HYPERBOLIC GRAPH-OF-AGENTS: A TRAINING-FREE FRAMEWORK FOR LLM-BASED MULTI-AGENT COLLABORATION IN HYPERBOLIC SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

With the continuous advancement of agents, the need to orchestrate multi-agent collaboration to enhance task performance has become increasingly urgent. Existing multi-agent collaboration methods typically use natural language text as the collaboration medium for multiple rounds of message passing, causing inference costs to grow rapidly with the number of collaboration rounds and agents, and introducing semantic drift through multiple rounds of comprehension and restatement. To this end, we propose Hyperbolic Graph-of-Agents (HGoA), a training-free framework for large language model-based multi-agent collaboration in hyperbolic space. Specifically, a novel hyperbolic-aware dual-channel graph construction method is proposed to adaptively construct an agent graph from observable outputs in a training-free manner. Next, a novel hyperbolic collaboration method based on consensus energy is proposed, which uses representation vectors instead of natural language text as the collaboration medium, thereby mitigating semantic drift while reducing inference costs. Furthermore, we leverage hyperbolic space to overcome the limitation of Euclidean space in failing to capture the hierarchical structures that exist among agent responses. Finally, experiments on six benchmarks demonstrate that, compared with mainstream methods, HGoA can improve the efficiency of multi-agent collaboration on various complex tasks.

1 INTRODUCTION

As task scenarios become increasingly diverse, single-agent systems are no longer capable of effectively handling complex tasks (Madaan et al., 2023; Agarwal & Yamada, 2026; Cookson et al., 2026). To overcome this limitation, collaborative paradigms based on multi-agent systems (MAS) have emerged (Zhang et al., 2025d; Emde et al., 2026; Kavathekar et al., 2026). Leveraging collective intelligence by coordinating different large language model-based (LLM-based) agents to achieve superior inference performance has now become a mainstream method for addressing complex real-world problems.

Existing studies have begun to enhance the collaborative performance of MAS from the perspective of topology optimization of agent graphs (Ding et al., 2024; Zhang et al., 2025c; Wang et al., 2026). Related methods can be categorized into three classes: methods based on edge importance measurement (Li et al., 2021; Zhang et al., 2025b), methods based on graph autoencoders (Salha et al., 2019; Zhang et al., 2025a), and methods based on reinforcement learning (Zhuge et al., 2024; Liu et al., 2026). These methods rely excessively on the training of agent graphs, the hidden states of LLMs, or additional environmental feedback signals to learn the agent graph structure. As a result, they may introduce extra computational and temporal costs, making them difficult to directly support purely test-time inference. **Therefore, the first motivation of this study is to explore whether an agent graph can be adaptively constructed without additional training on the agent graph, based solely on the observable outputs of agents for the current query.**

After obtaining the agent graph, achieving efficient collaboration on the graph is equally crucial. Recently, methods represented by Mixture-of-Agents (MoA) (Wang et al., 2025a) adopt a hierarchical aggregation mechanism, where the outputs of multi-agents are concatenated and passed to the next layer, as shown in Figure 1(a). Methods represented by Graph-of-Agents (GoA) (Yun et al.,

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

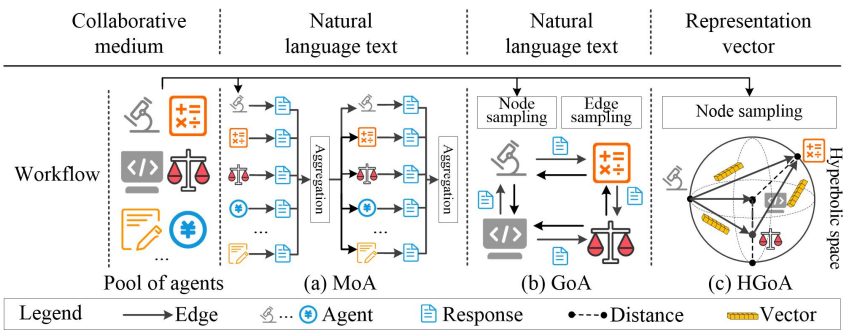


Figure 1: Comparison of the working mechanisms of MoA, GoA, and HGoA. (a) MoA integrates the responses from all agents and then generates the final response through various aggregation operations. The collaborative medium is natural language text, which tends to incur substantial inference costs. (b) GoA selects relevant agents through node sampling and subsequently constructs an agent graph via edge sampling. The collaborative medium is also natural language text. (c) HGoA likewise adopts the ideas of node sampling and graph construction for multi-agent collaboration. However, HGoA operates in hyperbolic space and employs representation vectors as the collaborative medium, thereby avoiding multiple rounds of natural language text interaction and effectively reducing inference costs.

2026) construct an agent graph through node and edge sampling, and perform bidirectional natural language message passing between adjacent agents, as shown in Figure 1(b). However, these methods rely on natural language text as the collaboration medium. On the one hand, after receiving responses from other agents, an agent typically needs to call the LLM again and regenerate its response, leading to inference cost that grows with the number of collaboration rounds and agents. On the other hand, as agents undergo multiple rounds of comprehension and restatement, semantic drift inevitably occurs.

More importantly, what multi-agent collaboration actually requires is not the exchange of complete reasoning chains, but rather the exchange of key information regarding the consistency, reliability, and related attributes of agent responses. Once such information is encapsulated and transmitted in the form of long text, the key information becomes intermixed with redundant information, hindering effective collaboration among agents. Additionally, multi-agent responses typically exhibit a hierarchical structure that progressively differentiates from shared consensus to individual dissent (Cetin et al., 2023; Qu et al., 2026). Euclidean space struggles to preserve both hierarchical distances and branching relationships when embedding such hierarchical structures (Chami et al., 2019), whereas hyperbolic space (Cetin et al., 2023) can maintain these hierarchical structures in low-dimensional representations, thereby more accurately capturing the hierarchical distances of agent responses, as shown in Figure 1(c). **Therefore, the second motivation of this study is to explore whether the repeated transmission of long texts between agents can be avoided by converting the collaborative medium from natural language text to representation vectors, and by employing hyperbolic space to characterize such hierarchical structures, thereby reducing inference costs while mitigating semantic drift.**

To this end, we propose a multi-agent collaboration framework (HGoA) for test-time inference. HGoA first performs agent selection based on model cards. Then, the responses of the agents are encoded and mapped into the Poincaré ball, constructing the agent graph structure without subjective mutual evaluation or topology training. Next, a node encoder is employed to convert the agent's initial response into a representation vector, and multi-agent collaboration in hyperbolic space is achieved by minimizing a consensus energy. In addition, we freeze the original responses to suppress semantic drift. Finally, two response parsing paths (Meta-free and Meta-select) are used to parse the outputs of multi-agent collaboration into the final response, with only a single call to the meta-LLM. The main contributions of this study are summarized as follows:

- Training-free topology optimization: We propose a novel hyperbolic-aware dual-channel graph construction method to model the topological structure of multi-agent collaboration.

- Collaborative paradigm upgrade: We propose a novel hyperbolic collaboration method based on consensus energy, which converts the collaboration medium from natural language text to representation vectors to improve the efficiency of multi-agent collaboration.
- Compared with GoA, HGoA achieves reductions of 54.55%, 59.55%, and 39.01% in the number of LLM calls, average token consumption, and average time consumption, respectively. Compared with existing mainstream methods, HGoA not only reduces inference costs but also effectively mitigates semantic drift.

