 2024), and MASRouter learns model routing (Yue et al., 2025). GraphSearch and GraphR1 apply graph reasoning or retrieval to task records (Liu et al., 2026, Luo et al., 2026). Task graphs represent records and relations. Communication graphs specify the interactions implemented by the adapter.

Communication-Topology Design. Here, the agent communication organization is constructed, optimized, or generated. GPTSwarm optimizes graph connectivity with prompts (Zhuge et al., 2024). GDesigner generates task-dependent graphs (Zhang et al., 2025), ARGDes generates roles and links (Li et al., 2026), and GoAgent generates group communication topologies (Chen et al., 2026). The learned structure in this category specifies communication among agents.

These learning categories are orthogonal to the execution-oriented adapter families in Table 4. Appendix B.4 specifies the fields for graph construction, runtime use, roles, verification, and recovery that describe both axes.

3.3 Graph Intervention Protocols

Each protocol changes one organizational factor under the controls in Equation 4.

Structure Intervention. The structure protocol swaps directed-edge destinations while preserving agents, in/out-degrees, edge count, and communication budget. It rejects self-loops, duplicate edges, and invalid role connections. Scores, calls, and messages measure performance and resource use.

Node Intervention. The node protocol removes selected units and incident edges. No critic and No specialist target different roles, while Expert only retains the expert-induced subgraph. Scores measure each variant's effect.

Information Intervention. The information protocol replaces eligible upstream records or candidates with format-valid incorrect content. These items follow the existing graph and routing rules, and final-answer accuracy measures the effect.

Execution Intervention. The execution protocol disables workers and resumes from the last valid shared state using configured retry, fallback, or rerouting. Increasing failures tests task completion under disruption. Appendix A.2 lists settings and controls.

4 Experiments and Analyses

Three questions organize the evaluation. Task utility covers Q1 to Q3. Organizational sensitivity covers Q4 to Q6. Failure robustness and resource use covers Q7 and Q8. The following experiments examine each group through specific comparisons. **Q1 (Utility)**. Which evaluated configurations improve task scores over single-agent baselines? **Q2 (Competitiveness)**. How do the best evaluated graph-enhanced configurations compare with ordinary multi-agent systems? **Q3 (Effectiveness)**. How do existing G-MAS methods perform across different task domains? **Q4 (Structure Intervention)**. Does topology matter when the degree sequence and communication budget limits are fixed? **Q5 (Unit Intervention)**. How much do role and collaboration-unit removals affect performance? **Q6 (Information Intervention)**. How does accuracy change as the fraction of incorrect upstream items increases? **Q7 (Execution Intervention)**. How robust are G-MAS executions as the number of failed workers increases under recovery? **Q8 (Efficiency)**. How efficient are G-MAS methods?

4.1 Experimental Setup

Datasets. We evaluate seven categories. (1) Mathematics uses GSM8K and AQuA (Cobbe et al., 2021, Ling et al., 2017). (2) Knowledge uses MMLU-Pro and StrategyQA (Wang et al., 2024, Geva et al., 2021). (3) Code uses HumanEval and LiveCodeBench-v6 (Chen et al., 2021a, Jain et al., 2024). (4) Biomedical QA uses MedQA and MedMCQA (Jin et al., 2021, Pal et al., 2022). (5) Finance uses ConvFinQA and FinQA (Chen et al., 2022, 2021b). (6) Tabular reasoning uses TabFact and WikiTableQuestions (WikiTableQ in Figure 4) (Chen et al., 2020, Pasupat & Liang, 2015). (7) G-MAS-Complex contains 400 frozen multi-document tasks in four tiers. Appendix A.1 describes the complete task collection and its metrics.

Baselines. We evaluate 17 configurations on the 29 public datasets and G-MAS-Complex through a common interface. Appendices A.2, B.5, and A.9 describe baseline implementations, validation records, and complete dataset results, respectively.

Execution settings. The broad suite follows fixed dataset-specific evaluation lists. The 13 datasets with repeated-sampling analysis use three matched samples of 500 instances, shared by all methods. Each configuration evaluates the same 400 frozen G-MAS-Complex tasks. Reference and intervention runs use matched task identifiers, input order, backend settings, prompt templates, parsers, routing policies, and runtime limits. The audit script checks these controls before aggregation. Appendices A.6 and A.7 give the recorded fields and intervention settings.

Scores and variation. The broad suite retains dataset-specific metrics, and G-MAS-Complex uses Strict EM. Table 1 reports central scores. Appendix A.8 distinguishes sample-set variation, run variation, and binomial standard errors. Rankings in Q1 to Q3 describe displayed central values. Intervention analyses report paired score changes and reference-score retention, while efficiency compares observed scores and resource use.

4.2 Utility of Multi-Agent Collaboration (Q1)

To answer Q1, Table 1 compares all 17 configurations on one dataset from each broad domain and G-MAS-Complex. Graph-enhanced configurations have the highest central score on 3 of these 7 datasets, single-agent configurations on 2, and ordinary multi-agent configurations on 2. Qwen leads on HumanEval and ConvFinQA, while DyLAN leads on MMLU-Pro and TabFact. The full dataset results appear in Appendix A.9. Our Conclusion (C1). Central-score advantages vary across tasks and configuration groups.

4.3 Competitiveness of Graph-Enhanced MAS (Q2)

To answer Q2, Figure 2 displays group-wise maxima over 3 ordinary and 12 graph-enhanced configurations. In each broad domain, we first take the unweighted mean of each configuration's dataset-level reported scores, then select the highest configuration within each group. For G-MAS-Complex,

Table 1. Central scores (%) for all 17 methods on six public datasets and G-MAS-Complex. Bold and underlined values mark the best and second-best scores. Complete results and uncertainty terms are in Appendices A.9 and A.8.

Method	Math GSM8K	Knowledge MMLU-Pro	Code HumanEval	Biomedical MedQA	Finance ConvFinQA	Tabular TabFact	Complex G-MAS-Complex
DeepSeek	<u>93.71</u>	73.07	84.76	88.45	60.13	92.00	26.75
Qwen	59.44	60.93	**90.24**	78.95	**61.81**	81.40	13.25
AutoGen	92.87	71.80	81.71	89.08	60.74	91.80	23.50
MADebate	92.87	71.87	83.54	89.08	61.01	91.47	14.00
DyLAN	93.33	**73.87**	85.98	88.77	60.60	**92.53**	4.50
GPTSwarm	93.56	70.73	84.76	89.63	60.67	92.20	15.50
GDesigner	93.63	71.87	85.37	88.45	<u>61.54</u>	91.73	14.50
SparseCT	**93.86**	71.33	84.76	89.24	60.81	91.93	15.25
GraphSearch	93.56	71.60	81.10	<u>89.71</u>	60.20	92.00	14.00
GraphR1	92.87	70.53	86.59	88.92	60.67	91.40	10.75
R-GFM	93.18	71.53	89.02	89.16	60.40	92.13	2.25
BigMAS	93.18	72.20	84.76	**89.87**	60.40	<u>92.33</u>	19.25
MASRouter	<u>93.71</u>	72.53	76.83	89.24	60.20	91.40	10.75
GoAgent	92.87	70.33	79.88	89.40	60.74	<u>92.33</u>	**50.25**
VeriMap	93.63	<u>73.53</u>	78.66	88.77	61.28	92.07	<u>29.50</u>
ARGDes	93.63	71.47	<u>89.63</u>	88.77	60.60	92.07	13.25
EIB	93.56	72.47	83.54	89.16	61.34	91.40	14.00

we select the highest task accuracy within each group and tier. The selected configuration can change across domains and tiers, so the comparison describes the best displayed score in each setting.

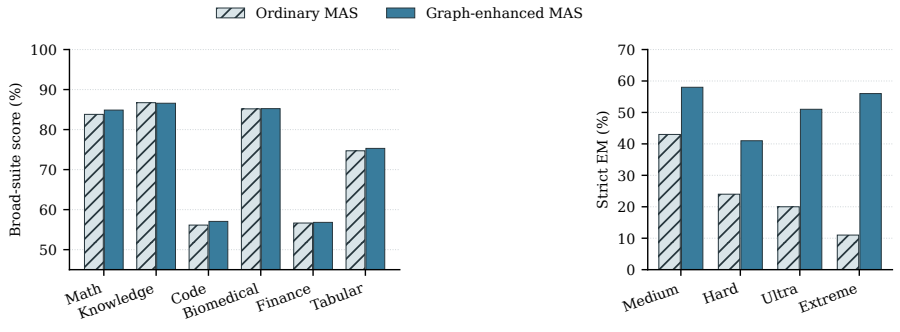

Figure 2. Group-wise maximum reported scores over evaluated Ordinary MAS and Graph-enhanced MAS configurations. Broad-domain bars use the highest per-configuration unweighted dataset macro-average within each group. Tier bars use the highest task accuracy within each group and tier. The maximizing configuration is selected separately in each setting.

On the broad suite, graph-enhanced group maxima lead in five of six domains. In Knowledge, the best ordinary-MAS macro-score is higher by 0.14 points. On the four G-MAS-Complex tiers, the plotted margins are 15, 17, 31, and 45%. Across all 400 tasks, GoAgent and VeriMap (Chen et al., 2026, Xu et al., 2025) score above the single-agent DeepSeek reference, while the other ten graph-enhanced configurations score below it. Our Conclusion (C2). GoAgent and VeriMap achieve the highest complex-task scores in the graph-enhanced group.

4.4 Performance Across Task Domains (Q3)

To answer Q3, Figure 3 compares four graph-enhanced configurations across six domains.

