CONCAT: Consensus- and Confidence-Driven Ad Hoc Teaming for Efficient LLM-Based Multi-Agent Systems

Ziyang Ma[1], Dingyi Zhang[1], Sichu Liang[1], Jiajia Chu[2], Pengfei Xia[2], Hui Zang[2]*, Deyu Zhou[1]*
[1]Southeast University [2]Huawei Technologies Ltd

Abstract

Although large language model (LLM) based multi-agent systems (MAS) show their capability to solve complex tasks and achieve higher performance over single agent systems, they lead to huge computational overheads because of heavy communication between agents. Previous research has made efforts to train a sparse multi-agent graph or fine-tune a planner to orchestrate the workflow better. However, such extra training processes introduce computational costs and limit MAS to specific domains, therefore compromising their generalizability. In this paper, we propose CONCAT, a training-free multi-agent collaboration framework based on CONsensus and Confidence-driven Ad hoc Teaming to efficiently organize agent interactions. Specifically, agents are clustered based on their initial answers, and leaders of each cluster are selected based on the agents' confidence. Then, a heuristic function based on the Theory of Mind is designed to predict the collaboration benefits between every two leaders according to their answers and confidence. Finally, an ad hoc multi-agent network is organized after evicting a percentage of communications based on the predicted benefits. Experiments across three LLMs and three benchmarks show that CONCAT achieves up to **2.02**× higher efficiency (accuracy/latency ratio) than LLM-Debate and outperforms training-aware methods such as AgentDropout, while reducing average latency by **50.1%** on Qwen2.5-14B-Instruct, without any task-specific training.[1]

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across diverse domains, from mathematical reasoning and code generation to complex problem-solving tasks (Jaech et al., 2024; DeepSeek-AI et al., 2025). Recent advances have extended these capabilities through LLM-based agents that can autonomously interact with external environments and utilize tools (Wang et al., 2025b; Li et al., 2025). These agent systems have shown that equipping LLMs with the ability to perceive, reason, and act substantially enhances their practical utility. However, individual agents often face inherent limitations in handling complex tasks that require diverse expertise, multiple perspectives, or iterative refinement.

To overcome these limitations, Multi-Agent Systems (MAS) have emerged as a promising paradigm where multiple LLM-based agents collaborate to solve challenging problems (Guo et al., 2024). Frameworks such as MetaGPT (Hong et al., 2023), Magnetic-One (Fourney et al., 2024), and AgentOrchestra (Zhang et al., 2025b) employ orchestrator agents to decompose tasks and coordinate specialized agents, while GPTSwarm (Zhuge et al., 2024) formalizes MAS as temporal-spatial directed acyclic graphs with distributed agent nodes. Specialized agent frameworks such as ChatDev (Qian et al., 2024) demonstrate that role-playing agents with structured communication protocols can tackle complex software development tasks. These systems leverage inter-agent communication to make agents share responses, challenge each other's reasoning (Du et al., 2024), and iteratively refine their own solutions (Shinn et al., 2023) to converge to high-quality answers.

Despite these successes, current MAS architectures face critical efficiency challenges stemming from communication overhead and redundancy. One prominent research direction focuses on workflow optimization through pruning strategies. Methods like AgentPrune (Zhang et al., 2024) and AgentDropout (Wang et al., 2025a) employ task-specific training to identify and eliminate redundant communication edges or underperforming agents from the collaboration graph. Another line of work introduces capable orchestrators that train specialized LLMs to dynamically determine op-

*Corresponding author
[1]The code will be available upon publication.

timal topologies and agent configurations (Dang et al., 2025; Ye et al., 2025b). While these approaches achieve performance improvements, they inherently rely on task-specific training data and incur high computational costs. Moreover, recent studies on cross-context communication optimization (Ye et al., 2025a) and parallel agent execution (Zhang et al., 2025a) reveal that, for both distributed and orchestrator-based MAS, much redundant computation exists in agent interactions such as repetitive key-value prefilling and non-optimal workflow execution, which contribute minimal value while consuming considerable resources.

The fundamental challenge lies in determining which agent interactions genuinely contribute to improved outcomes without requiring expensive task-aware training. Individual LLM agents also face inherent limitations from lacking external feedback and self-reflection mechanisms (Zhao et al., 2024), making principled communication topology design essential. Graph-based optimization methods demand downstream data for training to learn domain-aware pruning policies, while orchestrator-based approaches necessitate training meta-agents with sufficient capacity to reason about complex collaboration dynamics. Both paradigms struggle with controllability and generalization: trained policies may overfit to specific task distributions, and the training process itself introduces substantial latency and cost. Furthermore, recent work has shown that LLMs exhibit conformity biases in multi-agent settings (Weng et al., 2024; Zhu et al., 2025), where agents tend to change their initially correct answers when exposed to multiple incorrect peer responses. Such conformity among LLMs partly reflects the communication redundancy issue in multi-agent systems and indicates that ineffective interactions are probably foreseeable. Considering all the above, we raise a critical question: *Can we identify and eliminate redundant or ineffective communications in multi-agent systems without task-specific training*?

To address this challenge, we propose a training-free framework, **CON**sensus- and **C**onfidence-driven **A**d hoc **T**eaming (**CONCAT**), which leverages intrinsic agent information to achieve efficient collaboration. Our approach is motivated by two observations from empirical analysis: (i) More than 60% agent communications yield neutral or negative impacts, where agents frequently maintain their answers regardless of incoming information, or correct answers degrade to incorrect ones after collaboration. (ii) Collaboration effectiveness can be predicted using simple heuristics based on answer similarity and confidence scores of agents. Building on these insights, CONCAT realizes ad hoc networking via two modules: (1) **Consensus clustering and leader selection**: answer-based clustering groups agents by their responses; within each cluster, the highest-confidence agent is designated as a leader, and only leaders participate in subsequent exchanges; (2) **Benefit-driven edge pruning**: a Theory-of-Mind-inspired predictor estimates the utility of communication for each pair of leaders, and the lowest-scoring links are pruned. By constructing sparse topologies through leader selection and benefit-driven pruning, CONCAT maintains competitive accuracy while reducing latency and token usage relative to existing methods.

Our contributions are listed as follows:

- Our empirical analysis on multi-agent collaboration dynamics reveals the prevalence of ineffective communications in existing MAS frameworks and the predictability of collaboration benefits using intrinsic agent signals.
- We propose CONCAT, a training-free MAS framework that combines answer-based agent clustering with Theory-of-Mind-based collaboration benefit prediction to construct efficient communication topologies dynamically.
- Experiments across mathematics, general reasoning, and code generation benchmarks demonstrate that CONCAT achieves higher efficiency than both training-aware and -free baselines, without task-specific training.

2 Background

2.1 Task Definition

Following Zhuge et al. (2024), Zhang et al. (2024), and Wang et al. (2025a), we formalize an LLM-based Multi-Agent System (MAS) as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ represents the set of N agents, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the communication edges between agents. Each directed edge $(v_i, v_j) \in \mathcal{E}$ indicates that agent v_j can reference and incorporate the reasoning output of agent v_i during collaboration.

For each agent $v_i \in \mathcal{V}$, we define its state at round t as:

$$s_i^{(t)} = \{a_i^{(t)}, c_i^{(t)}\}, \tag{1}$$

where $a_i^{(t)}$ represents the agent's answer to the given task, and $c_i^{(t)} \in [0, 1]$ denotes its confidence. During the collaboration process, agent v_i updates its state by aggregating information from its incoming neighbors $\mathcal{N}_{\text{in}}(v_i) = \{v_j : (v_j, v_i) \in \mathcal{E}\}$:

$$s_i^{(t+1)} = f_\theta \left(s_i^{(t)}, \{ s_j^{(t)} : v_j \in \mathcal{N}_{\text{in}}(v_i) \} \right) \quad (2)$$

where f_θ is the LLM-based reasoning function parameterized by θ.

Communication Redundancy. The key challenge in designing efficient MAS lies in determining an optimal sparse graph structure $\mathcal{G}^*$ that maintains task performance while minimizing computational overhead. A fully-connected graph ($|\mathcal{E}| = N(N-1)$) enables maximal information exchange but incurs quadratic communication complexity. Our goal is to identify a sparse topology $\mathcal{G}^*$ where $|\mathcal{E}^*| \ll N(N-1)$, such that:

$$\mathcal{G}^* = \arg\max_{\mathcal{G}} \frac{\text{Performance}(\mathcal{G})}{\text{Latency}(\mathcal{G})} \quad (3)$$

In the following subsections, we present empirical observations that motivate our approach to constructing such sparse structures.

2.2 Observation 1: Ineffective Collaboration

To understand collaboration dynamics in MAS, we conduct statistical analysis using both AgentDropout (Wang et al., 2025a) and LLM-Debate (Du et al., 2024). Our analysis reveals a critical insight: referencing other agents frequently degrades performance rather than improving it.

Performance Drop with Collaborations As shown in Figure 1, agents with references are more frequently misled rather than correctly guided. Notably, when agents receive no references in Round 0 and get two references in Round 1, they change their answers from correct to wrong over 15% more than they correct their answers from wrong ones.

