HiVA: Self-organized Hierarchical Variable Agent via Goal-driven Semantic-Topological Evolution

Jinzhou Tang[1, 2*], **Jusheng Zhang**[1*], **Qinhan Lv**[1*], **Sidi Liu**[1, 3], **Jing Yang**[1], **Chengpei Tang**[1†], **Keze Wang**[1†]
[1]Sun Yat-sen University
[2]University of California, San Diego
[3]The Hong Kong University of Science and Technology (Guangzhou)
kezewang@gmail.com

Abstract

Autonomous agents play a crucial role in advancing Artificial General Intelligence, enabling problem decomposition and tool orchestration through Large Language Models (LLMs). However, existing paradigms face a critical trade-off. On one hand, reusable fixed workflows require manual reconfiguration upon environmental changes; on the other hand, flexible reactive loops fail to distill reasoning progress into transferable structures. We introduce Hierarchical Variable Agent (HiVA), a novel framework modeling agentic workflows as self-organized graphs with the Semantic-Topological Evolution (STEV) algorithm, which optimizes hybrid semantic-topological spaces using textual gradients as discrete-domain surrogates for backpropagation. The iterative process comprises Multi-Armed Bandit-infused forward routing, diagnostic gradient generation from environmental feedback, and coordinated updates that co-evolve individual semantics and topology for collective optimization in unknown environments. Experiments on dialogue, coding, long-context Q&A, mathematical, and agentic benchmarks demonstrate improvements of 5% to 10% in task accuracy and enhanced resource efficiency over existing baselines, establishing HiVA's effectiveness in autonomous task execution.

Code — https://github.com/tangjzh/HiVA

1 Introduction

The pursuit of general-purpose autonomous agents, which are capable of independently solving complex, open-ended tasks, represents a central goal of artificial intelligence. (Ye et al. 2025; Webb, Holyoak, and Lu 2022; Raman et al. 2025) Large Language Models (LLMs) have emerged as a powerful backbone for such agents, enabling them to decompose goals, plan actions, and invoke tools through natural language reasoning. (Li 2025) However, despite the success of LLM-based agents in applications such as automated software development and scientific discovery, their underlying multi-agent coordination paradigms remain fundamentally limited. (Jimenez-Romero, Yegenoglu, and Blum 2025; Zhang et al. 2025f)

Existing frameworks can be broadly categorized into two categories: (1) *Manually-designed workflows*, which rely on fixed, structured agent interactions to complete tasks. While these offer modularity and reuse, they suffer from poor generalization to unseen task formats. (Qin et al. 2022; Hu et al. 2025)(2) *Reactive agent loops* (*e.g.*, ReAct (Yao et al. 2023), AutoGPT (Yang, Yue, and He 2023)), which execute task solving as a sequence of language-based decisions. While more adaptable, these reactive agents fail to directly reuse reasoning patterns or continually improve themselves in dynamic environments (Mialon et al. 2023; Xi et al. 2023). Though recent efforts have introduced mechanisms to optimize either agent behaviors or routing strategies (Zhang et al. 2025a; He et al. 2024), they largely treat the coordination structure and semantics as independent. These approaches focus on incremental adaptation, i.e., tuning prompts, sampling structures, or adjusting routes from predefined templates, without a convincing mechanism to explore **how LLMs can spontaneously build a complex from a singleton** (Guo et al. 2024). We argue that general-purpose agents should be **evolutionary** systems for dynamic environments: they must learn not only (1) *what each agent should do* (i.e., semantic behavior), but also (2) *how agents should interact and organize* (i.e., structural topology). This motivates our central question: *Can a multi-agent system evolve both its internal semantics and collaborative structure* **from ZERO** *to achieve scalable, adaptive, and self-organized intelligence across diverse tasks?*

To this end, we propose Hierarchical Variable Agent (HiVA), a novel multi-agent framework based on the *Semantic-Topological Evolution (STEV)* algorithm. In HiVA, the coordination structure is modeled as a dynamic computational graph, and each agent is represented as a configurable LLM module. Optimization occurs in a hybrid space that consists of: (1) a *semantic space* for learning agent-level behaviors (*e.g.*, prompts, tool configurations) (Yuksekgonul et al. 2025), and (2) a *topological space* that encodes which agents are connected and how information flows among them (Zhou et al. 2025). Each optimization round in HiVA involves three steps: (1) a *Forward Pass*, where a task is routed through a dynamically constructed subgraph of agents; (2) a *Textual Gradient Feedback*, where language-based diagnostics are generated based on the environmental feedback to approximate gradient-like signals for

*Equal contribution
†Corresponding author

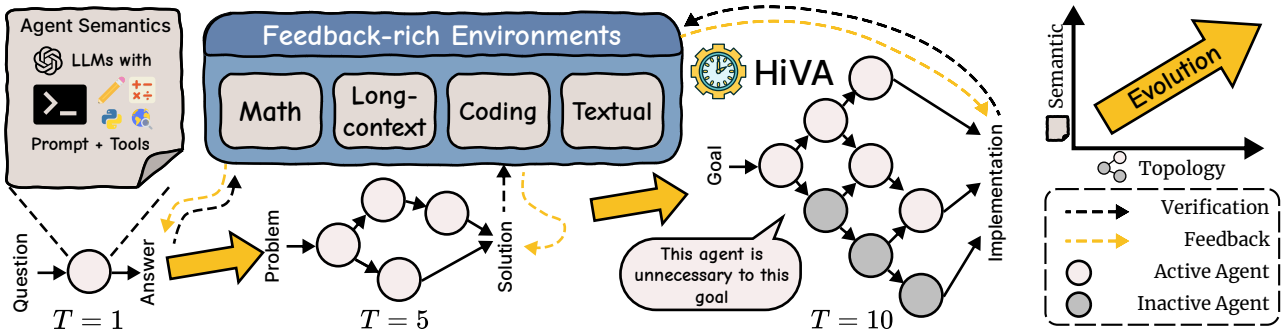

Figure 1: **Semantic-Topological Evolution from Singleton to Self-organized Complex Agents.** We explore how Large Language Models, when deployed in feedback-rich environments, can spontaneously form increasingly complex cognitive roles to refine their decision-making and tool use. Starting from a single agent with basic capabilities, HiVA fosters gradual evolution through semantic refinement and topological reconfiguration. Agents adapt to feedback-derived gradients from environmental interactions and develop into specialized yet interconnected sub-agents, thereby forming a complex system.

optimization in a non-differentiable space; and (3) a *Coordinated Update*, where each agent adjusts both its internal semantic parameters and its structural links to other agents. This loop enables HiVA to progressively refine its capabilities and collaboration structure across tasks, resulting in a self-optimizing, specialized multi-agent system. Experimental results across diverse benchmarks show that HiVA generally outperforms static workflows, reactive loops, and mainstream multi-agent optimization algorithms, achieving better transferability and greater efficiency.

