SafeSieve: From Heuristics to Experience in Progressive Pruning for LLM-based Multi-Agent Communication

Ruijia Zhang[1], Xinyan Zhao[1], Ruixiang Wang[1], Sigen Chen[1], Guibin Zhang[1], An Zhang[2], Kun Wang[3], Qingsong Wen[4]

[1] National University of Singapore, Singapore
[2] School of Public Policy and Administration, Chongqing University, China
[3] Nanyang Technological University, Singapore
[4] AI Research Institute, Squirrel Ai Learning, China
e0536882@u.nus.edu, zhao_xinyan@u.nus.edu, ruixiangw@u.nus.edu, e1538182@u.nus.edu,
guibinz@outlook.com, 20210101013@cqu.edu.cn, wk520529@mail.ustc.edu.cn, qingsongedu@gmail.com

Abstract

LLM-based multi-agent systems exhibit strong collaborative capabilities but often suffer from redundant communication and excessive token overhead. Existing methods typically enhance efficiency through pretrained GNNs or greedy algorithms, but often isolate pre- and post-task optimization, lacking a unified strategy. To this end, we present SafeSieve, a progressive and adaptive multi-agent pruning algorithm that dynamically refines the inter-agent communication through a novel dual-mechanism. SafeSieve integrates initial LLM-based semantic evaluation with accumulated performance feedback, enabling a smooth transition from heuristic initialization to experience-driven refinement. Unlike existing greedy Top-k pruning methods, SafeSieve employs 0-extension clustering to preserve structurally coherent agent groups while eliminating ineffective links. Experiments across benchmarks (SVAMP, HumanEval, etc.) showcase that SafeSieve achieves 94.01% average accuracy while reducing token usage by 12.4%-27.8%. Results further demonstrate robustness under prompt injection attacks (1.23% average accuracy drop). In heterogeneous settings, SafeSieve reduces deployment costs by 13.3% while maintaining performance. These results establish SafeSieve as an efficient, GPU-free, and scalable framework for practical multi-agent systems. Our code can be found below.

Code — https://github.com/csgen/SafeSieve

Introduction

Large language model (LLM) based multi-agent systems (MAS) have demonstrated impressive collaborative problem-solving capabilities (Wang et al. 2025a; Chang et al. 2024), fueling frameworks such as AutoGen and ChatDev for real-world applications (Wu et al. 2023; Chen et al. 2023). Nevertheless, the dense, round-robin conversations among agents often incur substantial token overhead and communication redundancy, which not only elevates inference cost but also dilutes attention over key information, leading to potential accuracy degradation (Liu et al. 2024a). Longer context windows further enlarge the attack surface for prompt-injection (Anil et al. 2024). Consequently, recent studies have begun to sparsify MAS communication topologies to improve both efficiency and robustness (Zhuge et al. 2024; Zhang, Yue et al. 2024b,a).

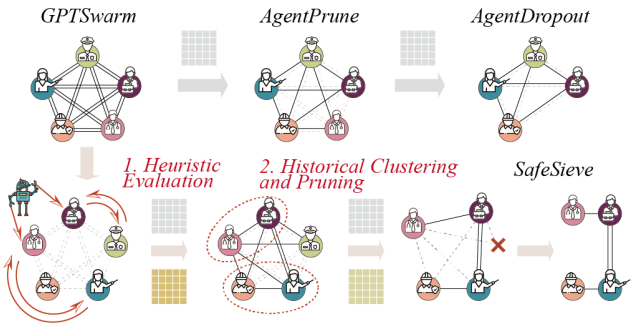

Figure 1: Comparison of **SafeSieve** with GPTSwarm, AgentPrune, and AgentDropout. It illustrates the evolutionary trajectory of post-pruning MAS, highlighting SafeSieve's novel contribution as a unified design that bridges early-stage heuristics and feedback-driven refinement.

Among communication sparsification strategies, one prevalent line of work directly constructs compact graph topologies prior to execution, such as G-Designer and GPTSwarm (Zhang, Yue et al. 2024b; Zhuge et al. 2024). These methods improve communication efficiency at initialization but exhibit limited generalizability and cannot adapt to run-time dynamics. A more recent and adaptive paradigm is on-the-fly pruning, where the communication graph is dynamically adjusted based on task performance feedback. Representative works like AgentPrune and AgentDropout (Zhang, Yue et al. 2024a; Wang, Wang et al. 2025) start from a basic or fully-connected topology and iteratively prune edges during execution. These approaches require no pretraining and offer strong task adaptability with minimal deployment burden. However, most of them still rely on greedy *top-k* pruning strategies, which may mistakenly remove critical communication paths, reducing system robustness. To date, *no method* has yet unified both heuristic-driven early filtering and performance-aware dynamic adaptation, form-

ing a full-spectrum optimization pipeline.

Motivated by the similarity between MAS collaboration and human team organization (Guo et al. 2024), we propose **SafeSieve**, a progressive and adaptive pruning algorithm. SafeSieve introduces a two-stage edge scoring scheme that ① uses LLM-based semantic compatibility to offer heuristic guidance at startup and ② gradually shifts weight to accumulated contribution during execution, emulating the "plan-then-adjust" paradigm of human teamwork, as shown in Figure 1. Instead of pruning edges individually, SafeSieve employs a *0-extension* based clustering mechanism (Fakcharoenphol et al. 2003), preserving structurally coherent agent groups while eliminating ineffective links. This design avoids the local sub-optimality of greedy top-k pruning and retains inter-agent complementarity.

Comprehensive experiments on six benchmarks (including GSM8K, SVAMP, HumanEval, AQuA, MMLU, MATH-500) showcase that SafeSieve reduces token usage by 12.4–27.8% and boosts accuracy by up to 2.22%, consistently outperforming prior sparsification methods. It further remains resilient under prompt-injection, suffering only a 1.23-1.94% accuracy drop, and supports heterogeneous collaboration where large LLMs guide smaller ones, thus expanding the real-world deployment space.