2 Related Work

LLM-based Multi-Agent System Existing studies have shown that LLM-based MAS can achieve superior performance over single-agent systems by introducing multi-round discussion (Du et al., 2024), opinion interaction (Dewes & Dimitrova, 2025), and result aggregation mechanisms (Ahmad et al., 2025). The advantages of LLM-based MAS have been demonstrated on different tasks such as knowledge reasoning (Qian et al., 2024), mathematical problem solving (Gou et al., 2024; Yan et al., 2025), and code generation (Jain et al., 2025; Yang et al., 2026). As the complexity of LLM-based MAS continues to grow, existing methods still face challenges in terms of computational cost and communication efficiency (Shi et al., 2026; Chen et al., 2026). To address these challenges, Zhang et al. (2025b) proposed AgentPrune, which eliminates redundant communications through low-rank constraints and a one-shot pruning strategy. Singh et al. (2026) proposed Grammar Search, which employs context-free grammar to optimize the structure of multi-agent systems. However, they overlooked the online adaptive capabilities of agents. To address this limitation, Xu et al. (2026) and Xia et al. (2026) respectively proposed an open-world multi-agent retrieval framework and an online adaptive mechanism based on contribution attribution. However, none of these methods have explored how to adaptively construct the agent graph.

Multi-Agent Systems as Graphs Before LLMs became widely adopted, some studies had already explored optimal communication topologies for MAS within reinforcement learning frameworks. Representative methods include graph diffusion mechanisms (Pesce & Montana, 2023), graph neural networks (Liu et al., 2022), and Transformer architectures (Hu et al., 2024). With the rise of LLM-based MAS, graph structures have once again been used to characterize the interactions among agents. For example, ChatEval (Chan et al., 2024) validated the effect of multiple rounds of debates on collaborative performance under graph topologies through diverse role configurations. GPTSwarm (Zhuge et al., 2024), G-Designer (Zhang et al., 2025c), and MacNet (Qian et al., 2025) all adopted directed acyclic graphs to model collaborative structures. However, these methods struggled to dynamically adjust the role configurations of agents. To address this, AgentDropout (Wang et al., 2025b) leveraged node pruning strategies to dynamically adjust the roles of agents. In addition, Graph-of-Agents (GoA) (Yun et al., 2026) and GraphCogent (Wang et al., 2026) further optimized the multi-agent collaboration through message passing mechanisms and graph sampling strategies, respectively. However, the aforementioned methods still require multiple rounds of natural language text interaction on the agent graph, and therefore cannot exploit the advantages of representation vectors to reduce inference costs.

3 Methodology

3.1 Preliminaries

MAS is defined as an agent graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ denotes the set of nodes, and $\mathcal{E} = \{(v_i, v_j)|1 \leq i, j \leq n\}$ denotes the set of directed edges. Each node corresponds to an agent. Given a query q, k relevant agents are selected from a pool of n agents to construct a new agent graph. The adjacency matrix of this graph is $\mathbf{A} \in \mathbb{R}^{k \times k}$. Finally, the MAS collaborates on this adjacency matrix and generates the final response according to the query q.

3.2 Overall Structure

HGoA mainly consists of four components: Agent Selection, Agent Graph Construction, Agent Collaboration, and Response Generation, as shown in Figure 2. These four components constitute the core pipeline of multi-agent collaboration.

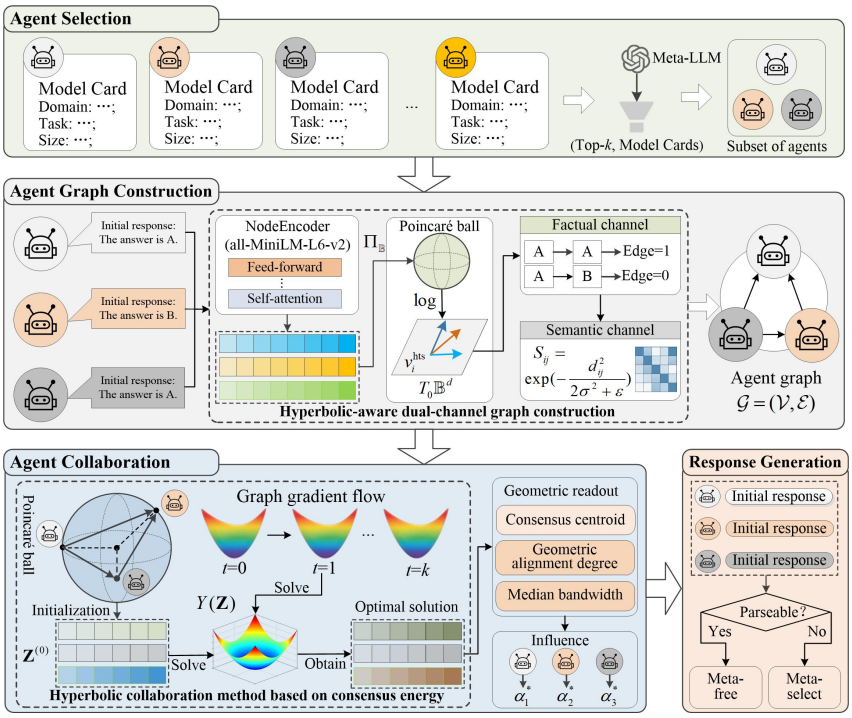


Figure 2: Overall structure of HGoA. (1) Agent selection: Relevant agents are selected based on model cards. (2) Agent graph construction: The initial responses of agents are mapped into the Poincaré ball, and the proposed hyperbolic-aware dual-channel graph construction method is employed to establish the topological structure for multi-agent collaboration. (3) Agent collaboration: The hyperbolic collaboration method based on consensus energy is employed to achieve agent collaboration in the hyperbolic space, thereby converting the collaboration medium from natural language text to representation vectors. (4) Response generation: The final response of multi-agent collaboration is obtained through two response parsing paths.

3.3 Agent Selection

To select agents relevant to the task, we leverage the model cards that are publicly available on Hugging Face. These model cards contain detailed descriptions of the agents, including task descriptions, specialized domains, and model sizes. Based on these model cards, we then employ the meta-LLM to generate a subset of agents $\mathcal{V}_{set} = \text{Meta-LLM}(\text{Top-}k,\ \text{Model Cards})$. Agent selection not only reduces the complexity of subsequent agent graph construction, but also improves the efficiency of multi-agent collaboration.