Our contributions are threefold: (i) We propose HiVA, the first framework to unify semantic and structural evolution from a singleton in LLM-based multi-agent systems; (ii) We introduce *Semantic-Topological Evolution*, a general principle for co-evolving agent behavior and collaboration; and (iii) We design a dynamic routing and update mechanism based on bandit exploration and textual gradients.

2 Related Works

LLM-Based Multi-Agent Systems Multi-agent systems (MAS) powered by Large Language Models (LLMs) excel at complex tasks like software development and scientific discovery (Junda He 2025; Su et al. 2025; Li, Zhang, and Safara 2023; Zhang et al. 2025d; Sheng 2025). Recent works like ReAct (Yao et al. 2023) and AutoGPT (Yang, Yue, and He 2023) emphasize the reasoning and long-term planning capabilities of LLM-driven MAS, while lacking the potential for scalability through reproducing their chain-of-thought (CoT). MetaGPT (Hong et al. 2024) uses shared message pools, and DyLAN (Liu et al. 2023) employs layered communication to boost efficiency. However, their static workflows limit adaptability to novel tasks. In contrast, our HiVA framework can dynamically evolve, thereby enhancing flexibility and scalability.

Dynamic Self-Improvement Mechanisms Dynamic self-improvement mechanisms enable agents to adapt through feedback. Existing approaches can be categorized into two paradigms: the first involves text-based optimization methods, including both single-agent semantic (Zhang et al. 2024) and multi-agent collaborative optimization (Liang, Xu, and Dong 2025); the second employs reinforcement learning approaches for agent improvement (Guo et al. 2025). In contrast, HiVA enables co-evolution of agent parameters and topology from a singleton, achieving effective test-time-scaling for diverse tasks and scenarios.

Hierarchies and Topological Optimization Hierarchical structures and topological optimization enhance MAS efficiency. MetaGPT (Hong et al. 2024) and MASAI (Wadhwa et al. 2024) streamline workflows with modular designs. MASS (Zhang et al. 2025b) proposes a Multi-Agent System Search for prompt and topology optimization. G-designer (Zhang et al. 2025c) uses graph neural networks for topology optimization without semantic integration. HiVA's Bayesian-driven hierarchical evolution enables better modeling of individuals in self-organized MAS.

3 Methodology

This paper introduces HiVA (Hierarchical Variable Agent), a novel framework designed to address the problem of adaptive task-flow optimization in Multi-Agent Systems (MAS). The core idea is to formulate this optimization problem as a process of generalized gradient descent within a **Hybrid Space**, which is constituted by both a semantic space and a topological space. The theoretical cornerstone of our work is the **Semantic-Topological Evolution (STEV)** algorithm, as illustrated in Figure 1, which enables optimization in the discrete and non-differentiable space of agentic graphs. Our STEV algorithm facilitates the simultaneous co-evolution of agents' internal parameters (semantics) and their collaborative structures (topology) by introducing **Textual Gradients** as a substitute for traditional gradients, combined with knowledge-aware dynamic computation graph construction.

3.1 Semantic-Topological TextGrad

We define a complete solution for a multi-agent system as a point s within a hybrid space $\mathcal{S} = \mathcal{G} \times \mathcal{P}_\Theta$, where $\mathcal{G}$ is the

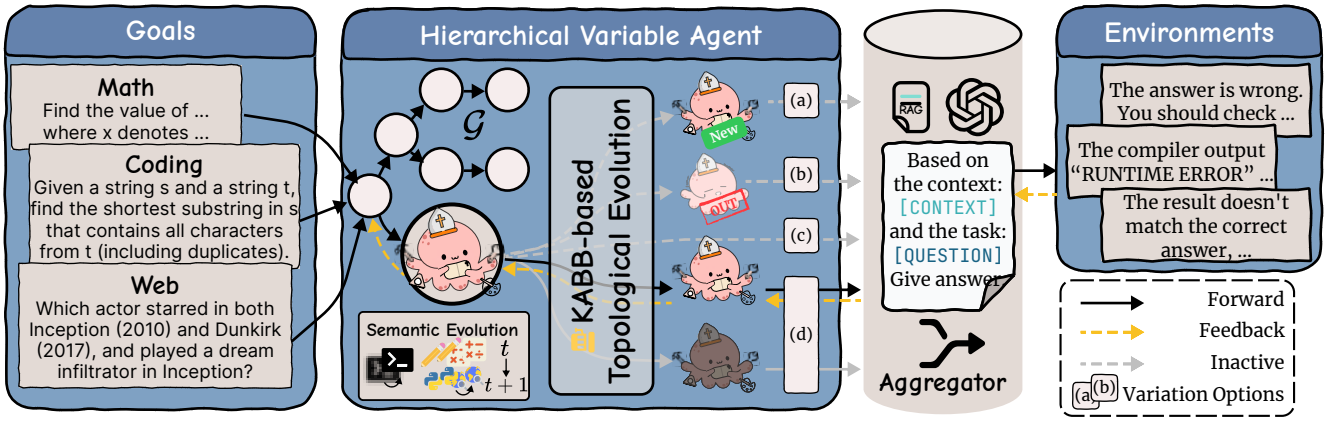

Figure 2: **Evolving mechanism of Hierarchical Variable Agent in a single iteration.** Driven by the goal, HiVA decomposes evolution into two stages: forward and backward propagation. In the forward pass, each agent in $\mathcal{G}$ utilizes KABB to select relevant successors and generate instructions for them. The aggregator then uses specific tools (*e.g.*, RAG) to access MAS context and generate final answers. In the backward pass, each agent receives feedback from its successor (notably, aggregator receives feedback from the environments) and evolves their semantics (prompts and tools) and topologies with four options: (a) add a successor, (b) delete a successor, (c) connect directly to the aggregator, and (d) do nothing.

space of graph topologies and $\mathcal{P}_\Theta$ is the space of semantic parameters Θ. The optimization objective is to find a solution s^* that minimizes a black-box objective function $\mathcal{L}(s)$.

$$s^* = \arg\min_{s \in \mathcal{S}} \mathcal{L}(s)$$

This optimization is non-trivial, as the discrete and non-Euclidean nature of $\mathcal{S}$ renders the traditional gradient $\nabla_s \mathcal{L}$ ill-defined. To navigate this, we introduce the **Textual Gradient** (Yuksekgonul et al. 2025) as a functional substitute. We operationalize it using a **Textual Gradient Parser**, implemented via an LLM, which translates raw textual feedback into a structured update instruction, Δs_t. The parser diagnoses whether the gradient applies to an agent's semantics or its topology, generating a corresponding command for either the semantic evolution function (f_P) or the topological evolution function (f_G). This reframes the optimization as a **Generalized Gradient Descent** process, with an update rule $s_{t+1} \leftarrow s_t \oplus \Delta s_t$, where the parsed gradient guides targeted updates across the hybrid space.

3.2 The Environment: Oracle and Adversary

The interaction space for our optimization problem is defined by the **Environment**, $\mathcal{E}_{\text{env}}$. Formally, the environment is a function that accepts the system's final output, $S_{\text{out}} = s(I_{\text{task}})$, and returns a potentially complex outcome or feedback. The environment plays a crucial dual role.