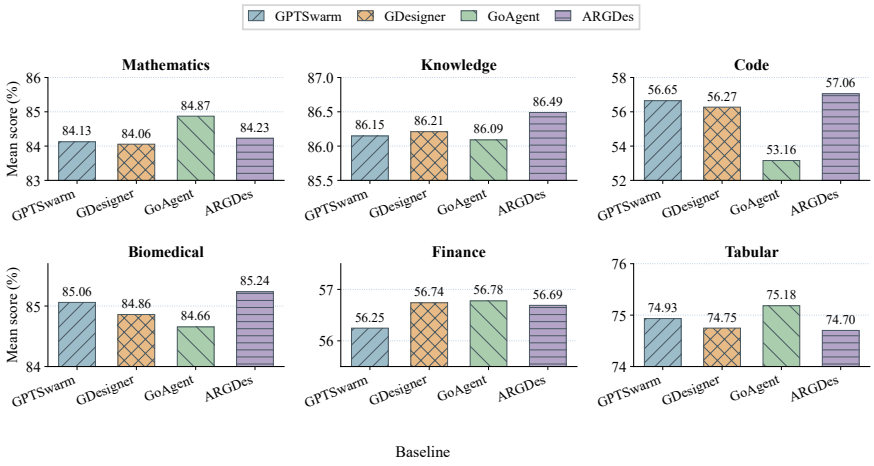

Figure 3. Domain means over the datasets in each domain, with independent panel scales. Each bar averages the corresponding central scores in Appendix A.9 with equal dataset weights.

Code ranges from 53.16% to 57.06%, giving the largest central-score spread of 3.90%. The spreads are 0.82 on Mathematics, 0.58 on Biomedical QA, 0.53 on Finance, 0.48 on Tabular, and 0.40 on Knowledge. Appendix A.1 reports dataset-level variation. Our Conclusion (C3). Displayed central-score dispersion varies substantially across task domains.

4.5 Sensitivity to Communication Structure (Q4)

To answer Q4, we rewire 10%, 30%, 50%, and 70% of communication edges while fixing agents, roles, degrees, edge count, models, prompts, routing policy, and budget limits. Figure 4(a) reports the five datasets separately. Their mean falls from 83.18% to 81.21%, 79.54%, 77.21%, and 74.99%. At 70% rewiring, the loss is 8.19%, leaving 90.15% of the reference score. Our Conclusion (C4). Degree-preserving rewiring reduces scores under the specified protocol.

4.6 Sensitivity to Collaboration-Unit Removal (Q5)

To answer Q5, we use GDesigner and compare Base with No critic, No specialist, and Expert only. The first two remove the selected role nodes and incident edges. Expert only keeps the expert-induced subgraph. The five-dataset mean falls from 83.18% to 81.30%, 78.30%, and 75.59%, giving losses of 1.88, 4.89, and 7.59%. Specialist removal incurs 2.59 times the mean loss of critic removal, while MMLU-Pro improves after critic removal. Our Conclusion (C5). Role removal produces asymmetric changes, with the largest loss under Expert only.

4.7 Robustness to Incorrect Messages (Q6)

To answer Q6, we replace eligible upstream records with format-valid incorrect content at rates of 0%, 20%, 40%, 60%, and 80%, keeping graph and execution controls fixed. Panel (c) reports a five-dataset mean over the tasks listed in panels (a) and (b). MAD falls from 82.81% to 44.41%, and VeriMap from 82.30% to 54.59%. At 80%, they retain 53.63% and 66.33% of their reference scores. VeriMap moves from 0.50 points below MAD to 10.18 points above it. Our Conclusion (C6). Similar reference scores conceal different corruption responses.

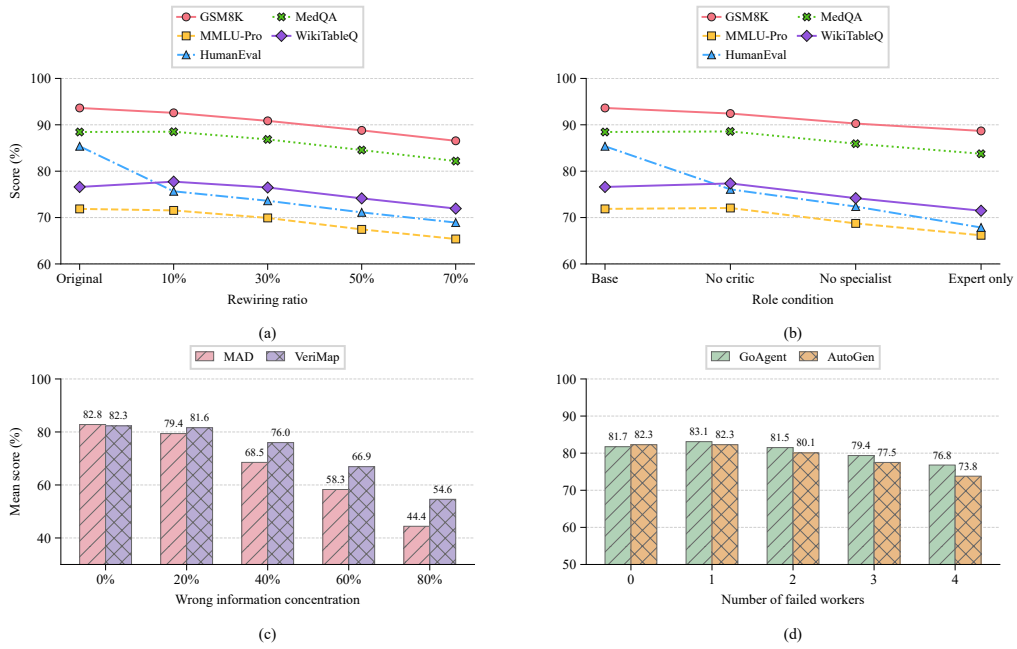

Figure 4. Responses to four organizational interventions. Panels (a) and (b) use GDesigner on GSM8K, MMLU-Pro, HumanEval, MedQA, and WikiTableQuestions (WikiTableQ). Panel (c) compares MADebate (MAD) with VeriMap, and (d) compares GoAgent with AutoGen. Appendix A.4.1 specifies tasks, aggregation, and worker populations.

4.8 Recovery Under Worker Failures (Q7)

To answer Q7, we disable 0 to 4 workers from initial populations of GoAgent 8 and AutoGen 8. Panel (d) reports a five-dataset mean over the same tasks as (a) and (b). GoAgent scores 81.74% without failures, 83.10% with one failure, and 76.80% with four. AutoGen changes from 82.28% to 73.80% across the sweep. Four failures produce losses of 4.94 and 8.48%, respectively. GoAgent moves from 0.54 points below AutoGen to 3.00 points above it. Our Conclusion (C7). GoAgent retains more reference accuracy than AutoGen in this worker-failure setting.

4.9 Efficiency (Q8)

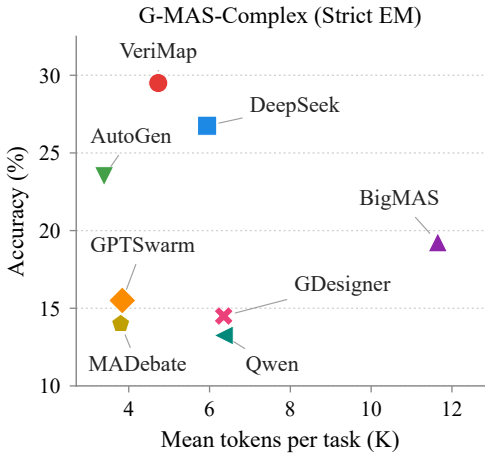

Figure 5. Accuracy and token use for eight G-MAS-Complex configurations.

To answer Q8, we compare strict exact match and mean tokens per task. Figure 5 presents eight configurations spanning the three system groups. Figure 6 in Appendix A.3 shows all 17 configurations. Table 5 adds latency, executed nodes and edges, and Strict EM/K Tokens, which divides the strict-match percentage by mean token use in thousands.

VeriMap achieves 29.50% with 4.73K tokens per task, exceeding DeepSeek's 26.75% with 5.94K tokens. AutoGen and VeriMap have the highest score-per-token ratios. GoAgent reaches 50.25% with 26.47K tokens in the complete comparison. Relative to DeepSeek, it gains 23.50% while using 4.46 times the tokens and 30.11 times the latency. Our Conclusion (C8). The highest accuracy and highest score per token occur at different operating points.

Each operating point is linked to an execution record containing raw and parsed predictions, task scores, prompt and organization hashes, token usage, model calls, executed nodes and edges, retries, latency, and termination status. Intervention records also identify the reference organization, changed components, injected records, failed workers, and selected recovery action. These records connect reported scores and resource measurements to the task and configuration used for each run.

Accuracy is computed over the same 400 frozen tasks, with 100 tasks in each difficulty tier. Strict exact match requires agreement on final-layer values, the mission result, the global checksum, the final-layer ranking, source-document identifiers, and critical checkpoints. Component accuracy is recorded separately for these six outputs. The validation record distinguishes format failures, where a response cannot be parsed into the required schema, from content failures, where a valid response contains incorrect values. These records accompany token and execution measurements, connecting the accuracy and resource coordinates of each configuration to its individual task results.

5 Conclusion and Future Directions

OpenMAS-GCom evaluates multi-agent organizations through broad task comparisons and paired interventions on communication structure, collaboration units, information quality, and worker availability. The results identify how configuration choice, task domain, organizational changes, and resource budgets affect task scores and the gains from graph enhancement.

Effectiveness (C1 to C3). Graph-enhanced, single-agent, and ordinary MAS configurations achieve the highest central score on 3, 2, and 2 of the 7 main-table datasets, respectively (C1). On G-MAS-Complex, graph-enhanced group maxima exceed ordinary-MAS maxima by 15 to 45% across tiers, yet only GoAgent and VeriMap outperform the DeepSeek reference across all 400 tasks (C2). Displayed central-score dispersion varies substantially across domains (C3).

Organizational sensitivity and robustness (C4 to C7). Degree-preserving rewiring decreases the five-dataset mean by 8.19% at 70% rewiring (C4). Specialist removal causes a larger mean loss than critic removal, while Expert only produces the largest decrease (C5). At 80% incorrect-message corruption, VeriMap leads MAD by 10.18 points (C6). With four worker failures, GoAgent retains more reference accuracy than AutoGen and leads by 3.00 points (C7). These comparisons use fixed tasks, models, and budget limits.

Efficiency and implications (C8). GoAgent attains the highest Strict EM on G-MAS-Complex, whereas AutoGen and VeriMap attain the highest score-per-token ratios. Thus, task performance and token efficiency favor different operating points (C8). The eight questions connect task scores with the conditions producing them. A method can lead on the broad task suite, lose accuracy after an organizational change, or achieve a competitive score with higher token and latency costs. Execution traces link the intervention, score change, and measured costs for each matched task.