3.4 Agent Graph Construction

An effective graph structure is key to improving the efficiency of multi-agent collaboration. To adaptively construct the agent graph, we propose a novel hyperbolic-aware dual-channel graph construction method based on the observable outputs (e.g., initial responses) of agents for the current query. This method enables the construction of agent graphs without requiring any model training (training-free). First, for each agent $v_i \in \mathcal{V}_{set}$, an initial response r_i is generated given the query q and an initial prompt. Then, we employ a node encoder to convert each agent's initial response into a fixed-length embedding, as shown in Eq. (1):

$$\mathbf{X}_i \leftarrow \text{NE}(r_i), \hat{\mathbf{X}}_i = \frac{\mathbf{X}_i}{||\mathbf{X}_i||_2 + \varepsilon}, \tag{1}$$

where $\mathbf{X}_i$ denotes the feature vector of agent v_i, $\hat{\mathbf{X}}_i$ denotes the unit direction vector of $\mathbf{X}_i$, NE denotes the node encoder, and ε denotes a smoothing term to prevent the denominator from being

zero. To enhance the discriminative capability of the node encoder across different responses, we inject the extracted response key (the extraction method follows the rules designed in GoA) into the beginning of the responses before feeding them into the node encoder. Next, since responses from multi-agents often exhibit a hierarchical structure that progressively differentiates from shared consensus to individual dissent, we construct a representation space capable of capturing the hierarchical structure. To this end, we employ hyperbolic space as the underlying architecture of this representation space to effectively model such hierarchical structure. Meanwhile, to avoid unstable optimization on the manifold, we perform metric computations in the hyperbolic tangent space. We map $\hat{\mathbf{X}}_i$ into the Poincaré ball $\mathbb{B}^d = \{x \in \mathbb{R}^d : \|x\|_2 < 1\}$ and apply the logarithmic map at the origin to obtain the coordinates v_i^{hts} of the hyperbolic tangent space, as shown in Eq. (2):

$$x_i = \Pi_{\mathbb{B}}\left((1-\delta)\hat{\mathbf{X}}_i\right), v_i^{\text{hts}} = \log(x_i) \in T_0\mathbb{B}^d, \tag{2}$$

where δ denotes a fixed contraction constant, $\Pi_{\mathbb{B}}$ denotes the projection operator, x_i denotes the semantic embedding of agent v_i, $T_0\mathbb{B}^d$ denotes the hyperbolic tangent space of the Poincaré ball at the origin, and d denotes the embedding dimension. We then compute the tangent space distance $d_{ij} = ||v_i^{\text{hts}} - v_j^{\text{hts}}||_2$. Subsequently, we construct the agent graph using dual channels, namely the factual channel and the semantic channel. The factual channel is shown in Eq. (3):

$$F_{ij} = \begin{cases} 0, a_i \neq \phi, a_j \neq \phi, a_i \neq a_j, \\ 1, a_i \neq \phi, a_j \neq \phi, a_i = a_j, \\ u, \text{otherwise}, \end{cases} \tag{3}$$

where a_i denotes the response key obtained by parsing the response of agent v_i, F_{ij} denotes the response consistency between agents v_i and v_j, $F_{ij} = 0$ denotes that the responses of agents v_i and v_j contain a factual conflict, $F_{ij} = 1$ denotes that the responses of agents v_i and v_j do not contain a factual conflict, and $F_{ij} = u$ denotes that the responses of agents v_i and v_j are submitted to the semantic channel for discrimination. The semantic channel is $S_{ij} = \exp(-\frac{d_{ij}^2}{2\sigma^2+\varepsilon}) \in (0, 1]$, where $\sigma = \text{median}\{d_{ij} : 1 \leq i < j \leq k\}$ denotes the bandwidth estimated from the distance distribution of the current query. Finally, the adjacency matrix $\mathbf{A}$ is constructed. If $F_{ij} = 1$, then $\mathrm{A}_{ij} = 1$. If $F_{ij} = 0$, then $\mathrm{A}_{ij} = 0$. If $F_{ij} = u$, then $\mathrm{A}_{ij} = S_{ij}$. A larger $\mathbf{A}_{ij}$ indicates that agents v_i and v_j should exert a stronger mutual attraction in subsequent agent collaboration. The semantic channel provides an adaptive structural prior in the absence of fact labels, while the factual channel provides hard topological constraints related to correctness.

3.5 Agent Collaboration

To avoid message passing among multi-agents on the agent graph via natural language text, we propose a novel hyperbolic collaboration method based on consensus energy. This method converts the collaborative medium from natural language text to representation vectors, improving upon the conventional collaborative paradigm based on the natural language text. By minimizing the consensus energy, we accomplish multi-agent collaboration, thereby improving inference accuracy while reducing inference costs.

Hierarchical Initialization To ensure that high-confidence nodes are pushed toward the central region and low-confidence nodes toward the the peripheral region, thereby manifesting hierarchical distinctions at the geometric level, we perform a hierarchical initialization:

$$s_i = \frac{\sum_{j \neq i} \mathbf{A}_{ij}}{\sum_{\xi \in \mathcal{V}_{set}} \sum_{j \neq \xi} \mathbf{A}_{\xi j} + \varepsilon}, \quad \textstyle\sum_{i \in \mathcal{V}_{set}} s_i = 1, \quad \rho_i = 1 - s_i(1-\delta) \in [\delta, 1), \tag{4}$$

where s_i denotes the hierarchical support degree, and ρ_i denotes the induced radial coordinate. Subsequently, ρ_i is substituted into Eq. (2) to obtain the initial matrix $\mathbf{Z}^{(0)} = [z_1^{(0)}, ..., z_k^{(0)}]^{\mathrm{T}}$ for multi-agent collaboration.

Consensus Energy Objective Function The collaborative paradigm based on natural language text lacks a globally optimizable objective, as the collaboration trajectory is composed of multiple

concatenated text segments. To make multi-agent collaboration analyzable and computable, and to avoid achieving agent consensus through multiple rounds of comprehension and restatement of reasoning chains, we formulate multi-agent collaboration as the minimization of a strictly convex energy: it preserves individual initial values of agents while pulling collaborating agents closer according to the graph consensus. First, we define the consensus energy objective function $Y(\mathbf{Z})$:

$$Y(\mathbf{Z}) = \frac{\lambda}{2}||\mathbf{Z} - \mathbf{Z}^{(0)}||_2 + \frac{\lambda}{2}\mathrm{tr}(\mathbf{Z}^{\mathrm{T}}\mathbf{L}\mathbf{Z}), \tag{5}$$

where λ denotes the weight of the consensus term, $||\mathbf{Z} - \mathbf{Z}^{(0)}||_2$ denotes the fidelity term that penalizes collaborative vectors deviating from the hierarchical initial values, thereby suppressing over-smoothing, $\mathrm{tr}(\mathbf{Z}^{\mathrm{T}}\mathbf{L}\mathbf{Z})$ denotes the consensus term that constrains message passing among nodes on the agent graph, $\mathbf{L}$ denotes the graph Laplacian matrix, and $\mathbf{Z}$ denotes the state matrix of multi-agent collaboration. Next, we need to solve for the optimal solution $\mathbf{Z}^*$ of multi-agent collaboration in the hyperbolic tangent space, as shown in Eq. (6):

$$\mathbf{Z}^* = \arg\min_{\mathbf{Z}\in\mathbb{R}^{k\times d}} Y(\mathbf{Z}). \tag{6}$$

The objective minimized in Eq. (6) is essentially the key information representation of agent responses in the hyperbolic tangent space, rather than the complete reasoning chain representation.

Graph Gradient Flow To solve for $\mathbf{Z}^*$, we leverage the graph gradient flow to find the unique minimizer along the descent direction of the consensus energy objective function under the agent graph constraint. This method transforms the traditional collaborative paradigm based on natural language text into one based on representation vectors in the hyperbolic tangent space. First, the gradient $\nabla_{\mathbf{Z}}Y(\mathbf{Z})$ of $\mathbf{Z}$ is shown in Eq. (7):

$$\nabla_{\mathbf{Z}}Y(\mathbf{Z}) = (\mathbf{Z} - \mathbf{Z}^{(0)}) + \lambda\mathbf{L}\mathbf{Z}, \tag{7}$$

where $\lambda\mathbf{L}\mathbf{Z}$ aggregates the information of neighboring nodes along edges, corresponding to message passing on the agent graph. Thus, we can obtain the optimal solution $\mathbf{Z}^*$. The discrete forms of the graph gradient flows are detailed in Appendix A.1.