First, it acts as an **oracle**, providing the ground-truth feedback necessary for learning. The textual loss required for optimization is computed by an objective function $\mathcal{L}$ that maps the environment's rich dynamics driven by s to a explainable signal: $\mathcal{L}(s) = \mathcal{L}(\mathcal{E}_{\text{env}}(s(I_{\text{task}})))$.

Second, the environment acts as an **adversary**, defining the complexity and challenge of the task space. The robustness and adaptability of a solution s are measured by its ability to consistently achieve low loss across the distribution of problems posed by $\mathcal{E}_{\text{env}}$. The nature of $\mathcal{E}_{\text{env}}$ can vary widely, encompassing: (a) programmatic environments with verifiable outcomes (*e.g.*, code compilation, unit test execution); (b) data-driven environments where feedback is derived from metrics on a held-out dataset (*e.g.*, QA); (c) agentic environments that provide interactive simulators or interfaces (*e.g.*, browser, game, IDE); and (d) mathematical environments where feedback is qualitative text from a verifiable math evaluator (*e.g.*, Lean). A successful optimization must yield a solution robust to the specific challenges of its target environment.

3.3 The Iterative Optimization of HiVA

The optimization process in HiVA is an iterative loop driving the solution $s_t = (G_t, \Theta_t)$ toward a minimum of the objective function, as illustrated in Algorithm 1. Each iteration consists of a coupled forward and backward pass. The function RepairTopology$(\cdot)$ removes isolated nodes and prevents cycles; more details can be found in the Algorithmic Details section of the Appendix.

The **Forward Pass**, as shown in Algorithm 2, probes the potential of the current solution s_t. It employs a dynamic routing mechanism to construct a task-specific execution subgraph $\mathcal{G}_{\text{exec},t}$, culminating in the generation of a final output $S_{\text{out},t}$.

This output is then subjected to the **Loss Evaluation** phase. It is passed to the environment, $\mathcal{E}_{\text{env}}$, and the resulting outcome is mapped to a textual loss:

$$\mathcal{L}_t = \mathcal{L}(\mathcal{E}_{\text{env}}(S_{\text{out},t})).$$

The loss $\mathcal{L}_t$ triggers the **Backward Pass**. The process begins by estimating a global textual gradient at the sink point v_a of $\mathcal{G}$ based on the environmental feedback. This gradient is then decomposed and propagated via a textual chain rule to each participating node, yielding localized textual gradients.

Algorithm 1: SEMANTIC-TOPOLOGICAL EVOLUTION

Definitions:
Agent $v_i \in \mathcal{V}$: Entity with prompt p_i, tool τ_i, mapping instruction x_i to output y_i.
Aggregator v_a: Aggregates outputs.
Network $\mathcal{G}$: DAG with source v_s and sink v_a.
Input: Instruction x_0, environment function $\mathcal{E}_{\text{env}}$.
Output: Optimized $\mathcal{G}$, result y.
Initialize: Create v_s (p_s, t_s), connect $v_s \to v_a$.
for $t = 1$ to T **do**
  Forward: Execute FORWARDPASS($\mathcal{G}, x_0$)
  Loss: $\mathcal{L}_t \leftarrow \mathcal{L}(\mathcal{E}_{\text{env}}(y))$, $\nabla_{\text{text}}\mathcal{L}_t \leftarrow \text{LLM}(y, \mathcal{L}_t)$
  Backward:
  for each $v_i \in \text{reverse}(\sigma)$ **do**
    $\frac{\partial \mathcal{L}_t}{\partial v_i} \leftarrow \nabla_{\text{text}}\mathcal{L}_t$ if $v_i = v_a$, else
    $\frac{\partial \mathcal{L}_t}{\partial v_i} \leftarrow \text{LLM}(\{\frac{\partial \mathcal{L}_t}{\partial v_j} \mid v_j \in \text{successors}(v_i)\}, y_i)$
    $p_i, \tau_i \leftarrow f_P(p_i, \tau_i, \frac{\partial \mathcal{L}_t}{\partial v_i})$
    $\mathcal{G} \leftarrow f_G(\mathcal{G}, \{\mathcal{R}_{ij}\}_{j \in \text{successor}(v_i)}, \frac{\partial \mathcal{L}_t}{\partial v_i})$
  end for
  $\alpha_i, \beta_i \leftarrow \alpha_i^{(t+1)}, \beta_i^{(t+1)}$
  $\mathcal{G} \leftarrow \text{RepairTopology}(\mathcal{G})$
end for
Return: $\mathcal{G}, y$

Algorithm 2: FORWARDPASS

Initialize $\mathcal{R}_{ij} \leftarrow 0, \forall v_i, v_j \in \mathcal{V}$
Compute topological order σ of $\mathcal{G}$
for each $v_i \in \sigma$ **do**
  if v_i receives no instruction **then**
    continue
  end if
  $y_i \leftarrow \tau_i(x_i)$
  $s_j \sim P_i, v_j \in \text{successor}(v_i)$
  $\mathcal{V}' = \text{top-}k_{v_j \in \text{successor}(v_i)}(s_j)$
  for each $v_j \in \mathcal{V}'$ **do**
    $\mathcal{R}_{ij} \leftarrow \mathcal{R}_{ij} + 1$
    Generate $x_j \leftarrow \text{LLM}(x_i, y_i, p_j)$
  end for
end for
$y \leftarrow v_a.\text{aggregate}(\{y_i \mid \deg^+(v_i) = 1, \mathcal{R}_{ia} = 1\})$
Return: y

This backward flow of information culminates in a **Coordinated Update** of the current solution s_t. The semantic parameters and graph topology are co-optimized via textual gradient descent. Finally, the dynamic routing policy is refined by updating its Bayesian belief parameters, following KABB (Zhang et al. 2025e).

3.4 Knowledge-Based Subgraph Generation

Dynamic routing underpins HiVA's efficiency and adaptability in selecting agents for task-specific execution subgraphs. It models agent selection as a Multi-Armed Bandit (MAB) problem, solved using Thompson Sampling. With topological order, at each decision point, the system assigns P_i to each agent A_i by sampling from a probability distribution proportional to:

$$P_i \propto \frac{\alpha_i^{(t)}}{\alpha_i^{(t)} + \beta_i^{(t)}} \cdot \exp\left(-\lambda \cdot \text{Dist}(A_i, I_{\text{task}})\right) \cdot \zeta(\mathcal{S}_t)^{\eta}$$

Here, $\zeta(\mathcal{S}_t) = \frac{1}{|\mathcal{S}_t|(|\mathcal{S}_t|-1)} \sum_{v_i, v_j \in \mathcal{S}_t, i \neq j} C_{\text{syn}}^{(t)}(v_i, v_j)$ quantifies the collaborative effect within the selected subset $\mathcal{S}_t$ which include previous selected agents and A_i, where $C_{\text{syn}}^{(t)}(v_i, v_j)$ is the synergy gain coefficient for pairwise agent interactions, and η adjusts its influence. This distribution balances historical performance (via Bayesian belief parameters α_i, β_i), task relevance, and team synergy, penalized by a **knowledge-based cost function**:

$$\text{Dist}(A_i, I_{\text{task}}) = \log(1 + d_I) \cdot \sum_{k=1}^{4} \omega_k \Psi_k$$

This cost function uses an external knowledge graph to measure the mismatch Ψ_k across four sub-indicators of an agent's capabilities against task requirements. The routing mechanism then constructs efficient, task-specific execution subgraphs $G_{\text{exec},t}$ through Thompson Sampling, which serve as the foundation for forward propagation. More details can be found in the Appendix on Knowledge-Based Cost Function.