Fine-Grained Analysis via LLM-Debate To further investigate this phenomenon, we employ LLM-Debate (Du et al., 2024) to track answer correctness transitions across agent configurations. As shown in Figure 2, non-beneficial outcomes dominate across all agent configurations: Correct→Correct, Wrong→Wrong, and Correct→Wrong together account for over 83% of all agent pairs, while Wrong→Correct never exceeds 17%. Therefore, we conclude this insight as:

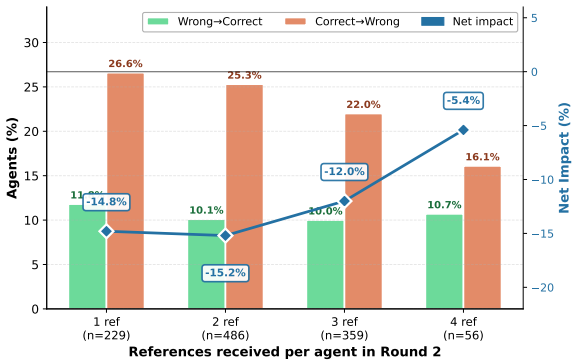

Figure 1: Statistics of two types of answer changes (correct to wrong and wrong to correct) with and without collaboration for agents. The agents are categorized based on the number of other agents they referenced in Round 2 (ranging from one to four references). The MAS is trained on GSM8K using AgentDropout (Wang et al., 2025a), with five agents and two rounds.

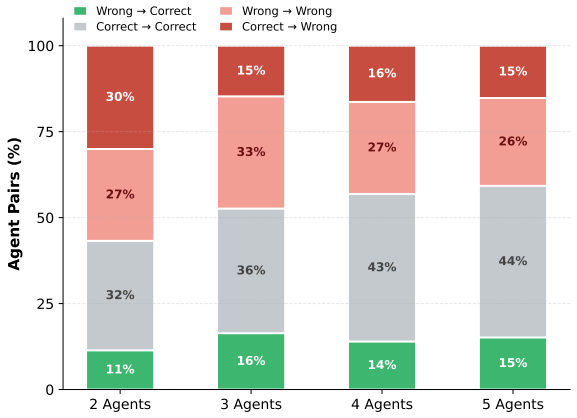

Figure 2: Collaboration outcome distribution on GSM8K using LLM-Debate (Du et al., 2024) with two, three, four, and five agents. Each agent pair is categorized by its answer transition across rounds: Wrong→Correct, Correct→Correct, Wrong→Wrong, or Correct→Wrong.

Observation 1: Referencing other agents has negative or neutral impacts more frequently than correcting errors.

2.3 Observation 2: Predictability of Collaboration Effectiveness

Building on the insight that many collaborations are ineffective, we investigate whether collaboration outcomes can be predicted *a priori* from observable agent signals, enabling proactive topology optimization. Specifically, given a focal agent v_i and a source agent v_j, we ask: does the collaboration benefit $b_{j \to i}$ correlate with observable quantities

such as answer similarity s_{ij} and confidence scores c_i, c_j? Here $b_{j\to i}$ is defined as the expected change in v_i's answer correctness after referencing v_j.

Predictability via Intrinsic Signals. Each agent pair is labeled as *helpful* (Wrong→Correct) or *not helpful* (all other transitions) based solely on the observed answer correctness before and after collaboration (Table 4 in Appendix D). We then define a simple, training-free *dissent strength* score: $d_{j\to k} = \bar{c}_j \cdot (1 - \text{agree}_{jk})$, where $\bar{c}_j$ is the mean confidence of source agent j across all focal agents in the same configuration and agree_{jk} indicates whether j and k share the same answer. Intuitively, a confident source who disagrees with the focal agent is more likely to provide a corrective signal. As shown in Figure 3, this single feature achieves ROC-AUC of 0.74–0.86 across all agent configurations and both benchmarks, substantially above the random baseline of 0.50. This demonstrates that collaboration benefit is predictable from observable signals alone, providing principled motivation for the benefit-based edge pruning in CONCAT.

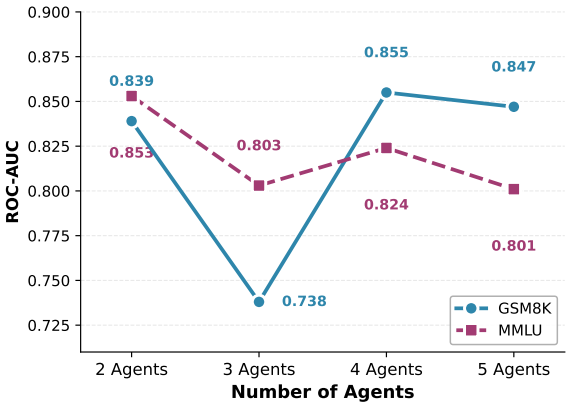

Figure 3: ROC-AUC of dissent strength ($d_{j\to k} = \bar{c}_j \cdot (1 - \text{agree}_{jk})$) for predicting helpful collaboration (Wrong→Correct) across 2–5 agent configurations on GSM8K and MMLU. All results computed on LLM-Debate (Du et al., 2024) based on Llama-3-8B-Instruct.

Observation 2: Collaboration effectiveness is predictable from answer similarity and agent confidence scores, enabling training-free and principled edge pruning.

3 Method

3.1 Overview

Building upon the two empirical observations, we propose CONCAT, Consensus- and Confidence-Driven Ad Hoc Teaming, to address the computational inefficiency and collaboration quality issues in multi-agent systems. As depicted in Figure 4, CONCAT operates in three phases: (1) **Initialization**: each agent independently generates an answer; (2) **Ad Hoc Teaming**: agents are grouped via **consensus clustering and leader selection**, reducing active participants from N to K cluster leaders, and **benefit-driven edge pruning** reconstructs the sparse communication graph at each round; (3) **Final Answer Aggregation**: an LLM synthesizer aggregates the answers of all agents to produce the final answer. The pseudo code of CONCAT is provided in Algorithm 1 in Appendix B.

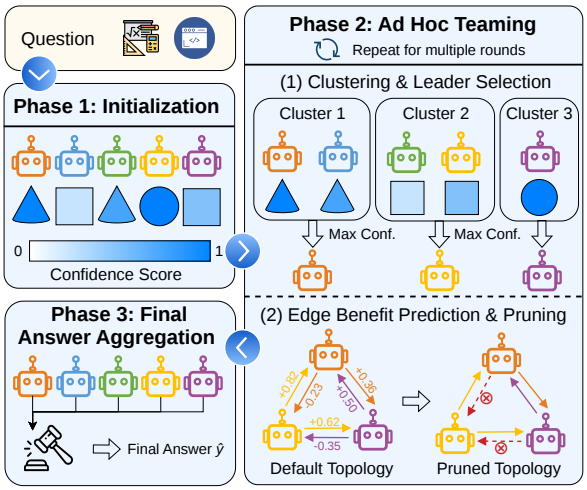

Figure 4: Overview of CONCAT. CONCAT operates through three phases: (1) *Initialization*, where each agent independently generates an answer; (2) *Ad Hoc Teaming*, where agents are grouped by consensus clustering and leader selection, followed by benefit-driven edge pruning to construct a sparse communication topology, repeated for $(m-1)$ rounds; (3) *Final Answer Aggregation*, where an LLM synthesizer aggregates the answers of all the agents and output the final answer. Circles, squares, and triangles denote different answers.

3.2 Phase 1: Initialization

The framework initializes with each agent $v_i \in \mathcal{V}$ independently generating an initial response without observing peer outputs:

$$s_i^{(0)} = \{a_i^{(0)}, c_i^{(0)}\} = f_\theta(q, \emptyset), \tag{4}$$

where the confidence score $c_i^{(0)} \in [0, 1]$ is computed as the average token probability across all generated output tokens. This independence ensures diverse initial perspectives by preventing premature consensus and the groupthink phenomenon.

3.3 Phase 2: Ad Hoc Teaming

The core of CONCAT resides in its adaptive debate mechanism, which iteratively refines answers through selective leader interactions. Each round $t \in \{1, \ldots, m-1\}$ executes six coordinated steps to construct and utilize a sparse topology.

Clustering and Leader Selection At each round t, we first partition the agent population into consensus clusters based on answer similarity:

$$\mathcal{C}^{(t)} = \text{ClusterBySimilarity}(\{a_i^{(t-1)}\}_{i=1}^N) \quad (5)$$

The similarity metric is task-dependent: for problems with deterministic answers (e.g., multiple-choice, mathematical reasoning), we employ exact matching; for code generation tasks, we use Jaccard similarity over node types of abstract syntax trees. Specifically, we parse code snippets into abstract syntax trees, extract all node types from each tree, and compute the Jaccard coefficient as the ratio of intersection to union of node type sets.

From each cluster $C_j \in \mathcal{C}^{(t)}$, we select a single leader via confidence maximization in Equation 6, which results in the leader set in Equation 7.

$$l_j = \arg\max_{v_i \in C_j} c_i^{(t-1)} \quad (6)$$

$$\mathcal{L}^{(t)} = \{l_1, l_2, \ldots, l_K\} \subseteq \mathcal{V} \quad (7)$$

Collaboration Benefit Prediction and Pruning For each ordered pair of leaders $(l_j, l_k) \in \mathcal{L}^{(t-1)} \times \mathcal{L}^{(t)}$ with $j \neq k$, we predict the collaboration benefit $b_{j\to k}$, defined as the expected utility of l_k referencing l_j, using a Theory-of-Mind (ToM) based heuristic (Baker et al., 2017). We first classify l_j as a *supporter* (answer similarity $s_{jk} \geq \theta_{\text{sim}}$) or *challenger*, and compute its effective signal strength $\hat{c}_j = c_j \cdot s_{jk}$ (supporter) or $\hat{c}_j = c_j(1 - s_{jk})$ (challenger). The benefit for the Challenger case is derived from a Bayesian Expected Utility of Communication (EUC) framework linearized via Taylor expansion:

$$b_{j\to k} = \underbrace{4c_k(1-c_k)\cdot \hat{c}_j}_{\text{correction gain}} - \underbrace{\alpha \cdot \frac{1+2c_k}{2+2c_k+2\hat{c}_j}}_{\text{inertia discount}} + \underbrace{\alpha \cdot (1-c_k)}_{\text{epistemic openness}}, \quad (8)$$

where the correction gain $4c_k(1-c_k)\hat{c}_j$ is the first-order Taylor approximation of the EUC, the inertia discount models LLM anchoring via a Beta-Binomial posterior (Gelman et al., 2013), and the epistemic openness term follows Value of Information theory (Howard, 1966). For the Supporter case: $b_{j\to k} = \alpha(\hat{c}_j - c_k)$. The full theoretical derivation is in Appendix F.

