GEMMAS: Graph-based Evaluation Metrics for Multi Agent Systems

Jisoo Lee[*]
Seoul National University
sally66890@snu.ac.kr

Raeyoung Chang[*]
Sogang University
icanry@sogang.ac.kr

Dongwook Kwon[*]
Kwangwoon University
dongwook.kwon@kw.ac.kr

Harmanpreet Singh[†]
LG Electronics, Toronto AI Lab
harmanpreet.singh@lge.com

Nikhil Verma[†]
LG Electronics, Toronto AI Lab
nikhil.verma@lge.com

Abstract

Multi-agent systems built on language models have shown strong performance on collaborative reasoning tasks. However, existing evaluations focus only on the correctness of the final output, overlooking how inefficient communication and poor coordination contribute to redundant reasoning and higher computational costs. We introduce GEMMAS, a graph-based evaluation framework that analyzes the internal collaboration process by modeling agent interactions as a directed acyclic graph. To capture collaboration quality, we propose two process-level metrics: Information Diversity Score (IDS) to measure semantic variation in inter-agent messages, and Unnecessary Path Ratio (UPR) to quantify redundant reasoning paths. We evaluate GEMMAS across five benchmarks and highlight results on GSM8K, where systems with only a 2.1% difference in accuracy differ by 12.8% in IDS and 80% in UPR, revealing substantial variation in internal collaboration. These findings demonstrate that outcome-only metrics are insufficient for evaluating multi-agent performance and highlight the importance of process-level diagnostics in designing more interpretable and resource-efficient collaborative AI systems.

1 Introduction

Large language models (LLMs) such as GPT-4 (Achiam et al., 2023), Llama (Touvron et al., 2023), and Qwen (Bai et al., 2023) demonstrate emergent reasoning capabilities and achieve state-of-the-art performance across a variety of NLP tasks. As multi-agent LLM systems become more prevalent (Shen et al., 2023; Chen et al., 2023), their evaluation remains narrowly focused on the correctness of the final answer. This outcome-centric view overlooks the underlying collaboration dynamics, how agents share information, coordinate reasoning, and avoid duplication of efforts.

Recent studies highlight the limitations of this evaluation and call for process-level metrics that assess the quality of intermediate reasoning steps (Liu et al., 2023). In practice, multi-agent systems often re-traverse the same inference paths or underutilize some agents entirely. Redundant messages can significantly inflate token usage, introducing a communication tax that increases both latency and computational cost (Zhang et al., 2024a).

Existing evaluation metrics fail to expose these inefficiencies. Representing agent interactions as a directed graph, where nodes denote agents and edges represent message passing, provides a structured view of coordination patterns (Zhang et al., 2024b). This view enables the identification of redundant reasoning paths and inactive agents, offering actionable insight into the behavior of the system. Furthermore, smaller open-source LLMs have demonstrated competitive performance with significantly reduced cost, in some cases up to 94% lower than proprietary models (Liu et al., 2023; Zhang et al., 2024a). These trends motivate the need for evaluation methods that go beyond accuracy and help practitioners build more interpretable and efficient multi-agent systems.

Motivated by this gap, we present GEMMAS, a graph-based evaluation framework for analyzing multi-agent LLM reasoning processes. GEMMAS encodes the entire reasoning trace as a directed acyclic graph (DAG), where each node represents an agent having (prompt, response) pair and each edge captures the flow of information between them. We introduce two structure-aware metrics: (1) Information Diversity Score (IDS), which quantifies the semantic uniqueness of agent contributions, and (2) Unnecessary Path Ratio (UPR), which measures the fraction of reasoning steps that do not contribute new information. Together, these metrics evaluate collaboration efficiency beyond

[*] Equal Contribution.
[†] Corresponding author.

