ENHANCING MULTI-AGENT COMMUNICATION THROUGH ATTENTION STEERING WITH CONTEXT RELEVANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-based multi-agent systems often accumulate long communication histories. As a result, relevant information is increasingly diluted by irrelevant context, leading to degraded performance. Existing context management methods compress or prune the history, which may discard subtle but critical information. Attention steering preserves the full history, but identifying relevant context remains challenging as multi-agent communication evolves at every reasoning step. To address this, we introduce AGENT-RADAR, an attention steering method that leverages the inherent structure of multi-agent communication to dynamically guide agents' attention toward relevant context. Across five benchmarks, AGENT-RADAR outperforms state-of-the-art methods, yielding average gains of 4.17 absolute points over the strongest baseline. Furthermore, our analysis shows that AGENT-RADAR remains effective and robust as the number of agents and interaction rounds increases. Finally, the ablations show that the core components in AGENT-RADAR are crucial to performance.

1 INTRODUCTION

LLM-based multi-agent systems (MAS) have demonstrated remarkable performance on complex tasks (Liang et al., 2024; Wang et al., 2024b; Du et al., 2024; Wu et al., 2024). Despite these successes, agent communication often accumulates long histories (Zhang et al., 2025a). As a result, the original task description, key constraints, and relevant intermediate results are often buried within the context, making it difficult for LLMs to reliably identify and attend to them (Wang et al., 2025b; Zhang et al., 2025b). This phenomenon, also known as the "lost-in-the-middle" effect, frequently induces hallucinations and logical errors during generation (Liu et al., 2024a). This issue is further exacerbated as agentic systems scale. Increasing numbers of agents and interaction rounds cause communication histories to grow rapidly, amplifying context dilution and undermining the reliability of large-scale agent collaborations (Cemri et al., 2025; Becker et al., 2026).

Prior context management methods have sought to mitigate this challenge in two ways. One line of work applies periodic summarization to compress the evolving history (Wang et al., 2025a; Wu et al., 2025). Another adopts pruning strategies that remove redundant agents or communication edges (Zhang et al., 2025a; Wang et al., 2025b). While effective at improving efficiency, these methods can introduce information loss. Specifically, compressing or pruning messages can discard subtle but critical signals, potentially misleading downstream reasoning and propagating errors across subsequent interactions (Shen et al., 2025; Acharjee et al., 2025).

Attention steering methods provide a promising way to address this problem without the need to prune or compress context information. They keep the full input and steer model attention to selected tokens during generation (Zhang et al., 2024b; Tian & Zhang, 2025). However, existing attention steering methods are either too computationally expensive or require manual context selection. As multi-agent communication evolves, relevant context constantly shifts as the task progresses.

To address this limitation, we propose AGENT-RADAR, a lightweight attention steering method that dynamically guides model attention toward relevant context without manual context selection. As multi-agent communication evolves, a fundamental challenge is to identify the relevant context at each step against the complex and heterogeneous communication history, which interleaves instruc-

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

tions, planning or reasoning messages, tool executions, intermediate outputs, etc. from different agents. A naive approach is to apply semantic retrieval to the communication history. However, this approach treats the history as flat text and may retrieve outdated information or get distracted by messages from agents that are not directly related to the current context. By contrast, AGENT-RADAR leverages the inherent communication structure and identifying relevant context based on multiple facets including semantic similarity, temporal recency, and spatial proximity in the communication graph. AGENT-RADAR then applies a lightweight attention approximation via logit arithmetic to steer the model's attention toward the relevant context.

We evaluate AGENT-RADAR on five benchmarks using three base LLMs. Compared with SOTA context management methods, including compression-based approaches (Wang et al., 2025a) and pruning methods (Zhang et al., 2025a; Wang et al., 2025b), AGENT-RADAR achieves average gains of 4.17 absolute points over the strongest baseline. We also demonstrate that the performance gain generalizes across three different agentic systems, including a debate-based system (Du et al., 2024), GPTSwarm (Zhuge et al., 2024), and AutoGen (Wu et al., 2024). AGENT-RADAR consistently improves the performance of these systems (4.54 absolute points on average). Furthermore, AGENT-RADAR remains robust as systems scale to larger numbers of agents and longer interaction rounds. Ablations show that spatial and temporal components together improve over semantic-only context selection by 3.60 to 4.80 absolute points.

2 PROBLEM FORMULATION

We model the multi-agent system architecture as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$. Let $\mathcal{V} = \{v_i\}_{i=1}^{N}$ be the set of N nodes, where each node v_i represents an autonomous LLM agent. The edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ defines the communication topology. An edge $(v_i, v_j) \in \mathcal{E}$ denotes the communication path from the agent v_i to the agent v_j. The set of mapping functions $\mathcal{F} = \{f_i\}_{i=1}^{N}$ represents the inference processes of the agents. We further define the hop distance $d_{\mathcal{G}}(v_j, v_i)$ as the length of the shortest path from v_j to v_i in $\mathcal{G}$. The k-hop neighborhood of v_i is $\mathcal{N}_k(v_i) = \{v_j \in \mathcal{V} \mid d_{\mathcal{G}}(v_j, v_i) \leq k\}$, and $\mathcal{N}_{N-1}(v_i)$ denotes the set of all agents that can reach v_i.