With predicted benefits computed for all leader pairs, we construct a sparse communication topology through adaptive thresholding. The pruning threshold is computed in Equation 9, where $p \in [0, 1]$ controls edge retention rate and $\tau_{\min}$ enforces an absolute quality floor. Therefore, the edge set is defined accordingly in Equation 10.

$$\tau^{(t)} = \max\left(\text{Percentile}(\{b_{j\to k}\}, p \times 100), \tau_{\min}\right) \quad (9)$$

$$\mathcal{E}^{(t)} = \{(l_j, l_k) \mid b_{j\to k} \geq \tau^{(t)}\} \quad (10)$$

Leader Answer Refinement For each leader $l_k \in \mathcal{L}^{(t)}$, we construct a personalized debate context by collecting responses from beneficial peers:

$$\mathcal{R}_k^{(t)} = \{a_j^{(t-1)} \mid (l_j, l_k) \in \mathcal{E}^{(t)}\} \quad (11)$$

Each leader then refines its answer by invoking the language model with the original query and curated peer responses as depicted in Equation 12, while non-leader agents maintain their previous state without model invocation in Equation 13.

$$s_j^{(t)} = \{a_j^{(t)}, c_j^{(t)}\} = f_\theta(q, \mathcal{R}_j^{(t)}) \quad (12)$$

$$s_i^{(t)} = s_i^{(t-1)}, \quad \forall v_i \notin \mathcal{L}^{(t)} \quad (13)$$

3.4 Final Answer Aggregation

After $m-1$ debate rounds, we aggregate all agent responses $\{a_i^{(m-1)}\}_{i=1}^N$ to produce the final answer. LLM-based synthesizers are employed to generate the final answer by reasoning over all agent responses:

$$\hat{y} = f_\theta\left(P_{final} \oplus \{a_i^{(m-1)}\}_{i=1}^N\right), \quad (14)$$

where P_{final} is the aggregating prompt.

4 Experiment

This section will include comparative experiments and an ablation study to verify the effectiveness of CONCAT and its core modules. §4.2 answers the question: Is the proposed method CONCAT more efficient than other baselines of multi-agent systems? §4.3 investigates the importance of clustering and heuristic edge pruning in CONCAT to improve the efficiency of multi-agent collaboration.

4.1 Experiments Setup

Datasets We conduct experiments on three benchmarks covering diverse domains, which include GSM8k (Cobbe et al., 2021) for mathematical tasks, MMLU (Hendrycks et al., 2020) for general reasoning, and HumanEval (Chen et al., 2021) for code generation.

Baselines We compare our method, CONCAT, with the following baselines: (1) Chain-of-Thought (CoT) (Wei et al., 2022); (2) CoT with self-consistency (SC-CoT) (Wang et al., 2022); (3) LLM-Debate (Du et al., 2024); (4) Vanilla MAS; (5) AgentDropout (Wang et al., 2025a). Among these baselines, (1)-(4) are training-free, and (5) requires task-specific training. Besides, (4) and (5) are applied with five topologies (i.e., Star, Chain, Random, Layered, and Fully Connected).

Backbone LLMs Three LLMs, Llama-3-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-14B-Instruct, and Qwen2.5-72B-Instruct (Yang et al., 2024) are used as backbone models of multi-agent systems, which have different model sizes.

Implementation Details We use vLLM (Kwon et al., 2023) to deploy Llama-3-8B-Instruct on one NVIDIA RTX 3090 24G GPU, and Qwen2.5-14B-Instruct on four NVIDIA RTX 3090 24G GPUs with a tensor parallel size of 4. Meanwhile, we use vLLM-Ascend (Vllm-Project, 2025) to serve Qwen2.5-72B-Instruct on four Ascend 910B NPUs with a tensor parallel size of 4. For every LLM, we set temperature as 0.7 and top_p as 0.8. The single empirical parameter α in the benefit predictor is set to 0.2, selected via hyperparameter search (see §4.4). More implementation details are provided in Appendix E. To ensure reliability, all reported results are averaged over three repetitive runs.

4.2 Main Result

We compare CONCAT with five baselines on three LLMs and three datasets to validate our framework across varying domains and model sizes.

CONCAT achieves superior efficiency while maintaining competitive performance. As shown in Table 1 and Figure 5, CONCAT consistently outperforms baseline methods in terms of efficiency across different base models. Specifically, on Llama3-8B-Instruct, CONCAT achieves an efficiency score of 1.56, substantially higher than LLM-Debate (0.70), all Vanilla MAS variants (0.59-0.76), and all AgentDropout variants (0.67-0.92). The efficiency gain is even more pronounced on Qwen2.5-14B-Instruct, where CONCAT reaches an efficiency of 2.85, which is 2.02× higher than LLM-Debate (1.41), 1.94× higher than the best Vanilla MAS variant (1.47), and 1.58× higher than the best AgentDropout variant (1.80). Importantly, CONCAT maintains competitive average accuracy of 64.97% on Llama3-8B and 86.02% on Qwen2.5-14B, demonstrating its ability to balance accuracy and computational cost.

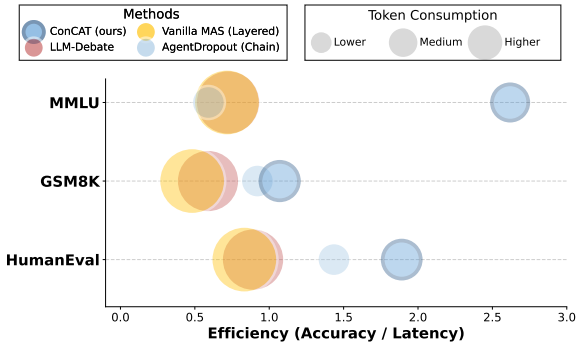

Figure 5: Efficiency comparison of multi-agent methods on Llama-3-8B-Instruct across three benchmarks. Bubble size indicates token consumption relative to other methods within each dataset.

CONCAT substantially reduces latency and token consumption. On Llama3-8B-Instruct, CONCAT reduces average latency by 55.5% vs. LLM-Debate and 40.4% vs. the best AgentDropout variant. On Qwen2.5-14B-Instruct, the latency reduction reaches 50.1% vs. LLM-Debate and 35.6% vs. the best AgentDropout variant. Token consumption follows the same trend: CONCAT averages 1.9M total tokens on Llama3-8B, representing 45.7% and 48.6% reductions compared to LLM-Debate (3.5M) and Vanilla MAS Layered (3.7M), respectively. Notably, CONCAT achieves slightly higher accuracy than Vanilla MAS Layered (64.97% vs. 64.12%) while reducing latency by 33.3%. Detailed per-benchmark token statistics are provided in Appendix G.

4.3 Ablation Study

To validate the effectiveness of each component in CONCAT, we conduct ablation experiments by removing or randomizing the answer-based clustering and edge pruning modules individually.

Impact of Edge Pruning. As shown in Table 2, removing clustering increases average latency by 54.9% while reducing accuracy by 1.50%, confirming that consensus-based leader selection is the pri-