Our main contributions are threefold:

- **Unified Framework.** We categorize MAS communication optimization into *pre-design* and *post-prune* paradigms, and propose SafeSieve—the first post-pruning framework that integrates LLM-based semantic evaluation, cumulative historical feedback, and 0-extension clustering to achieve progressive graph sparsification while preserving agent complementarity.
- **Efficiency with Robustness.** Extensive experiments across six benchmarks demonstrate that SafeSieve reduces token consumption by 12.4%—27.8% compared to peer methods while maintaining or improving accuracy by up to 2.22%. Uniquely among post-prune optimizers, SafeSieve exhibits inherent adversarial resilience, detecting and mitigating malicious agents with 1.23% degradation.
- **Heterogeneous Deployment.** We pioneer heterogeneous multi-agent evaluation by systematically analyzing cross-model collaboration with real-time cost tracking, revealing that SafeSieve's clustering mechanism effectively leverages model diversity to reduce deployment costs by up to 13.3% in production settings.

Related Work and Preliminary

Communication Efficiency in MAS. Recent research in LLM-based MAS has explored two primary paradigms for optimizing communication efficiency: *pre-design* approaches that construct optimized communication structures before the task, and *post-prune* methods that start with various basic topologies and iteratively remove redundant connections. They reflect the trade-off between upfront design complexity and runtime adaptability in MAS.

Pre-design methods, including GPTSwarm (Zhuge et al. 2024), G-Designer (Zhang, Yue et al. 2024b), AnyMAC (Wang et al. 2025b), EvoMAC (Hu et al. 2024), and DyLAN (Liu et al. 2024c), focus on directly generating efficient communication topologies—whether through GNN-based graph construction, autoregressive agent selection, evolutionary adaptation, or two-stage team formation.

Post-pruning approaches begin with dense communication topology and progressively sparsify them: AgentPrune (Zhang, Yue et al. 2024a) introduces dynamic edge pruning via one-hot mask matrices, AgentDropout (Wang, Wang et al. 2025) extends this to node-level pruning, Adaptive Graph Pruning (Li et al. 2025) jointly learns node selection and edge connectivity via end-to-end GNN training, and Adaptive Prompt Pruning (Dong, Chen, and Chi 2024) reduces per-agent prompt lengths to save token.

Graph Clustering and 0-extension. Recent studies have shown that LLMs exhibit human-like collaborative patterns, where agents with similar or complementary capabilities naturally form effective working groups (Wang et al. 2024; Hong et al. 2023). Inspired by this observation, we propose that clustering-based pruning offers a more principled approach than direct edge removal in MAS collaboration.

While various clustering methods exist—including spectral clustering (Schaeffer 2007), hierarchical clustering (Xue et al. 2024), and density-based approaches (Birant and Kut 2007)—the k-terminal *0-extension problem* (Calinescu, Karloff, and Rabani 2003; Fakcharoenphol et al. 2003) presents unique advantages for our setting. 0-extension has been successfully applied in computing system for its computational efficiency ($O(n \log n)$ complexity), simple deployment, and strong theoretical guarantees in preserving graph connectivity (Englert et al. 2014; Chen and Gopalakrishnan 1998). By formulating agent clustering as a 0-extension problem, we replace aggressive top-k pruning with a connectivity-aware approach that maintains critical communication paths between agent communities while achieving similar sparsification rates.

MAS as a Communication Graph. Building upon the graph-based paradigm for multi-agent systems, GPTSwarm (Zhuge et al. 2024) first formalized multi-agent orchestration as a differentiable computational graph. At each communication round t, the interaction among N agents is represented as a directed communication graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$, where $\mathcal{V}$ denotes the set of agents and each edge $e_{ij} \in \mathcal{E}_t$ represents a message from agent i to agent j. While GPTSwarm generates static graph structures during inference, AgentPrune (Zhang, Yue et al. 2024a) introduces dynamic sparsification through a mask matrix $\mathbf{M}^{(t)} \in \{0,1\}^{N \times N}$, where each entry controls edge activation:

$$\tilde{\mathcal{E}}_t = \{e_{ij} \in \mathcal{E}_t : M_{ij}^{(t)} = 1\} \tag{1}$$

This mask effectively defines a candidate set of communication links from which the actual runtime communication graph is sampled. AgentDropout (Wang, Wang et al. 2025) extends this framework by incorporating node-level pruning and real-time feedback mechanisms, enabling more aggressive sparsification across communication rounds.