Geometric Readout Geometric readout aims to map $\mathbf{Z}^* = [z_1^*, ..., z_k^*]^{\mathrm{T}}$ to the agent influence α^*, which serves as the basis for final response generation and helps the meta-LLM learn the responses of different agents more effectively. First, the consensus centroid $\bar{\mathbf{z}}^* = \sum_{i\in\mathcal{V}_{set}} s_i z_i^* \in \mathbb{R}^d$, the geometric alignment degree $\beta_i = \exp(-\frac{||z_i^* - \bar{z}^*||_2^2}{2\sigma_\beta^2+\varepsilon})$, and the median bandwidth $\sigma_\beta = \mathrm{median}\left\{\|z_u^* - \bar{z}^*\|_2\right\}_{u\in\mathcal{V}_{set}}$ are defined. Then, we can obtain the agent influence $\alpha^* = \frac{s_i\beta_i}{\sum_{u\in\mathcal{V}_{set}} s_u\beta_u+\varepsilon}$.

3.6 Response Generation

Since the output of the previous section consists of $\mathbf{Z}^*$ and α^*, it still needs to be parsed into the final response. To improve parsing efficiency, we design two response parsing paths.

Path A: Meta-free. When the response key is parseable, HGoA should aim to increase the probability that the correct response key is selected, rather than requiring the meta-LLM to re-solve the query. Specifically, for each candidate response y, the confidence quality of y is $M(y) = \sum_{i\in\mathcal{V}_{set}:a_i=y} \alpha_i^* c_i$ (where c_i denotes the confidence score obtained through self-evaluation when the agent generates the initial response). Let $y^* \in \arg\max_y M(y)$ be the response with the maximum confidence quality, and let $\mathcal{Y}^* = \{y : M(y) = M(y^*)\}$ be the set of responses achieving the maximum confidence quality. If $|\mathcal{Y}^*| = 1$, then the final response is $\hat{r} = y^*$.

Path B: Meta-select. When the response key is not parseable, the meta-LLM is called to generate the final response. Specifically, we define a template translator ψ that maps the optimized geometric indicators into a geometric prompt $gp = \psi(\{M(y)\}, \{a_i\}, \{c_i\}, s, \alpha^*, \{F_{ij}\})$. Then, the meta-LLM f_{meta} is called to generate the final response $\hat{r} = f_{meta}(q, gp, \{r_i\}_{i\in\mathcal{T}_K})$, where $\mathcal{T}_K = \text{Top-}k(\alpha^*)$ represents the set of Top-k agents. Since the meta-LLM possesses its own factual knowledge, combined with the geometric indicators provided by the template translator, it can achieve effective response generation with low token consumption and a single LLM call. The core idea of this stage is to leverage the key information (geometric indicators) from the agent responses as prior information for the meta-LLM, thereby improving the final response quality of HGoA.

Table 1: Results of the effectiveness experiments. The best performance is in **bold**, and the second-best performance is underlined. The evaluation metric for HumanEval is pass@1, while the evaluation metric for all other datasets is accuracy (Acc). The multi-agent baseline includes Debate (Du et al., 2024), Refine (Madaan et al., 2023), GPTSwarm (Zhuge et al., 2024), AgentDropout (Wang et al., 2025b), MoA (Wang et al., 2025a), Self-MoA (Li et al., 2026), MAGRPO (Liu et al., 2026), Grammar (Singh et al., 2026), and GoA(Yun et al., 2026).

		Multi-Domain			**Domain-Specific**		
		MMLU	MMLU-Pro	GPQA	MATH	Human-Eval	MedM-CQA
Single-Agent	General	77.61	53.90	32.83	68.92	81.50	55.22
	Code	68.04	42.33	33.84	59.04	85.37	45.57
	Math	63.47	37.19	30.81	48.24	57.93	45.25
	Biomedicine	46.60	27.90	25.25	18.84	20.73	46.55
	Finance	54.11	25.52	28.28	13.76	27.44	42.08
	Law	55.86	27.57	30.30	12.92	36.59	41.50
Multi-Agent (6 Agents)	Debate	72.53	47.05	29.29	70.08	40.24	53.05
	Refine	77.40	54.71	38.92	71.60	80.49	54.94
	GPTSwarm	77.93	55.48	40.91	68.32	83.54	59.89
	AgentDropout	75.44	54.38	38.38	65.52	79.88	58.40
	MoA	75.71	53.33	32.83	65.84	76.22	54.94
	Self-MoA	78.14	54.19	33.84	69.16	79.27	55.56
	MAGRPO	74.39	54.52	40.40	65.36	75.61	52.43
	Grammar	73.44	53.90	37.88	68.08	83.54	60.12
Multi-Agent (3 Agents)	GoA	**79.18**	54.78	39.98	69.76	84.67	60.04
	HGoA	78.25	**57.43**	**43.43**	**72.20**	**86.59**	**62.06**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUPS

Experimental Environment All experiments were implemented in Python 3.10. HGoA was evaluated on an x86_64 server running Ubuntu 22.04.5 LTS with approximately 512GB of memory, dual AMD EPYC 9654 96-core processors, and multiple NVIDIA RTX PRO 6000 GPUs. Each LLM was served independently via vLLM (Virtual Large Language Model), and the node encoder was a locally frozen all-MiniLM-L6-v2 model (Wang et al., 2020). δ, ε and λ were set to 10^{-3}, 10^{-8}, and 1, respectively. Meta-LLM was Qwen2.5-7B-Instruct. In addition, the results of the category-wise comparative experiments are detailed in Appendix A.2. The results of the ablation study are detailed in Appendix A.3.

Benchmarks and Baselines We evaluate the performance of HGoA and the baselines on three multi-domain benchmarks (MMLU (Hendrycks et al., 2021), MMLU-Pro (Wang et al., 2024), GPQA (Zhang et al., 2026)) and three domain-specific mainstream benchmarks (MATH (Lightman et al., 2024), HumanEval (Zhuge et al., 2024), MedMCQA (Pal et al., 2022)). For MMLU and MMLU-Pro, due to their large scale, we strictly follow the stratified sampling method of GoA. For the pool of agents, we mainly utilize six LLMs with 7-8 billion (B) parameters, covering diverse domains including general (Qwen2.5-7B-Instruct), code (Qwen2.5-Coder-7B-Instruct), math (Mathstral-7B-v0.1), biomedicine (Bio-Medical-Llama3-8B), finance (Finance-Llama3-8B), and law (Saul-7B-Instruct-v1). All of these LLMs can be downloaded from Hugging Face. The specific baselines are listed in Table 1.

4.2 MAIN RESULTS

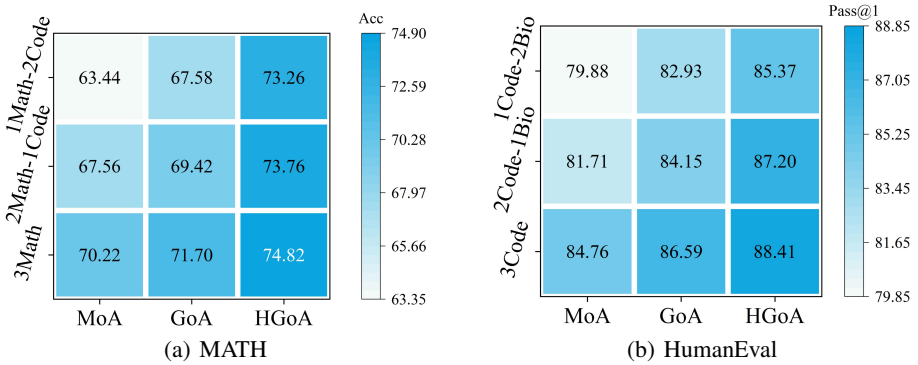
Effectiveness Experiments Table 1 compares the performance of single-agent and multi-agent baselines across six datasets. As listed in Table 1, the accuracy (Acc) of single-agent models is gen-

Table 2: Results of the efficiency experiments. k denotes thousand. s denotes second.