Method	Topology	TF	Accuracy↑				Latency↓				Eff.↑
			MMLU	GSM8K	HumanEval	Avg.	MMLU	GSM8K	HumanEval	Avg.	
Backbone LLM: Llama3-8B-Instruct											
CoT	-	✓	48.15	64.64	63.91	58.90	5.69	9.20	3.28	6.05	-
SC-CoT	-	✓	$50.54_{\uparrow 2.40}$	$74.19_{\uparrow 9.56}$	$63.36_{\downarrow 0.55}$	$62.70_{\uparrow 3.80}$	17.66	40.77	16.89	25.11	-
LLM-Debate	Debate	✓	$56.43_{\uparrow 8.28}$	$78.80_{\uparrow 14.17}$	$60.88_{\downarrow 3.03}$	$65.37_{\uparrow 6.47}$	63.32	133.92	84.11	93.78	0.70
Vanilla MAS	Star	✓	$53.59_{\uparrow 5.45}$	$75.49_{\uparrow 10.86}$	$56.91_{\downarrow 7.00}$	$62.00_{\uparrow 3.10}$	65.90	147.11	103.28	105.43	0.59
	Chain	✓	$54.68_{\uparrow 6.54}$	$73.70_{\uparrow 9.06}$	$58.13_{\downarrow 5.79}$	$62.17_{\uparrow 3.27}$	56.95	122.11	65.29	81.45	0.76
	Random	✓	$51.63_{\uparrow 3.49}$	$75.57_{\uparrow 10.94}$	$56.20_{\downarrow 7.71}$	$61.14_{\uparrow 2.24}$	62.62	135.14	80.85	92.87	0.66
	Layered	✓	$59.26_{\uparrow 11.11}$	$76.35_{\uparrow 11.72}$	$56.75_{\downarrow 7.16}$	$64.12_{\uparrow 5.22}$	71.17	158.44	79.27	102.96	0.62
	FullConnected	✓	$57.30_{\uparrow 9.15}$	$75.60_{\uparrow 10.96}$	$53.99_{\downarrow 9.92}$	$62.30_{\uparrow 3.40}$	71.81	149.22	90.60	103.87	0.60
AgentDropout	Star	✗	$55.99_{\uparrow 7.84}$	$76.28_{\uparrow 11.64}$	$57.30_{\downarrow 6.61}$	$63.19_{\uparrow 4.29}$	48.67	107.69	56.97	71.11	0.89
	Chain	✗	$56.86_{\uparrow 8.71}$	$78.57_{\uparrow 13.93}$	$61.93_{\downarrow 1.98}$	$65.79_{\uparrow 6.89}$	39.62	85.37	104.80	76.60	0.86
	Random	✗	$60.35_{\uparrow 12.20}$	$76.32_{\uparrow 11.68}$	$57.02_{\downarrow 6.89}$	$64.56_{\uparrow 5.66}$	50.33	98.48	61.14	69.98	0.92
	Layered	✗	$54.90_{\uparrow 6.75}$	$76.25_{\uparrow 11.61}$	$52.47_{\downarrow 11.44}$	$61.21_{\uparrow 2.31}$	48.52	100.57	113.07	87.39	0.70
	FullConnected	✗	$54.25_{\uparrow 6.10}$	$75.09_{\uparrow 10.45}$	$56.75_{\downarrow 7.16}$	$62.03_{\uparrow 3.13}$	52.92	118.11	105.81	92.28	0.67
CONCAT (ours)	Hybrid	✓	$55.56_{\uparrow 7.41}$	$77.08_{\uparrow 12.44}$	$62.26_{\downarrow 1.65}$	$64.97_{\uparrow 6.07}$	**29.39**	**72.02**	**23.77**	**41.73**	**1.56**
Backbone LLM: Qwen2.5-14B-Instruct											
CoT	-	✓	73.20	67.97	83.20	74.79	6.07	13.78	3.06	7.64	-
SC-CoT	-	✓	$73.86_{\uparrow 0.65}$	$93.05_{\uparrow 25.08}$	$88.43_{\uparrow 5.23}$	$85.11_{\uparrow 10.32}$	20.78	54.00	16.49	30.42	-
LLM-Debate	Debate	✓	$75.60_{\uparrow 2.40}$	$94.24_{\uparrow 26.28}$	$86.18_{\uparrow 2.98}$	$85.34_{\uparrow 10.55}$	50.00	112.34	18.96	60.43	1.41
Vanilla MAS	Star	✓	$76.25_{\uparrow 3.05}$	$94.06_{\uparrow 26.09}$	$85.12_{\uparrow 1.93}$	$85.15_{\uparrow 10.36}$	51.32	131.60	22.06	68.33	1.25
	Chain	✓	$75.16_{\uparrow 1.96}$	$93.57_{\uparrow 25.60}$	$85.95_{\uparrow 2.75}$	$84.89_{\uparrow 10.10}$	41.82	110.68	20.17	57.56	1.47
	Random	✓	$76.03_{\uparrow 2.83}$	$93.98_{\uparrow 26.02}$	$84.57_{\uparrow 1.38}$	$84.86_{\uparrow 10.08}$	48.65	117.23	20.32	62.07	1.37
	Layered	✓	$73.42_{\uparrow 0.22}$	$94.14_{\uparrow 26.17}$	$86.50_{\uparrow 3.31}$	$84.69_{\uparrow 9.90}$	51.87	119.64	16.27	62.60	1.35
	FullConnected	✓	$74.73_{\uparrow 1.53}$	$93.98_{\uparrow 26.02}$	$84.57_{\uparrow 1.38}$	$84.43_{\uparrow 9.64}$	55.12	132.10	20.21	69.14	1.22
AgentDropout	Star	✗	$74.95_{\uparrow 1.74}$	$94.53_{\uparrow 26.56}$	$85.67_{\uparrow 2.48}$	$85.05_{\uparrow 10.26}$	37.05	88.62	18.44	48.04	1.77
	Chain	✗	$76.91_{\uparrow 3.70}$	$94.01_{\uparrow 26.04}$	$81.82_{\downarrow 1.38}$	$84.24_{\uparrow 9.46}$	34.17	82.01	24.45	46.88	1.80
	Random	✗	$74.51_{\uparrow 1.31}$	$94.14_{\uparrow 26.17}$	$81.44_{\downarrow 1.75}$	$83.36_{\uparrow 8.58}$	36.57	85.58	18.41	46.85	1.78
	Layered	✗	$74.29_{\uparrow 1.09}$	$94.24_{\uparrow 26.28}$	$85.67_{\uparrow 2.48}$	$84.74_{\uparrow 9.95}$	37.71	95.85	20.33	51.30	1.65
	FullConnected	✗	$75.82_{\uparrow 2.61}$	$93.98_{\uparrow 26.02}$	$84.85_{\uparrow 1.65}$	$84.88_{\uparrow 10.09}$	39.77	93.51	**12.78**	48.69	1.74
CONCAT (ours)	Hybrid	✓	$77.78_{\uparrow 4.57}$	$94.10_{\uparrow 26.13}$	$86.19_{\uparrow 2.99}$	$86.02_{\uparrow 11.23}$	**25.76**	**51.19**	13.55	**30.17**	**2.85**
Backbone LLM: Qwen2.5-72B-Instruct											
CoT	-	✓	77.56	93.85	84.30	85.24	23.85	37.26	10.99	24.03	-
SC-CoT	-	✓	$79.74_{\uparrow 2.18}$	$94.06_{\uparrow 0.21}$	$87.05_{\uparrow 2.75}$	$86.95_{\uparrow 1.71}$	84.06	131.48	58.82	91.45	-
LLM-Debate	Debate	✓	$79.52_{\uparrow 1.96}$	$93.93_{\uparrow 0.08}$	$85.95_{\uparrow 1.65}$	$86.47_{\uparrow 1.23}$	149.85	303.27	57.46	170.19	0.51
Vanilla MAS	Star	✓	$81.70_{\uparrow 4.14}$	$93.49_{\downarrow 0.36}$	$88.71_{\uparrow 4.41}$	$87.96_{\uparrow 2.73}$	235.17	341.80	59.79	212.25	0.41
	Chain	✓	$81.05_{\uparrow 3.49}$	$92.86_{\downarrow 0.99}$	$87.88_{\uparrow 3.58}$	$87.26_{\uparrow 2.03}$	217.53	299.55	60.95	192.68	0.45
	Random	✓	$81.48_{\uparrow 3.92}$	$93.44_{\downarrow 0.41}$	$90.08_{\uparrow 5.79}$	$88.33_{\uparrow 3.10}$	148.03	308.52	59.27	171.94	0.51
	Layered	✓	$80.39_{\uparrow 2.83}$	$93.78_{\downarrow 0.07}$	$88.71_{\uparrow 4.41}$	$87.62_{\uparrow 2.39}$	156.18	312.79	53.20	174.06	0.50
	FullConnected	✓	$79.30_{\uparrow 1.74}$	$93.28_{\downarrow 0.57}$	$88.15_{\uparrow 3.86}$	$86.91_{\uparrow 1.68}$	196.93	334.57	59.12	196.87	0.44
AgentDropout	Star	✗	$82.57_{\uparrow 5.01}$	$93.72_{\downarrow 0.13}$	$85.95_{\uparrow 1.65}$	$87.42_{\uparrow 2.18}$	180.62	232.63	52.73	155.33	0.56
	Chain	✗	$78.43_{\uparrow 0.87}$	$93.36_{\downarrow 0.49}$	$84.57_{\uparrow 0.27}$	$85.45_{\uparrow 0.21}$	176.02	228.89	71.63	158.85	0.54
	Random	✗	$80.83_{\uparrow 3.27}$	$92.76_{\downarrow 1.09}$	$87.05_{\uparrow 2.75}$	$86.88_{\uparrow 1.64}$	116.22	231.72	49.90	132.61	0.66
	Layered	✗	$81.48_{\uparrow 3.92}$	$93.26_{\downarrow 0.60}$	$86.50_{\uparrow 2.20}$	$87.08_{\uparrow 1.84}$	121.07	243.10	49.34	137.84	0.63
	FullConnected	✗	$81.92_{\uparrow 4.36}$	93.85	$88.98_{\uparrow 4.68}$	$88.25_{\uparrow 3.01}$	184.71	244.47	**38.26**	155.81	0.57
CONCAT (ours)	Hybrid	✓	$79.52_{\uparrow 1.96}$	$93.49_{\downarrow 0.36}$	84.30	$85.77_{\uparrow 0.53}$	**107.59**	**141.72**	46.31	**98.54**	**0.87**

Table 1: Performance comparison between CONCAT and other baselines. **Bold** and underline indicate the best and second-best performance for latency and efficiency metrics among MAS baselines (LLM-Debate, Vanilla MAS, and AgentDropout). TF stands for Training-Free, and Eff. denotes Efficiency (Avg. Accuracy/Avg. Latency).

mary driver of computational efficiency. Removing benefit-prediction edge pruning yields a smaller latency increase of 6.2% but causes a 0.76% accuracy drop, most notably on MMLU (−1.96%), indicating that edge pruning functions as a quality gate that filters harmful communications. The two modules are thus complementary: clustering reduces interaction scale while edge pruning preserves interaction quality.