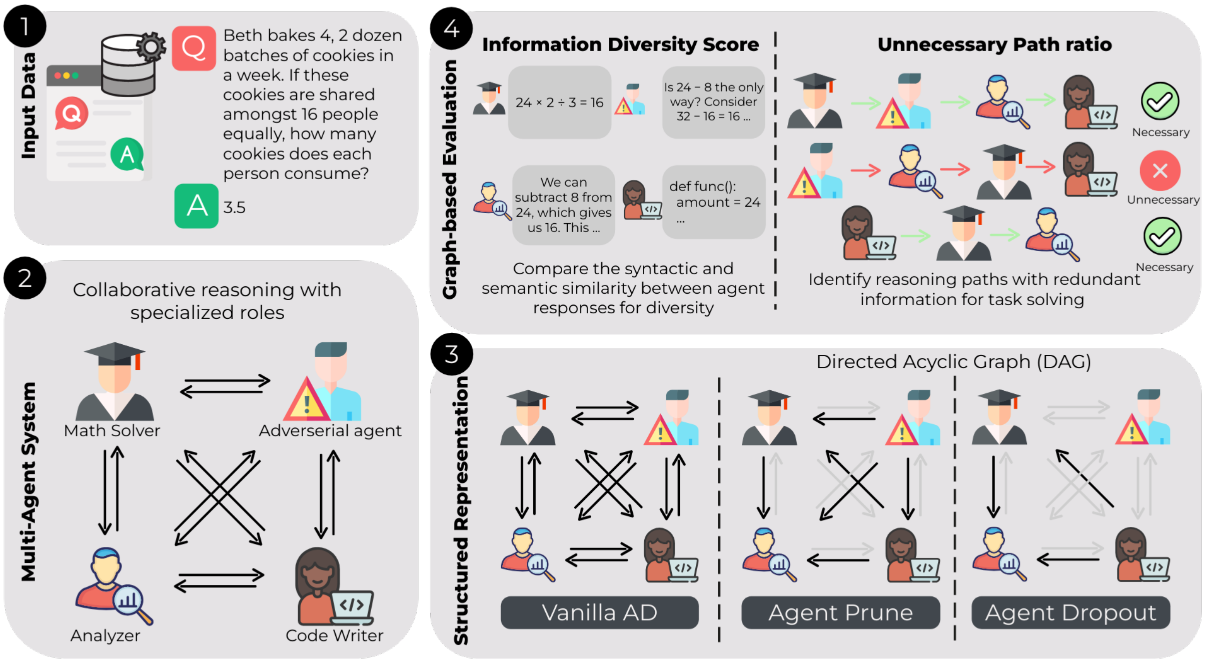

1
Input Data
Q
Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?
A
3.5
2
Multi-Agent System
Collaborative reasoning with specialized roles
Math Solver
Adverserial agent
Analyzer
Code Writer
4
Graph-based Evaluation
Information Diversity Score
24 × 2 ÷ 3 = 16
Is 24 − 8 the only way? Consider 32 − 16 = 16 ...
We can subtract 8 from 24, which gives us 16. This ...
def func(): amount = 24 ...
Compare the syntactic and semantic similarity between agent responses for diversity
Unnecessary Path ratio
Necessary
Unnecessary
Necessary
Identify reasoning paths with redundant information for task solving
3
Structured Representation
Directed Acyclic Graph (DAG)
Vanilla AD
Agent Prune
Agent Dropout

Figure 1: **Overview of the GEMMAS evaluation framework.** The process begins with input mathematical problems, which are solved collaboratively by a multi-agent system composed of specialized agents. Their interactions are represented as a DAG, capturing both communication flow and reasoning structure. From this DAG, GEMMAS computes structural metrics—Information Diversity Score (IDS) and Unnecessary Path Ratio (UPR)—to evaluate collaboration quality and efficiency beyond final-task accuracy.

what is captured by task accuracy alone.

We apply GEMMAS to five mathematical reasoning benchmarks: GSM8K, AQuA, MultiArith, SVAMP, and MMLU using lightweight open-source LLMs. Our findings show that naïve agent pipelines suffer from high redundancy and low diversity, while configurations optimized for higher IDS and lower UPR improve both accuracy and token efficiency under fixed computational budgets. GEMMAS thus surfaces hidden inefficiencies in multi-agent collaboration and offers practical design signals for building interpretable and cost-effective agent systems, an especially important concern in real-world industry deployments.

Our contributions are as follows:

- We introduce GEMMAS, a graph-based evaluation framework for multi-agent LLM reasoning, and propose two novel structural metrics: Information Diversity Score (IDS) and Unnecessary Path Ratio (UPR), which together assess collaboration quality beyond task accuracy.

- We evaluate multiple small open-source LLMs, using GEMMAS to compare their collaborative behavior and identify efficiency trade-offs.

- We conduct systematic evaluations across five mathematical reasoning benchmarks, revealing substantial differences in collaboration quality among systems with similar final accuracies.

2 Related Work

2.1 Agent-Level Diversity Metrics

Standard metrics (e.g., BLEU, cosine similarity) capture surface-level variation (Zhu et al., 2018; Li et al., 2015) but ignore semantic roles and the graph structure critical for coordination (Li et al., 2015; Park et al., 2024). Well-orchestrated lightweight models can match the performance of larger models in reasoning and planning (Liu et al., 2023), making efficient evaluation of agent-level diversity increasingly important.

Agent-level diversity in multi-agent systems requires specialized evaluation that considers how individual agents contribute unique perspectives to collaborative reasoning through their structural connectivity within the communication graph. These limitations have motivated recent efforts to develop graph-aware metrics that consider both content and structural diversity in multi-agent reasoning.