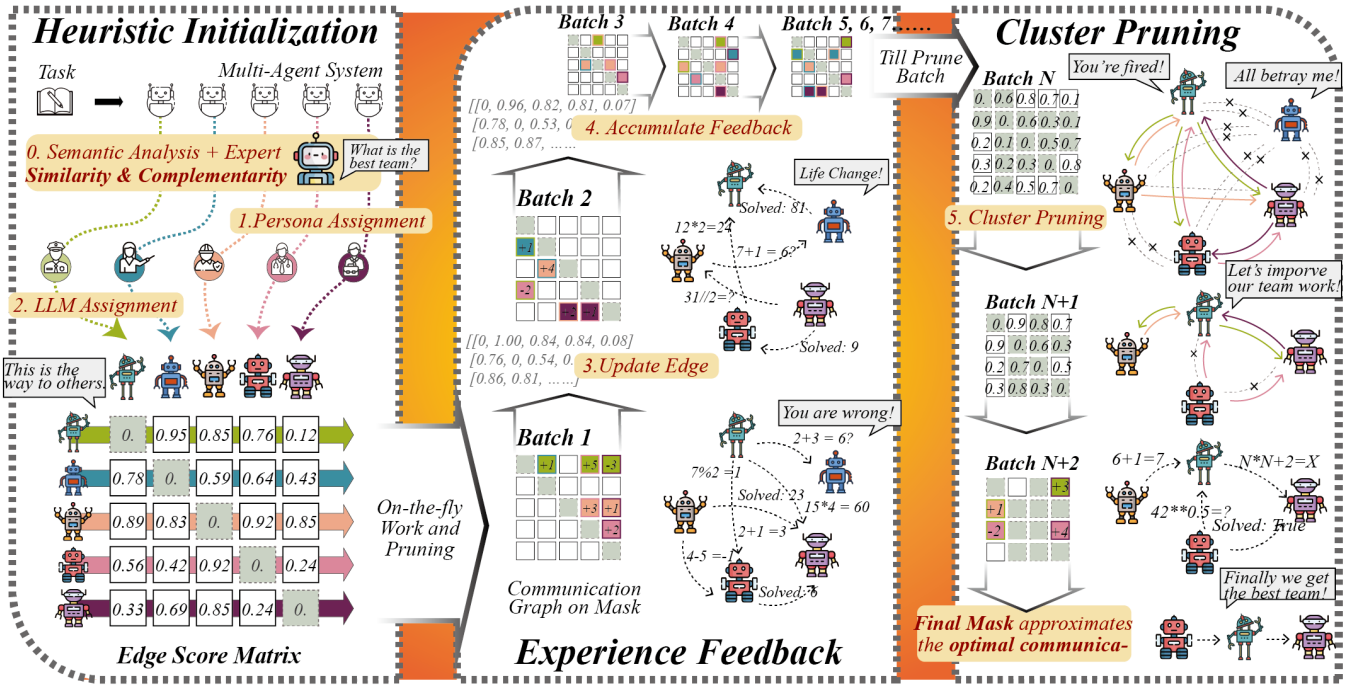

Figure 2: ***SafeSieve Pipeline***. The process begins by constructing a complete communication graph based on semantic relevance among agent roles. During task execution, edge importance is updated based on reasoning success, enabling adaptive pruning via 0-extension clustering. The final communication structure reflects a task-aware, resource-efficient collaboration topology.

SafeSieve: Integrated Pruning Strategy

SafeSieve progressively prunes communication graphs in MAS by assessing inter-agent link quality through a dynamic edge scoring matrix $E \in \mathbb{R}^{n \times n}$. We combine semantic compatibility—obtained from expert LLM assessments—with historical complementarity based on interaction outcomes. This dual mechanism enables adaptive edge pruning based on **time-varying** thresholds, while isolated nodes are naturally removed. 0-extension clustering is integrated to preserve coherent community structures and information flow. Figure 2 illustrates the whole pruning process.

Semantic Heuristic Initialization

While semantic similarity facilitates basic cooperation (Deng et al. 2025), functional complementarity plays a more critical role in complex multi-hop reasoning tasks (Zhang et al. 2024; Zhang, Chen, and Kumar 2025). SafeSieve initializes the communication edge score between agents i and j by combining embedding-based similarity with expert-assessed compatibility:

$$S_{ij}^{\text{compat}} = \gamma \cdot \frac{\mathbf{e}_i \cdot \mathbf{e}_j}{||\mathbf{e}_i|| \cdot ||\mathbf{e}_j||} + (1-\gamma) \cdot \mathcal{Q}(S_{ij}^{\text{expert}}) \quad (2)$$

where $\mathbf{e}_i, \mathbf{e}_j \in \mathbb{R}^d$ are pre-trained role embeddings that capture agent capabilities, $S_{ij}^{\text{expert}} \in [0,1]$ represents the functional compatibility score assessed by an expert LLM, and $\mathcal{Q}(\cdot)$ denotes a 5-level quantization function that discretizes continuous scores into categorical levels. The balance parameter $\gamma \in [0,1]$ controls the relative importance of semantic similarity versus expert-assessed complementarity.

Progressive Pruning with Historical Feedback

To capture the dynamic nature of agent cooperation, SafeSieve progressively shifts from static semantic initialization to experience-based scoring through a unified temporal mechanism (Zhang, Yang, and Başar 2022).

Historical Complementarity. The system tracks each edge's contribution to successful task completion over time. The historical complementarity score quantifies the accumulated value of edge (i,j) relative to all edges in the graph:

$$C_{ij}^{\text{hist}}(t) = \frac{\sum_{\tau=1}^{t} \mathbf{1}_{ij}^{\text{correct}}(\tau)}{\sum_{(k,l)\in E_t} \sum_{\tau=1}^{t} \mathbf{1}_{kl}^{\text{correct}}(\tau) + n^2\varepsilon} \quad (3)$$

Here, $\mathbf{1}_{ij}^{\text{correct}}(\tau) \in \{0,1\}$ indicates whether edge (i,j) contributed to a correct answer at time step τ, E_t is the set of active edges at time t, and n is the total number of agents.

Throughout SafeSieve, we use $\varepsilon > 0$ as a small constant to ensure numerical stability and prevent division by zero. The normalization term $n^2\varepsilon$ also accounts for the maximum possible number of edges in the complete graph.

Integrated Edge Scoring. The overall edge score dynamically combines semantic initialization with accumulated historical feedback, gradually emphasizing learned patterns over static heuristics with chronological weights:

$$E_{ij}(t) = \left(1 - \frac{t}{T}\right) \cdot \alpha_0 \cdot S_{ij}^{\text{compat}} + \left[\beta_0 + (\beta_{\max} - \beta_0) \cdot \frac{t}{T}\right] \cdot C_{ij}^{\text{hist}}(t) \tag{4}$$

where $\alpha_0 > 0$ is the initial weight for semantic compatibility, $\beta_0 \geq 0$ and $\beta_{\max} > \beta_0$ define the range of historical contribution weights, t is the current time step, and T is the total number of time steps. This formulation ensures a smooth transition from heuristics to experience evaluation.

0-Extension Clustering for Pruning Decisions

SafeSieve employs a principled clustering approach based on the 0-extension framework (Fakcharoenphol et al. 2003) to make globally-informed pruning decisions. This method provides theoretical approximation guarantees while maintaining computational and communicative efficiency.

Dynamic Threshold. The pruning threshold adapts over time to balance exploration and exploitation, starting conservative and growing aggressive as the system evolves:

$$\theta(t) = \theta_0 + (\theta_{\max} - \theta_0) \cdot \left[1 - \exp^{-k \cdot \max\left(\frac{t}{T}, 0\right)}\right] \tag{5}$$

where $\theta_0 \geq 0$ and $\theta_{\max} > \theta_0$ are the initial and maximum threshold values, and $k > 0$ is the growth rate parameter controlling how quickly the threshold increases.