Impact of Random Baselines. To further isolate the contribution of each algorithmic choice, we compare CONCAT against two random baseline variants on Qwen2.5-14B-Instruct. *w/ Rand. Edge Pruning* retains confidence-driven leader selection but replaces benefit-prediction pruning with random edge removal, achieving accuracy of 85.79% with average latency 26.95s (efficiency 3.18). *w/ Rand. Selection & Pruning* additionally replaces confidence-driven leader selection with random agent selection, yielding accuracy of 84.68% with latency 26.72s (efficiency 3.17). Both random variants achieve slightly lower latency than CONCAT, yet fall 0.23–1.34 points below CONCAT in average accuracy (86.02%), demonstrating that the ToM-based benefit predictor and confidence-driven leader selection provide meaningful accuracy gains over random alternatives. Notably, *w/*

Method	Accuracy↑				Latency↓				Efficiency↑
	MMLU	GSM8K	HumanEval	Avg.	MMLU	GSM8K	HumanEval	Avg.	
CONCAT	77.78	94.10	**86.19**	**86.02**	25.76	51.19	13.55	30.17	2.85
w/o Edge Pruning	75.82	**94.57**	85.40	85.26	25.99	52.41	17.72	32.04	2.66
w/o Clustering	74.51	94.22	84.85	84.53	40.02	85.93	14.25	46.74	1.81
w/ Rand. Edge Pruning	**79.08**	93.98	84.30	85.79	**22.07**	46.10	12.67	26.95	**3.18**
w/ Rand. Selection & Pruning	75.82	93.91	84.30	84.68	22.52	**45.04**	**12.60**	**26.72**	3.17

Table 2: Ablation study on Qwen2.5-14B-Instruct. CONCAT is the full model with confidence-driven leader selection (α=0.20) and benefit-prediction edge pruning. The first two ablations remove individual components: edge pruning (retaining all leader-to-leader edges) and consensus-based clustering (allowing all agents to participate directly). The last two replace learned components with random counterparts: random edge removal instead of benefit-prediction pruning, and additionally random leader assignment instead of confidence-driven selection. Here, "w/" and "w/o" denote "with" and "without", respectively. **Bold** and underline indicate the best and second-best accuracy, latency, and efficiency. Efficiency is average accuracy divided by average latency.

α	Accuracy↑				Latency↓				Efficiency↑
	MMLU	GSM8K	HumanEval	Avg.	MMLU	GSM8K	HumanEval	Avg.	
0.10	75.38	94.30	81.71	83.80	30.70	80.01	22.82	44.51	1.88
0.15	76.14	**94.45**	87.02	85.87	**25.48**	51.53	13.18	**30.06**	**2.86**
0.20	**77.78**	94.10	86.19	**86.02**	25.76	**51.19**	13.55	30.17	2.85
0.25	76.47	94.06	85.40	85.31	25.78	51.66	13.07	30.17	2.83
0.30	76.14	94.22	**87.05**	85.80	26.13	51.22	**12.88**	30.08	2.85

Table 3: Sensitivity of inertia-discount weight α on Qwen2.5-14B-Instruct. Accuracy (%), latency (s), and mean efficiency (average accuracy divided by average latency) are reported. **Bold** and underline indicate the best and second-best value per column. The highlighted row is the selected configuration.

Rand. Edge Pruning outperforms CONCAT on MMLU (79.08% vs. 77.78%), which we attribute to imperfect confidence calibration on heterogeneous knowledge tasks.

4.4 Hyperparameter Sensitivity Analysis

The benefit predictor in CONCAT contains a single empirical parameter $\alpha = 0.20$, which jointly controls the inertia-discount weight and the epistemic-openness weight in the Challenger formula (Eq. 8). We evaluate five candidate values $\alpha \in \{0.10, 0.15, 0.20, 0.25, 0.30\}$ on Qwen2.5-14B-Instruct across all three benchmarks, using mean efficiency as the selection criterion, where mean efficiency is defined as average accuracy divided by average latency.

As shown in Table 3, $\alpha = 0.10$ leads to significantly degraded efficiency (Mean Eff. = 1.88 vs. $\geq$2.82 for other values), driven by anomalously high GSM8K latency (80.01s vs. $\sim$51s). A very small α renders the inertia discount negligible, causing the benefit predictor to over-retain edges and reverting toward dense communication. For $\alpha \geq 0.15$, mean efficiency remains within a narrow range of 2.83–2.86, demonstrating low sensitivity to α in this regime. We select $\alpha = 0.20$ as it achieves the overall highest average accuracy across all benchmarks (86.02%) while maintaining competitive mean efficiency (2.85).

5 Conclusion

This paper explores methods to enhance the efficiency of LLM-based multi-agent systems by eliminating redundant communications in a training-free manner. We propose CONCAT, Consensus- and Confidence-Driven Ad Hoc Teaming, a self-organization MAS framework that realizes comparable accuracy while reducing the overall latency.

Our study points to several directions for further improvement of LLM-based multi-agent systems, including developing more robust ad hoc networking mechanisms that precompute the sparse topology for MAS inference, scaling up multi-agent systems through self-organization, and realizing controllable multi-agent collaborations using the Bayesian Theory of Mind model to supervise communication processes. Furthermore, Improving the confidence calibration of individual agents through methods such as temperature scaling or Bayesian calibration may further enhance the reliability of the ToM-inspired benefit predictor on heterogeneous knowledge benchmarks. Moreover, extending training-free and self-organizational multi-agent systems to broader multimodal domains and diverse real-world tasks such as web search is also promising for future research.

Limitations

Our work focuses on achieving efficient LLM-based multi-agent systems through consensus-based agent teaming and heuristic edge pruning for sparse communication. While our approach demonstrates promising results, several limitations should be acknowledged.

Confidence Quantification In our current implementation, an agent's confidence for a given response is calculated as the average token probability across output tokens. However, this represents a simplified measure of uncertainty. Alternative confidence quantification methods, such as entropy (Si et al., 2022) and Chain-of-Embeddings (Wang et al., 2024), may provide more accurate uncertainty estimates and warrant further investigation.

Extension to Real-World Scenarios We validate CONCAT on three common domains: mathematics, general reasoning, and code generation. However, the generalizability of CONCAT to other complex benchmarks, such as GAIA (Mialon et al., 2023), requires further verification. Additionally, our current implementation uses predefined, domain-specific agent roles that remain fixed throughout execution. Consequently, applying CONCAT to new domains necessitates careful design of targeted roles through prompt engineering, which may require domain expertise and iterative refinement.

Acknowledgments

We used Claude Sonnet 4.6 and Deepseek V4 Pro to polish the writing in this paper, including rephrasing sentences for clarity, improving grammatical fluency, and smoothing transitions. All scientific content, arguments, and conclusions were conceived and written by the authors. No AI-generated content was introduced beyond surface-level language editing.

References

Chris L. Baker, Julian Jara-Ettinger, Rebecca Saxe, and Joshua B. Tenenbaum. 2017. Rational quantitative attribution of beliefs, desires and percepts in human mentalizing. *Nature Human Behaviour*, 1:0064.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating Large Language Models Trained on Code. *Preprint*, arXiv:2107.03374.

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

Logan Cross, Violet Xiang, Agam Bhatia, Daniel Yamins, and Nick Haber. 2025. Hypothetical minds: Scaffolding theory of mind for multi-agent tasks with large language models. In *International Conference on Learning Representations*, volume 2025, pages 6507–6546.

Yufan Dang, Chen Qian, Xueheng Luo, Jingru Fan, Zihao Xie, Ruijie Shi, Weize Chen, Cheng Yang, Xiaoyin Che, Ye Tian, Xuantang Xiong, Lei Han, Zhiyuan Liu, and Maosong Sun. 2025. Multi-Agent Collaboration via Evolving Orchestration. *Preprint*, arXiv:2505.19591.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *Preprint*, arXiv:2501.12948.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving Factuality and Reasoning in Language Models through Multiagent Debate. In *Forty-First International Conference on Machine Learning*.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang, Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. 2024. Magentic-One: A Generalist Multi-Agent System for Solving Complex Tasks. *Preprint*, arXiv:2411.04468.

Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2013. *Bayesian Data Analysis*, 3rd edition. CRC Press.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv preprint arXiv:2402.01680*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2023. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *The Twelfth International Conference on Learning Representations*.

Ronald A. Howard. 1966. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Adam Kostka and Jarosław A Chudziak. 2025. Evaluating theory of mind and internal beliefs in llm-based multi-agent systems. In *International Conference on Computational Collective Intelligence*, pages 18–32. Springer.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Huao Li, Yu Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and Katia Sycara. 2023. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. 2025. WebThinker: Empowering Large Reasoning Models with Deep Research Capability. *Preprint*, arXiv:2504.21776.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. GAIA: A benchmark for General AI Assistants. *Preprint*, arXiv:2311.12983.

Chunjiang Mu, Ya Zeng, Qiaosheng Zhang, Kun Shao, Chen Chu, Hao Guo, Danyang Jia, Zhen Wang, and Shuyue Hu. 2026. Adaptive theory of mind for llm-based multi-agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29608–29616.

David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and Brain Sciences*, 1(4):515–526.

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. 2024. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Haojun Shi, Suyu Ye, Xinyu Fang, Chuanyang Jin, Leyla Isik, Yen-Ling Kuo, and Tianmin Shu. 2025. Muma-tom: Multi-modal multi-agent theory of mind. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1510–1519.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Lee Boyd-Graber, and Lijuan Wang. 2022. Prompting GPT-3 To Be Reliable. In *The Eleventh International Conference on Learning Representations*.

Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131.