2.2 Evaluation in Multi-Agent Reasoning

Evaluation of multi-agent reasoning remains in its early stages. Traditional metrics such as final accuracy or task success rate often overlook the internal dynamics of agent collaboration.

To address limitations of conventional evaluations, graph-based methods have been proposed. *AgentPrune* (Zhang et al., 2024a) prunes low impact edges in the communication graph, while *AgentDropout* (Wang et al., 2025) removes underperforming agents, both with the aim of reducing redundancy without compromising output quality. *VillagerAgent* (Dong et al., 2024) incorporates DAG based planning to assess workload balance and depth of reasoning.

These graph-based approaches highlight the growing emphasis on structural analysis to uncover inefficiencies and redundant communication patterns in collaborative reasoning among agents.

3 Evaluation Framework: GEMMAS

In this section, we introduce **GEMMAS** (General Evaluation Metrics for Multi-Agent Systems), a comprehensive evaluation framework for graph-based multi-agent LLM systems. Unlike conventional approaches that focus solely on task outcomes, GEMMAS evaluates both the final results and the internal reasoning process of multi-agent collaboration.

3.1 Task Definition and Problem Setup

We consider the problem of multi-agent collaboration for mathematical reasoning tasks, where multiple language model agents interact through structured communication to solve problems jointly. Each multi-agent system (MAS) is modeled as a DAG, capturing the flow of information across agents throughout the reasoning process.

Formally, let $G = (V, E, F)$ denote the communication graph:

- $V = \{v_1, v_2, ..., v_N\}$ is the set of N agent nodes;
- $E \subseteq V \times V$ represents directed edges, where $(v_i, v_j) \in E$ indicates that the output of agent v_i is available to agent v_j;
- $F = \{f_1, f_2, ..., f_N\}$ denotes the set of agent-specific reasoning functions, where f_i defines the behavior or prompt processing logic of agent v_i.

To analyze both the communication structure and the temporal execution dynamics of the system, we maintain two adjacency matrices. The *spatial adjacency matrix* $S \in \{0, 1\}^{N \times N}$ encodes direct communication links between agents, indicating which agents can exchange information. Complementarily, the *temporal adjacency matrix* $T \in \{0, 1\}^{N \times N}$ captures the causal or time-ordered dependencies among agent outputs, allowing us to trace how intermediate reasoning steps influence one another across time.

GEMMAS evaluates multi-agent systems beyond final-task accuracy by analyzing the spatial and temporal structures of agent communication. This reveals inefficiencies such as redundant reasoning, low diversity, shallow chains, and idle agents, patterns not captured by conventional metrics. While we report traditional baselines including accuracy (correct task completion rate) and token efficiency (prompt and completion token usage) (Wang et al., 2025), GEMMAS provides process-level insights that quantify collaboration quality and structural resource utilization. By modeling both the topology and semantic content of communication graphs, it addresses a key limitation in current evaluation practices for multi-agent LLM systems.

3.2 DAG-specific Metrics

Information Diversity Score (IDS). This metric quantifies the heterogeneity of information generated by different agents by measuring the degree of similarity between their responses. It addresses a fundamental question in collaborative systems: *Do agents contribute unique perspectives, or do they merely repeat similar reasoning?*

Existing evaluation metrics are limited in multi-agent contexts, as they primarily capture surface-level correctness and ignore semantic intent or the structural role of each agent within the communication graph. To address this gap, we combine syntactic analysis using TF-IDF (Sparck Jones, 1972) with semantic similarity computed via BERT embeddings (Devlin et al., 2019), while incorporating structural context from the DAG. To account for the topology of the agent graph, we weight each agent pair (i, j) based on their spatial and temporal proximity within the DAG. The Information Diversity

Score is defined as:

$$IDS = \frac{\sum_{i,j} w_{ij} \cdot (1 - SS_{\text{total}}[i,j])}{\sum_{i,j} w_{ij}} \tag{1}$$

$$w_{ij} = \max(S_{ij}, S_{ji}) + \max(T_{ij}, T_{ji}) \tag{2}$$

Here, $SS_{\text{total}}[i,j]$ represents the average syntactic-semantic similarity between agents i and j, computed as the cosine similarity of their TF-IDF and BERT representations, weighted equally with $\lambda_1 = \lambda_2 = 0.5$. The weight w_{ij} captures the relevance of agent pair (i,j) based on their direct or indirect communication, using spatial adjacency matrix S and temporal adjacency matrix T.

The complete algorithmic procedure for computing IDS, including similarity calculation and structure-aware weighting, is described in Algorithm 1. Sensitivity analysis for different weighting schemes is provided in Appendix A.1.