	Metric	MMLU	MMLU-Pro	GPQA	MATH	HumanEval	MedMCQA
MoA (6 Agents)	Calls	19	19	19	19	19	19
	Token (k)	58.49	56.05	56.87	49.98	29.44	42.84
	Time (s)	56.32	70.83	85.09	60.78	38.52	46.86
GoA (3 Agents)	Calls	11	11	11	11	11	11
	Token (k)	14.23	16.19	15.91	11.34	7.75	15.29
	Time (s)	25.52	32.65	37.12	33.07	19.73	26.83
HGoA (3 Agents)	Calls	5	5	5	5	5	5
	Token (k)	3.35	10.72	5.73	5.28	3.74	3.84
	Time (s)	10.42	25.63	23.17	21.25	11.57	14.64

erally lower than that of multi-agent models, indicating that multi-agent baselines have an advantage in handling complex tasks. Furthermore, HGoA achieves the best results on five of the six benchmarks using only three agents. The average score (consisting of both accuracy and pass@1 (Gou et al., 2024)) of HGoA across the six datasets is 66.66%, which is higher than that of GoA (64.74%). Compared with GoA, HGoA improves the scores on MMLU-Pro, GPQA, MATH, HumanEval, and MedMCQA by 2.65%, 3.45%, 2.44%, 1.92%, and 2.02%, respectively. These results demonstrate that the four components of HGoA constitute an effective multi-agent collaboration framework that can enhance the performance of multi-agent systems across different tasks. On MMLU, the Acc of HGoA (78.25%) is slightly lower than that of GoA (79.18%). The main reason is that MMLU primarily consists of general-knowledge multiple-choice questions, and GoA leverages mutual evaluation among LLMs to achieve further response correction, which benefits improving the performance of multi-agent collaboration.

Efficiency Experiments Table 2 compares the Calls (the number of LLM calls), Token (average token consumption per sample), and Time (average time consumption per sample) among MoA (6 agents), GoA (3 agents), and HGoA (3 agents). The experimental results are as follows: (1) The number of LLM calls for MoA, GoA, and HGoA is 19, 11, and 5, respectively, with HGoA reducing the number of LLM calls by 73.68% and 54.55% compared with MoA and GoA, respectively. (2) The average token consumption of MoA, GoA, and HGoA across the six datasets is 48.95k, 13.45k, and 5.44k, respectively, with HGoA reducing average token consumption by 88.89% and 59.55% compared with MoA and GoA, respectively. (3) The average time consumption of MoA, GoA, and HGoA across the six datasets is 59.73s, 29.15s, and 17.78s, respectively, with HGoA reducing average time consumption by 70.23% and 39.01% compared with MoA and GoA, respectively. The primary reason is that HGoA eliminates natural language message passing and mutual evaluation among LLMs during multi-agent collaboration. HGoA converts the collaboration medium from nat-



(a) MATH (b) HumanEval

Figure 3: Robustness experiment results. Code denotes Qwen2.5-Coder-7B-Instruct, Math denotes Mathstral-7B-v0.1, and Bio denotes Bio-Medical-Llama-3-8B.

ural language text to representation vectors, retaining only one initial response and one meta-LLM call. Therefore, this reduces the number of LLM calls, token consumption, and time consumption of HGoA.

Robustness Experiments on Agent Selection To evaluate the robustness of agent selection, we conduct experiments on MATH and HumanEval, as shown in Figure 3. Taking HumanEval as an illustrative example, since it is a benchmark dataset for code generation, we fix the pool of agents to three code agents (e.g., Qwen2.5-Coder-7B-Instruct) and then progressively replace code agents with noisy agents (e.g., Bio-Medical-Llama-3-8B) to simulate erroneous agent selection scenarios. Specifically, HGoA achieves the best performance under all three configurations. Moreover, it is the most robust when a single noisy agent is added, with a performance drop of only 1.21%, whereas the performance of GoA and MoA drops by 2.44% and 3.05%, respectively. The main reason is that the fidelity term in HGoA suppresses over-smoothing, while freezing the initial responses prevents noise from being repeatedly propagated. As the number of noisy agents increases, the performance of HGoA, GoA, and MoA declines cumulatively by 3.04%, 3.66%, and 4.88%, respectively. These results indicate that HGoA can still limit performance degradation through adaptive graph construction and geometric readout, demonstrating superior fault tolerance compared with GoA and MoA.

4.3 Scalability Experiments

To evaluate the scalability of HGoA, we take GPT-5.6-Terra as an illustrative example and replicate 2, 3, and 6 agents, respectively, to construct multi-agent systems of different sizes. Scalability experiments are then conducted on GPQA, MATH, and MedMCQA, respectively (the scalability experiments on MATH and MedMCQA are detailed in Appendix A.4). The experimental results are as follows: (1) As listed in Table 3, increasing the number of agents can improve the performance of the multi-agent system. For instance, on GPQA, the single-agent GPT-5.6-Terra achieves an accuracy of 83.33%, whereas AgentDropout with 6 agents achieves an accuracy of 84.34%. Meanwhile, as the number of agents grows, the performance of HGoA also improves. The primary reason is that multi-agent systems can handle complex tasks more flexibly through mechanisms such as task division and information sharing. (2) As listed in Table 3, on GPQA, although the accuracy of HGoA increases as the number of agents increases, its inference cost also increases. For instance, the number of LLM calls increases from 4 to 8, token consumption increases from 1.49k to 2.46k, and time consumption increases from 55.81s to 109.95s. When the number of agents is expanded from 3 to 6, the accuracy improves by only 1.01%, whereas the number of LLM calls, token consumption, and time consumption increase by 60.00%, 38.20%, and 58.82%, respectively, exhibiting diminishing marginal returns. Therefore, a trade-off between effectiveness and cost should be considered when selecting the number of agents.

Table 3: Specific experimental results regarding model scalability on GPQA.

Pool of agents	Acc	Calls	Token (k)	Time (s)
GPT-5.6-Terra	83.33	1	0.44	25.34
Debate(6 Agents)	84.85	12	32.41	263.49
Refine(6 Agents)	86.36	13	45.78	220.78
MoA(6 Agents)	84.85	19	59.85	360.53
AgentDropout(6 Agents)	84.34	9	31.23	160.25
GoA(6 Agents)	85.35	19	18.78	231.45
HGoA (2 Agents)	83.33	4	1.49	55.81
HGoA (3 Agents)	87.37	5	1.78	69.23
HGoA (6 Agents)	88.38	8	2.46	109.95

5 Conclusion

To improve the efficiency of multi-agent collaboration, we propose a training-free framework named HGoA. HGoA leverages hyperbolic space to capture the hierarchical structures underlying agent responses and then employs a training-free approach to model the topological structure of multi-agent collaboration. Subsequently, we formulate multi-agent collaboration as the minimization of the strictly convex energy, thereby converting the collaboration medium from natural language text to representation vectors. Finally, HGoA utilizes two response parsing paths to generate the final response of the multi-agent system, offering a new solution for the advancement of multi-agent collaboration.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks with required disclosure. We have not used generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. No other required-disclosure tasks are applicable to this work.