The routing policy evolves by updating the Bayesian belief parameters for each agent based on performance and task alignment:

$$\alpha_i^{(t+1)} = \gamma^{\Delta t} \alpha_i^{(t)} + \left[r_i^{(t)} + \delta \cdot \text{KM}(A_i, I_{\text{task}}) \right] \cdot \mathbb{I}_{\{A_i \in \mathcal{S}_t\}}$$

$$\beta_i^{(t+1)} = \gamma^{\Delta t} \beta_i^{(t)} + \left[1 - r_i^{(t)} + \delta \cdot \text{KD}(A_i, I_{\text{task}}) \right] \cdot \mathbb{I}_{\{A_i \in \mathcal{S}_t\}}$$

Here, $\text{KM}(A_i, I_{\text{task}}) = \rho_{\text{overlap}} \cdot \zeta(\{A_i, I_{\text{task}}\})$ measures task alignment and $\text{KD}(A_i, I_{\text{task}}) = 1 - \text{KM}(A_i, I_{\text{task}})$ plays the opposite, incorporating a synergy term scaled by a task relevance factor ρ_{overlap}. The updates combine a reward signal $r_i^{(t)}$ reflecting the agent's contribution, a knowledge-driven adjustment KM, and an exponential decay factor $\gamma^{\Delta t} = e^{-\kappa \Delta t}$ to prioritize recent performance. The indicator function $\mathbb{I}_{\{A_i \in \mathcal{S}_t\}}$ ensures only agents in the selected subset are updated. This continuous learning process refines the policy, favoring agents that perform well and align closely with the task while promoting collaborative subsets, leading to increasingly effective subgraph generation over time.

3.5 Multi-agent Structure as Memory

In our HiVA framework, the multi-agent structure serves not only as an organizational scheme for computational units but also functions as the core memory mechanism of the system. Unlike traditional multi-agent systems that confine memory to internal states of individual agents, HiVA encodes collective memory within the network topology $\mathcal{G}$ and inter-agent connection weights, achieving distributed memory storage and retrieval.

Specifically, each edge $(v_i, v_j) \in \mathcal{E}$ in the network topology $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ carries a weight $C_{\text{syn}}^{(t+1)}(v_i, v_j)$ that encodes the historical accuracy and task relevance of information

transfer from agent v_i to v_j. These weights evolve continuously through a Bayesian update mechanism:

$$C_{\text{syn}}^{(t+1)}(v_i, v_j) = C_{\text{syn}}^{(t)}(v_i, v_j) + \gamma \cdot \frac{\alpha_{ij}^{(t)}}{\alpha_{ij}^{(t)} + \beta_{ij}^{(t)}} \cdot \mathcal{R}_{ij}^{(t)}$$

where $\mathcal{R}_{ij}^{(t)}$ represents the task contribution of edge (v_i, v_j) in the t-th iteration, and γ is the learning rate.

Notably, HiVA's memory mechanism exhibits hierarchical characteristics. At the macro level, the topological structure $\mathcal{G}$ stores long-term memory of collaboration patterns among agents; at the meso level, edge weights w_{ij} record the effectiveness of specific collaboration paths; at the micro level, agents' semantic parameters Θ_i preserve individual specialized knowledge. This ensures that the system can utilize historical experience at different granularities.

3.6 Semantic-Topological Evolution

HiVA's core innovation lies in simultaneously optimizing agent semantic parameters and network topology from singleton through two complementary functions f_P and f_G, as illustrated in Figure 2. Detailed prompting strategies can be found in the Prompting Strategies section of the Appendix.

Semantic evolution function f_P handles semantic parameter evolution. Given textual gradient $\frac{\partial \mathcal{L}_t}{\partial v_i}$, current semantic parameters p_t, and tool configuration t_i, it produces updated parameters. The LLM analyzes the feedback to identify potential improvements in the agent's prompt and tool definition, and then generates refined prompts that address issues.

Topological evolution function f_G modifies the local topology at successor nodes of v_i based on textual gradients, topological connections, and task contribution matrix $\mathcal{R}$. Further, f_G instructs an LLM to analyze local topological neighborhoods and determine optimal structural modifications, including adding connections, removing redundant edges, or restructuring subgraphs.

4 Experiment

To comprehensively assess the capabilities of our **HiVA** framework, we have conducted experiments across diverse task-driven environments. These experiments evaluate HiVA's performance, efficiency, and the contributions of its key components through ablation studies, comparing it against state-of-the-art baselines.

4.1 Experimental Setup

We evaluated **HiVA** across mathematical reasoning, long-context multi-hop question answering, programmatic tasks, textual reasoning, and complex agentic environments, comparing it against state-of-the-art baselines. Ablation studies assessed the impact of HiVA's components: Topological Evolution (TEV), Semantic Evolution (SEV), Knowledge-Aware Bayesian-Bandit Routing (KABB), environment feedback (Env), and tool integration (Tool). Adaptability and scalability experiments are conducted on the MBPP dataset to assess performance at different optimization steps. Experiments are conducted on both open-sourced models (*e.g.*, `Qwen2.5-72B-Instruct-Turbo`) and close-sourced APIs (*e.g.*, `GPT-4o-mini`). All experiments are conducted with a fixed temperature of 1.0.

Datasets We used the following benchmark datasets to cover diverse task domains: (1) Mathematical Reasoning: MATH (Hendrycks et al. 2021b) for complex problems and GSM-8K (Cobbe et al. 2021) for elementary problems; (2) Long-context Question Answering: HotpotQA (Yang et al. 2018) and 2WikihopQA (Ho et al. 2020) for multi-hop reasoning; (3) Programmatic Tasks: HumanEval (Chen and et al. 2021) and MBPP (Odena and et al. 2021) for code generation and problem-solving, with MBPP also used for adaptability and scalability experiments; (4) Textual Reasoning: MMLU (Hendrycks et al. 2021a) for professional knowledge and BBH (Suzgun and Scales 2023) for challenging reasoning; and (5) Complex Agentic Environments: GAIA (Mialon et al. 2024) for interactive and agent-driven scenarios.