Vllm-Project. 2025. Vllm-ascend.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

Yiming Wang, Pei Zhang, Baosong Yang, Derek F. Wong, and Rui Wang. 2024. Latent Space Chain-of-Embedding Enables Output-free LLM Self-Evaluation. In *The Thirteenth International Conference on Learning Representations*.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. 2025a. 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 (Volume 1: Long Papers)*, pages 24013–24035, Vienna, Austria. Association for Computational Linguistics.

Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, Xiaojian Ma, and Yitao Liang. 2025b. JARVIS-1: Open-World Multi-Task Agents With Memory-Augmented Multimodal Language Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(3):1894–1907.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,

and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhiyuan Weng, Guikun Chen, and Wenguan Wang. 2024. Do as We Do, Not as You Think: The Conformity of Large Language Models. In *The Thirteenth International Conference on Learning Representations*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Hancheng Ye, Zhengqi Gao, Mingyuan Ma, Qinsi Wang, Yuzhe Fu, Ming-Yu Chung, Yueqian Lin, Zhijian Liu, Jianyi Zhang, Danyang Zhuo, and Yiran Chen. 2025a. KVCOMM: Online Cross-context KV-cache Communication for Efficient LLM-based Multi-agent Systems. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Rui Ye, Shuo Tang, Rui Ge, Yaxin Du, Zhenfei Yin, Siheng Chen, and Jing Shao. 2025b. MAS-GPT: Training LLMs to Build LLM-based Multi-Agent Systems. In *Forty-Second International Conference on Machine Learning*.

Enhao Zhang, Erkang Zhu, Gagan Bansal, Adam Fourney, Hussein Mozannar, and Jack Gerrits. 2025a. Optimizing Sequential Multi-Step Tasks with Parallel LLM Agents. *Preprint*, arXiv:2507.08944.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. 2024. Cut the Crap: An Economical Communication Pipeline for LLM-based Multi-Agent Systems. In *The Thirteenth International Conference on Learning Representations*.

Wentao Zhang, Liang Zeng, Yuzhen Xiao, Yongcong Li, Ce Cui, Yilei Zhao, Rui Hu, Yang Liu, Yahui Zhou, and Bo An. 2025b. AgentOrchestra: A Hierarchical Multi-Agent Framework for General-Purpose Task Solving. *Preprint*, arXiv:2506.12508.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Xiaochen Zhu, Caiqi Zhang, Tom Stafford, Nigel Collier, and Andreas Vlachos. 2025. Conformity in Large Language Models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3854–3872, Vienna, Austria. Association for Computational Linguistics.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. 2024. GPTSwarm: Language Agents as Optimizable Graphs. In *Forty-First International Conference on Machine Learning*.

A Related Work

LLM-Based Multi-Agent System LLM-based multi-agent systems have emerged as a promising paradigm for solving complex tasks through collaborative intelligence, primarily adopting two architectural patterns: orchestrator-worker and distributed collaboration. In orchestrator-worker architectures, a central planner agent decomposes complex tasks into manageable subtasks and coordinates specialized worker agents to complete their respective responsibilities. Magnetic-One (Fourney et al., 2024) employs a dedicated orchestrator agent that dynamically plans task execution and delegates subtasks to domain-specific agents with heterogeneous capabilities. Hong et al. (Hong et al., 2023) propose MetaGPT, an innovative meta-programming framework that encodes standardized operating procedures into prompt sequences, enabling agents with human-like domain expertise to verify intermediate results through an assembly line paradigm. AgentOrchestra (Zhang et al., 2025b) enhances orchestration flexibility by allowing the planner to adaptively adjust agent compositions and workflow structures based on task requirements and intermediate execution feedback.

In contrast, distributed multi-agent architectures formalize collaboration through decentralized communication patterns without centralized control. GPTSwarm (Zhuge et al., 2024) represents multi-agent systems as temporal-spatial directed acyclic graphs, where different agents serve as distributed nodes that communicate and collaborate through graph edges, enabling flexible information flow and parallel task execution. This graph-based formalization allows agents to interact dynamically based on task dependencies and information requirements, supporting more scalable and fault-tolerant collaboration patterns compared to centralized orchestration approaches.

Efficient LLM-Based Multi-Agent System Recent research has explored various strategies to improve the efficiency of LLM-based multi-agent systems. For orchestrator-worker architectures, several approaches focus on optimizing the central coordination mechanism and execution paradigm. Ye et al. (2025b) propose MAS-GPT, which trains a 32B model to generate complete multi-agent systems adaptively for different queries, achieving efficiency through single-inference generation rather than iterative planning. Puppet (Dang et al., 2025) enhances coordination flexibility through an evolving orchestration mechanism that dynamically adjusts agent workflows based on intermediate execution states. Zhang et al. (2025a) propose to utilize parallel plan execution in M1-Parallel, which simultaneously runs different workflow plans and selects the fastest completion to reduce overall latency.

In contrast, distributed multi-agent systems have witnessed efficiency improvements through graph-based communication optimization and KV-cache management techniques. AgentPrune (Zhang et al., 2024) reduces communication overhead by employing graph neural networks to learn and prune unnecessary communication edges in the agent interaction topology. Wang et al. (2025a) propose to utilize dynamic agent elimination strategies in AgentDropout, which removes redundant nodes and their associated communications to minimize token consumption while preserving collaboration effectiveness. Li et al. (Ye et al., 2025a) introduce KVCOMM, an online cross-context KV-cache communication mechanism that enables efficient sharing and reuse of cached representations across agents, avoiding redundant processing of shared contextual information.

Despite these advances, existing graph-based communication optimization methods for distributed multi-agent systems face critical limitations. These approaches are typically designed for specific topology structures and require training graph neural networks on task-specific datasets with carefully annotated communication patterns. This training-based paradigm suffers from prolonged training time, limited generalization to unseen task distributions and agent configurations, and persistent redundant communications that remain even after optimization. Additionally, orchestrator improvement methods necessitate large-scale, high-quality datasets capturing diverse multi-agent interactions and substantial training resources, while parallel inference acceleration incurs high computational costs with weak controllability over efficiency-resource trade-offs, collectively limiting their practical applicability across diverse application scenarios.

Theory of Mind in Multi-Agent Systems Theory of Mind (ToM) (Premack and Woodruff, 1978) refers to the ability to attribute mental states, including beliefs, desires, and intentions, to oneself and others. Recent work has explored leveraging ToM in LLM-based multi-agent systems to improve coordination. Li et al. (2023) evaluate LLM agents

in cooperative text games requiring ToM inference, revealing that explicit ToM modeling can improve collaborative task performance. Cross et al. (2025) scaffold ToM reasoning for multi-agent tasks, enabling LLMs to hypothesize about other agents' states and coordinate more effectively. More recent work (Mu et al., 2026) proposes adaptive ToM for real-time multi-agent coordination, while Shi et al. (2025) extends ToM evaluation to multimodal embodied settings via a dedicated benchmark. Kostka and Chudziak (2025) provide a systematic evaluation of BDI-based ToM architectures in multi-agent LLM systems. Distinct from these approaches that directly model agent intent, CONCAT employs a lightweight ToM-inspired heuristic (Baker et al., 2017) to predict communication benefit without explicit mental state representation, enabling efficient training-free topology optimization.

LLM Conformity The conformity behavior of large language models, where LLMs adjust their responses based on information from previous interactions or peer outputs, has recently attracted research attention due to its implications for multi-agent collaboration. Zhu et al. (2025) investigate conformity in large language models and reveal that repeated wrong answers can mislead LLMs from correct responses to incorrect ones, while LLMs with confident initial responses tend to resist conforming to alternative answers. Do as We Do (Weng et al., 2024) explores how LLMs conform or resist conforming according to the consistency of previous discussions, demonstrating that larger models exhibit less conformative tendencies and maintain their original judgments more robustly. Both studies identify critical factors influencing conformity, including response confidence, answer repetition frequency, and model scale. However, existing investigations primarily employ rigid experimental settings that present LLMs with either unanimous consensus or controlled opposition scenarios, largely ignoring realistic situations where models are provided with multiple diverse answers or mixed signals containing both correct and incorrect information. This limitation restricts our understanding of how LLMs navigate complex, heterogeneous information landscapes in practical multi-agent systems, where agents may produce varied outputs with different confidence levels and correctness.

B Algorithmic Description of CONCAT

Algorithm 1 describes the workflow of our proposed framework, CONCAT.

C ToM-based Collaboration Benefit Prediction

The algorithm classifies the inter-agent relationship based on answer similarity, distinguishing between supporters ($s_{jk} \geq \theta_{\text{sim}}$) and challengers. For each case, the effective signal strength $\hat{c}_j$ captures how strongly agent v_j's signal challenges or supports agent v_k:

$$\hat{c}_j = \begin{cases} c_j \cdot s_{jk} & \text{if SUPPORTER} \\ c_j \cdot (1 - s_{jk}) & \text{if CHALLENGER} \end{cases} \tag{15}$$

Challenger case. The benefit formula is derived from a Bayesian Expected Utility of Communication (EUC) framework (Baker et al., 2017), linearized via Taylor expansion around $\hat{c}_j = 0.5$:

$$b_{j\to k} = \underbrace{4c_k(1-c_k)\cdot \hat{c}_j}_{\text{correction gain}} - \underbrace{0.2 \cdot \frac{1+2c_k}{2+2c_k+2\hat{c}_j}}_{\text{inertia discount}} + \underbrace{0.2\cdot(1-c_k)}_{\text{epistemic openness}}, \tag{16}$$

The coefficient $4c_k(1-c_k)$ is the derivative of the exact EUC formula at $\hat{c}_j$=0.5, capturing the intuition that agents with moderate confidence are most receptive to correction. The inertia discount $p_{\text{stay}} = \frac{1+2c_k}{2+2c_k+2\hat{c}_j}$ is the Beta-Binomial posterior mean (Gelman et al., 2013), modeling LLM anchoring effects (Tversky and Kahneman, 1974). The epistemic openness term $(1-c_k)$ follows Value of Information theory (Howard, 1966): uncertain agents gain more from additional signals. The benefit threshold $\tau_{\min} = 0$ is theoretically justified by the exact EUC formula (Appendix F): $b_{j\to k} < 0$ when the source signal $\hat{c}_j$ falls below the focal agent's correction threshold $\hat{c}_j^*(c_k)$, meaning communication is predicted to reduce expected correctness. Negative-benefit edges are therefore absolutely filtered regardless of the percentile cutoff p.