Additionally, we used generative AI tools for a task with recommended disclosure: editing the research paper to improve readability, i.e., language polishing. The tools were used only to suggest improvements to grammar, wording, clarity, and style. They were not used to generate or alter the scientific content, research ideas, hypotheses, data, code, analyses, figures, references, or conclusions. We have reviewed all AI-assisted work. All AI-assisted edits were checked and revised by the authors to ensure accuracy, originality, and fidelity to the authors' intended meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide an anonymous repository at `https://anonymous.4open.science/r/HGoA-5CD0/`. This repository contains our code, datasets, data preprocessing scripts, model configurations, and detailed instructions for reproducing the main experimental results. All necessary implementation details, hyperparameters, and evaluation procedures are included in the repository.

REFERENCES

Aakash Kumar Agarwal and Moyuru Yamada. Hypothesis-driven reasoning for large language models. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence*, pp. 1686–1693. 2026.

H. M. Sabbir Ahmad, Ehsan Sabouni, Alexander Wasilkoff, Param Budhraja, Zijian Guo, Songyuan Zhang, Chuchu Fan, Christos G. Cassandras, and Wenchao Li. HMARL-CBF - hierarchical multi-agent reinforcement learning with control barrier functions for safety-critical autonomous systems. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems*, pp. 82284–82317. 2025.

Edoardo Cetin, Benjamin Paul Chamberlain, Michael M. Bronstein, and Jonathan J. Hunt. Hyperbolic deep reinforcement learning. In *Proceedings of the 11th International Conference on Learning Representations*. 2023.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, pp. 4869–4880. 2019.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. In *Proceedings of the 12th International Conference on Learning Representations*. 2024.

Nuo Chen, Yicheng Tong, Yuzhe Yang, Yufei He, Xueyi Zhang, Qingyun Zou, Qian Wang, and Bingsheng He. Diversity collapse in multi-agent LLM systems: Structural coupling and collective failure in open-ended idea generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 251–306. 2026.

Benjamin Cookson, Soroush Ebadian, and Nisarg Shah. Fairness perceptions of large language models. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence*, pp. 35393–35401. 2026.

Rafael Dewes and Rayna Dimitrova. Contract-based design and verification of multi-agent systems with quantitative temporal requirements. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, pp. 23152–23159. 2025.

Shifei Ding, Wei Du, Ling Ding, Jian Zhang, Lili Guo, and Bo An. Robust multi-agent communication with graph information bottleneck optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:3096–3107, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 11733–11763. 2024.

Cornelius Emde, Alexander Rubinstein, Anmol Goel, Ahmed Heakl, Sangdoo Yun, Seong Joon Oh, and Martin Gubri. Maseval: Extending multi-agent evaluation from models to systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 345–356. 2026.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving. In *Proceedings of the 12th International Conference on Learning Representations*. 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the 9th International Conference on Learning Representations*. 2021.

Shengchao Hu, Li Shen, Ya Zhang, and Dacheng Tao. Learning multi-agent communication from graph modeling perspective. In *Proceedings of the 12th International Conference on Learning Representations*. 2024.

Arnav Kumar Jain, Gonzalo Gonzalez-Pumariega, Wayne Chen, Alexander M. Rush, Wenting Zhao, and Sanjiban Choudhury. Multi-turn code generation through single-step rewards. In *Proceedings of the 42nd International Conference on Machine Learning*. 2025.

Ishan Kavathekar, Hemang Jain, Ameya Rathod, Ponnurangam Kumaraguru, and Tanuja Ganu. TAMAS: benchmarking adversarial risks in multi-agent LLM systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 31238–31268. 2026.

Wenzhe Li, Yong Lin, Mengzhou Xia, and Chi Jin. Rethinking mixture-of-agents: Is mixing different large language models beneficial? *Transactions on Machine Learning Research*, 2026.

Yiming Li, Shunli Ren, Pengxiang Wu, Siheng Chen, Chen Feng, and Wenjun Zhang. Learning distilled collaboration graph for multi-agent perception. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, pp. 29541–29552. 2021.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *Proceedings of the 12th International Conference on Learning Representations*. 2024.

Shuo Liu, Zeyu Liang, Xueguang Lyu, and Christopher Amato. LLM collaboration with multi-agent reinforcement learning. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence*, pp. 32150–32158. 2026.

Yuntao Liu, Yong Dou, Yuan Li, Xinhai Xu, and Donghong Liu. Temporal dynamic weighted graph convolution for multi-agent reinforcement learning. In *Proceedings of the 44th Annual Meeting of the Cognitive Science Society*. 2022.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems*, pp. 46534–46594. 2023.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Proceedings of the Conference on Health, Inference, and Learning 2022*, pp. 248–260. 2022.

Emanuele Pesce and Giovanni Montana. Learning multi-agent coordination through connectivity-driven communication. *Machine Learning*, 112:483–514, 2023.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 15174–15186. 2024.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large language model-based multi-agent collaboration. In *Proceedings of the 13th International Conference on Learning Representations*. 2025.

Bohao Qu, Xiaofeng Cao, Bing Li, Menglin Zhang, Tuan-Anh Vu, Di Lin, and Qing Guo. Exploiting geometric structures for modeling multi-agent behaviors: A new thinking. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence*, pp. 25027–25035. 2026.

Guillaume Salha, Stratis Limnios, Romain Hennequin, Viet-Anh Tran, and Michalis Vazirgiannis. Gravity-inspired graph autoencoders for directed link prediction. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 589–598. 2019.

Yuchen Shi, Shihong Duan, Cheng Xu, Ran Wang, Fangwen Ye, and Chau Yuen. Dynamic deep factor graph for multi-agent reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48:3417–3431, 2026.

Mayank Singh, Vikas Yadav, Shiva Krishna Reddy Malay, Shravan Nayak, Sai Rajeswar, Sathwik Tejaswi Madhusudhan, and Eduardo Blanco. Grammar search for multi-agent systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 1640–1655. 2026.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *Proceedings of the 13th International Conference on Learning Representations*. 2025a.

Rongzheng Wang, Shuang Liang, Qizhi Chen, Yihong Huang, Muquan Li, Yizhuo Ma, Dongyang Zhang, Ke Qin, and Man-Fai Leung. Graphcogent: Mitigating llms' working memory constraints via multi-agent collaboration in complex graph understanding. In *Proceedings of the 35th ACM Web Conference*, pp. 3811–3822. 2026.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*, pp. 5776–5788. 2020.

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. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems*, pp. 95266–95290. 2024.

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

Yihan Xia, Taotao Wang, Shengli Zhang, Zhangyuhua Weng, Bin Cao, and Soung Chang Liew. Hivemind: Contribution-guided online prompt optimization of LLM multi-agent systems. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence*, pp. 29767–29774. 2026.

Jiasheng Xu, Mingda Li, Yongqiang Tang, Peijie Wang, and Wensheng Zhang. Towards open-world retrieval-augmented generation on knowledge graph: A multi-agent collaboration framework. In *Proceedings of the 35th ACM Web Conference*, pp. 2171–2182. 2026.