Terminal Selection and Cluster Assignment. The clustering process begins by selecting a subset of terminal agents that serve as cluster centers. Terminals are selected to maximize overall connectivity transferred from edge scores:

$$T = \arg \max_{S \subseteq V, |S| = |T|} \sum_{v \in S} \sum_{u \in V} \frac{1}{(E_{vu}(t) + \varepsilon)^{-1}} \tag{6}$$

where V is the set of all agents. The number of terminals $|T|$ is determined adaptively based on the graph size: $|T| = \max(2, \min(\sqrt{n}, \lfloor n/3 \rfloor))$, ensuring at least two clusters while avoiding over-fragmentation. Each agent is then assigned to a terminal by solving the 0-extension problem:

$$f^* = \arg \min_{f: V \to T} \sum_{(i,j) \in E} (E_{ij}(t) + \varepsilon)^{-1} \cdot \mathbf{1}\{f(i) \neq f(j)\} \tag{7}$$

This optimization finds the cluster assignment f^* that minimizes the total distance of edges crossing cluster boundaries, where distance is inversely proportional to edge score.

Structured Edge Pruning. Pruning occurs at time steps t when both conditions are met: $t \geq B_{\text{start}}$ (warm-up period completed) and $\mathcal{R}(t) < R_{\max}$ (current pruning rate below maximum). The pruning set is constructed hierarchically to meet the target pruning rate $r \in (0, 1)$:

$$|\mathcal{E}_{\text{prune}}(t)| = r \cdot |\mathcal{E}_{\text{active}}^{(t)}| = |\mathcal{E}_{\text{rule}}(t) \cup \mathcal{E}_{\text{budget}}(t)| \tag{8}$$

The rule-based set $\mathcal{E}_{\text{rule}}(t)$ consists of edges that both cross cluster boundaries and fall below the threshold, specifically those satisfying the condition (i, j) such that $f^(i) \neq f^(j)$, $i, j \notin T$, and $\hat{E}_{ij}(t) < \theta(t)$. If this set is less than the pruning budget, $\mathcal{E}_{\text{budget}}(t)$ supplements it by adding the lowest-scoring remaining edges until the target is reached.

Mask and Node Update. The communication mask matrix $\mathbf{M} \in \{0, 1\}^{n \times n}$ is updated to reflect pruned edges:

$$\mathbf{M}_{ij}^{(t+1)} = \mathbf{M}_{ij}^{(t)} \cdot \mathbf{1}\{(i, j) \notin \mathcal{E}_{\text{prune}}(t)\} \tag{9}$$

After edge pruning, the node set is updated to remove isolated nodes only if a minimum viable graph is maintained:

$$V_{t+1} = \begin{cases} V_t \setminus \mathcal{V}_{\text{iso}}^{(t+1)} & \text{if } |V_t \setminus \mathcal{V}_{\text{iso}}^{(t+1)}| > 2 \\ V_t & \text{otherwise} \end{cases} \tag{10}$$

where $\mathcal{V}_{\text{iso}}^{(t+1)} = \{v \in V : \sum_{u \in V} \mathbf{M}_{vu}^{(t+1)} = 0\}$ denotes the set of isolated nodes after pruning.

Post-Pruning Regularization. To adapt to the changed graph structure, edge scores are normalized and historical weights are adjusted after each pruning step:

$$\hat{E}_{ij}(t) = \frac{E_{ij}(t) - \mu_t}{\sigma_t + \varepsilon}, \quad \hat{\beta}(t) = \beta(t) \cdot \frac{\Delta_{\text{before}}}{\Delta_{\text{after}} + \varepsilon} \tag{11}$$

where μ_t and σ_t are the mean and standard deviation of edge scores before pruning, and Δ_{before} and Δ_{after} represent the score range before and after pruning respectively. This regularization ensures that the scoring mechanism remains calibrated despite the evolving graph topology.

Further implementation details, including pseudocode and case study, are provided in Supplementary Material.

Experiments

Experiment Setup

Models & Benchmarks. In our main experiments, we adopt Deepseek-V3 (671B) (Liu et al. 2024b) as the primary backbone model. For smaller-model ablation, we use GPT-4o-mini. In heterogeneous settings, we additionally incorporate LLaMA3-8B (AI@Meta 2024), Qwen2.5-72B (Team 2024), and Kimi-K2 (AI 2024) to validate cross-model communication robustness. Build upon these, we evaluate general and mathematical reasoning using six standard benchmarks: MMLU (Hendrycks et al. 2021), GSM8K (Cobbe et al. 2021), SVAMP (Patel, Bhattamishra, and Goyal 2021), HumanEval (Chen et al. 2021), AQuA (Ling et al. 2017) and MATH-500 (Lightman et al. 2023).

Baselines. In the main experiments, we compare SafeSieve with prompting-based strategies including Vanilla (direct reasoning) and Chain-of-Thought (CoT) (Wei et al. 2023)(referred to as *single*), as well as collaborative frameworks such as GPT-Swarm (Zhuge et al. 2024) and G-Designer (Zhang, Yue et al. 2024b), which enhance agent outputs prior to inference (referred to as *pre-design*). We also include pruning-based baselines AgentPrune (Zhang,

Method	Paradigm	MMLU	GSM8K	SVAMP	HumanEval	AQuA	MATH-500	Avg.
Base model: deepseek-V3-671B								
Vanilla	single	87.97%	94.68%	93.67%	88.43%	84.58%	88.20%	89.59%
CoT	single	89.31%	95.15%	93.94%	89.26%	85.42%	90.41%	90.58%
GPTSwarm	pre-design	90.52%	94.83%	92.43%	90.14%	88.40%	90.56%	91.15%
G-Designer	pre-design	91.13%	95.47%	93.79%	90.93%	89.63%	91.02%	92.00%
AgentPrune	post-prune	90.99%	95.30%	95.40%	92.91%	90.30%	91.76%	92.78%
AgentDropout	post-prune	90.17%	95.16%	96.01%	93.16%	91.37%	89.82%	92.62%
SafeSieve (ours)	post-prune	**92.39%**	**96.27%**	**96.60%**	**95.01%**	**91.89%**	**91.90%**	**94.01%**
Base model: gpt-4o-mini (∼8B)								
Vanilla	single	77.81%	87.45%	88.26%	87.08%	71.42%	70.14%	80.36%
CoT	single	78.43%	87.10%	86.24%	88.13%	65.00%	77.18%	80.35%
GPTSwarm	pre-design	82.80%	89.14%	87.02%	**89.32%**	78.40%	80.70%	84.56%
G-Designer	pre-design	**87.20%**	93.97%	90.29%	87.50%	80.07%	81.76%	86.80%
AgentPrune	post-prune	83.30%	93.58%	93.05%	86.25%	84.71%	82.81%	87.28%
AgentDropout	post-prune	80.01%	93.25%	92.90%	85.41%	84.87%	81.97%	86.40%
SafeSieve (ours)	post-prune	82.32%	**93.61%**	**93.29%**	88.20%	**84.90%**	**83.33%**	**87.61%**

Table 1: Performance comparison between SafeSieve and baseline reasoning frameworks. Results for Vanilla and CoT under DeepSeek-V3 are adapted from AgentDropout (Wang, Wang et al. 2025) except MMLU and MATH-500; GPT-4o-mini results are taken from AGP (Li et al. 2025) except for MATH-500 and post-prune paradigm. Other results are evaluated by us under the same computing environment. All methods that start with a basic topology are based on **full-connected** graph.