Supporter case. When v_j agrees with v_k ($s_{jk} \geq \theta_{\text{sim}}$), the benefit quantifies confidence reinforcement:

$$b_{j\to k} = \alpha \cdot (\hat{c}_j - c_k), \quad \hat{c}_j = c_j \cdot s_{jk} \tag{17}$$

A supporter with higher effective confidence than v_k provides positive value; an uncertain supporter yields negative benefit.

This lightweight heuristic contains a single empirical parameter $\alpha = 0.2$, selected via hyperparameter search in §4.4. It operates purely on observable signals without training data, and is motivated by the empirical finding that dissent strength predicts collaboration benefit with ROC-AUC of 0.74–0.86, as shown in Figure 3. The full theoretical derivation, covering the exact EUC formula, three strict propositions, Taylor linearization, and Beta-Binomial conjugate update, is provided in Appendix F.

D Collaboration Outcome Labels

Table 4 defines the four collaboration outcome categories used in Figure 2 and for computing the helpful/not-helpful binary label in Figure 3. Each agent pair is classified solely by the focal agent's answer correctness before (S_1) and after (S_2) collaboration, without reference to confidence scores.

Table 4: Collaboration outcome categories based on answer correctness transition.

S_1 **correct**	S_2 **correct**	**Category**	**Helpful?**
No	Yes	Wrong→Correct	Yes
Yes	Yes	Correct→Correct	No
No	No	Wrong→Wrong	No
Yes	No	Correct→Wrong	No

E Supplementary Implementation Details

For Llama-3-8B-Instruct, Qwen2.5-14B-Instruct, and Qwen2.5-72B-Instruct, we configure the maximum sequence length for model outputs to 32,768 tokens. For CONCAT, we set the edge retention rate as 0.7 and the code similarity threshold for clustering as 0.45. The clustering algorithm on the code generation task is hierarchical clustering. For SC-CoT, five responses for each question are sampled for majority voting. For multi-agent methods, we utilize a unified setting with five agents and two-round collaboration. The answer aggregation prompts on MMLU, GSM8K, and HumanEval are shown in Figure 6, 7, and 8.

F Theoretical Derivation of the Benefit Predictor

This appendix provides the complete mathematical derivation underlying the ToM-based benefit predictor in Algorithm 2.

F.1 Problem Setup and Assumptions

For a focal agent v_k with confidence c_k and a source agent v_j with effective signal strength $\hat{c}_j$ (Eq. of the Challenger case), we define the *Expected Utility of Communication* (EUC):

$$b_{j\to k}^{\text{exact}} \triangleq \mathbb{E}[U \mid s_j] - \mathbb{E}[U] \tag{18}$$

where U is a 0-1 utility that equals 1 if v_k answers correctly. The key assumptions are: (A1) $c_i \triangleq P_i(a_i = a^*)$: we interpret confidence as a proxy for the subjective probability of correctness; in practice, c_i is computed as the average token probability of the response (see §3.2), which serves as an empirical approximation of this quantity; (A2) v_k makes a binary choice between $\{a_k, a_j\}$; (A3) conditional independence of beliefs given a^*; (A4) v_k uses c_j as its estimate of v_j's correctness, following first-order ToM (Baker et al., 2017).

F.2 Exact EUC Formula (Challenger Case)

Before communication: $\mathbb{E}[U]_{\text{before}} = c_k$.

After observing v_j's signal, v_k performs Bayesian belief update. Two hypotheses compete:

Hypothesis	Prior	Likelihood	Joint
a_k correct	c_k	$(1-\hat{c}_j)$	$c_k(1-\hat{c}_j)$
a_j correct	$(1-c_k)$	$\hat{c}_j$	$(1-c_k)\hat{c}_j$

The normalization factor is $Z = c_k(1-\hat{c}_j) + (1-c_k)\hat{c}_j$. The Bayesian-optimal decision chooses the larger posterior:

$$b_{j\to k}^{\text{exact}} = \frac{\max\big(c_k(1-\hat{c}_j),\ (1-c_k)\hat{c}_j\big)}{Z} - c_k \tag{19}$$

Proposition 1 (Sign): $b_{j\to k}^{\text{exact}} > 0 \iff \hat{c}_j > \hat{c}_j^*(c_k)$, where the threshold

$$\hat{c}_j^*(c_k) = \frac{c_k^2}{1 - 2c_k + 2c_k^2}$$

is a monotonically increasing function of c_k that equals c_k only at $c_k = 0.5$. When $c_k > 0.5$, the threshold exceeds c_k: a highly confident focal agent requires a correspondingly stronger challenge signal to benefit. When $c_k < 0.5$, the threshold falls

Figure 6: MMLU Answer Aggregation Prompt

System Prompt:

```
You are the top decision-maker and are good at analyzing and summarizing other people's opinions,
finding errors and giving final answers.

I will ask you a question. I will also give you 4 answers enumerated as A, B, C and D. Only one
answer out of the offered 4 is correct. You must choose the correct answer to the question. Your
response must be one of the 4 letters: A, B, C or D, corresponding to the correct answer. I will
give you some other people's answers and analysis. Your reply must only contain one letter and
cannot have any other characters. For example, your reply can be A.
```

User Prompt:

```
{few-shot examples} The task is: {question}. At the same time, the output of other agents is as
follows: {agent responses}
```

Figure 7: GSM8K Answer Aggregation Prompt

System Prompt:

```
You are the top decision-maker. Good at analyzing and summarizing mathematical problems, judging
and summarizing other people's solutions, and giving final answers to math problems.

You will be given a math problem, analysis and code from other agents. Please find the most
reliable answer based on the analysis and results of other agents. Give reasons for making
decisions. The last line of your output contains only the final result without any units, for
example: The answer is 140
```

User Prompt:

```
{few-shot examples} The task is: {question}. At the same time, the output of other agents is as
follows: {agent responses}
```

Figure 8: HumanEval Answer Aggregation Prompt

System Prompt:

```
You are the top decision-maker and are good at analyzing and summarizing other people's opinions,
finding errors and giving final answers. And you are an AI that only responds with only python
code.

You will be given a function signature and its docstring by the user. You may be given the
overall code design, algorithm framework, code implementation or test problems. Write your full
implementation (restate the function signature). If the prompt given to you contains code that
passed internal testing, you can choose the most reliable reply. If there is no code that has
passed internal testing in the prompt, you can change it yourself according to the prompt. Use a
Python code block to write your response. For example:
```python
print('Hello world!')
```
Do not include anything other than Python code blocks in your response.
```

User Prompt:

```
The task is: {question}. At the same time, the outputs and feedbacks of other agents are as
follows: {agent responses with test results}
```

below c_k: an uncertain focal agent benefits even from a moderately confident dissenter.

Proposition 2 (Monotonicity): $b_{j\to k}^{\text{exact}}$ is monotonically increasing in $\hat{c}_j$ and monotonically decreasing in c_k.

Proposition 3 (Zero point): $b_{j\to k}^{\text{exact}} = 0 \iff \hat{c}_j = \hat{c}_j^*(c_k)$.

These propositions justify setting $\tau_{\min} = 0$: by

Proposition 1, edges with $b_{j\to k} < 0$ correspond to source signals too weak to overcome the focal agent's prior, and are therefore predicted to be harmful or neutral on average.

F.3 Taylor Linearization

The exact formula is numerically unstable when $c_k \to 1$ because the denominator $Z \to 0$. It also does not model LLM anchoring effects (Tversky and Kahneman, 1974). We expand around $\hat{c}_j = 0.5$:

$$b_{j\to k}^{\text{exact}} \approx (1-2c_k) + 4c_k(1-c_k)\cdot(\hat{c}_j - 0.5) \quad (20)$$

The coefficient $4c_k(1-c_k)$ has an elegant interpretation: it is maximized at $c_k = 0.5$, where the agent is most receptive to correction, and approaches 0 as $c_k \to 0$ or $c_k \to 1$, where extreme confidence makes the agent insensitive to challenges. Absorbing the constant $-2c_k(1-c_k)$ into the intercept and simplifying gives the correction gain term $4c_k(1-c_k)\cdot\hat{c}_j$ in Eq. 8.

F.4 Beta-Binomial Inertia Discount

LLMs exhibit anchoring effects: even when $\hat{c}_j > c_k$, the agent does not always switch answers. We model the probability of v_k maintaining its answer via Beta-Binomial conjugate updating (Gelman et al., 2013):

- Prior strength: $\alpha_0 = 1 + 2c_k$, where confidence acts as an equivalent prior sample count.
- Challenge evidence: $\beta_0 = 1 + 2\hat{c}_j$, where challenge strength acts as equivalent counter-evidence.
- Posterior mean: $p_{\text{stay}} = \frac{1+2c_k}{2+2c_k+2\hat{c}_j}$.