Practical implications. OpenMAS-GCom supports a staged diagnostic workflow for future G-MAS studies. A method should first be compared with single-agent and ordinary multi-agent references on the same task lists, then examined across domains and task difficulty, and finally tested under topology, role, information, and worker interventions. This sequence distinguishes a general accuracy gain from an improvement that depends on a particular collaboration unit or communication pattern. It also relates accuracy to resource use and robustness. A system can attain a strong score with high token or latency cost, while a cheaper system can lose accuracy under incorrect messages or worker failures. Reporting scores, traces, resource use, and uncertainty together gives a more complete account of system behavior.

Future benchmarks should extend this protocol to longer-horizon, multimodal, interactive, and changing environments while preserving frozen task manifests and matched controls. Organization learning should jointly consider topology, roles, routing, verification, and recovery under token, latency, and reliability budgets. Stronger provenance tracking, conflict resolution, error correction, and standardized cost accounting are also needed to compare reliability, energy, monetary cost, and reproducibility alongside accuracy. These extensions can identify which communication structures and role assignments suit particular task dependencies, input formats, and resource budgets.

AI USE STATEMENT

Generative AI tools were used during the preparation of this manuscript for language refinement, literature retrieval and discovery, and research ideation support. Specifically, AI tools assisted with improving writing clarity, organizing related work, exploring potential research directions, and discussing possible methodological designs and experimental considerations. We also used AI tools to assist in constructing the G-MAS-Complex dataset.

REPRODUCIBILITY STATEMENT

We provide the anonymized OpenMAS-GCom code package at `https://anonymous.4open.science/r/OpenMAS-GCom-123456/`. The package contains method configurations, execution adapters, intervention scripts, dataset preparation utilities, and evaluation tools. The main paper describes the benchmark formulation, four intervention protocols, evaluation settings, and experimental results. The appendix details the datasets, G-MAS-Complex task construction, scoring rules, intervention parameters, and execution records. The code documentation describes environment setup, data preparation, external model dependencies, experiment commands, and the additional source components required for execution. Datasets and pretrained checkpoints are obtained separately. The package also includes checks for anonymization and source syntax.

A APPENDIX

This appendix describes datasets, baseline adapters, metrics, interventions, evaluation strategies, execution metadata, parameter settings, supplementary results, and reproducibility records.

A.1 DATASET DESCRIPTION

OpenMAS-GCom contains a broad task suite and a separately constructed G-MAS-Complex stress suite. The broad suite contains 29 datasets across six domains. Mathematics evaluates arithmetic, symbolic, and competition-style reasoning with GSM8K, SVAMP, MultiArith, AQuA, ASDiv-A, Game-of-24, and MATH (Cobbe et al., 2021, Patel et al., 2021, Roy & Roth, 2015, Ling et al., 2017, Miao et al., 2020, Yao et al., 2023, Hendrycks et al., 2021b). Knowledge and commonsense uses MMLU, MMLU-Redux, MMLU-Pro, StrategyQA, CommonsenseQA, ARC-Easy, and ARC-Challenge to test subject knowledge, robust multiple-choice answering, implicit reasoning, and science QA (Hendrycks et al., 2021a, Gema et al., 2025, Wang et al., 2024, Geva et al., 2021, Talmor et al., 2019, Clark et al., 2018). Code uses HumanEval, HumanEval++, LiveCodeBench-v6, and MultiAgentBench-Coding for executable program synthesis and collaborative coding evaluation (Chen et al., 2021a, Liu et al., 2023, Jain et al., 2024, Zhu et al., 2025). Biomedical tasks include PubMedQA, MedQA, MedMCQA, and the medical subset of MMLU (Jin et al., 2019, 2021, Pal et al., 2022, Hendrycks et al., 2021a). Finance uses TAT-QA, ConvFinQA, and FinQA to evaluate numerical reasoning over financial text and tables (Zhu et al., 2021, Chen et al., 2022, 2021b). Tabular reasoning uses TabFact, WikiSQL, WikiTableQuestions, and SQA for table verification, semantic parsing, and sequential table QA (Chen et al., 2020, Zhong et al., 2017, Pasupat & Liang, 2015, Iyyer et al., 2017).

G-MAS-Complex task composition. The evaluated G-MAS-Complex release is v6.3.2. We combine its Test-A and Test-B sets to obtain 400 independent tasks, with 100 tasks in each of the Medium, Hard, Ultra, and Extreme tiers. Each task combines multiple source documents, dependencies among records, records that must be rejected, and an Oracle graph. Solving a task requires selecting relevant records and following their dependencies to compute the outputs.

Difficulty progression. Table 3 makes the additional work at each tier explicit. The reported Medium statistics include 11,036 prompt characters, 6 rejected records, 27 Oracle nodes, and 38 Oracle edges. Hard increases all four task-scale measures. Ultra raises the rejected-record count to 21 and the Oracle edge count to 79, increasing record selection and dependency tracking work. Extreme has the longest input and the largest Oracle graph, with 37,061 prompt characters and 98 edges. Across the four tiers, a solver must inspect more text, exclude more unsuitable records,

Table 2. BROAD-SUITE DATASET SUMMARY. Domain-level statistics summarize the broad suite. Per-dataset results and sample counts are reported by domain in Appendix A.9.

Domain	#Datasets	Datasets	Primary evaluation target
Mathematics	7	GSM8K, SVAMP, MultiArith, AQuA, ASDiv-A, Game-of-24, MATH	Numeric answer, multiple-choice answer, valid expression, or competition-style final answer
Knowledge and commonsense	7	MMLU, MMLU-Redux, MMLU-Pro, StrategyQA, CommonsenseQA, ARC-Easy, ARC-Challenge	Multiple-choice or yes/no answer after robust answer extraction
Code	4	HumanEval, HumanEval++, LiveCodeBench-v6, MultiAgentBench-Coding	Unit-test pass rate, executable correctness, or code-task quality score
Biomedical	4	PubMedQA, MedQA, MedMCQA, MMLU-Med	Biomedical yes/no/maybe answer or medical multiple-choice answer
Finance	3	TAT-QA, ConvFinQA, FinQA	Normalized numerical or textual financial answer
Tabular	4	TabFact, WikiSQL, WikiTableQuestions, SQA	Table entailment, table-query answer, or sequential table answer

Table 3. G-MAS-COMPLEX DIFFICULTY STATISTICS. Reported task-scale characteristics of the four tiers in v6.3.2. Prompt length is measured in characters. Oracle node and edge counts describe the task's Oracle graph.

Tier	Tasks	Prompt chars	Rejected records	Oracle nodes	Oracle edges
Medium	100	11,036	6	27	38
Hard	100	17,066	13	36	57
Ultra	100	25,056	21	48	79
Extreme	100	37,061	27	59	98

Table 4. BASELINE FAMILIES AND ORGANIZATION ADAPTERS.

Family	Methods	Adapter representation
Single-agent	DeepSeek, Qwen (DeepSeek-AI, 2025, Qwen Team, 2025)	One runnable unit, empty edge set, direct answer parser
Ordinary MAS	AutoGen, MADebate, DyLAN (Wu et al., 2023, Du et al., 2023, Liu et al., 2024)	Predefined multi-agent protocol. DyLAN dynamically selects units within its configured candidate pool
Graph construction and routing	GPTSwarm, GDesigner, GraphSearch, GraphR1, MASRouter (Zhuge et al., 2024, Zhang et al., 2025, Liu et al., 2026, Luo et al., 2026, Yue et al., 2025)	Explicit graph or search/routing module mapped to typed directed relations
Sparse or role-specialized G-MAS	SparseCT, R-GFM, BigMAS, ARGDes, EIB (Li et al., 2024, Hao et al., 2026, Li et al., 2026, Shen et al., 2025)	Specialist units and sparse communication structures mapped to runnable nodes and edges
Verification and recovery-oriented G-MAS	GoAgent, VeriMap (Chen et al., 2026, Xu et al., 2025)	Verification, fallback, or rerouting policies exposed as execution-policy components

and combine information across more dependencies. These Oracle statistics characterize the tasks. Executed agent counts and communication edges are measured separately for each method.

Shared output requirements. All four tiers use the same six-component answer contract. For task i, we write the reference answer as

$$z_i = (v_i, m_i, \chi_i, r_i, \ell_i, h_i), \tag{7}$$

where v_i contains final-layer values, m_i is the mission result, χ_i is the global checksum, r_i is the final-layer ranking, ℓ_i contains source-document identifiers, and h_i contains critical checkpoints. A task passes strict exact match only when all six components agree with the reference. Thus, correct final values must be accompanied by the correct ranking, sources, checkpoints, and checksum. Appendix B.2 describes the component checks.

Illustrative record dependency. The small inventory example in Appendix B.1 illustrates record selection and calculation. One document states an initial inventory of 12 units, another revises it to 9, and a third reserves 4. Selecting the revised record gives $9 - 4 = 5$ available units. This example explains how a record-selection error changes a downstream result. The evaluated tasks apply the complete six-component contract above across the tier sizes in Table 3.

A.2 BASELINE DESCRIPTION

Multi-agent baselines use the official implementations released by their authors. The benchmark interface wraps each method as an executable organization $\mathcal{O} = (V, E, S, \pi)$ before evaluation. This adapter layer separates method-specific collaboration behavior from shared benchmark components such as dataset loading, backend calls, answer parsing, metric calculation, trace logging, and cost accounting. Single-agent baselines contain one runnable unit and no communication edges. Ordinary MAS baselines contain multiple units with predefined collaboration patterns. Graph-enhanced MAS baselines expose organization graphs, learned or searched routing, sparse communication, verification links, or recovery policies through the same runner interface.

Organization configuration and execution. The configured organization and the realized execution trace are recorded separately. The organization specifies available units, roles, communication relations, routing policy, and recovery policy, while the trace records which units actually execute for a particular input. Execution can differ from the static configuration as messages arrive, candidate answers are rejected, workers become unavailable, or recovery procedures are triggered. This dis-

tinction separates available units from executed units and connects the graph specified by a method with the communication actually used to produce a prediction.

A.3 Metric Description

OpenMAS-GCom reports the dataset metrics listed in the complete result tables. For classification and question-answering datasets, the parser maps free-form generations to the answer space before applying exact agreement with the reference. For code tasks, generated programs are evaluated with the corresponding pass@1-style executable test protocol. For tabular and financial reasoning, answers are normalized before comparison, including number formatting, option markers, and short textual aliases when supported by the original benchmark metric.