Method	Acc.	Prom.	Comp.	↓Tokens
Full-connected (No Prune)	95.50%	321K	123K	—
No-Heuristic Cluster Prune	94.41%	233K	104K	24.2%
No-Historic Cluster Prune	93.78%	219K	92K	30.0%
Combined + Top-k Prune	93.13%	207K	107K	29.3%
SafeSieve (Ours)	95.01%	223K	98K	27.8%

Table 2: Ablation results on HumanEval. Compared to full-connected baseline, SafeSieve achieves comparable accuracy while reducing token usage by 27.8%.

Yue et al. 2024a) and AgentDropout (Wang, Wang et al. 2025), which dynamically remove redundant links in multi-agent communication graphs (referred to as *post-prune*). Since SafeSieve also belongs to the pruning paradigm, these two methods are further compared under prompt injection and heterogeneous-agent settings.

Main Results

Takeaway 1: Carefully designed communication graphs outperform single-agent methods, with post-pruning paradigms comprehensively surpassing pre-design approaches. Multi-agent collaboration demonstrates significant performance improvements, with SafeSieve achieving 94.01% average accuracy on DeepSeek-V3, substantially exceeding single-agent Vanilla (89.59%) and CoT (90.58%) baselines as shown in Table 1. Post-pruning methods universally outperform pre-design approaches, with AgentPrune (92.78%), AgentDropout (92.62%), and SafeSieve (94.01%) all surpassing GPTSwarm (91.15%) and G-Designer (92.00%). SafeSieve achieves the best performance among post-pruning methods, reaching 94.01% on DeepSeek-V3 and 87.61% on GPT-4o-mini, establishing itself as the optimal solution across both model scales.

Takeaway 2: Task-dependent performance improvements exhibit differentiated characteristics. Mathematical reasoning tasks (GSM8K, SVAMP) show stable improvements of 2-3 percentage points, with GSM8K improving from 94.68% to 96.27% (+1.59 points) and SVAMP from 93.67% to 96.60% (+2.93 points). Complex collaborative tasks demonstrate more significant gains, with MMLU improving by 4.42 points (87.97%→92.39%), HumanEval by 6.58 points (88.43%→95.01%), and AQuA by 7.31 points (84.58%→91.89%). This reflects SafeSieve's capability to identify and preserve critical heterogeneous communication paths through semantic similarity scoring.

Takeaway 3: Large and small models show differentiated improvements across task types, providing necessity for heterogeneous deployment. Large models (DeepSeek-V3) demonstrate greater improvements on complex tasks, with MMLU improving by 4.42 points and HumanEval by 6.58 points, showcasing advantages in handling complex collaboration. Small models (GPT-4o-mini) show higher improvements on structured tasks, with SVAMP achieving 5.7% improvement (88.26%→93.29%) and GSM8K achieving 7.1% relative improvement (87.45%→93.61%). This complementarity establishes the foundation for heterogeneous deployment, where large models excel at complex decision-making while small models are more efficient for structured tasks.

Takeaway 4: SafeSieve achieves superior efficiency-accuracy trade-off positioning across all benchmarks.

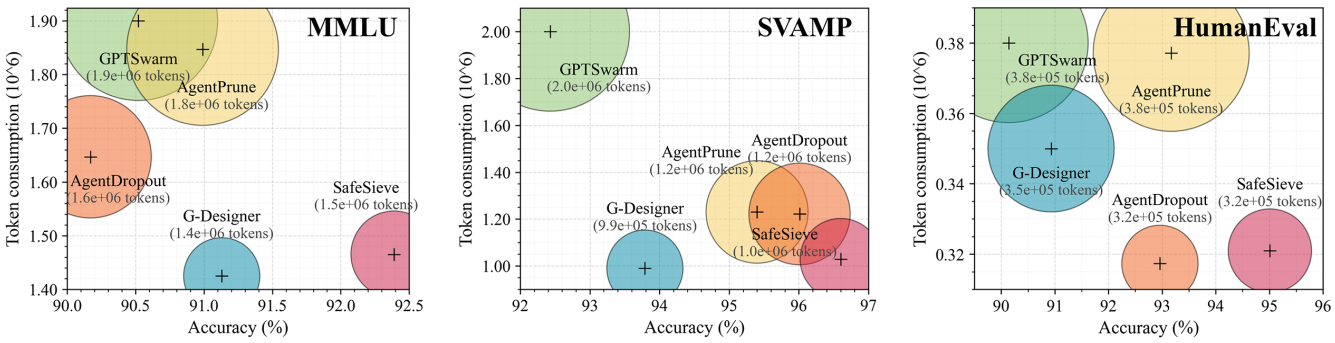

Figure 3: Accuracy–efficiency trade-off across benchmarks. Each graph represents MAS method's performance on one of three datasets: MMLU, SVAMP and HumanEval. It shows SafeSieve's superior task-specific pruning capabilities.

As demonstrated in Figure 3, SafeSieve consistently occupies the optimal position in the accuracy-token consumption space. On MMLU, SafeSieve achieves 92.39% accuracy with 1.47M tokens, outperforming GPTSwarm (90.52%, 1.90M tokens) and AgentPrune (90.99%, 1.85M tokens). On SVAMP, SafeSieve reaches 96.60% accuracy with only 1.03M tokens compared to AgentPrune's 95.40% at 1.23M tokens, achieving 16.3% token reduction with 1.2 point accuracy improvement. For HumanEval, SafeSieve attains 95.01% accuracy with 321K tokens versus AgentPrune's 93.17% at 377K tokens. This efficiency advantage stems from SafeSieve's dual-stage scoring mechanism that precisely eliminates redundant paths while preserving critical collaborative links.