The system proceeds over communication rounds. At round t, the active agent v_i generates the response

$$o_i^{(t)} = f_i\Big(q_i^{(t)},\ \mathcal{Z}_i^{(t)}\Big), \tag{1}$$

where $q_i^{(t)}$ is the query received by the current agent at round t, and $\mathcal{Z}_i^{(t)}$ is the accumulated communication history available to v_i. The context available to the current agent v_i at round t satisfies

$$\mathcal{Z}_i^{(t)} \subseteq \{\ o_j^{(\tau)} \mid v_j \in \mathcal{N}_{N-1}(v_i),\ 1 \leq \tau < t\ \}, \tag{2}$$

where $o_j^{(\tau)}$ is the output produced by agent v_j in communication round τ.

For each agent generation, AGENT-RADAR keeps the full history $\mathcal{Z}_i^{(t)}$ as input and identifies a subset $\hat{\mathcal{Z}}_i^{(t)} \subseteq \mathcal{Z}_i^{(t)}$. This subset does not replace the input, it indicates which tokens in $\mathcal{Z}_i^{(t)}$ receive attention steering. In the following section, we describe how AGENT-RADAR identifies $\hat{\mathcal{Z}}_i^{(t)}$ from the accumulated communication history and passes it to the inference function f_i for attention steering.

3 METHODOLOGY

We introduce AGENT-RADAR (Figure 1), a training-free method designed to optimize how agents process long communication histories. Instead of treating histories as flat text, AGENT-RADAR leverages the inherent communication structure and identifies relevant context based on multiple facets including semantic relevance, spatial proximity in the communication graph, and temporal recency. Then, it steers the agent's attention toward the relevant context during generation.

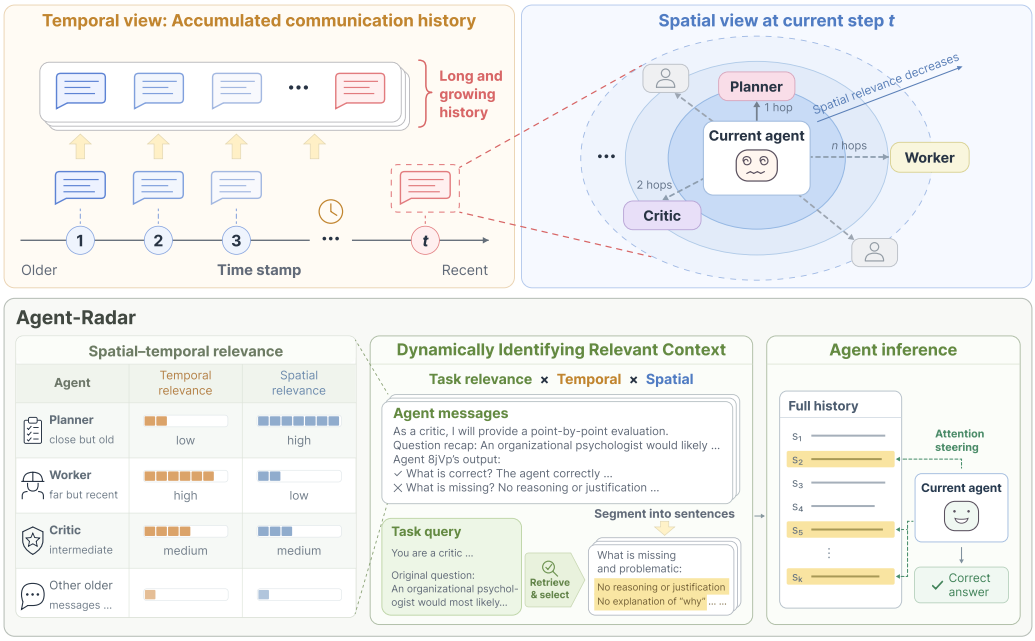

Figure 1: Overview of AGENT-RADAR. ***(Top)*** MAS interactions rapidly accumulate long communication histories, where useful information is buried in the middle, receiving insufficient attention. ***(Bottom)*** AGENT-RADAR preserves the full transcript and topology, scores sentence-level context by semantic relevance weighted with temporal and spatial decay, and steers the agent's attention toward the relevant context during inference.

3.1 DYNAMICALLY IDENTIFYING RELEVANT CONTEXT

While standard semantic retrieval is effective in many cases, applying it naively to the dynamic MAS context is insufficient.[1] Unlike a flat document collection, multi-agent communication is generated by different agents at different communication rounds, and these source and temporal signals affect whether a previous response is likely to be useful for the current active agent. In particular, a sentence from a previously generated output can remain semantically relevant to the inference objective of the current agent while being outdated or not directly related to the current context. Pure semantic retrieval ignores these signals and may therefore identify stale or indirectly related context. To this end, we formulate AGENT-RADAR as identifying relevant context over the spatio-temporal communication history. For each generated output $o_j^{(\tau)}$ in the communication history, we assign a scalar relevance score that combines a spatial decay factor and a temporal decay factor:

$$r_i(o_j^{(\tau)};\, t) \;=\; \underbrace{\phi_s(v_j,\, v_i)}_{\text{spatial decay}} \cdot \underbrace{\phi_t(\tau;\, t)}_{\text{temporal decay}} \;, \tag{3}$$

where ϕ_s and ϕ_t are the spatial and temporal decay factors. Given these scores, we retrieve fine-grained, sentence-level relevant context by combining this spatio-temporal score with semantic similarity to the query received by the current agent.