For a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ and method F, the broad-suite score is

$$s(F, \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} m(\text{parse}(F(x_i)), y_i), \tag{8}$$

where m is the dataset-specific metric. For QA datasets, m is accuracy or exact match. For code datasets, m is executable correctness. For MultiAgentBench-Coding, m is CodeQ.

For G-MAS-Complex, strict exact match is

$$\text{StrictEM}(\hat{z}_i, z_i) = \mathbf{1}[\hat{z}_i \models \Omega_i \wedge \hat{z}_i = z_i], \tag{9}$$

and the suite score is

$$s_{\text{complex}}(F) = \frac{1}{400} \sum_{i=1}^{400} \text{StrictEM}(\hat{z}_i, z_i). \tag{10}$$

A prediction is correct only when final-layer values, the mission result, the global checksum, the final-layer ranking, source-document identifiers, and critical checkpoints all match the reference. Component accuracy records agreement for each of these six outputs separately. A mismatch in any component gives zero strict exact-match credit for that task.

Resource records contain token usage, model calls, executed nodes and edges, retries, latency, and available cost statistics for each method. Latency comparisons group runs by their backend and execution environment. For G-MAS-Complex, Table 5 additionally reports Strict EM/K Tokens, the strict-match percentage per 1K tokens.

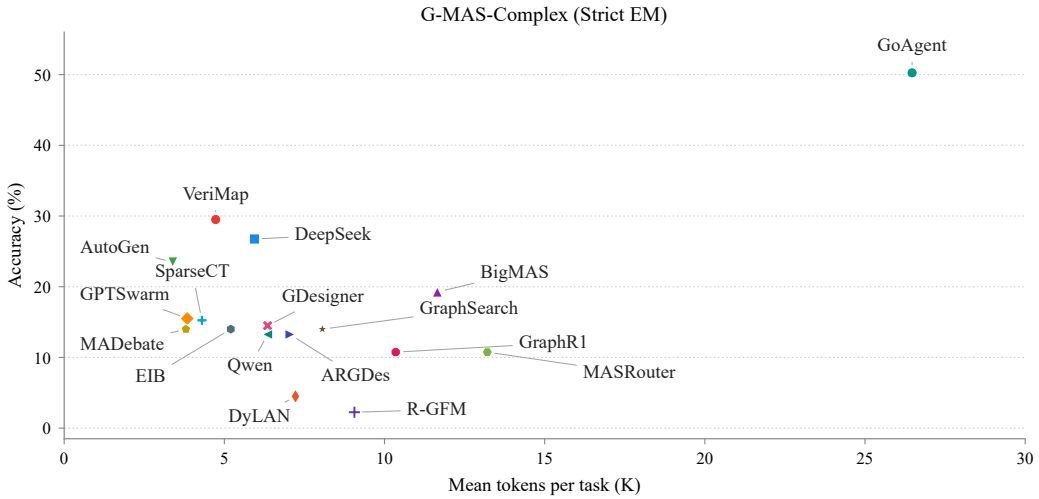

Figure 6. G-MAS-Complex accuracy and token use for all 17 configurations. Each labeled point shows strict exact match and mean tokens per task for one method.

Table 5. Observed operating points on G-MAS-Complex. Strict EM is in percent, tokens per task in K, and latency in seconds. Nodes and edges are execution averages. Strict EM/K Tokens describes score per 1K tokens at the reported setting. Qwen latency comes from local deployment, and the other latency measurements come from server deployment.

Method	Strict EM (%)	Tokens/task (K)	Strict EM/K Tokens	Latency (s)	Nodes	Edges
GoAgent	50.25	26.47	1.90	45.46	17.25	67.15
VeriMap	29.50	4.73	6.24	7.09	4.00	4.00
DeepSeek	26.75	5.94	4.50	1.51	1.00	0.00
AutoGen	23.50	3.39	6.93	6.42	3.00	2.00
BigMAS	19.25	11.65	1.65	17.11	14.30	30.90
GPTSwarm	15.50	3.84	4.04	6.48	3.95	2.95
SparseCT	15.25	4.30	3.55	9.25	4.00	3.00
GDesigner	14.50	6.35	2.28	7.95	6.15	9.05
EIB	14.00	5.20	2.69	9.25	6.15	8.05
GraphSearch	14.00	8.06	1.74	11.84	9.15	18.40
MADebate	14.00	3.80	3.68	5.85	4.00	3.00
ARGDes	13.25	7.05	1.88	8.75	7.15	11.25
Qwen	13.25	6.35	2.09	6.41	1.00	0.00
GraphR1	10.75	10.35	1.04	12.60	11.15	28.55
MASRouter	10.75	13.21	0.81	10.76	21.60	38.20
DyLAN	4.50	7.22	0.62	12.67	9.15	12.10
R-GFM	2.25	9.06	0.25	21.27	10.15	13.10

A.4 Robustness Intervention Description

For every intervention, the task list, backend configuration, prompt template, answer parser, metric, routing policy, and runtime budget remain fixed unless the intervention explicitly targets the corresponding organization component. Let

$$\mathcal{C} = \{\mathcal{D}_{\mathcal{T}}, \Theta, M_{\mathcal{T}}, P\} \tag{11}$$

denote the shared controls. For intervention type r, we construct

$$\mathcal{O}'_r = \mathcal{I}_r(\mathcal{O}, \Delta\mathcal{O}_r), \qquad r \in \{\mathrm{str}, \mathrm{node}, \mathrm{info}, \mathrm{exec}\}, \tag{12}$$

and measure

$$\Delta_r(F, \mathcal{T}) = s(F, \mathcal{O}'_r, \mathcal{T}) - s(F, \mathcal{O}, \mathcal{T}). \tag{13}$$

A negative value indicates lower performance after modification under the shared controls.

Table 6. Controls preserved by diagnostic interventions. Yes means the factor is kept fixed relative to the reference run.

Intervention	Samples	Backend	Prompts	Parser	Budget	Roles	Edges	Degree seq.
Structure rewiring	Yes	Yes	Yes	Yes	Yes	Yes	Changed	Yes
Node removal	Yes	Yes	Yes	Yes	Yes	Changed	Changed	No
Information injection	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Execution failure	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Structure intervention. We evaluate the Original graph and degree-preserving rewiring ratios of 10%, 30%, 50%, and 70%. Directed edge pairs $u \to a$ and $v \to b$ are proposed for the swap $u \to b$ and $v \to a$. Proposals that create self-loops, duplicate edges, or invalid role connections are rejected. The actual ratio and random seed are recorded.

Node intervention. The No critic and No specialist variants remove the corresponding role nodes and all incident edges. The Expert only variant retains expert nodes and their induced communication edges. This intervention changes collaboration-unit composition while keeping the task, model, prompt, parser, routing, and budget controls fixed.

Information intervention. For concentration $c \in \{0, .2, .4, .6, .8\}$, a fraction c of eligible upstream records or candidate items is replaced with a format-valid but incorrect candidate and forwarded through the unchanged downstream graph. We report task accuracy and whether final answers follow the injected candidate. The trace further records whether the injected candidate reaches downstream units and whether it is adopted, corrected, or ignored before the final answer.

Execution intervention. For each failure count $k \in \{0, 1, 2, 3, 4\}$, k workers are marked unavailable at selected execution points. The last valid shared state is recorded and the method resumes through its configured retry, fallback, or rerouting mechanism. $k = 0$ denotes no failure.

A.4.1 Figure 4 experimental settings

Figure 4 evaluates four interventions using a matched reference within each panel. Score changes are computed relative to that panel's zero-intervention condition.

Panels (a) and (b). Both panels use the same GDesigner reference configuration and display GSM8K, MMLU-Pro, HumanEval, MedQA, and WikiTableQuestions separately. The role composition is four experts, two domain specialists, and two critics. For each task, its generated reference graph is stored before either intervention. Panel (a) modifies its communication edges, and panel (b) removes the specified role groups. Expert only keeps the four experts and their links.

Panels (c) and (d). Both panels use the five datasets in panels (a) and (b). Panel (c) compares MADebate and VeriMap. Panel (d) compares GoAgent and AutoGen. The aggregation rule is equal-weight dataset mean. For dataset scores $s_1, \ldots, s_5$, this rule gives $\bar{s} = \frac{1}{5}\sum_{j=1}^{5} s_j$, using each dataset's original scoring metric. Every condition uses the same task list within each dataset. The reported percentage scale is retained when averaging the five scores.

Worker population and failure schedule. The initial populations are 8 task workers for GoAgent and 8 for AutoGen. A worker is a task-executing agent. Coordinators and tool processes are excluded from this count. The failure count k denotes original workers that are simultaneously unavailable, so a configuration starting with N_0 workers retains $N_0 - k$ available original workers. The schedule disables a uniformly selected set after initial task assignment and before their outputs are returned, keeps them unavailable until task termination, and retains the last valid shared state for recovery. Each method then follows its configured retry, fallback, or rerouting procedure within the original budget. Configured initial workers and executed-node averages describe the initial population and realized execution, respectively.

A.5 Evaluation Strategies

The benchmark follows four evaluation perspectives, matching the main questions in Section 4. Utility compares single-agent, ordinary MAS, and graph-enhanced MAS under identical task and metric settings. Effectiveness compares method behavior across the six broad-suite domains and the four G-MAS-Complex tiers. Robustness applies the four controlled interventions above to examine communication topology, role composition, information quality, and execution recovery. Efficiency reports quality together with token usage, model calls, latency, executed nodes and edges, and retries.

A.6 Experiment Environment

Sections 4.1 and 4.9 describe execution settings and run records. Table 8 lists the complete fields connecting tasks, configurations, predictions, and resource measurements.

Before aggregation, the audit script checks that reference and intervention runs share the same task identifiers, parser version, backend configuration, prompt template, and stopping budget. Runs that fail these consistency checks are excluded from aggregate tables until regenerated.

Validation protocol. The validation procedure in Section 4.9 separates format failures from content failures for G-MAS-Complex. Each run produces a record containing dataset and sample identifiers,

Table 7. EVALUATION STRATEGY SUMMARY.