Takeaway 5: Ablation experiments validate the effectiveness of three core components. As table 2 shows, historic feedback contributes significantly, with its removal causing accuracy to drop to 93.78% (-1.23 points) while saving 30.0% tokens. Heuristic initialization proves crucial, with its removal reducing accuracy to 94.41% (-0.60 points) while saving 24.2% tokens. The 0-extension clustering outperforms Top-k pruning, as replacing it with Top-k reduces accuracy to 93.13% (-1.88 points), demonstrating the superiority of structure-aware clustering. The combination of all three components achieves optimal performance with 95.01% accuracy and 27.8% token savings, realizing the best balance between efficiency and performance.

Robustness Analysis

Takeaway 1: Task characteristics determine vulnerability patterns to malicious agents. Knowledge-intensive MMLU proves most vulnerable, with SafeSieve accuracy dropping from 92.39% to 89.50% (-2.89 points), yet still outperforming AgentPrune's 7.32-point decline (90.99%→83.67%) as shown in Figure 4. SVAMP mathematical reasoning shows minimal degradation, with SafeSieve declining only 0.56 points (96.60%→96.04%) while AgentPrune drops 4.66 points and AgentDropout drops 3.76 points. HumanEval programming tasks demonstrate moderate protection, with SafeSieve declining 1.84 points (95.01%→93.17%) compared to AgentPrune's 5.41-point drop and AgentDropout's 2.56-point drop.

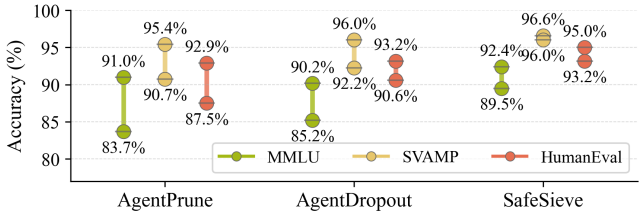

Figure 4: Accuracy drop of AgentPrune, AgentDropout, and SafeSieve when injecting low-quality agents into MMLU, SVAMP, and HumanEval tasks.

Method	MMLU	SVAMP	H.E.	Avg.	Rate
AgentPrune	↓4.99	↓3.80	↓4.97	↓4.59	↓5.14%
AgentDropout	↓1.67	↓2.81	↓2.16	↓2.21	↓2.40%
SafeSieve	**↓1.19**	**↓1.60**	**↓0.91**	**↓1.23**	**↓1.33%**

Table 3: Accuracy drop (–) under malicious agent intervention. SafeSieve shows minimal average drop.

Takeaway 2: SafeSieve's triple defense mechanism ensures minimal degradation and superior robustness. Overall performance remains optimal with average drops of 1.23% for SafeSieve, 2.21% for AgentDropout, and 4.59% for AgentPrune across three tasks, achieving a relative degradation rate of 1.33% as detailed in Table 3. Preventive defense assigns low weights (0.1-0.3) to suspicious agents compared to normal agents (0.7-0.9) through LLM semantic scoring. Responsive defense typically identifies malicious agents within 30 batches, automatically triggering pruning when cumulative scores fall below thresholds. Structural defense maintains network connectivity through 0-extension clustering, with accuracy fluctuations remaining under 3% before and after pruning in MMLU experiments, avoiding *information island* formation.

Heterogeneous Agent Collaboration

Experimental Setup Note. DeepSeek-V3 serves as evaluation expert, chief commander, and answer extractor, while other subtasks are allocated to models including Qwen-72B,

Method	Model	MMLU			SVAMP			HumanEval			Overall		
		Toks	Cost	Δ	Toks	Cost	Δ	Toks	Cost	Δ	Toks	Cost	Δ
Agent Prune	DeepSeek-V3	491K	¢23.46	–	287K	¢13.73	–	397K	¢18.94	–	1,175K	¢56.13	–
	GPT-4o-mini	168K	¢4.41	–	84K	¢2.21	–	86K	¢2.26	–	338K	¢8.87	–
	Llama-8B	131K	¢1.31	–	44K	¢0.44	–	141K	¢1.41	–	316K	¢3.15	–
	Qwen2.5-72B	171K	¢6.19	–	43K	¢1.56	–	54K	¢1.97	–	268K	¢9.72	–
	Kimi-K2	229K	¢16.88	–	139K	¢10.26	–	130K	¢9.57	–	498K	¢36.71	–
	Total	**1,190K**	**¢52.25**	–	**597K**	**¢28.19**	–	**808K**	**¢34.15**	–	**2,595K**	**¢114.59**	–
Agent Dropout	DeepSeek-V3	483K	¢23.04	-1.8%	233K	¢11.12	-19.0%	371K	¢17.72	-6.4%	1,087K	¢51.88	-7.6%
	GPT-4o-mini	164K	¢4.29	-2.6%	85K	¢2.23	+1.1%	101K	¢2.65	+17.4%	350K	¢9.17	+3.4%
	Llama-8B	122K	¢1.22	-6.4%	56K	¢0.56	+28.9%	108K	¢1.08	-23.5%	286K	¢2.86	-9.2%
	Qwen2.5-72B	203K	¢7.36	+18.9%	66K	¢2.39	+53.1%	99K	¢3.58	+81.5%	368K	¢13.33	+37.1%
	Kimi-K2	201K	¢14.85	-12.0%	52K	¢3.81	-62.9%	76K	¢5.63	-41.2%	329K	¢24.28	-33.9%
	Total	**1,173K**	**¢50.77**	**-2.8%**	**492K**	**¢20.11**	**-28.6%**	**755K**	**¢30.65**	**-10.2%**	**2,420K**	**¢101.53**	**-11.4%**
Safe Sieve	DeepSeek-V3	489K	¢23.35	-0.5%	215K	¢10.25	-25.3%	291K	¢13.89	-26.7%	995K	¢47.49	-15.4%
	GPT-4o-mini	133K	¢3.49	-20.9%	69K	¢1.82	-17.3%	158K	¢4.16	+84.0%	360K	¢9.47	+6.7%
	Llama-8B	230K	¢2.3	+75.6%	70K	¢0.70	+60.7%	88K	¢0.88	-37.3%	388K	¢3.88	+23.1%
	Qwen2.5-72B	178K	¢6.46	+4.4%	51K	¢1.83	+17.2%	120K	¢4.36	+121.4%	349K	¢12.65	+30.1%
	Kimi-K2	203K	¢14.97	-11.4%	65K	¢4.79	-53.3%	82K	¢6.06	-36.7%	350K	¢25.82	-29.7%
	Total	**1,233K**	**¢50.56**	**-3.2%**	**470K**	**¢19.40**	**-31.2%**	**740K**	**¢29.35**	**-14.0%**	**2,442K**	**¢99.31**	**-13.3%**

Table 4: Comparison of token usage and cost across heterogeneous LLM models under three pruning paradigms. Δ indicates relative cost difference w.r.t. AgentPrune baseline.