Unnecessary Path Ratio (UPR). This metric assesses the structural efficiency of the MAS by identifying reasoning paths that provide negligible or redundant contributions to solving the task. While IDS focuses on diversity, UPR addresses efficiency, quantifying the proportion of communication paths that fail to add meaningful information. It serves as an indicator of communication overhead and redundancy in agent interactions.

Formally, UPR is defined as:

$$UPR = 1 - \frac{|\mathcal{P}_{\text{necessary}}|}{|\mathcal{P}_{\text{all}}|} \tag{3}$$

where $\mathcal{P}_{\text{all}}$ represents the total number of reasoning paths in the spatial communication graph, and $\mathcal{P}_{\text{necessary}}$ includes only those paths that yield contribution scores above a predefined threshold.

A path is deemed necessary if it facilitates the production of correct or informative responses by downstream agents, based on a contribution function defined over message impact. The detailed algorithmic pipeline, which includes path enumeration, contribution analysis, and threshold filtering, is outlined in Algorithm 2.

3.3 Evaluation Setup

We systematically evaluate different MAS architectures that vary in their communication graph topologies. Our evaluation encompasses both baseline and structurally optimized approaches to assess the effectiveness of DAG-based modifications.

Algorithm 1 Information Diversity Score

```
Input: Agent responses $O = \{o_1, \dots, o_N\}$,
       Spatial adjacency matrix $S$,
       Temporal adjacency matrix $T$
Output: Information diversity score $IDS \in [0,1]$
  /* Calculating syntactic-semantic similarity */
  Obtain syntactic features $\Phi \leftarrow$ TF-IDF(O)
  Obtain semantic features $\Psi \leftarrow$ BERT(O)
  $SS_{\text{syn}} \leftarrow$ pairwise_cosine($\Phi$)    {syntactic similarity}
  $SS_{\text{sem}} \leftarrow$ pairwise_cosine($\Psi$)    {semantic similarity}
  $SS_{\text{total}} \leftarrow \lambda_1 \cdot SS_{\text{syn}} + \lambda_2 \cdot SS_{\text{sem}}$
  /* Calculating diversity score */
  Initialize weighted diversity $D_w \leftarrow 0$
  Initialize DAG connection weights $W \leftarrow 0$
  for $i = 1$ to $N - 1$ do
    for $j = i + 1$ to $N$ do
      $w \leftarrow \max(S_{ij}, S_{ji}) + \max(T_{ij}, T_{ji})$ {connection weights}
      if $w > 0$ then
        $D_w \leftarrow D_w + w \cdot (1 - SS_{total}[i,j])$
        $W \leftarrow W + w$
      end if
    end for
  end for
  return $D_w / W$
```

Baseline. We employ a fully connected multi-agent system without structural optimization, where all agents can directly communicate with each other. We refer to this baseline configuration as Vanilla-AD, as a reference point for measuring the effectiveness of communication graph modifications.

3.3.1 Multi-Agent System Architecture

We adopt the agent-role configuration established in AgentPrune (Zhang et al., 2024a), which defines a structured collaboration protocol among specialized agents to promote diversity and robustness in reasoning.

Our MAS setup involves four specialized agents with distinct roles. These are 1) AnalyzeAgent focuses on problem decomposition and structured plan generation, 2) CodeWritingAgent formulates code-based computational reasoning strategies, 3) MathSolverAgent performs formal mathematical operations and symbolic solving and 4) AdversarialAgent introduces plausible but intentionally incorrect solutions to stress-test robustness.

The collaborative process unfolds in three stages. First, the input problem is distributed to all four agents concurrently. Second, the agents engage in two rounds of communication based on the speci-

Algorithm 2 Unnecessary Path Ratio

```
Input: Spatial communication graph $G = (V, E_{\text{spatial}})$,
       Correct answer $\alpha$
Output: Unnecessary path ratio UPR $\in [0, 1]$
  /* Path enumeration */
  $\mathcal{P}_{all} \leftarrow \{p \mid p \text{ is a subpath in } G\}$
  $\mathcal{P}_{necessary} \leftarrow \emptyset$
  /* Path contribution assessment */
  for each path $p \in \mathcal{P}_{all}$ do
      Initialize correct count $c \leftarrow 0$
      Initialize total count $t \leftarrow 0$
      for each agent $v \in p$ do
          $a \leftarrow \text{ExtractAnswer}(\text{output}(v))$
          if $a == \alpha$ then
              $c \leftarrow c + 1$
          end if
          $t \leftarrow t + 1$
      end for
      score $\leftarrow \frac{c}{t}$ if $t > 0$ else 0        {contribution score}
      if score $\geq 0.5$ then
          $\mathcal{P}_{\text{necessary}} \leftarrow \mathcal{P}_{\text{necessary}} \cup \{p\}$
      end if
  end for
  return $1 - |\mathcal{P}_{necessary}|/|\mathcal{P}_{all}|$
```

fied DAG topology, sharing intermediate reasoning traces from their specialized perspectives. Finally, a FinalRefer agent aggregates these outputs to generate the final answer through collective reasoning.