Evaluation Metrics The performance is measured using accuracy (%) averaged over five runs on randomly sampled data subsets. For the GAIA dataset, we evaluated performance using Accuracy (Acc) for task completion and Cost-efficiency Score (CS), calculated as Accuracy divided by the LLM's cost in dollars. Higher CS values indicate better cost-efficiency. The CS calculation method and sample strategy are detailed in the Appendix on Experimental Setup Details.

Baselines We compared HiVA against: **CoT** (Wei et al. 2022): Step-by-step reasoning. **Self-Consistency** (Wang et al. 2023): Majority voting over multiple reasoning paths. **Self-Refine** (Madaan et al. 2023): Iterative output refinement. **Multi-Agent Debate** (Du et al. 2023): Collaborative multi-agent reasoning. **DyLAN** (Liu et al. 2023): Dynamic learning agent network. **AgentVerse** (Chen et al. 2024): Multi-agent coordination framework. **ADAS** (Hu, Lu, and Clune 2025): Adaptive agent optimization. **MaAS** (Zhang et al. 2025a): Multi-agent system with specialized roles.

4.2 Main Results

We evaluated HiVA and baseline methods across mathematical, long-context, programmatic, and textual reasoning tasks, as shown in Table 1. The results highlight HiVA's superior performance, achieving the highest average accuracy of 89.2% (+8.0% over Vanilla), with leading scores in GSM-8K (94.5%, +3.6%), HotpotQA (79.7%, +18.3%), 2WikihopQA (86.5%, +13.5%), MMLU (91.7%, +6.5%), and BBH (93.4%, +8.2%). HiVA's semantic-topological evolution and knowledge-aware routing excel in tasks requiring multi-step reasoning and complex interactions, particularly in web-based and textual domains. Compared to MaAS (86.3% average), which performs strongly in programmatic tasks (94.2% on HumanEval, 90.1% on MBPP), and ADAS (85.7% average), which leads in MATH (86.1%), HiVA demonstrates broader robustness. However, HiVA's performance on MATH (81.2%, -1.8%) is slightly below Vanilla. As in our qualitative case study (Figure 4), this performance drop can be attributed to the aggregator's difficulty in resolving conflicting answers generated by different agents during

Method	Mathematical		Long-context		Programmatic		Textual		Avg.
	MATH	GSM-8K	HotpotQA	2WikihopQA	HumanEval	MBPP	MMLU	BBH	
Vanilla	82.7	91.2	67.4	76.2	86.1	86.7	86.1	86.3	82.6
CoT	$84.3_{\uparrow 1.6\%}$	$92.1_{\uparrow 0.9\%}$	$68.8_{\uparrow 1.9\%}$	$77.6_{\uparrow 1.7\%}$	$87.4_{\uparrow 1.5\%}$	$87.3_{\uparrow 0.9\%}$	$85.2_{\downarrow 0.7\%}$	$87.6_{\uparrow 1.4\%}$	$83.8_{\uparrow 1.5\%}$
Self-Consistency	$84.7_{\uparrow 2.2\%}$	$92.6_{\uparrow 1.4\%}$	$69.1_{\uparrow 2.4\%}$	$78.3_{\uparrow 2.4\%}$	$88.2_{\uparrow 2.4\%}$	$88.2_{\uparrow 1.5\%}$	$85.6_{\downarrow 0.3\%}$	$88.1_{\uparrow 2.0\%}$	$84.4_{\uparrow 2.2\%}$
Self-Refine	$85.1_{\uparrow 2.8\%}$	$93.2_{\uparrow 2.0\%}$	$69.4_{\uparrow 3.1\%}$	$78.6_{\uparrow 3.0\%}$	$86.2_{\uparrow 0.1\%}$	$87.1_{\uparrow 0.3\%}$	$85.1_{\downarrow 1.3\%}$	$87.2_{\uparrow 0.8\%}$	$84.0_{\uparrow 1.7\%}$
Multi-Agent Debate	$85.4_{\uparrow 3.1\%}$	$93.3_{\uparrow 2.2\%}$	$70.2_{\uparrow 3.9\%}$	$79.1_{\uparrow 3.7\%}$	$88.1_{\uparrow 2.3\%}$	$87.4_{\uparrow 0.9\%}$	$85.4_{\downarrow 0.7\%}$	$87.6_{\uparrow 1.4\%}$	$84.6_{\uparrow 2.4\%}$
DyLAN	$85.3_{\uparrow 3.0\%}$	$92.9_{\uparrow 1.8\%}$	$69.7_{\uparrow 3.1\%}$	$78.8_{\uparrow 3.3\%}$	$89.7_{\uparrow 4.2\%}$	$87.3_{\uparrow 0.6\%}$	$85.2_{\downarrow 1.3\%}$	$87.1_{\uparrow 0.8\%}$	$84.5_{\uparrow 2.3\%}$
AgentVerse	$85.6_{\uparrow 3.3\%}$	$93.1_{\uparrow 2.1\%}$	$70.1_{\uparrow 3.9\%}$	$79.3_{\uparrow 3.9\%}$	$89.6_{\uparrow 4.1\%}$	$87.6_{\uparrow 0.9\%}$	$85.7_{\downarrow 0.7\%}$	$87.4_{\uparrow 1.4\%}$	$84.8_{\uparrow 2.7\%}$
ADAS	$\mathbf{86.1}_{\uparrow \mathbf{4.0\%}}$	$93.4_{\uparrow 2.5\%}$	$72.3_{\uparrow 6.8\%}$	$80.7_{\uparrow 5.6\%}$	$85.2_{\downarrow 1.0\%}$	$89.2_{\uparrow 2.8\%}$	$87.2_{\uparrow 1.0\%}$	$88.6_{\uparrow 2.5\%}$	$85.3_{\uparrow 3.3\%}$
MaAS	$85.7_{\uparrow 3.6\%}$	$94.1_{\uparrow 3.2\%}$	$76.2_{\uparrow 13.1\%}$	$81.1_{\uparrow 6.4\%}$	$92.3_{\uparrow 7.2\%}$	$90.1_{\uparrow 3.9\%}$	$89.4_{\uparrow 3.8\%}$	$90.6_{\uparrow 5.0\%}$	$87.4_{\uparrow 5.8\%}$
HiVA (ours)	$81.2_{\downarrow 1.8\%}$	$\mathbf{94.5}_{\uparrow \mathbf{3.6\%}}$	$\mathbf{79.7}_{\uparrow \mathbf{18.3\%}}$	$\mathbf{86.5}_{\uparrow \mathbf{13.5\%}}$	$\mathbf{94.2}_{\uparrow \mathbf{9.4\%}}$	$\mathbf{92.1}_{\uparrow \mathbf{6.2\%}}$	$\mathbf{91.7}_{\uparrow \mathbf{6.5\%}}$	$\mathbf{93.4}_{\uparrow \mathbf{8.2\%}}$	$\mathbf{89.2}_{\uparrow \mathbf{8.0\%}}$

Table 1: Performance comparison across different task-driven environments on `Qwen-2.5-72B-Instruct-Turbo`. Results show accuracy scores (%) for each method across programmatic, textual, long-context, and mathematical reasoning tasks. Bold indicates best performance. Subscript values denote standard deviation across three runs.

parallel verification. When faced with contradictory results, the aggregator can get "stuck" and fail to produce a final answer, revealing a limitation in handling tasks that require strict logical consistency across multiple reasoning paths. Despite this, its strong results in MBPP (92.1%, +6.2%) and knowledge-intensive reasoning tasks underscore its effectiveness in handling other types of intricate dependencies.