When c_k is high, p_{stay} is large and the agent is harder to dislodge. When $\hat{c}_j$ is high, p_{stay} is small and a strong challenge overcomes inertia. This is consistent with empirical conformity findings (Zhu et al., 2025).

F.5 Epistemic Openness Term

By Value of Information (VoI) theory (Howard, 1966), an agent with higher uncertainty benefits more from additional information. The term $(1-c_k)$ directly proxies v_k's epistemic uncertainty, providing a floor for communication benefit even when $\hat{c}_j$ is not particularly high. Combining all components with empirical weight $\alpha = 0.2$, selected via hyperparameter search in §4.4, gives the final Challenger formula in Eq. 8.

G Token Consumption

As shown in Table 5, CONCAT consumes substantially fewer tokens than full-communication baselines across all three benchmarks. On average, CONCAT uses 1.9M total tokens on Llama-3-8B-Instruct, representing a 45.7% reduction compared to LLM-Debate (3.5M) and a 48.6% reduction compared to Vanilla MAS Layered (3.7M). The savings are most pronounced on GSM8K, where CONCAT's 5.1M total tokens compare favorably against LLM-Debate's 8.9M and Vanilla MAS Layered's 9.6M. Among all MAS methods, AgentDropout (Chain) achieves the second-lowest token consumption (1.7M avg.) owing to its sparse topology, yet CONCAT surpasses it in efficiency by maintaining higher average accuracy (64.97% vs. 65.79%) with a lower-latency communication schedule.

Method	Topology	MMLU			GSM8K			HumanEval			Avg.		
		Ptok.	**Ctok.**	**Ttok.**	**Ptok.**	**Ctok.**	**Ttok.**	**Ptok.**	**Ctok.**	**Ttok.**	**Ptok.**	**Ctok.**	**Ttok.**
CoT	-	21K	12K	32K	634K	149K	783K	18K	6K	24K	224K	56K	280K
SC-CoT	-	103K	46K	149K	3.2M	751K	3.9M	146K	39K	185K	1.1M	279K	1.4M
LLM-Debate	Debate	566K	106K	673K	7.2M	1.7M	8.9M	619K	190K	808K	2.8M	656K	3.5M
Vanilla MAS	Star	613K	94K	708K	8.1M	1.7M	9.8M	529K	114K	643K	3.1M	633K	3.7M
	Chain	427K	104K	531K	6.2M	1.7M	7.9M	392K	149K	542K	2.3M	654K	3.0M
	Random	527K	98K	626K	6.9M	1.8M	8.7M	567K	180K	747K	2.7M	687K	3.4M
	Layered	634K	100K	734K	7.7M	1.9M	9.6M	666K	173K	839K	3.0M	716K	3.7M
	FullConnected	686K	98K	784K	8.2M	1.7M	9.9M	828K	194K	1.0M	3.2M	670K	3.9M
AgentDropout	Star	302K	62K	364K	5.0M	1.3M	6.4M	376K	104K	480K	1.9M	498K	2.4M
	Chain	210K	59K	269K	3.4M	1.1M	4.5M	188K	96K	284K	1.3M	405K	1.7M
	Random	271K	57K	328K	4.2M	1.1M	5.3M	337K	113K	450K	1.6M	421K	2.0M
	Layered	297K	59K	356K	4.5M	1.1M	5.6M	348K	114K	462K	1.7M	440K	2.1M
	FullConnected	328K	56K	384K	5.0M	1.3M	6.3M	447K	127K	574K	1.9M	491K	2.4M
CONCAT (ours)	Hybrid	253K	66K	319K	4.0M	1.1M	5.1M	227K	95K	322K	1.5M	413K	1.9M

Table 5: Token consumption comparison on Llama-3-8B-Instruct. Ptok. = prompt tokens, Ctok. = completion tokens, Ttok. = total context length.

Algorithm 1 CONCAT: Consensus- and Confidence-Driven Ad Hoc Teaming for Multi-Agent Systems

Require: Query q, Agent set $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$, Number of rounds m, Pruning percentile p, Benefit threshold τ_{min}

Ensure: Final answer $\hat{y}$

// Round 0: Independent Answer Generation (Hello Packet)
for each agent $a_i \in \mathcal{A}$ **do**
  $(r_i^{(0)}, c_i^{(0)}) \leftarrow a_i(q)$ {Generate answer $r_i^{(0)}$ and confidence $c_i^{(0)}$}
end for
// Round 1 to $m-1$: Iterative Leader Debate with ToM-based Pruning
for $t = 1$ to $m-1$ **do**
  // Step 1: Answer Clustering
  $\mathcal{C}^{(t)} \leftarrow \text{ClusterBySimilarity}(\{r_i^{(t-1)}\}_{i=1}^N)$ {Group agents by answer similarity}
  // Step 2: Leader Selection
  $\mathcal{L}^{(t)} \leftarrow \emptyset$ {Initialize leader set}
  for each cluster $C_j \in \mathcal{C}^{(t)}$ **do**
    $l_j \leftarrow \arg\max_{a_i \in C_j} c_i^{(t-1)}$ {Select agent with highest confidence}
    $\mathcal{L}^{(t)} \leftarrow \mathcal{L}^{(t)} \cup \{l_j\}$
  end for
  // Step 3: ToM-based Collaboration Benefit Prediction (Parallel)
  $\mathcal{B}^{(t)} \leftarrow \emptyset$ {Initialize benefit matrix}
  for each leader pair $(l_j, l_k) \in \mathcal{L}^{(t)} \times \mathcal{L}^{(t)}$ where $j \neq k$ **in parallel do**
    // Apply Theory of Mind heuristic:
      // 1. Classify l_j as supporter or challenger of l_k based on answer similarity
      // 2. Compute Bayesian posterior belief and correction potential
      // 3. Weight by confidence values
    $b_{j \to k} \leftarrow \text{ToM-Predict}(r_k^{(t-1)}, c_k^{(t-1)}, r_j^{(t-1)}, c_j^{(t-1)})$ {Predict collaboration benefit}
    $\mathcal{B}^{(t)} \leftarrow \mathcal{B}^{(t)} \cup \{((l_j, l_k), b_{j \to k})\}$
  end for
  // Step 4: Benefit-based Edge Pruning
  $\tau^{(t)} \leftarrow \max\left(\text{Percentile}(\{b \mid (e, b) \in \mathcal{B}^{(t)}\}, p \times 100), \tau_{min}\right)$
  $\mathcal{E}^{(t)} \leftarrow \{(l_j, l_k) \mid ((l_j, l_k), b_{j \to k}) \in \mathcal{B}^{(t)}, b_{j \to k} \geq \tau^{(t)}\}$ {Keep high-benefit edges only}
  // Step 5: Construct Sparse Leader Debate Topology
  for each leader $l_k \in \mathcal{L}^{(t)}$ **do**
    $\mathcal{R}_k^{(t)} \leftarrow \{r_j^{(t-1)} \mid (l_j, l_k) \in \mathcal{E}^{(t)}\}$ {Collect answers from beneficial leaders only}
  end for
  // Step 6: Leader Answer Refinement
  for each leader $l_j \in \mathcal{L}^{(t)}$ **do**
    $(r_j^{(t)}, c_j^{(t)}) \leftarrow l_j(q, \mathcal{R}_j^{(t)})$ {Refine answer via selective debate}
  end for
  // Non-leaders maintain their previous answers
  for each agent $a_i \notin \mathcal{L}^{(t)}$ **do**
    $(r_i^{(t)}, c_i^{(t)}) \leftarrow (r_i^{(t-1)}, c_i^{(t-1)})$
  end for
end for
// Final Aggregation
$\hat{y} \leftarrow \text{Aggregate}(\{r_i^{(m-1)}\}_{i=1}^N)$ {E.g., majority voting or LLM summarization}
return $\hat{y}$

Algorithm 2 ToM-based Collaboration Benefit Prediction

Require: Target agent's answer r_k and confidence c_k, Source agent's answer r_j and confidence c_j, Similarity threshold θ_{sim}, Weight parameter α

Ensure: Collaboration benefit $b_{j \to k} \in \mathbb{R}$

// Step 1: Classify relationship via answer similarity
$s_{jk} \leftarrow \text{Similarity}(r_j, r_k)$ {Compute answer similarity}
if $s_{jk} \geq \theta_{sim}$ **then**
    type $\leftarrow$ SUPPORTER
else
    type $\leftarrow$ CHALLENGER
end if
// Step 2: Compute effective signal strength
if type $=$ SUPPORTER **then**
    $\hat{c}_j \leftarrow c_j \cdot s_{jk}$ {Supporter: high similarity amplifies effective confidence}
    $b_{j \to k} \leftarrow \alpha \cdot (\hat{c}_j - c_k)$ {Confidence reinforcement (Supporter formula)}
else
    $\hat{c}_j \leftarrow c_j \cdot (1 - s_{jk})$ {Challenger: high dissimilarity amplifies effective challenge strength}
    // Step 3: Beta-Binomial inertia discount (Bayesian posterior mean)
    $p_{\text{stay}} \leftarrow \frac{1+2c_k}{2+2c_k+2\hat{c}_j}$ {Probability of v_k maintaining current answer}
    // Step 4: Challenger benefit (Taylor-linearized EUC formula)
    $b_{j \to k} \leftarrow 4c_k(1 - c_k) \cdot \hat{c}_j - \alpha \cdot p_{\text{stay}} + \alpha(1 - c_k)$ {Correction gain $-$ inertia discount $+$ epistemic openness}
end if
return $b_{j \to k}$ {Negative values indicate predicted harmful communication}