Perspective	Main comparison	Reported results
Utility	Single-agent vs. ordinary MAS vs. graph-enhanced MAS	Main seven-dataset comparison and full 29-dataset broad-suite results
Effectiveness	Method behavior across domains and complex-task tiers	Domain averages, G-MAS-Complex tiers, and complete method rankings
Robustness	Structure, node, information, and execution interventions	Performance changes under controlled organizational perturbations
Efficiency	Task quality relative to execution cost	Strict EM, tokens, latency, executed nodes, executed edges, and Strict EM/K tokens

Table 8. RECORDED ENVIRONMENT AND RUN METADATA. The record links task, organization, backend, prediction, cost, and intervention fields.

Category	Recorded fields
Task state	Dataset id, sample id, split or frozen-list identifier, gold answer, parser version
Organization state	Method id, organization hash, node list, edge list, role configuration, routing policy
Backend state	Backend configuration id, prompt-template hash, runtime limits, random seed when used
Prediction state	Raw response, parsed prediction, invalid-parse flag, metric result, final answer
Cost state	Token usage, model calls, executed nodes, executed edges, retries, latency, termination reason
Intervention state	Reference organization hash, intervention type, changed nodes/edges, injected records, failed workers, recovery action

method configuration, parser version, raw and parsed answers, metric value, and trace hash before entering an aggregate table.

A.7 HYPERPARAMETER SETTINGS

The shared controls and sampling settings are described in Section 4.1. Table 9 gives the settings for each comparison track and intervention sweep.

Table 9. BENCHMARK-LEVEL HYPERPARAMETER AND SWEEP SETTINGS.

Component	Setting
Broad-suite sampling	Full evaluation split where used, otherwise fixed ordered subset, with three matched 500-instance samples for repeated-sampling datasets
G-MAS-Complex sampling	400 frozen tasks, 100 tasks each for Medium, Hard, Ultra, and Extreme
Structure rewiring	Original, 10%, 30%, 50%, and 70% degree-preserving rewiring
Node intervention	Base, No critic, No specialist, Expert only
Information intervention	Incorrect-information concentration $c \in \{0, .2, .4, .6, .8\}$
Execution intervention	Failed workers $k \in \{0, 1, 2, 3, 4\}$
Uncertainty reporting	Sample-set or run-seed standard deviation, or binomial standard error for binary outcomes

A.8 Uncertainty Sources

The detailed tables report central scores and available standard deviations or standard errors. Table 10 identifies their statistical sources. Ranking markers order the central scores, including equal numerical values. Sample-set standard deviations describe variation across task samples, run-seed standard deviations describe variation across runs, and binomial standard errors describe sampling uncertainty for binary task outcomes.

The supplementary file `full_results_reported_statistics.csv` records each dataset and configuration, central score, original uncertainty quantity, and displayed uncertainty term. The original quantities are retained in `full_results_variance_sources.csv`. Central scores are reported without uncertainty terms for the 34 weighted empirical estimates and 14 CodeQ binomial-form quantities. This reporting rule preserves the scores while distinguishing measured variation from quantities without an applicable SD or SE interpretation.

Source	Cells	Recorded quantity	Repetition information
Sample-set SD	221	Mean and sample standard deviation across sampled task sets. Each set contains 500 instances.	Three sample sets
Run-seed SD	152	Fixed reference score with an attached run-seed standard deviation.	Run count unrecorded
Central score only	34	Original uncertainty estimates combine dataset and method components at weights 0.70 and 0.30. These estimates are omitted from the result tables.	Component counts unrecorded
Binomial SE	86	Reference score with a binomial standard-error estimate computed from that score and the displayed evaluation count N, for binary task outcomes.	Analytic estimate
Central score only	14	CodeQ averages normalized judge ratings. Binomial-form quantities are omitted because the task outcome is a graded score.	Binomial SE inapplicable
Backup run-seed summary	3	Run-seed mean and standard deviation for cells with an empty reference metric.	Run count unrecorded

Table 10. Statistical sources and reporting policy for the 510 entries. Uncertainty is reported for 462 entries, and 48 entries report central scores only.

For the three percentage scores y_1, y_2, y_3, the sample-set summary uses

$$\bar{y} = \frac{1}{3}\sum_{j=1}^{3} y_j, \qquad s = \sqrt{\frac{1}{2}\sum_{j=1}^{3}(y_j - \bar{y})^2}. \tag{14}$$

For a reported binary score $p \times 100\%$ and evaluation count N, the binomial standard error in percentage points is $100\sqrt{p(1-p)/N}$ under an independent binary-outcome model. The MultiAgentBench-Coding CodeQ metric averages four judge ratings from 1 to 5 and normalizes the result by five. The CodeQ table reports run-seed standard deviations for DeepSeek, BigMAS, and MASRouter, and central scores for the remaining configurations. Source records preserve the central value, original uncertainty quantity, and reporting status as separate fields.

A.9 Supplementary Results

The complete 29-dataset and 17-configuration comparison is organized by domain in the tables below. Repeated-sampling summaries are reported for the 13 datasets with matched three-sample-set evaluations. Each domain has a labeled result table.

Reading the detailed results. Each entry reports a central score in percent. Parentheses give a standard deviation or standard error in percentage points where reported. Appendix A.8 defines the reporting rule and statistical sources. Bold and underlined scores mark the highest and second-highest distinct values within each dataset, including ties. Horizontal rules separate single-agent, ordinary multi-agent, and graph-enhanced configurations.

Method	GSM8K Acc. $N = 1319$	SVAMP Acc. $N = 1000$	MultiArith Acc. $N = 600$	AQuA Acc. $N = 254$	ASDiv-A Acc. $N = 500$	Game-of-24 Acc. $N = 1362$	MATH Acc. $N = 500$
DeepSeek	93.71 (0.33)	94.00 (0.10)	98.50 (0.25)	89.76 (0.45)	88.53 (1.10)	58.88 (1.33)	55.73 (3.64)
Qwen	59.44	90.90 (0.91)	97.17 (0.68)	68.90	86.87 (0.42)	21.00 (1.10)	45.20 (2.27)
AutoGen	92.87 (0.34)	93.70 (0.77)	98.50 (0.19)	88.98 (1.72)	88.53 (0.99)	67.91 (1.26)	56.13 (2.66)
MADebate	92.87 (0.24)	94.20 (0.74)	**98.83** (0.00)	89.37 (0.23)	88.53 (0.99)	62.70 (1.31)	55.27 (3.42)
DyLAN	93.33	94.00 (0.25)	98.67 (0.29)	89.76 (0.99)	**88.80** (1.40)	62.41 (1.31)	56.07 (3.24)
GPTSwarm	93.56 (0.43)	94.10 (0.46)	98.50 (0.25)	**91.34**	88.67 (0.81)	65.71 (0.85)	57.00 (2.84)
GDesigner	93.63 (0.27)	94.00 (0.06)	**98.83** (0.10)	88.98 (0.82)	88.60 (1.31)	67.62 (1.27)	56.73 (3.37)
SparseCT	**93.86**	93.90 (0.76)	98.67 (0.10)	89.76 (0.68)	**88.80** (0.92)	64.54 (1.30)	57.13 (3.87)
GraphSearch	93.56 (0.13)	94.10 (0.21)	98.67 (0.00)	89.37 (1.20)	88.67 (0.90)	67.55 (1.27)	56.40 (3.46)
GraphR1	92.87 (0.27)	93.60 (0.55)	98.50 (0.10)	90.55 (0.23)	88.60 (1.06)	66.89 (1.28)	56.73 (2.72)
R-GFM	93.18 (0.20)	93.80 (0.26)	98.50 (0.10)	89.76 (0.91)	**88.80** (1.44)	66.08 (1.28)	55.73 (3.88)
BigMAS	93.18 (0.19)	94.30 (0.44)	98.67 (0.10)	90.55 (0.45)	88.53 (1.51)	65.93 (1.28)	**57.47** (3.59)
MASRouter	93.71 (0.23)	93.80 (0.17)	98.50 (0.10)	88.98 (0.39)	88.40 (1.59)	63.14 (1.31)	56.00 (3.33)
GoAgent	92.87 (0.04)	94.30 (0.31)	98.50 (0.00)	89.76 (0.79)	88.53 (0.99)	**74.08** (1.19)	56.07 (2.89)
VeriMap	93.63 (0.39)	93.60 (0.25)	**98.83** (0.10)	89.76 (0.68)	88.73 (0.50)	65.71 (0.28)	56.33 (3.87)
ARGDes	93.63 (0.38)	**94.60** (0.71)	98.67 (0.17)	90.55 (0.99)	88.53 (1.17)	67.11 (1.27)	56.53 (3.01)
EIB	93.56 (0.08)	94.30 (0.10)	98.50 (0.10)	89.76 (0.39)	**88.80** (1.40)	67.84 (1.27)	56.60 (3.29)

Table 11. Mathematical reasoning results with available uncertainty terms.