Figure 2 illustrates the structural evolution of the DAG topology, comparing the initial structure at Iteration 1 with the optimized structure at Iteration 10 produced by the G-Designer method. To analyze the effect of structural optimization, we implement and compare three state-of-the-art methods called 1) AgentPrune prunes communication links with low marginal impact, 2) AgentDropout dynamically removes underperforming agents and their associated links to reduce redundancy and 3) G-Designer learns optimal DAG topologies over multiple iterations, aiming to improve reasoning efficiency by minimizing communication overhead.

3.3.2 Benchmarks and Language Models

We evaluate DAG-based MAS architectures on five mathematical reasoning benchmarks: GSM8K (Cobbe et al., 2021), which contains 1,000 grade-school math problems requiring multistep numeric reasoning; AQuA (Ling et al., 2017), consisting of 254 algebraic word problems with multiple-choice answers; MMLU (Hendrycks et al., 2021), where we select 748 questions from the mathematics subsets covering elementary to college-level topics; MultiArith (Roy and Roth, 2015), with 180 arithmetic word problems; and SVAMP (Patel et al., 2021), comprising 300 elementary-level problems designed to test reasoning variation. To examine how GEMMAS performs across different model scales, we use two small open-source instruction-tuned language models: Llama 3.1–8B-Instruct (Grattafiori et al., 2024) and Qwen 2.5–7B-Instruct (Yang et al., 2025).

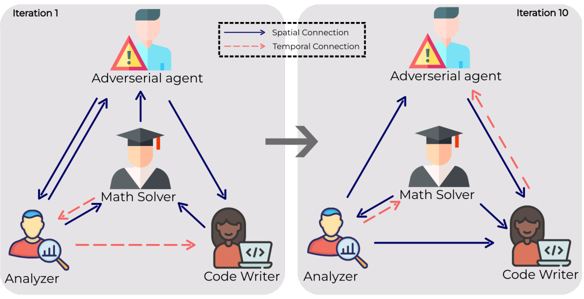


Figure 2: **Comparison of Multi-Agent DAG Structures Before and After Optimization.** The figure shows the evolution of the DAG structure from the initial setup (Iteration 1) to the final optimized configuration (Iteration 10). Solid blue lines denote spatial communication links, while dashed orange arrows indicate temporal dependencies.

3.3.3 Implementation Details

All experiments are run under consistent hyperparameters: a learning rate of 0.1, a dropout rate of 0.1, 40 training examples, 10 sampling iterations, and two communication rounds per task. We set the generation temperature to zero for deterministic outputs. Unless stated otherwise, all other parameters follow the default configurations of the original MAS frameworks.

4 Results and Analysis

We apply GEMMAS to assess the structural quality and collaborative behavior of multi-agent systems (MAS). Tables 1 and 2 present the results for both Llama3.1–8B-Instruct and Qwen2.5–7B-Instruct models across five reasoning benchmarks.

4.1 Revealing Hidden Inefficiencies

Conventional evaluation metrics, such as final answer accuracy, fail to capture inefficiencies in multi-agent reasoning. For instance, on GSM8K with Qwen2.5-7B-Instruct, the Vanilla-AD configuration achieves 85.6% accuracy, while G-Designer reaches 87.4%. Although these results appear similar in terms of performance, GEMMAS reveals

Method	Accuracy↑	Ptok↓	Ctok↓	IDS↑	UPR↓
GSM8K					
Vanilla - AD	0.7961	10.18	3.16	0.33	0.39
AgentDropout	0.6727	**08.01**	**2.84**	**0.52**	0.33
AgentPrune	0.6688	12.68	4.11	0.33	0.32
G-Designer	**0.8391**	11.09	3.43	0.32	**0.14**
AQuA					
Vanilla - AD	**0.6333**	**2.14**	**0.94**	**0.38**	0.46
AgentDropout	0.5833	2.37	1.07	**0.38**	0.47
AgentPrune	0.5833	2.57	1.25	0.36	**0.44**
G-Designer	0.5625	2.76	1.22	0.37	0.47
MultiArith					
Vanilla - AD	**0.9875**	1.21	0.31	0.40	0.13
AgentDropout	0.8688	**0.99**	**0.26**	0.40	0.24
AgentPrune	0.8125	1.83	0.58	**0.42**	0.06
G-Designer	0.9625	1.42	0.40	0.36	**0.01**
SVAMP					
Vanilla - AD	**0.8536**	**1.12**	**0.45**	0.63	**0.39**
AgentDropout	0.8000	1.35	0.58	**0.66**	0.46
AgentPrune	0.8107	2.86	0.82	0.38	0.96
G-Designer	0.8286	2.86	0.81	0.37	0.42
MMLU					
Vanilla - AD	0.5278	3.48	0.83	0.34	0.66
AgentDropout	0.5389	**2.47**	**0.68**	0.34	**0.62**
AgentPrune	0.5792	3.35	0.88	0.36	0.66
G-Designer	**0.7042**	5.09	2.00	**0.53**	0.70