In the complex agentic environment (GAIA), we assessed both Accuracy (Acc) and Cost-efficiency Score (CS), as shown in Figure 3. HiVA consistently outperforms MaAS and AutoGPT in Acc across all task levels (Level-1: 26.2%, Level-2: 24.3%, Level-3: 11.1% vs. MaAS: 25.2%, 22.0%, 6.3% and AutoGPT: 13.21%, 0.0%, 3.9%). The largest performance gap is in Level-2, where HiVA achieves 24.3% Acc compared to MaAS's 22.0% and AutoGPT's 0.0%. Additionally, HiVA achieves the highest average CS (5.5) compared to MaAS (5.2) and AutoGPT (1.3), indicating superior cost-efficiency with fewer LLM calls. HiVA's dynamic routing and adaptive agent coordination enable it to balance high accuracy with efficient resource use, outperforming workflow-based and reactive-loop-based methods in GAIA's multi-step tasks. More experiments on optimization cost can be found in the Scalability Analysis section of the Appendix.

To evaluate HiVA's adaptability and scalability, we conducted experiments on the MBPP dataset with tasks of increasing complexity (Introductory, Interview, Competition). Figure 5 illustrates performance trends over 10 iterations for HiVA, MaAS, and TextGrad. HiVA demonstrates superior adaptability, improving from 86.3% to 91.7% (+5.4%) over 10 iterations, surpassing MaAS (86.3% to 90.6%, +4.3%) after iteration 4 and TextGrad (86.3% to 87.4%, +1.1%), which plateaus early. HiVA's steady performance gains, driven by its topological optimization and knowledge-aware routing, highlight its ability to scale effectively with task complexity while maintaining high accuracy.

4.3 Qualitative Case Study

To gain deeper insights into HiVA's behavior, we conducted a qualitative case study on representative tasks from each domain. For mathematical reasoning (MATH), we observed that HiVA's textual gradient mechanism effectively

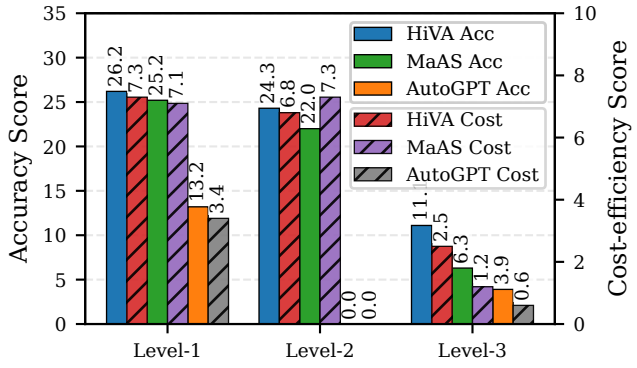

Figure 3: Evaluations in complex agentic environments. We compare two mainstream agentic frameworks (*i.e.*, MaAS, AutoGPT) with HiVA in an open, complex benchmark (*i.e.*, GAIA) and evaluate their performance through Accuracy (Acc) and Cost-efficiency Score (CS).

refines intermediate solutions, improving accuracy. In multi-hop question answering (HotpotQA), its knowledge-aware bandit-based routing dynamically selects relevant agents, enhancing multi-hop reasoning. For programmatic tasks (HumanEval), HiVA's topological optimization ensures efficient agent collaboration for robust code generation. In textual reasoning (MMLU), integrating environment feedback and tools enables precise knowledge retrieval. These observations highlight HiVA's ability to synergistically leverage its components across diverse tasks. Two illustrative cases, shown in Figure 4, demonstrate HiVA's capabilities and limitations. More cases can be found in the Appendix on Qualitative Case Study.

The contrasting cases reveal critical factors determining HiVA's effectiveness: **Task Decomposability**: Multi-hop QA benefits from natural sequential decomposition, while complex bug fixing requires simultaneous reasoning across multiple interdependent components. **Context Scalability**: Success correlates with tasks fitting within LLM context windows and agent specialization boundaries. **Feed-**

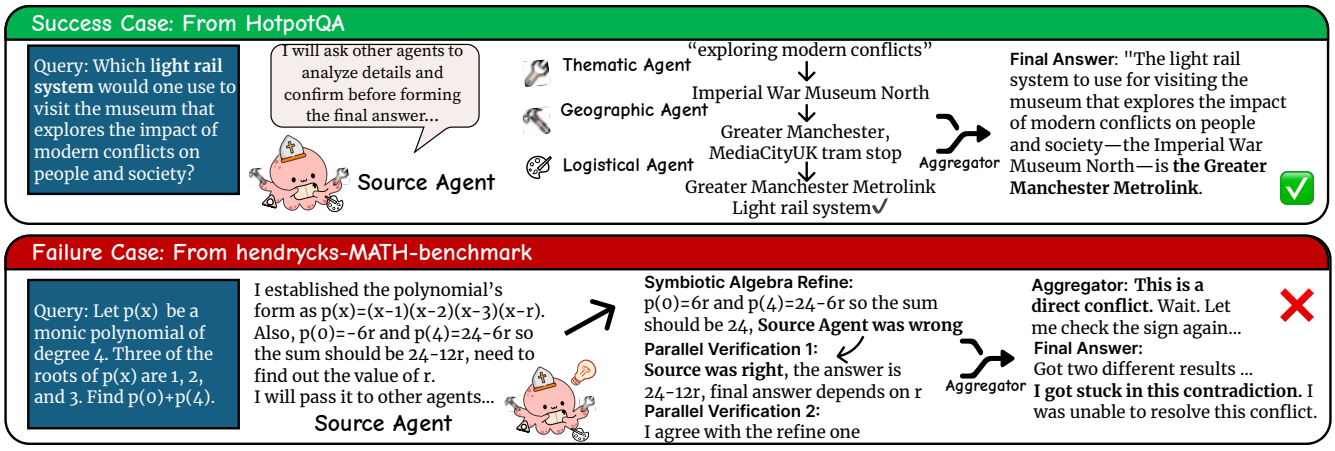

Figure 4: **Comparative Evolution Trajectories: Success vs. Failure Cases**. We evaluate our algorithm on HotpotQA and MATH, and choose two evolution trajectories (*i.e.*, success and failure cases) from them.