Method	MMLU Acc. $N = 500$	MMLU-Redux Acc. $N = 500$	MMLU-Pro Acc. $N = 500$	StrategyQA Acc. $N = 500$	Commonsense QA Acc. $N = 500$	ARC-Easy Acc. $N = 500$	ARC-Challenge Acc. $N = 1418$
DeepSeek	85.40 (0.92)	83.73 (0.50)	73.07 (0.76)	84.60 (0.80)	85.40 (0.53)	94.27 (0.61)	96.26 (0.50)
Qwen	82.33 (2.76)	79.80 (0.92)	60.93 (0.90)	79.40 (1.93)	85.67 (1.33)	94.07 (0.76)	96.05 (0.52)
AutoGen	85.80 (0.35)	84.00 (1.73)	71.80 (1.20)	84.13 (0.70)	85.67 (1.30)	94.27 (0.76)	96.61 (0.15)
MADebate	85.93 (0.12)	83.60 (0.53)	71.87 (1.14)	84.07 (1.96)	85.20 (1.56)	94.07 (0.81)	96.54 (0.49)
DyLAN	86.00 (1.39)	83.93 (1.50)	**73.87** (1.21)	**86.67** (0.70)	85.73 (0.61)	94.07 (0.76)	96.76 (0.19)
GPTSwarm	85.60 (0.53)	84.40 (0.35)	70.73 (0.61)	85.67 (2.84)	85.53 (0.64)	94.27 (0.76)	**96.83** (0.19)
GDesigner	85.67 (1.51)	84.27 (1.17)	71.87 (0.31)	85.00 (2.03)	86.13 (1.36)	94.20 (1.06)	96.33 (0.50)
SparseCT	85.20 (0.87)	83.73 (0.50)	71.33 (1.10)	85.80 (0.80)	86.13 (1.21)	93.93 (0.64)	96.69 (0.48)
GraphSearch	85.80 (0.80)	84.40 (1.91)	71.60 (0.92)	85.60 (0.20)	86.07 (1.27)	94.27 (0.58)	96.40 (0.49)
GraphR1	85.33 (1.30)	84.00 (1.06)	70.53 (1.33)	85.60 (0.69)	85.47 (1.67)	94.20 (0.72)	**96.83** (0.18)
R-GFM	85.47 (1.27)	84.27 (1.62)	71.53 (0.50)	85.13 (0.50)	85.93 (1.30)	93.93 (0.83)	96.76 (0.36)
BigMAS	86.00 (1.11)	84.40 (0.92)	72.20 (0.20)	85.07 (1.62)	85.87 (1.03)	93.93 (1.03)	**96.83** (0.16)
MASRouter	86.53 (1.03)	84.33 (0.83)	72.53 (1.81)	85.67 (0.58)	86.07 (0.70)	**94.33** (0.64)	96.61 (0.25)
GoAgent	85.93 (1.14)	83.87 (1.33)	70.33 (0.50)	85.27 (2.05)	**86.47** (1.45)	94.00 (1.06)	96.76 (0.18)
VeriMap	85.87 (0.46)	84.13 (1.51)	73.53 (1.70)	83.53 (1.17)	85.40 (1.00)	93.87 (0.92)	96.76 (0.12)
ARGDes	**87.27** (1.10)	83.60 (1.04)	71.47 (0.58)	86.00 (1.59)	86.20 (2.31)	94.13 (0.64)	96.76 (0.25)
EIB	86.13 (1.67)	**84.53** (1.42)	72.47 (0.12)	85.40 (1.40)	85.73 (1.17)	93.93 (0.70)	96.61 (0.22)

Table 12. Knowledge and commonsense results with available uncertainty terms.

Method	HumanEval Acc. $N = 164$	HumanEval++ Acc. $N = 164$	LiveCodeBench v6 Pass@1 $N = 175$	MultiAgentBench Coding CodeQ $N = 102$
DeepSeek	84.76 (0.00)	**80.49** (2.14)	20.57 (2.01)	41.55 (1.06)
Qwen	**90.24**	76.83 (0.70)	**26.86**	**60.54**
AutoGen	81.71	73.78 (3.43)	18.86	42.01
MADebate	83.54 (2.11)	77.44 (2.46)	21.71 (2.88)	40.69
DyLAN	85.98	78.66 (3.20)	22.86	37.06
GPTSwarm	84.76 (0.93)	**80.49** (5.21)	19.43 (0.00)	41.91
GDesigner	85.37	76.83 (1.76)	<u>23.43</u>	39.46
SparseCT	84.76 (1.83)	76.83 (1.22)	21.14 (1.44)	<u>43.82</u>
GraphSearch	81.10 (3.06)	74.39 (1.06)	22.29 (3.15)	40.64
GraphR1	86.59 (2.66)	75.61 (1.06)	20.00 (3.02)	38.09
R-GFM	89.02	75.61 (2.54)	19.43 (0.87)	40.74
BigMAS	84.76 (0.35)	76.22 (3.32)	21.71 (1.44)	37.79 (1.04)
MASRouter	76.83 (3.29)	71.95 (3.51)	20.57 (0.57)	37.45 (2.48)
GoAgent	79.88	74.39 (2.46)	20.57 (1.51)	37.79
VeriMap	78.66 (3.20)	77.44 (3.66)	21.14 (0.33)	40.15
ARGDes	<u>89.63</u>	<u>79.27</u> (3.17)	19.43 (1.84)	39.90
EIB	83.54 (2.20)	76.83 (1.27)	21.71 (1.44)	42.35

Table 13. Code generation results. Entries give scores and available uncertainty terms. The recorded CodeQ metric is retained for MultiAgentBench-Coding.

Method	PubMedQA Acc. $N = 3000$	MedQA Acc. $N = 1273$	MedMCQA Acc. $N = 500$	MMLU-Med Acc. $N = 1561$
DeepSeek	81.33 (0.71)	88.45 (0.68)	79.67 (0.99)	89.75 (0.77)
Qwen	80.70 (1.25)	78.95	73.53 (0.61)	87.51 (0.84)
AutoGen	81.43 (0.71)	89.08 (0.55)	79.67 (1.70)	89.94 (0.15)
MADebate	81.87 (0.70)	89.08 (0.53)	78.53 (1.14)	<u>90.26</u> (0.75)
DyLAN	**82.43** (0.69)	88.77 (0.08)	79.93 (1.10)	89.56 (0.38)
GPTSwarm	81.67 (0.71)	89.63	79.00 (0.87)	89.94 (0.24)
GDesigner	81.37 (0.71)	88.45 (0.48)	79.93 (0.76)	89.69 (0.77)
SparseCT	81.60 (0.67)	89.24 (0.21)	79.67 (1.67)	89.88 (0.39)
GraphSearch	81.53 (0.44)	<u>89.71</u> (0.85)	78.73 (1.01)	89.88 (0.27)
GraphR1	81.30 (0.71)	88.92 (0.88)	79.13 (0.83)	**90.39** (0.75)
R-GFM	81.53 (0.21)	89.16 (0.70)	79.53 (0.95)	89.88 (0.13)
BigMAS	81.20 (0.30)	**89.87**	78.73 (1.85)	90.07 (0.19)
MASRouter	81.60 (0.06)	89.24 (0.33)	78.80 (1.91)	89.56 (0.10)
GoAgent	80.53 (0.72)	89.40	79.13 (0.92)	89.56 (0.33)
VeriMap	81.27 (0.71)	88.77 (0.52)	<u>80.07</u> (1.51)	90.07 (0.11)
ARGDes	<u>82.03</u> (0.70)	88.77 (0.42)	**80.53** (0.70)	89.62 (0.77)
EIB	81.20 (0.46)	89.16 (0.48)	78.87 (1.33)	90.20 (0.75)

Table 14. Biomedical reasoning results. Entries give scores and available uncertainty terms. PubMedQA retains the recorded source sample count.

Method	TAT-QA Acc. $N = 1663$	ConvFinQA Acc. $N = 1490$	FinQA Acc. $N = 1147$
DeepSeek	70.48 (0.63)	60.13	38.10
Qwen	69.51 (1.13)	**61.81**	28.51
AutoGen	70.29 (1.12)	60.74 (0.66)	36.70
MADebate	70.66 (1.12)	61.01	36.97
DyLAN	70.48 (0.23)	60.60 (0.55)	<u>38.88</u>
GPTSwarm	70.41 (0.33)	60.67 (0.24)	37.66 (1.13)
GDesigner	70.23 (0.44)	<u>61.54</u>	38.45 (1.02)
SparseCT	70.41 (0.38)	60.81	**39.23**
GraphSearch	**71.26** (1.11)	60.20 (0.49)	38.45 (0.18)
GraphR1	69.75 (1.13)	60.67 (0.34)	38.01 (1.43)
R-GFM	<u>71.20</u> (1.11)	60.40 (0.24)	38.45 (0.46)
BigMAS	70.41 (0.68)	60.40 (0.21)	38.36
MASRouter	69.99 (0.24)	60.20 (0.34)	37.14 (1.43)
GoAgent	71.14 (1.11)	60.74 (0.47)	38.45 (0.13)
VeriMap	70.23 (0.33)	61.28 (1.26)	37.66 (0.83)
ARGDes	71.02 (1.11)	60.60 (0.12)	38.45 (0.31)
EIB	69.99 (0.67)	61.34	38.54

Table 15. Financial reasoning results with uncertainty terms.

Method	TabFact Acc. $N = 500$	WikiSQL Acc. $N = 500$	WikiTable Questions Acc. $N = 500$	SQA Acc. $N = 500$
DeepSeek	92.00 (1.11)	74.07 (0.61)	76.73 (2.16)	55.20 (1.44)
Qwen	81.40 (1.56)	<u>74.93</u> (1.27)	65.27 (1.50)	46.53 (2.14)
AutoGen	91.80 (1.22)	74.20 (1.78)	75.93 (1.03)	54.67 (1.97)
MADebate	91.47 (0.42)	74.27 (1.10)	76.67 (1.33)	54.60 (1.93)
DyLAN	**92.53** (1.53)	73.87 (2.53)	<u>77.40</u> (1.44)	55.00 (1.74)
GPTSwarm	92.20 (1.40)	74.80 (1.97)	76.60 (1.93)	56.13 (3.13)
GDesigner	91.73 (0.61)	<u>74.93</u> (0.83)	76.60 (1.06)	55.73 (4.12)
SparseCT	91.93 (1.10)	74.87 (1.22)	**77.47** (0.76)	56.53 (1.86)
GraphSearch	92.00 (1.56)	74.67 (0.81)	77.13 (1.62)	55.60 (1.44)
GraphR1	91.40 (1.71)	74.40 (0.69)	76.80 (0.20)	56.13 (1.81)
R-GFM	92.13 (2.30)	**75.60** (1.25)	<u>77.40</u> (1.56)	56.07 (3.67)
BigMAS	<u>92.33</u> (1.29)	74.73 (0.90)	76.80 (1.11)	55.33 (3.03)
MASRouter	91.40 (1.11)	74.00 (2.55)	75.67 (0.81)	<u>57.60</u> (1.51)
GoAgent	<u>92.33</u> (0.81)	74.47 (1.79)	76.20 (0.80)	**57.73** (2.34)
VeriMap	92.07 (1.21)	74.33 (1.63)	76.93 (0.90)	54.87 (1.27)
ARGDes	92.07 (0.83)	74.47 (0.81)	77.00 (1.60)	55.27 (1.92)
EIB	91.40 (0.72)	74.60 (1.59)	76.13 (2.19)	55.53 (3.16)

Table 16. Tabular reasoning results with uncertainty terms.