Table 1: Performance comparison of multi-agent systems on GSM8K, AQuA, MultiArith, SVAMP, and MMLU test dataset using Llama 3.1-8B-Instruct.

Method	Accuracy↑	Ptok↓	Ctok↓	IDS↑	UPR↓
GSM8K					
Vanilla - AD	0.8563	10.15	2.59	0.39	0.40
AgentDropout	0.7797	**06.99**	**1.58**	0.40	0.41
AgentPrune	0.7508	10.01	2.68	0.41	0.16
G-Designer	**0.8742**	09.87	2.24	**0.44**	**0.08**
AQuA					
Vanilla - AD	**0.5958**	**1.82**	**0.73**	0.38	**0.31**
AgentDropout	0.5917	2.02	0.83	0.38	0.32
AgentPrune	0.5417	1.98	0.77	0.49	0.34
G-Designer	0.5042	2.06	0.80	**0.52**	0.36
MultiArith					
Vanilla - AD	0.9938	1.20	0.24	0.43	0.16
AgentDropout	0.9688	**0.86**	**0.12**	0.46	0.16
AgentPrune	0.9938	1.45	0.38	**0.57**	**0.00**
G-Designer	**1.0000**	1.12	0.21	0.54	**0.00**
SVAMP					
Vanilla - AD	0.8893	**1.02**	0.32	0.67	0.42
AgentDropout	**0.9071**	1.05	**0.29**	**0.69**	0.36
AgentPrune	0.8714	2.39	0.45	0.41	0.97
G-Designer	0.9036	2.37	0.45	0.46	**0.32**
MMLU					
Vanilla - AD	0.7153	3.13	0.61	0.54	0.51
AgentDropout	0.7181	**2.17**	**0.47**	0.63	0.53
AgentPrune	0.7319	2.71	0.62	0.49	**0.43**
G-Designer	**0.7806**	4.18	1.34	**0.72**	0.61

Table 2: Performance comparison of multi-agent systems on GSM8K, AQuA, MultiArith, SVAMP, and MMLU test dataset using Qwen2.5-7B-Instruct.

that G-Designer operates with significantly higher structural efficiency, recording a UPR of just 0.08 compared to 0.40 for Vanilla-AD, a fivefold improvement in redundant reasoning reduction.

Moreover, models with identical task accuracy may exhibit distinct internal collaboration patterns. On MultiArith with Qwen2.5-7B-Instruct, both Vanilla-AD and AgentPrune reach 99.4% accuracy. However, AgentPrune demonstrates greater semantic diversity (IDS of 0.57 versus 0.43) and higher structural efficiency (UPR of 0.00 versus 0.16), highlighting that quality of reasoning is not reflected by accuracy alone.

4.2 Identifying Optimal Configurations

Building on these findings, we identify recurring trends in structural metrics that support actionable system design decisions. Systems that simultaneously exhibit high IDS and low UPR are particularly desirable, as they combine semantic diversity with efficient communication.

For example, AgentPrune on MultiArith consistently demonstrates this pattern. With Llama3.1–8B-Instruct, it achieves IDS 0.42 and UPR 0.06, while with Qwen2.5–7B-Instruct, it yields IDS 0.57 and UPR 0.00. In contrast, systems with low IDS and high UPR represent the least efficient configurations. On SVAMP, AgentPrune records IDS 0.38 and UPR 0.96 with Llama3.1–8B-Instruct, and IDS 0.41 and UPR 0.97 with Qwen2.5–7B-Instruct, signaling redundant or repetitive communication behavior.

From a performance efficiency perspective, some configurations manage to achieve strong accuracy with minimal communication overhead. For example, G-Designer on MMLU achieves 70.4% precision with IDS 0.53 using Llama3.1–8B-Instruct, and 78.1% precision with IDS 0.72 using Qwen2.5–7B-Instruct.

These results confirm that GEMMAS exposes structural trade-offs that traditional metrics cannot capture. It offers MAS designers a set of complementary signals to guide configuration choices based on specific goals, whether that is, maximizing computational efficiency (via low UPR), enhancing semantic richness (via high IDS), or achieving a balance across both dimensions depending on the deployment scenario.