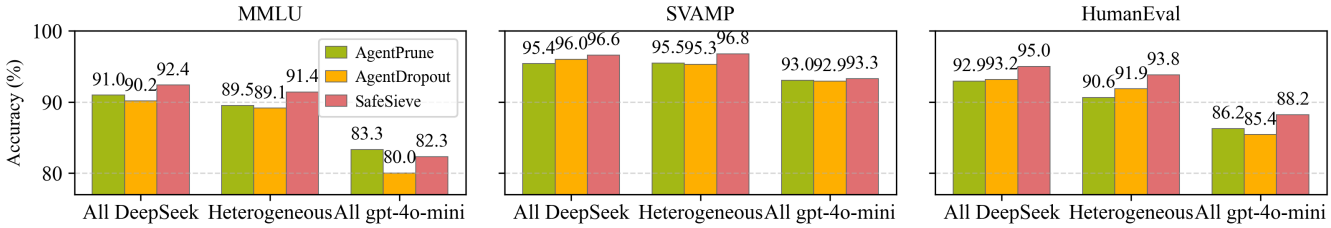

Figure 5: Performance and cost in heterogeneous settings. We compare AgentPrune, AgentDropout, and SafeSieve.

Kimi-K2, GPT-4o-mini, and LLaMA-8B.

Takeaway 1: Large model *commander* effect significantly reduces system costs. Total costs decrease by 13.3% from AgentPrune's ¢115 to SafeSieve's ¢99.3 as demonstrated in Table 4. Token allocation becomes intelligent, with DeepSeek-V3 consuming 995K tokens (40.7% of total) for core reasoning and final answer extraction while small models handle 59.3% of computational load. Cost optimization is significant, with DeepSeek-V3 costs reducing 15.4% (¢56.1→¢47.5) and Kimi-K2 reducing 29.7% (¢36.7→¢25.8). Small model usage increases but total costs decrease, with LLaMA-8B tokens increasing 23.1% but costs rising only ¢0.01, and GPT-4o-mini tokens increasing 6.7%. The *1+4* collaboration mode achieves optimal cost-effectiveness ratio on SVAMP as shown in Figure 5.

Takeaway 2: Task-dependent heterogeneous effects exhibit *barrel principle* characteristics. SVAMP mathematical reasoning shows heterogeneous advantages with 96.77% accuracy, slightly exceeding homogeneous configuration's 96.60% (+0.17 points) while reducing costs by 31.2% (¢28.2 → ¢19.4). MMLU knowledge tasks are limited by weaker models, with heterogeneous accuracy at 91.42% falling below homogeneous 92.39% (-0.97 points), validating the "barrel effect" in knowledge-intensive tasks. HumanEval programming tasks maintain competitiveness with 93.78% accuracy, declining only 1.23 points (vs. homogeneous 95.01%) while reducing costs by 14.0%.

Conclusion

We propose **SafeSieve**, a principled pruning framework for multi-agent collaboration that unifies semantic initialization with experience-guided refinement. It provides GPU-free sparsing strategy. Experiments across six benchmarks, including reasoning and coding tasks, show SafeSieve outperforms baselines with up to 6.58% accuracy gains and 30% token reduction. Furthermore, it demonstrates robustness against agent injection and excels in heterogeneous settings varying in model scale, validating the efficacy of structure-aware pruning for efficient LLM cooperation.

References

AI, M. 2024. Kimi K2: Open-source Instruction-tuned Model. Accessed: 2025-12-04.

AI@Meta. 2024. Llama 3 Model Card. Accessed: 2025-12-04.

Anil, C.; et al. 2024. Many-shot Jailbreaking: Exploring the Attack Surface of Long Contexts. *Anthropic Research*. Accessed: 2025-12-04.

Birant, D.; and Kut, A. 2007. ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering*, 60(1): 208–221.

Calinescu, G.; Karloff, H. J.; and Rabani, Y. 2003. An approximation algorithm for the 0-extension problem. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*.

Chang, Y.; Wang, X.; Wang, J.; Wu, Y.; Yang, L.; Zhu, K.; Chen, H.; Yi, X.; Wang, C.; Wang, Y.; et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3): 1–45.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; de Oliveira Pinto, H. P.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; Ray, A.; Puri, R.; Krueger, G.; Petrov, M.; Khlaaf, H.; Sastry, G.; Mishkin, P.; Chan, B.; Gray, S.; Ryder, N.; Pavlov, M.; Power, A.; Kaiser, L.; Bavarian, M.; Winter, C.; Tillet, P.; Such, F. P.; Cummings, D.; Plappert, M.; Chantzis, F.; Barnes, E.; Herbert-Voss, A.; Guss, W. H.; Nichol, A.; Paino, A.; Tezak, N.; Tang, J.; Babuschkin, I.; Balaji, S.; Jain, S.; Saunders, W.; Hesse, C.; Carr, A. N.; Leike, J.; Achiam, J.; Misra, V.; Morikawa, E.; Radford, A.; Knight, M.; Brundage, M.; Murati, M.; Mayer, K.; Welinder, P.; McGrew, B.; Amodei, D.; McCandlish, S.; Sutskever, I.; and Zaremba, W. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374.

Chen, Q.; Liu, W.; Liu, H.; Chen, N.; Dang, Y.; Li, J.; Yang, C.; Chen, W.; Su, Y.; Cong, X.; Xu, J.; Li, D.; Liu, Z.; and Sun, M. 2023. ChatDev: Communicative Agents for Software Development. *arXiv preprint arXiv:2307.07924*. Accessed: 2025-12-04.