Yibo Yan, Shen Wang, Jiahao Huo, Philip S. Yu, Xuming Hu, and Qingsong Wen. Mathagent: Leveraging a mixture-of-math-agent framework for real-world multimodal mathematical error detection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 69–82. 2025.

Jingxiao Yang, Ping He, Tianyu Du, Sun Bing, and Xuhong Zhang. Hogvul: Black-box adversarial code generation framework against lm-based vulnerability detectors. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence*, pp. 1435–1443. 2026.

Sukwon Yun, Jie Peng, Pingzhi Li, Wendong Fan, Jie Chen, James Zou, Guohao Li, and Tianlong Chen. Graph-of-agents: A graph-based framework for multi-agent LLM collaboration. In *Proceedings of the 14th International Conference on Learning Representations*. 2026.

Duzhen Zhang, Zhong-Zhi Li, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhijiang Guo, Le Song, and Cheng-Lin Liu. From system 1 to system 2: A survey of reasoning large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48:3335–3354, 2026.

Guibin Zhang, Muxin Fu, Kun Wang, Frank Wan, Miao Yu, and Shuicheng Yan. G-memory: Tracing hierarchical memory for multi-agent systems. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems*, pp. 12988–13018. 2025a.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. In *Proceedings of the 13th International Conference on Learning Representations*. 2025b.

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

Lan Zhang, Marco Valentino, and André Freitas. MASA: llm-driven multi-agent systems for autoformalization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 615–624. 2025d.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 62743–62767. 2024.

A Appendix

A.1 Graph Gradient Flow

The discrete form of the graph gradient flow is shown in Eq. (8):

$$\mathbf{Z}^{(t+1)} = \mathbf{Z}^{(t)} - \eta \nabla_{\mathbf{Z}} Y(\mathbf{Z}^{(t)}) = \mathbf{Z}^{(t)} - \eta((\mathbf{Z}^{(t)} - \mathbf{Z}^{(0)}) + \lambda \mathbf{L}\mathbf{Z}^{(t)}), \tag{8}$$

where t denotes the iteration step index, and η denotes the gradient flow step size. Since Y is strictly convex with respect to $\mathbf{Z}$, a unique minimizer exists, and its closed-form solution is given by the following linear system, as shown in Eq. (9):

$$(\mathbf{I} + \lambda \mathbf{L})\mathbf{Z}^* = \mathbf{Z}^{(0)} \Rightarrow \mathbf{Z}^* = (\mathbf{I} + \lambda \mathbf{L})^{-1}\mathbf{Z}^{(0)}, \tag{9}$$

where $\mathbf{I}$ is the identity matrix. Thus, we can obtain the optimal solution $\mathbf{Z}^*$.

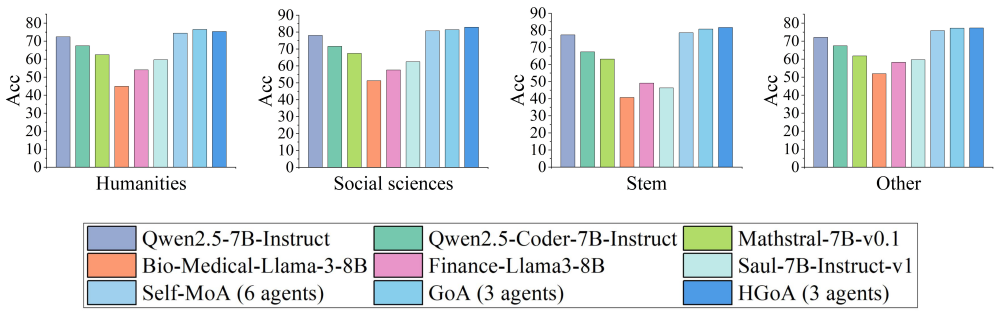


Figure A1: Category-wise comparison experiments on MMLU. The labels such as Humanities, denote different sample categories in MMLU. The accuracy of a model (e.g., Qwen2.5-7B-Instruct) is obtained by running it on these different categories.

A.2 Category-wise Comparative Experiments

As shown in Figures A1 and A2, although Self-MoA integrates all six agents and can smooth out fluctuations of individual models on several categories, its performance in most categories remains inferior to that of HGoA. Furthermore, HGoA outperforms the domain-specific models, Self-MoA, and GoA in terms of performance on most categories. The reasons are as follows: (1) The agent selection retrieves relevant agents according to the current query. For example, for legal queries, it prioritizes legal agents over other irrelevant agents, such as code and math agents. (2) The hyperbolic-aware dual-channel graph construction method characterizes semantic proximity using tangent space distance, bringing nodes with similar viewpoints within the same category closer to the consensus centroid, thereby adaptively adjusting communication strength when categories switch. (3) The hyperbolic collaboration method based on consensus energy can capture the key information of different categories, thereby facilitating effective multi-agent collaboration for HGoA. Therefore, HGoA can achieve better performance with only three agents on the majority of categories in MMLU and MMLU-Pro.

A.3 Ablation Study

To verify the necessity of each component of HGoA, we conduct an ablation study on six datasets. HGoA w/o AS denotes the model with the Agent Selection module removed from HGoA. HGoA w/o GC denotes the model with the hyperbolic-aware dual-channel Graph Construction method removed from HGoA (using the graph construction method of GoA instead). HGoA w/o RV denotes the model with the Representation Vectors replaced by natural language text in HGoA. The detailed experimental results are listed in Table A1.

As listed in Table A1, compared with the performance of HGoA, the performance of HGoA w/o GC exhibits the largest overall decrease, followed by HGoA w/o AS, and HGoA w/o RV. This indicates that agent selection, the hyperbolic-aware dual-channel graph construction method, and

Table A1: Experimental results of the ablation study. The best performance is in **bold**, and the second-best performance is underlined. Avg_Token denotes the average token consumption per sample across the six datasets. The evaluation metric for HumanEval is pass@1, while the evaluation metric for all other datasets is accuracy (Acc).

Model	MMLU	MMLU-Pro	GPQA	MATH	Human Eval	Med MCQA	Avg_Token
HGoA w/o AS	75.79	55.86	39.39	70.42	83.53	58.07	5.56k
HGoA w/o GC	74.67	54.67	40.40	68.78	85.37	57.30	8.48k
HGoA w/o RV	77.75	56.29	42.42	71.72	84.76	60.63	13.45k
HGoA	**78.25**	**57.43**	**43.43**	**72.20**	**86.59**	**62.06**	**5.44k**

Table A2: Experimental results regarding model scalability. The best performance is in **bold**, and the second-best performance is underlined. The evaluation metrics for GPQA, MATH, and MedMCQA are accuracy (Acc).