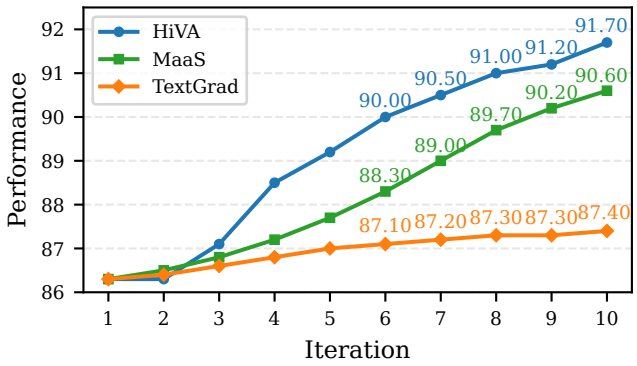

Figure 5: Adaptability and scalability trends of HiVA. Our HiVA is compared with MaAS and TextGrad across coding tasks in the MBPP of increasing iteration steps.

back Quality: Clear, actionable environmental feedback enables effective textual gradient propagation, while ambiguous signals lead to local optimization traps.

4.4 Ablation Studies

To understand the contribution of HiVA's key components, we conducted ablation studies by removing Topological Evolution (TEV), Semantic Evolution (SEV), Knowledge-Aware Bandit-Based Routing (KABB), environment feedback (Env), and tool integration (Tool). Table 2 presents the results on HotpotQA, MBPP, and MMLU tasks.

The results validate our core hypothesis that both semantic and topological evolution are essential for adaptive intelligence. Semantic Evolution (SEV) proves most critical, with removal causing the largest degradation (10.7% on HotpotQA, 5.2% on MMLU), confirming that agents must learn "what each agent should do." Topological Evolution (TEV) is equally important, particularly for multi-hop reasoning (7.3% drop on HotpotQA), validating our claim that systems must evolve "how agents should interact and organize." The synergistic effect of TEV and SEV demonstrates our Semantic-Topological Evolution (STEV) algorithm. Neither component alone achieves optimal performance. Knowledge-Aware Bandit-Based Routing (KABB) consistently contributes (4.4% decreases), confirming effective exploration of the evolved topological space. Environment feedback (Env) provides steady improvements, validating our textual gradient approach. More details can be found in the Appendix on Ablation Study Details.

Method	HotpotQA	MBPP	MMLU	Avg.
HiVA (ours)	**79.7**	**92.1**	**91.7**	**87.8**
HiVA w/o TEV	$74.0_{\downarrow 7.3\%}$	$88.9_{\downarrow 3.5\%}$	$88.3_{\downarrow 3.7\%}$	83.7
HiVA w/o SEV	$71.2_{\downarrow 10.7\%}$	$88.4_{\downarrow 4.0\%}$	$86.9_{\downarrow 5.2\%}$	82.2
HiVA w/o KABB	$76.2_{\downarrow 4.4\%}$	$88.1_{\downarrow 4.4\%}$	$90.6_{\downarrow 1.2\%}$	85.0
HiVA w/o Env	$75.2_{\downarrow 5.7\%}$	$89.3_{\downarrow 3.1\%}$	$89.5_{\downarrow 2.4\%}$	84.7
HiVA w/o Tool	$74.8_{\downarrow 6.1\%}$	$94.1_{\uparrow 2.2\%}$	$89.1_{\downarrow 2.8\%}$	84.3

Table 2: Ablation Study of HiVA. Results show accuracy scores (%) for each method across HotpotQA, programmatic (MBPP), and textual (MMLU) reasoning tasks. Bold indicates best performance. Subscript arrows denote relative change from full HiVA.

5 Conclusion

We introduce HiVA, a framework for self-organized multi-agent systems that jointly evolves agent behaviors (semantics) and their collaboration structure (topology). Guided by environmental feedback, HiVA consistently outperforms baselines in accuracy and cost-efficiency on complex tasks like mathematical reasoning and code generation. Our work confirms that this co-evolution of semantics and topology is critical for optimal adaptation. Future work may develop more operable and effective tool-calling methods to better handle the challenges of dynamic environments.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62276283, in part by the China Meteorological Administration's Science and Technology Project under Grant CMA-JBGS202517, in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515012985, in part by Guangdong-Hong Kong-Macao Greater Bay Area Meteorological Technology Collaborative Research Project under Grant GHMA2024Z04, in part by Fundamental Research Funds for the Central Universities, Sun Yat-sen University under Grant 23hytd006, and in part by Guangdong Provincial High-Level Young Talent Program under Grant RL2024-151-2-11.

References

Chen, M.; and et al., J. T. 2021. Evaluating Large Language Models Trained on Code.

Chen, W.; Su, Y.; Zuo, J.; Yang, C.; Yuan, C.; Chan, C.-M.; Yu, H.; Lu, Y.; Hung, Y.-H.; Qian, C.; Qin, Y.; Cong, X.; Xie, R.; Liu, Z.; Sun, M.; and Zhou, J. 2024. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. In *ICLR*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*.

Du, Y.; Li, S.; Torralba, A.; Tenenbaum, J. B.; and Mordatch, I. 2023. Improving factuality and reasoning in language models through multiagent debate. In *ICML*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N. V.; Wiest, O.; and Zhang, X. 2024. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In *IJCAI*.

He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; and Yu, D. 2024. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *ACL*, 6864–6890. Bangkok, Thailand: Association for Computational Linguistics.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021a. Measuring Massive Multitask Language Understanding. *ICLR*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021b. Measuring Mathematical Problem Solving With the MATH Dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Ho, X.; Duong Nguyen, A.-K.; Sugawara, S.; and Aizawa, A. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6609–6625. Barcelona, Spain (Online): International Committee on Computational Linguistics.

Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Wang, J.; Zhang, C.; Wang, Z.; Yau, S. K. S.; Lin, Z.; Zhou, L.; Ran, C.; Xiao, L.; Wu, C.; and Schmidhuber, J. 2024. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *The Twelfth International Conference on Learning Representations*.

Hu, M.; Zhou, Y.; Fan, W.; Nie, Y.; Xia, B.; Sun, T.; Ye, Z.; Jin, Z.; Li, Y.; Chen, Q.; Zhang, Z.; Wang, Y.; Ye, Q.; Ghanem, B.; Luo, P.; and Li, G. 2025. OWL: Optimized Workforce Learning for General Multi-Agent Assistance in Real-World Task Automation. In *ICML*.

Hu, S.; Lu, C.; and Clune, J. 2025. Automated Design of Agentic Systems. In *The Thirteenth International Conference on Learning Representations*.

Jimenez-Romero, C.; Yegenoglu, A.; and Blum, C. 2025. Multi-agent systems powered by large language models: applications in swarm intelligence. *Frontiers in Artificial Intelligence*, Volume 8 - 2025.

Junda He, D. L., Christoph Treude. 2025. LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision, and the Road Ahead. *ACM TRANSACTIONS ON SOFTWARE ENGINEERING AND METHODOLOGY*, –.