5 Conclusion

We introduced GEMMAS, a graph-based evaluation framework for multi-agent language model systems that assesses collaboration quality beyond final-task accuracy. By modeling agent interactions as a DAG, GEMMAS defines two structural metrics—Information Diversity Score (IDS) and Unnecessary Path Ratio (UPR)—to capture semantic uniqueness and reasoning redundancy. Experiments on five mathematical benchmarks reveal that systems with similar accuracy can vary significantly in internal collaboration patterns. GEMMAS thus enables process-level diagnostics to guide the development of interpretable and efficient multi-agent systems.

Limitations

While GEMMAS provides a structural lens to evaluate multi-agent LLM systems, it is currently limited to mathematical reasoning tasks and small open-source models. Extending this framework to broader domains, integrating runtime adaptivity, and coupling it with system-level metrics such as latency and memory footprint represent promising directions. Additionally, incorporating human-in-the-loop assessments and evaluating dynamic DAG topologies could further enhance the utility of GEMMAS for real-world deployment.

Acknowledgement

This work was conducted as part of a collaboration between CARTE and LG Electronics, Toronto AI Lab. We gratefully acknowledge the invaluable supervision and guidance of Homa Fashandi, Manasa Bharadwaj, and Kevin Ferreira throughout the course of this project. This research was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) [RS-2022-00143911, AI Excellence Global Innovative Leader Education Program].

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, and 1 others. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848*, 2(4):6.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186.

Yubo Dong, Xukun Zhu, Zhengzhe Pan, Linchao Zhu, and Yi Yang. 2024. VillagerAgent: A graph-based multi-agent framework for coordinating complex task dependencies in Minecraft. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16290–16314, Bangkok, Thailand. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

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

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.

Sungjin Park, Xiao Liu, Yeyun Gong, and Edward Choi. 2024. Ensembling large language models with process reward-guided tree search for better complex reasoning. *arXiv preprint arXiv:2412.15797*.

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

Association for Computational Linguistics: Human Language Technologies, pages 2080–2094.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. 2025. Agentdropout: Dynamic agent elimination for token-efficient and high-performance llm-based multi-agent collaboration. *arXiv preprint arXiv:2503.18891*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024a. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. *arXiv preprint arXiv:2410.02506*.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. 2024b. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

A Appendix

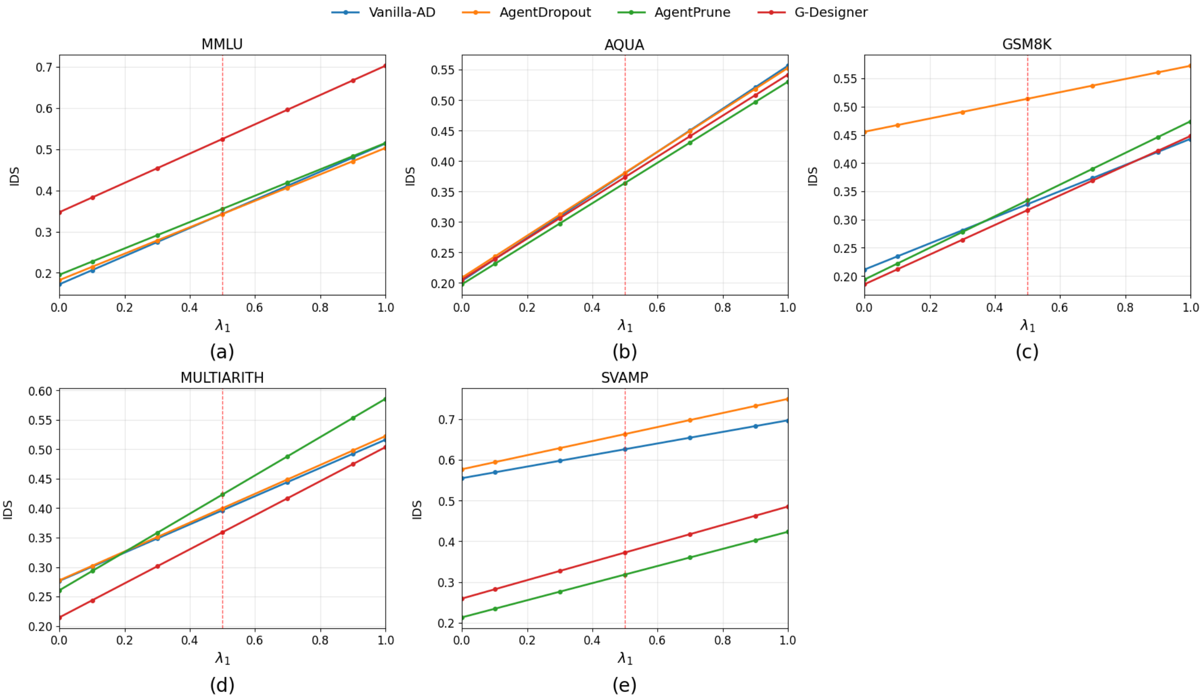
A.1 Information Diversity Score Analysis

Figures 3 and 4 show how IDS values vary with different combinations of syntactic-semantic weights between benchmarks and models. The weight balance parameter λ_1 controls the contribution of syntactic characteristics (TF-IDF), while semantic features (BERT embeddings) are weighted as $\lambda_2 = 1 - \lambda_1$.