Pool of agents	GPQA	MATH	MedMCQA
GPT-5.6-Terra (1 Agent)	83.33	88.80	91.25
Debate(6 Agents)	84.85	90.56	88.76
Refine(6 Agents)	86.36	92.42	89.87
MoA(6 Agents)	84.85	91.78	92.34
AgentDropout(6 Agents)	84.34	93.02	93.81
GoA(6 Agents)	85.35	99.24	92.06
HGoA(2 Agents)	83.33	98.58	92.52
HGoA(3 Agents)	87.37	99.62	94.74
HGoA(6 Agents)	**88.38**	**99.78**	**95.70**

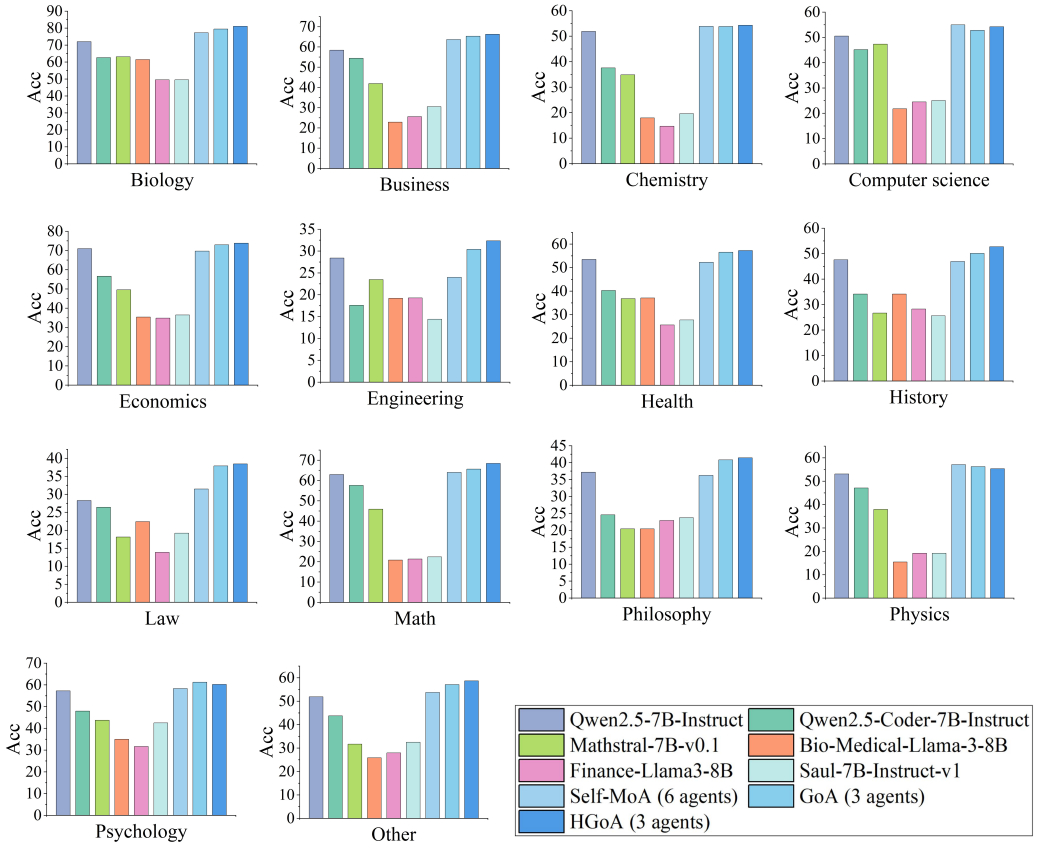


Figure A2: Category-wise comparison experiments on MMLU-Pro. The labels such as Biology denote different sample categories in MMLU-Pro. The accuracy of a model (e.g., Qwen2.5-7B-Instruct) is obtained by running it on these different categories.

the hyperbolic collaboration method based on consensus energy are all indispensable components of HGoA. In addition, we find that HGoA w/o RV has the highest Avg_Token, followed by HGoA w/o GC. The primary reason is that both of these models use natural language text as the collaboration medium during multi-agent collaboration, thereby increasing token consumption.

A.4 SCALABILITY EXPERIMENTS

To evaluate the scalability of HGoA, we take GPT-5.6-Terra as an illustrative example and replicate 2, 3, and 6 agents, respectively, to construct multi-agent systems of different sizes. Scalability experiments are then conducted on GPQA, MATH, and MedMCQA, respectively. As listed in Table A2, increasing the number of agents can improve the performance of the multi-agent system. For instance, on GPQA, MATH, and MedMCQA, the single-agent GPT-5.6-Terra achieves accuracies of 83.33%, 88.80%, and 91.25%, respectively, whereas AgentDropout with 6 agents achieves accuracies of 84.34%, 93.02%, and 93.81%, respectively.

A.5 PROMPTS

The prompts included in HGoA are as follows:

- Agent selection prompt.

Model Card Extraction - System Prompt

```
You are an expert in analyzing and summarizing AI model documentation.
```

Agent Selection

```
You are an AI model selection expert. Your task is to select the most relevant AI models to answer a given question based on their domain, specialized capabilities, and overall performance.

Select {top_k} models best suited for this question.

Question: {question}

Available models:
{model_cards}

Selection criteria (in priority order):
    1. Domain match — prefer models trained in the question's domain
    2. Task specialization — prefer models fine-tuned for the required skill
    3. Include at least one generalist model if applicable
    4. Prefer larger models only when the size gap is significant

Rules:
    1. Output exactly {top_k} comma-separated indices from [0, {max_index}]
    2. If the model is highly relevant, this index can be prioritized
    3.  No explanations

Example:
    top_k=1: "0",
    top_k=2: "0,3",
    top_k=3: "0,1,5",
    top_k=4: "0,1,2,3"

Answer:
```

- Initial response prompt.

Initial Response - Select the Corresponding Question Type

```
Question: {question}
Provide your step-by-step reasoning first, and then print "The answer is (x)", where x is the answer choice (one capital letter), at the end of your response.

Provide brief reasoning (2-3 key sentences), then output your final answer in JSON format:
{"reasoning": "<brief reasoning>", "answer": "<one of the answer choices: {choices}>", "confidence_level": "<a float between 0.0 and 1.0>"}
Please strictly output in JSON format.


Question: {question}
Provide your step-by-step reasoning first, and then print "The answer is \boxed{X}", where X is the final answer, at the end of your response.

Provide brief reasoning (2-3 key sentences), then output your final answer in JSON format:
{"reasoning": "<brief reasoning>", "answer": "<a mathematical expression or number (e.g., \"42\" or \"3/4\")>", "confidence_level": "<a float between 0.0 and 1.0>"}
Please strictly output in JSON format.


Given the following Python function signature, complete the function to match the expected behavior.
{function_signature}

Completed Function:
Provide brief reasoning (2-3 key sentences), then output your final answer in JSON format:
{"reasoning": "<brief reasoning>", "answer": "<the completed Python function code>", "confidence_level": "<a float between 0.0 and 1.0>"}
Please strictly output in JSON format.
```

- Response generation prompt.

Response Generation

```
Synthesize these model responses into one final answer.
Question: {question}
Collaboration context (representation-space α*/priors; NOT peer-review):
## Hyperbolic space collaboration report
### Soft prior M(y)=Σα* (NOT ground truth)
{M(y)}

### Parsed answers
{a_i}

| Agent | conf | s | α* | parsed_answer |
| {i} | {c_i} | {s_i} | {α*_i} | {a_i} |

### Fact conflicts (edges severed)
{F_ij}

Allowed final answers: {a_i}

Model responses:
{r_i}_{i ∈ T_K}/{drafts}

Produce an accurate, coherent answer integrating the best insights. Be critical — some drafts may be incorrect. If two or more drafts agree on the same allowed answer, prefer that unless you find a clear factual/reasoning error. When drafts disagree, verify each option against the question stem; prefer the draft whose reasoning is most consistent with the facts asked. Do not trust a specialty-model label alone when its reasoning is weak. End with a clear final answer choice.
{final_answer}
```