Li, X. 2025. A Review of Prominent Paradigms for LLM-Based Agents: Tool Use, Planning (Including RAG), and Feedback Learning. In Rambow, O.; Wanner, L.; Apidianaki, M.; Al-Khalifa, H.; Eugenio, B. D.; and Schockaert, S., eds., *Proceedings of the 31st International Conference on Computational Linguistics*, 9760–9779. Abu Dhabi, UAE: Association for Computational Linguistics.

Li, X.; Zhang, J.; and Safara, F. 2023. Improving the Accuracy of Diabetes Diagnosis Applications through a Hybrid Feature Selection Algorithm. *Neural Processing Letters*, 55: 153–169.

Liang, S.; Xu, K.; and Dong, Z. 2025. A Multi-Agent Approach to Modeling Task-Oriented Dialog Policy Learning. *IEEE Access*, 13: 11754–11764.

Liu, Z.; Zhang, Y.; Li, P.; Liu, Y.; and Yang, D. 2023. Dynamic LLM-Agent Network: An LLM-agent Collaboration Framework with Agent Team Optimization. *CoRR*, abs/2310.02170.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; et al. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *NeurIPS*, 36: 46534–46594.

Mialon, G.; Dessi, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; Roziere, B.; Schick, T.; Dwivedi-Yu, J.; Celikyilmaz, A.; Grave, E.; LeCun, Y.; and Scialom, T. 2023. Augmented Language Models: a Survey. *Transactions on Machine Learning Research*. Survey Certification.

Mialon, G.; Fourrier, C.; Wolf, T.; LeCun, Y.; and Scialom, T. 2024. GAIA: a benchmark for General AI Assistants. In *ICLR*.

Odena, A.; and et al., C. S. 2021. Program Synthesis with Large Language Models. In *Communications of the ACM, Volume 68, Issue 1*, n/a. n/a. N/a.

Qin, R.; Chen, F.; Wang, T.; Yuan, L.; Wu, X.; Zhang, Z.; Zhang, C.; and Yu, Y. 2022. Multi-Agent Policy Transfer via Task Relationship Modeling. In *AAMAS*, XXX–XXX. IFAAMAS.

Raman, R.; Kowalski, R.; Achuthan, K.; Iyer, A.; Parthasarathi, S.; Rangan, K.; and Borthakur, D. 2025. Navigating artificial general intelligence development: societal, technological, ethical, and brain-inspired pathways. *Scientific Reports*, 15(1): 8443.

Sheng, J. Z. 2025. GAM-Agent: Game-Theoretic and Uncertainty-Aware Collaboration for Complex Visual Reasoning. https://arxiv.org/abs/2505.23399. ArXiv:2505.23399, arXiv:2505.23399.

Su, H.; Chen, R.; Tang, S.; Yin, Z.; Zheng, X.; Li, J.; Qi, B.; Wu, Q.; Li, H.; Ouyang, W.; Torr, P.; Zhou, B.; and Dong, N. 2025. Many Heads Are Better Than One: Improved Scientific Idea Generation by A LLM-Based Multi-Agent System. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *ACL*, 28201–28240. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Suzgun, M.; and Scales, N. e. a. 2023. Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of ACL*, 13003–13051. Toronto, Canada: Association for Computational Linguistics.

Wadhwa, N.; Sonwane, A.; Arora, D.; Mehrotra, A.; Utpala, S.; Bairi, R. B.; Kanade, A.; and Natarajan, N. 2024. MASAI: Modular Architecture for Software-engineering AI Agents. In *NeurIPS Workshop*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q. V.; Chi, E. H.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *ICLR*.

Webb, T.; Holyoak, K. J.; and Lu, H. 2022. Emergent Analogical Reasoning in Large Language Models. *PNAS Nexus*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS*, 35: 24824–24837.

Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; Zheng, R.; Fan, X.; Wang, X.; Xiong, L.; Zhou, Y.; Wang, W.; Jiang, C.; Zou, Y.; Liu, X.; Yin, Z.; Dou, S.; Weng, R.; Cheng, W.; Zhang, Q.; Qin, W.; Zheng, Y.; Qiu, X.; Huang, X.; and Gui, T. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. *CoRR*, abs/2309.07864.

Yang, H.; Yue, S.; and He, Y. 2023. Auto-GPT for Online Decision Making: Benchmarks and Additional Opinions. *CoRR*, abs/2306.02224.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *EMNLP*.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K. R.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *ICLR*.

Ye, R.; Tang, S.; Ge, R.; Du, Y.; Yin, Z.; Chen, S.; and Shao, J. 2025. MAS-GPT: Training LLMs to Build LLM-based Multi-Agent Systems. In *ICML*.

Yuksekgonul, M.; Bianchi, F.; Boen, J.; Liu, S.; Lu, P.; Huang, Z.; Guestrin, C.; and Zou, J. 2025. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639: 609–616.

Zhang, C.; Yang, K.; Hu, S.; Wang, Z.; Li, G.; Sun, Y.; Zhang, C.; Zhang, Z.; Liu, A.; Zhu, S.; Chang, X.; Zhang, J.; Yin, F.; Liang, Y.; and Yang, Y. 2024. ProAgent: Building Proactive Cooperative Agents with Large Language Models. In *AAAI*. Peer-reviewed full paper.

Zhang, G.; Niu, L.; Fang, J.; Wang, K.; BAI, L.; and Wang, X. 2025a. Multi-agent Architecture Search via Agentic Supernet. In *ICML*.

Zhang, G.; Yue, Y.; Sun, X.; Wan, G.; Yu, M.; Fang, J.; Wang, K.; Chen, T.; and Cheng, D. 2025b. G-Designer: Architecting Multi-agent Communication Topologies via Graph Neural Networks. In *ICLR Workshop*.

Zhang, G.; Yue, Y.; Sun, X.; Wan, G.; Yu, M.; Fang, J.; Wang, K.; Chen, T.; and Cheng, D. 2025c. G-Designer: Architecting Multi-agent Communication Topologies via Graph Neural Networks. In *ICLR Workshop*.

Zhang, J.; Fan, Y.; Cai, K.; and Wang, K. 2025d. Kolmogorov-Arnold Fourier Networks. *arXiv preprint arXiv:2502.06018*.

Zhang, J.; Huang, Z.; Fan, Y.; Liu, N.; Li, M.; Yang, Z.; Yao, J.; Wang, J.; and Wang, K. 2025e. KABB: Knowledge-Aware Bayesian Bandits for Dynamic Expert Coordination in Multi-Agent Systems. In *ICML*.

Zhang, S.; Yin, M.; Zhang, J.; Liu, J.; Han, Z.; Zhang, J.; Li, B.; Wang, C.; Wang, H.; Chen, Y.; and Wu, Q. 2025f. Which Agent Causes Task Failures and When? On Automated Failure Attribution of LLM Multi-Agent Systems. In *ICML*.

Zhou, H.; Wan, X.; Sun, R.; Palangi, H.; Iqbal, S.; Vulic, I.; Korhonen, A.; and Arik, S. O. 2025. Multi-Agent Design: Optimizing Agents with Better Prompts and Topologies. *CoRR*, abs/2502.02533.