As λ_1 increases toward 1.0, emphasizing syntactic similarity, IDS values generally increase between models and benchmarks. This suggests that syntactic diversity (e.g., different vocabulary usage, sentence structures) tends to be more pronounced than semantic diversity in multi-agent communications.

Different multi-agent systems exhibit varying sensitivity to weight balance. For example, on SVAMP with both models, Vanilla-AD and Agent-Dropout show relatively higher IDS values compared to AgentPrune and G-Designer, indicating more diverse communication patterns across different syntactic-semantic weight settings.

We selected equal weights to balance the syntactic and semantic contribution without favoring either aspect of the similarity measurement. This balanced approach provides a neutral baseline for comparing multi-agent systems across different collaboration patterns.



(a)

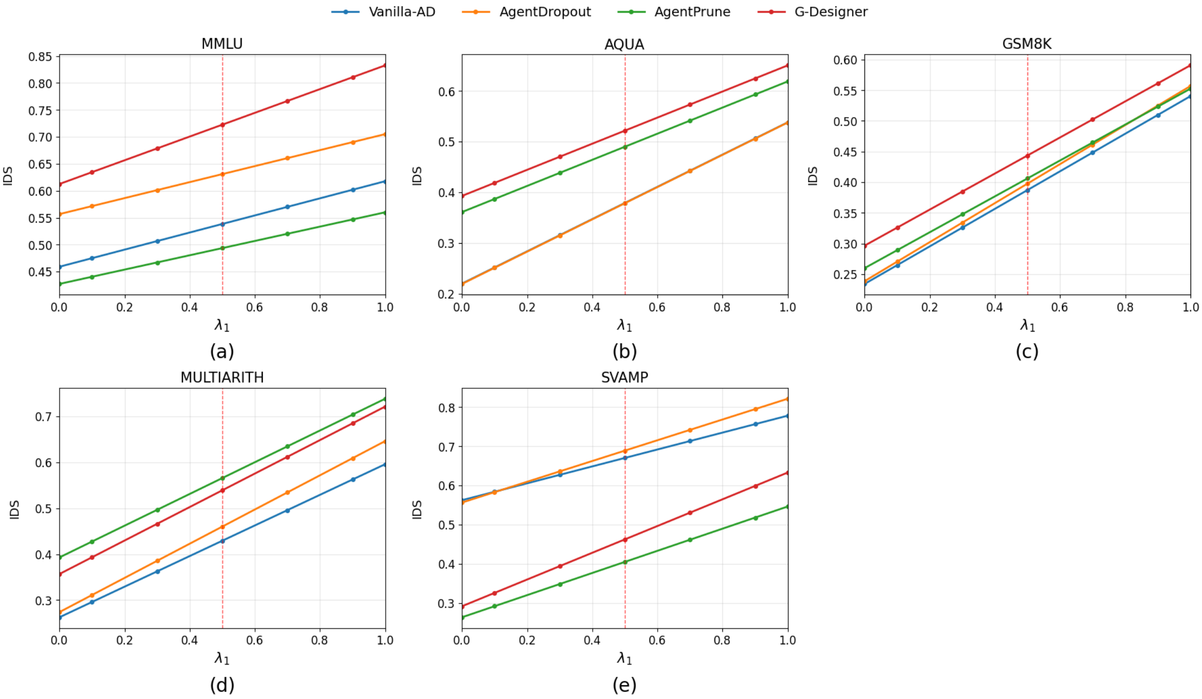
(b)

(c)

(d)

(e)

Figure 3: Sensitivity of Information Diversity Score (IDS) to syntactic-semantic weight balance across five benchmarks using Llama3.1-8B-Instruct. The x-axis shows λ_1 (syntactic weight), with vertical dashed line indicating $\lambda_1 = 0.5$ used in our main experiments.



(a)

(b)

(c)

(d)

(e)

Figure 4: Sensitivity of Information Diversity Score (IDS) to syntactic-semantic weight balance across five benchmarks using Qwen2.5-7B-Instruct. The x-axis shows λ_1 (syntactic weight), with vertical dashed line indicating $\lambda_1 = 0.5$ used in our main experiments.