Method	G-MAS-Complex Strict EM $N = 400$
DeepSeek	26.75 (2.21)
Qwen	13.25 (1.70)
AutoGen	23.50 (2.12)
MADebate	14.00 (1.73)
DyLAN	4.50 (1.04)
GPTSwarm	15.50 (1.81)
GDesigner	14.50 (1.76)
SparseCT	15.25 (1.80)
GraphSearch	14.00 (1.73)
GraphR1	10.75 (1.55)
R-GFM	2.25 (0.74)
BigMAS	19.25 (1.97)
MASRouter	10.75 (1.55)
GoAgent	**50.25** (2.50)
VeriMap	<u>29.50</u> (2.28)
ARGDes	13.25 (1.70)
EIB	14.00 (1.73)

Table 17. G-MAS-Complex results. Entries give strict exact match in percent with a binomial standard-error estimate in parentheses over 400 tasks.

Sampling variability. The following tables report means and sample standard deviations across three 500-instance sample sets. Their variation is across sampled task sets, and their entries reproduce the corresponding sample-set results in the detailed tables. Bold and underlined means identify the highest and second-highest distinct means in each dataset, including ties.

Method	ASDiv-A	MATH
DeepSeek	88.53±1.10	55.73±3.64
Qwen	86.87±0.42	45.20±2.27
AutoGen	88.53±0.99	56.13±2.66
MADebate	88.53±0.99	55.27±3.42
DyLAN	**88.80**±1.40	56.07±3.24
GPTSwarm	88.67±0.81	57.00±2.84
GDesigner	88.60±1.31	56.73±3.37
SparseCT	**88.80**±0.92	<u>57.13</u>±3.87
GraphSearch	88.67±0.90	56.40±3.46
GraphR1	88.60±1.06	56.73±2.72
R-GFM	**88.80**±1.44	55.73±3.88
BigMAS	88.53±1.51	**57.47**±3.59
MASRouter	88.40±1.59	56.00±3.33
GoAgent	88.53±0.99	56.07±2.89
VeriMap	<u>88.73</u>±0.50	56.33±3.87
ARGDes	88.53±1.17	56.53±3.01
EIB	**88.80**±1.40	56.60±3.29

Table 18. Mathematical reasoning under sample-set variation. Entries report mean ± sample standard deviation in percent across three 500-instance sample sets.

Method	MMLU	MMLU-Redux	MMLU-Pro	StrategyQA	Commonsense QA	ARC-Easy
DeepSeek	85.40±0.92	83.73±0.50	73.07±0.76	84.60±0.80	85.40±0.53	94.27±0.61
Qwen	82.33±2.76	79.80±0.92	60.93±0.90	79.40±1.93	85.67±1.33	94.07±0.76
AutoGen	85.80±0.35	84.00±1.73	71.80±1.20	84.13±0.70	85.67±1.30	94.27±0.76
MADebate	85.93±0.12	83.60±0.53	71.87±1.14	84.07±1.96	85.20±1.56	94.07±0.81
DyLAN	86.00±1.39	83.93±1.50	**73.87**±1.21	**86.67**±0.70	85.73±0.61	94.07±0.76
GPTSwarm	85.60±0.53	84.40±0.35	70.73±0.61	85.67±2.84	85.53±0.64	94.27±0.76
GDesigner	85.67±1.51	84.27±1.17	71.87±0.31	85.00±2.03	86.13±1.36	94.20±1.06
SparseCT	85.20±0.87	83.73±0.50	71.33±1.10	85.80±0.80	86.13±1.21	93.93±0.64
GraphSearch	85.80±0.80	84.40±1.91	71.60±0.92	85.60±0.20	86.07±1.27	94.27±0.58
GraphR1	85.33±1.30	84.00±1.06	70.53±1.33	85.60±0.69	85.47±1.67	94.20±0.72
R-GFM	85.47±1.27	84.27±1.62	71.53±0.50	85.13±0.50	85.93±1.30	93.93±0.83
BigMAS	86.00±1.11	84.40±0.92	72.20±0.20	85.07±1.62	85.87±1.03	93.93±1.03
MASRouter	86.53±1.03	84.33±0.83	72.53±1.81	85.67±0.58	86.07±0.70	**94.33**±0.64
GoAgent	85.93±1.14	83.87±1.33	70.33±0.50	85.27±2.05	**86.47**±1.45	94.00±1.06
VeriMap	85.87±0.46	84.13±1.51	73.53±1.70	83.53±1.17	85.40±1.00	93.87±0.92
ARGDes	**87.27**±1.10	83.60±1.04	71.47±0.58	86.00±1.59	86.20±2.31	94.13±0.64
EIB	86.13±1.67	**84.53**±1.42	72.47±0.12	85.40±1.40	85.73±1.17	93.93±0.70

Table 19. Knowledge and commonsense under sample-set variation. Entries report mean ± sample standard deviation in percent across three 500-instance sample sets.

Method	MedMCQA
DeepSeek	79.67±0.99
Qwen	73.53±0.61
AutoGen	79.67±1.70
MADebate	78.53±1.14
DyLAN	79.93±1.10
GPTSwarm	79.00±0.87
GDesigner	79.93±0.76
SparseCT	79.67±1.67
GraphSearch	78.73±1.01
GraphR1	79.13±0.83
R-GFM	79.53±0.95
BigMAS	78.73±1.85
MASRouter	78.80±1.91
GoAgent	79.13±0.92
VeriMap	80.07±1.51
ARGDes	**80.53**±0.70
EIB	78.87±1.33

Table 20. Biomedical reasoning under sample-set variation. Entries report mean ± sample standard deviation in percent across three 500-instance sample sets.

Method	TabFact	WikiSQL	WikiTable Questions	SQA
DeepSeek	92.00±1.11	74.07±0.61	76.73±2.16	55.20±1.44
Qwen	81.40±1.56	74.93±1.27	65.27±1.50	46.53±2.14
AutoGen	91.80±1.22	74.20±1.78	75.93±1.03	54.67±1.97
MADebate	91.47±0.42	74.27±1.10	76.67±1.33	54.60±1.93
DyLAN	**92.53**±1.53	73.87±2.53	77.40±1.44	55.00±1.74
GPTSwarm	92.20±1.40	74.80±1.97	76.60±1.93	56.13±3.13
GDesigner	91.73±0.61	74.93±0.83	76.60±1.06	55.73±4.12
SparseCT	91.93±1.10	74.87±1.22	**77.47**±0.76	56.53±1.86
GraphSearch	92.00±1.56	74.67±0.81	77.13±1.62	55.60±1.44
GraphR1	91.40±1.71	74.40±0.69	76.80±0.20	56.13±1.81
R-GFM	92.13±2.30	**75.60**±1.25	77.40±1.56	56.07±3.67
BigMAS	92.33±1.29	74.73±0.90	76.80±1.11	55.33±3.03
MASRouter	91.40±1.11	74.00±2.55	75.67±0.81	57.60±1.51
GoAgent	92.33±0.81	74.47±1.79	76.20±0.80	**57.73**±2.34
VeriMap	92.07±1.21	74.33±1.63	76.93±0.90	54.87±1.27
ARGDes	92.07±0.83	74.47±0.81	77.00±1.60	55.27±1.92
EIB	91.40±0.72	74.60±1.59	76.13±2.19	55.53±3.16

Table 21. Tabular and structured reasoning under sample-set variation. Entries report mean ± sample standard deviation in percent across three 500-instance sample sets.

A.10 Artifact and Reproducibility

The benchmark artifact separates task loading, baseline construction, backend execution, answer parsing, metric calculation, cost accounting, and intervention diagnosis. Each method is invoked through a common runner, making the raw prediction, parsed answer, metric result, trace, and cost fields comparable across single-agent, ordinary MAS, and graph-enhanced MAS configurations.

Each execution trace stores a method identifier, dataset identifier, sample identifier, organization hash, prompt-template hash, backend configuration, random seed when used, final prediction, parsed prediction, metric result, and cost fields. For every executed unit, the trace records role name, incoming message identifiers, outgoing message identifiers, model-call metadata, token counts, latency, retry status, and termination reason. For intervention runs, the trace additionally stores intervention parameters and changed graph or runtime elements.

The artifact specification includes dataset loaders, evaluation samples, baseline adapters, organization configurations, intervention operators, predictions, traces, cost logs, and aggregation scripts.

The final G-MAS-Complex result package records version v6.3.2 / 3.3.2-rc5, 400 frozen tasks, 17 configurations, and 6,800 execution traces. All 136 evaluation batches passed audit. The direct reference mappings are DeepSeek Direct to deepseek-v4-flash and Qwen Direct to qwen-plus. All other named methods use their official implementations through the common benchmark interface. All methods use the same frozen tasks. Qwen latency comes from local deployment, and the other latency measurements come from server deployment.

Reproducibility records. The artifact contains three linked record types. A configuration record stores the dataset list, sample manifest, backend setting, prompt template, organization hash, and intervention parameters. A scoring record stores the parser version, metric name, metric input, parsed prediction, invalid-parse flag, and score. An aggregation record stores the included datasets, metric weights, uncertainty source, method grouping, and the rule for selecting representative configurations. These records make the full tables auditable and preserve dataset-level metric identities even when results are summarized by domain or method family.

B SUPPLEMENTARY ORGANIZATIONAL DIAGNOSIS

B.1 ILLUSTRATIVE MULTI-DOCUMENT TASK

This constructed example illustrates the dependency and conflict logic of G-MAS-Complex. The query asks for the available inventory of product P and the identifiers of the records supporting the calculation. The document collection contains the following records.

Record	Content
D1	Initial inventory of product P is 12 units.
D2	Revised inventory of product P is 9 units. This record supersedes D1.
D3	The reservation for product P is 4 units.

The correct answer gives `available_units` as 5. It sets `inventory_source` to D2 and `reservation_source` to D3. Extraction locates the records, revision checking selects D2, calculation obtains $9 - 4 = 5$, and verification checks the value and source identifiers. Two illustrative error cases distinguish these requirements. Selecting D1 produces $12 - 4 = 8$, while returning 5 without D3 omits a required source.

A collaboration graph can assign extraction, revision checking, calculation, and integration to separate units. The illustrative records expose the task's dependency structure and distinguish record selection, numerical calculation, and source reporting within the answer contract.