Chen, S. F.; and Gopalakrishnan, P. S. 1998. Clustering via the bayesian information criterion with applications in speech recognition. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 2, 645–648. IEEE.

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*.

Deng, X.; Zhou, L.; Dong, D.; and Wei, J. 2025. Semantic Information Extraction and Multi-Agent Communication Optimization Based on Generative Pre-Trained Transformer. *IEEE Transactions on Cognitive Communications and Networking*. Introduces GPT-based semantic info extraction to optimize multi-agent communication.

Dong, H.; Chen, B.; and Chi, Y. 2024. Prompt-prompted Adaptive Structured Pruning for Efficient LLM Generation. In *Proceedings of the 1st Conference on Language Modeling (COLM)*. OpenReview preprint; training-free structured pruning via "flocking" in feedforward blocks.

Englert, M.; Gupta, A.; Krauthgamer, R.; Räcke, H.; Talgam-Cohen, I.; and Talwar, K. 2014. Vertex sparsifiers: New results from old techniques. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 152–166. Springer.

Fakcharoenphol, J.; Harrelson, C.; Rao, S.; and Talwar, K. 2003. An Improved Approximation Algorithm for the 0-Extension Problem. In *Proceedings of the 14th Annual ACM–SIAM Symposium on Discrete Algorithms (SODA)*, 257–265. New Orleans, LA, USA: ACM / SIAM. DOI available via dblp.

Guo, X.; Huang, K.; Liu, J.; and et al. 2024. Embodied LLM Agents Learn to Cooperate in Organized Teams. *arXiv preprint arXiv:2403.12482*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2021. Measuring Massive Multitask Language Understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Hong, S.; Zhuge, M.; Chen, J.; Zheng, X.; Cheng, Y.; Zhang, C.; Wang, J.; Wang, Z.; Yau, S. K. S.; Lin, Z.; et al. 2023. Metagpt: Meta programming for a multi-agent collaborative framework. *arXiv preprint arXiv:2308.00352*.

Hu, Y.; Cai, Y.; Du, Y.; Zhu, X.; Liu, X.; Yu, Z.; Hou, Y.; Tang, S.; and Chen, S. 2024. Self-Evolving Multi-Agent Collaboration Networks for Software Development. *arXiv preprint arXiv:2410.16946*. V1 posted Oct 22 2024.

Li, B.; Zhao, Z.; Lee, D.; and Wang, G. 2025. Adaptive Graph Pruning for Multi-Agent Communication. *arXiv preprint arXiv:2506.02951*. V1 posted Jun 3 2025; task-adaptive hard/soft pruning of agent communication topology.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*.

Ling, W.; Yogatama, D.; Dyer, C.; and Blunsom, P. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*.

Liu, N.; Lin, K.; Hewitt, J.; et al. 2024a. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*.

Liu, Y.; et al. 2024b. DeepSeek-V3: Scaling Open Models to 670B. *arXiv preprint arXiv:2406.09680*. Accessed: 2025-12-04.

Liu, Z.; Zhang, Y.; Li, P.; Liu, Y.; and Yang, D. 2024c. A Dynamic LLM-Powered Agent Network for Task-Oriented Agent Collaboration. *arXiv preprint arXiv:2310.02170*. Task-oriented collaboration framework.

Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP Models really able to Solve Simple Math Word Problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2080–2094. Online: Association for Computational Linguistics.

Schaeffer, S. E. 2007. Graph clustering. *Computer science review*, 1(1): 27–64.

Team, Q. 2024. Qwen2.5: A Party of Foundation Models. Accessed: 2025-12-04.

Wang, K.; Zhang, G.; Zhou, Z.; Wu, J.; Yu, M.; Zhao, S.; Yin, C.; Fu, J.; Yan, Y.; Luo, H.; et al. 2025a. A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment. *arXiv preprint arXiv:2504.15585*.

Wang, S.; Tan, Z.; Chen, Z.; Zhou, S.; Chen, T.; and Li, J. 2025b. AnyMAC: Cascading Flexible Multi-Agent Collaboration via Next-Agent Prediction. *arXiv preprint arXiv:2506.17784*. Submitted Jun 2025.

Wang, Z.; Mao, S.; Wu, W.; Ge, T.; Wei, F.; and Ji, H. 2024. Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, 7103–7126.

Wang, Z.; Wang, Y.; et al. 2025. AgentDropout: Dynamic Agent Elimination for Token-Efficient and High-Performance LLM-Based Multi-Agent Collaboration. *arXiv preprint arXiv:2503.18891*. Accessed: 2025-12-04.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; and Zhou, D. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903.

Wu, Q.; Bansal, G.; Zhang, J.; et al. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv preprint arXiv:2308.08155*.

Xue, J.; Xing, L.; Wang, Y.; et al. 2024. A comprehensive survey of fast graph clustering. *Vicinagearth*, 1: 7.

Zhang, G.; Yue, Y.; et al. 2024a. CUT THE CRAP: An Economical Communication Pipeline for LLM-Based Multi-Agent Systems. *arXiv preprint arXiv:2410.02506*. Accessed: 2025-12-04.

Zhang, G.; Yue, Y.; et al. 2024b. G-Designer: Architecting Multi-Agent Communication Topologies via Graph Neural Networks. *arXiv preprint arXiv:2410.11782*.

Zhang, K.; Yang, Z.; and Başar, T. 2022. Multi-Agent Deep Reinforcement Learning: A Survey. *Artificial Intelligence Review*, 55(3): 895–943. Comprehensive survey on current developments in MADRL.

Zhang, L.; Chen, Y.; and Kumar, R. 2025. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. *arXiv preprint arXiv:2502.12345*. Comprehensive survey of LLM-based multi-agent collaboration frameworks.

Zhang, Y.; Sun, R.; Chen, Y.; Pfister, T.; Zhang, R.; and Arik, S. 2024. Chain of Agents: Large Language Models Collaborating on Long-Context Tasks. In *Proceedings of NeurIPS 2024*. Demonstrates up to 10

Zhuge, M.; Wang, W.; Kirsch, L.; Faccio, F.; Khizbullin, D.; and Schmidhuber, J. 2024. GPTSwarm: Language Agents as Optimizable Graphs. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. To appear.