B.2 SIX-COMPONENT SCORING

G-MAS-Complex records accuracy for each of the six output components and strict exact match for their joint agreement. The same checks apply to all 400 tasks. Strict exact match counts a task only when every component matches.

Output component	Comparison with the reference
Final-layer values	All requested final-layer values agree.
Mission result	The overall task result agrees.
Global checksum	The returned global checksum agrees.
Final-layer ranking	The returned ordering of final-layer items agrees.
Source documents	The returned source-document identifiers agree.
Critical checkpoints	The returned critical checkpoint records agree.

These checks distinguish numerical completion from satisfying the full task. For example, a correct mission result accompanied by an incorrect source-document set passes the mission-result component and fails strict exact match. Similarly, correct final-layer values accompanied by an incorrect ranking fail the complete task. The shared contract requires systems to retain source identifiers and checkpoint information while combining intermediate results.

The evaluated collection is the union of Test-A and Test-B from v6.3.2, with 100 tasks per tier. Table 3 gives the input length, rejected-record count, and Oracle graph size associated with each tier. Every method is evaluated on this same 400-task collection, linking component scores, strict exact match, and resource measurements to the same task set.

B.3 REPRODUCIBILITY RECORD SCHEMA

The following schema defines the information needed to reconstruct a comparison. It specifies fields for dataset manifests, adapter configurations, and execution records. A comparison identifier joins these sources with predictions and scores.

Record group	Required fields
Dataset	Version, split, filters, sample identifiers, order, sampling seed, frozen manifest, and file hashes.
Model	Exact version, request date, reasoning mode, decoding parameters, context limit, and output limit.
Adapter	Implementation revision, configuration hash, retained mechanisms, changes, role prompts, and stopping rules.
Learning	Training and validation manifests, test isolation, objective, checkpoint, and graph-generation procedure.
Tools	Tool permissions, verification access, candidate selection, and format-repair rules.
Shared state	State entries readable and writable by each unit, with message delivery and state access distinguished.
Run and score	Run identifier, random seed, parser and metric versions, score unit, aggregation rule, and reference answers.
Cost and failure	Tokens, calls, retries, latency, timeouts, parsing and tool failures, termination status, concurrency, caching, and hardware.

For candidate replacement, the record includes eligible items, selected items, replacement content, and credibility markers in both reference and modified conditions. For worker failures, it includes the worker population, selection rule, injection times, duration, and whether the count represents simultaneous unavailability or accumulated events. A paired record joins reference and modified executions through the same task and configuration identifiers. It records the intervention type, selected items, random seed, schedule, and achieved modification level. Predictions, scores, and resource records share the execution identifier.

B.4 Organization Analysis Protocol

Graph construction and execution describe different properties. GPTSwarm optimizes prompts and connectivity (Zhuge et al., 2024). GDesigner learns task-dependent graph generation (Zhang et al., 2025). ARGDes sequentially generates roles and links (Li et al., 2026). Configuration records can describe these properties through the following attributes.

Attribute	Recorded configuration
Topology	Fixed, generated for each task, or updated during execution.
Learning	Optimized prompts, graph parameters, routing rules, or policies.
Roles	Unit definitions and differences in prompts, tools, and models.
Verification	Checks available to each unit and their invocation rules.
Recovery	Retry, fallback, or rerouting rules and termination conditions.

Graph records associate task identifiers with initial nodes, directed edges, updates, and executed messages. For a simple directed graph with $n \geq 2$ nodes and m edges, density is $m/[n(n-1)]$. Let $\mathcal{R}$ contain reachable ordered pairs of distinct nodes. The reachable-pair fraction and average reachable path length are

$$R_G = \frac{|\mathcal{R}|}{n(n-1)}, \qquad L_G = \frac{1}{|\mathcal{R}|} \sum_{(u,v)\in\mathcal{R}} d_G(u,v). \tag{15}$$

Here d_G is shortest directed path length, and L_G requires $|\mathcal{R}| > 0$. Comparing constructed edges with executed message edges connects available routes with their use. For executions containing messages, let p_v be the fraction originating from node v. Normalized source entropy is

$$H_G = -\frac{1}{\log n} \sum_{v\in V} p_v \log p_v, \tag{16}$$

with zero-probability terms contributing zero. Role-level message counts and temporal records describe communication before and after an intervention.

B.5 Official Implementation Validation Protocol

Validation of official implementations records the upstream source revision, task split, backend, prompts, graph construction, optimization, decoding, stopping rule, and metric used for each

method. Its comparison record contains the published score, reproduced score, their difference, and repetition count. A subsequent benchmark evaluation applies the shared task presentation and scoring controls. Method-specific roles and coordination rules remain part of the adapter description. This sequence links validation in the original setting to benchmark evaluation.

B.6 Follow-up Comparisons and Statistical Records

Future budget-matched comparisons can evaluate direct single-agent execution, iterative planning and verification, independent candidate generation, and representative multi-agent configurations under common model, tool, input, and resource limits. Actual consumption accompanies task scores. Candidate selection uses information available to the evaluated system. Role-replacement comparisons can preserve nodes and connections while replacing specialist prompts with general roles or exchanging role prompts. These settings test role definitions.

Further rewiring comparisons can check output reachability, scheduling validity, and schema compatibility for each proposed organization, alongside degree preservation. Repeated graph samples record seeds, accepted swaps, actual replacement ratios, and execution costs. Recovery comparisons can apply an identical failure schedule to the same configuration with recovery enabled and disabled, recording accuracy, retries, tokens, and latency.

Paired statistical analysis resamples common task identifiers and computes score differences within each replicate. The resampling specification groups generated tasks by template family to account for dependence among related instances. The analysis record specifies the resampling unit, repetition count, interval method, and central estimate. Separate run identifiers track execution randomness, while sample manifests track variation from dataset selection.

References

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38, 2026.

Hongjiang Chen, Xin Zheng, Yixin Liu, Pengfei Jiao, Shiyuan Li, Huan Liu, Zhidong Zhao, Ziqi Xu, Ibrahim Khalil, and Shirui Pan. Goagent: Group-of-agents communication topology generation for llm-based multi-agent systems, 2026. URL https://arxiv.org/abs/2603.19677.

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, Josh 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, 2021a. URL https://arxiv.org/abs/2107.03374.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification, 2020. URL https://arxiv.org/abs/1909.02164.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. FinQA: A dataset of numerical reasoning over financial data. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, Online and Punta Cana, Dominican Republic, November 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300/.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6279–6292, Abu

Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.421. URL https://aclanthology.org/2022.emnlp-main.421/.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023. URL https://arxiv.org/abs/2305.14325.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile van Krieken, and Pasquale Minervini. Are we done with mmlu?, 2025. URL https://arxiv.org/abs/2406.04127.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21/.

Guangfu Hao, Yuming Dai, Xianzhe Qin, and Shan Yu. Brain-inspired graph multi-agent systems for llm reasoning, 2026. URL https://arxiv.org/abs/2603.15371.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021a. URL https://arxiv.org/abs/2009.03300.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *CoRR*, abs/2103.03874, 2021b. URL https://arxiv.org/abs/2103.03874.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structures for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 209–220. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1020. URL https://aclanthology.org/P17-1020/.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421. URL https://www.mdpi.com/2076-3417/11/14/6421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2567–2577, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1259. URL https://aclanthology.org/D19-1259/.

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, 2026. doi: 10.1609/aaai.v40i28.39481. URL https://ojs.aaai.org/index.php/AAAI/article/view/39481.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.427. URL https://aclanthology.org/2024.findings-emnlp.427/.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, Vancouver, Canada, July 2017. Association for

Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015/.

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

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, 2023. URL https://arxiv.org/abs/2305.01210.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration, 2024. URL https://arxiv.org/abs/2310.02170.

Haoran Luo, Haihong E, Guanting Chen, Qika Lin, Yikai Guo, Fangzhi Xu, Zemin Kuang, Meina Song, Xiaobao Wu, Yifan Zhu, and Luu Anh Tuan. Graph-r1: Towards agentic graphrag framework via end-to-end reinforcement learning, 2026. URL https://arxiv.org/abs/2507.21892.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 975–984, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.92. URL https://aclanthology.org/2020.acl-main.92/.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann (eds.), *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 07–08 Apr 2022. URL https://proceedings.mlr.press/v174/pal22a.html.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In Chengqing Zong and Michael Strube (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1142. URL https://aclanthology.org/P15-1142/.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.168. URL https://aclanthology.org/2021.naacl-main.168/.

Qwen Team. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In Lluís Màrquez, Chris Callison-Burch, and Jian Su (eds.), *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1743–1752, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1202. URL https://aclanthology.org/D15-1202/.

Xu Shen, Yixin Liu, Yiwei Dai, Yili Wang, Rui Miao, Yue Tan, Shirui Pan, and Xin Wang. Understanding the information propagation effects of communication topologies in llm-based multi-agent systems, 2025. URL https://arxiv.org/abs/2505.23352.

Haochen Sun, Shuwen Zhang, Lujie Niu, Lei Ren, Hao Xu, Hao Fu, Fangkun Zhao, Caixia Yuan, and Xiaojie Wang. Collab-overcooked: Benchmarking and evaluating large language models as collaborative agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4922–4951, 2025.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

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 conversation, 2023. URL https://arxiv.org/abs/2308.08155.

Tianyang Xu, Dan Zhang, Kushan Mitra, and Estevam Hruschka. Verification-aware planning for multi-agent systems, 2025. URL https://arxiv.org/abs/2510.17109.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL https://arxiv.org/abs/2305.10601.

Yanwei Yue, Guibin Zhang, Boyang Liu, Guancheng Wan, Kun Wang, Dawei Cheng, and Yiyan Qi. Masrouter: Learning to route llms for multi-agent systems, 2025. URL https://arxiv.org/abs/2502.11133.

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, 2025. URL https://arxiv.org/abs/2410.11782.

Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning, 2017. URL https://arxiv.org/abs/1709.00103.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 3277–3287, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.254. URL https://aclanthology.org/2021.acl-long.254/.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. MultiAgentBench : Evaluating the collaboration and competition of LLM agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8580–8622, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.421. URL https://aclanthology.org/2025.acl-long.421/.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Language agents as optimizable graphs, 2024. URL https://arxiv.org/abs/2402.16823.