Spatial Decay. The organizational network theory (Borgatti et al., 2009; Allen, 1984; Park et al., 2023) finds that the relevance of information in collaborative networks typically declines with structural distance. In a multi-agent workflow, responses from nearby collaborators are more likely to contain direct, actionable evidence, whereas information propagating from distant agents often introduces indirect noise or redundant context. Based on this insight, we use an exponential form

[1]We present ablation studies in Section 4.4 comparing AGENT-RADAR against semantic-only retrieval and variants with spatial or temporal decay individually removed.

because each additional relay can be viewed as retaining only a fixed fraction of the useful signal. If an extra hop retains a fraction λ_s of relevance, then k additional hops retain λ_s^k. Appendix A.3 provides further justification and compares this choice with linear, logarithmic, and power-law alternatives. Therefore, the first term in Equation 3 models spatial relevance as a decay over the distance of the graph:

$$\phi_s(v_j, v_i) = \lambda_s^{\max\{d_{\mathcal{G}}(v_j, v_i)-1, 0\}}, \quad 0 < \lambda_s < 1, \tag{4}$$

where $d_{\mathcal{G}}(v_j, v_i)$ is the distance from Section 2 and λ_s controls the decay rate. For direct neighbors or self-generated critiques, we set $\phi_s = 1$. Generated outputs from agents further away receive exponentially lower weights.

Temporal Decay. As the discussion evolves, earlier responses should not be treated as equally informative as recent exchanges, which often refine, correct, or supersede earlier reasoning. This intuition is consistent with the recency effect observed in human memory (Finkenbinder, 1913) and with decay mechanisms in long-horizon sequence modeling (Sun et al., 2023; Ma et al., 2023). We use the same idea for temporal decay that discounts responses by their age:

$$\phi_t(\tau;\, t) = \lambda_t^{\left(t-\tau-1\right)}, \quad 0 < \lambda_t < 1, \tag{5}$$

where t is the current communication round, τ is the round when the response was produced, $t-\tau-1 \geq 0$ is the age of the generated output, and λ_t governs the decay rate. The most recent response receives a weight of 1, and older responses are discounted exponentially.

Retrieval. On top of these two decay mechanisms, we perform fine-grained sentence-level retrieval. Unlike previous methods that retrieve whole message chunks, steering attention to an entire generated response can still introduce irrelevant details, since only part of the message may be useful for the current query. We therefore retain only the relevant context from each generated output.

Specifically, we first segment each generated output $o_j^{(\tau)}$ into a sequence of sentences $\mathcal{S}_{j,\tau} = \{s_k\}_{k=1}^{K_{j,\tau}}$. Each sentence s_k is then encoded by a pre-trained sentence encoder $\mathrm{Enc}(\cdot)$[2]:

$$\mathbf{e}_k = \mathrm{Enc}(s_k) \in \mathbb{R}^d, \quad \mathbf{e}_{q_i^{(t)}} = \mathrm{Enc}(q_i^{(t)}) \in \mathbb{R}^d. \tag{6}$$

The semantic relevance of sentence s_k to the current query $q_i^{(t)}$ is measured by cosine similarity:

$$\phi_{\mathrm{sem}}(s_k,\, q_i^{(t)}) = \frac{\mathbf{e}_k^\top \mathbf{e}_{q_i^{(t)}}}{\|\mathbf{e}_k\|\, \|\mathbf{e}_{q_i^{(t)}}\|} \in [-1, 1]. \tag{7}$$

The final relevance score for sentence s_k from generated output $o_j^{(\tau)}$ is obtained by combining the spatio-temporal scalar with semantic similarity:

$$\mathrm{score}(s_k \mid o_j^{(\tau)},\, v_i,\, t,\, q_i^{(t)}) \;=\; r_i(o_j^{(\tau)};\, t) \;\cdot\; \phi_{\mathrm{sem}}(s_k,\, q_i^{(t)}). \tag{8}$$

This factored design ensures that a relevant sentence is recent, close to the receiving agent, and semantically aligned with the query $q_i^{(t)}$ received by the current agent.

The relevant context is constructed by retaining all sentences whose combined relevance score exceeds a threshold θ, with the current query $q_i^{(t)}$ included[3]:

$$\hat{\mathcal{Z}}_i^{(t)} = \{q_i^{(t)}\} \;\cup\; \Big\{ s_k \in \mathcal{S}_{j,\tau} \;\Big|\; o_j^{(\tau)} \in \mathcal{Z}_i^{(t)},\; \mathrm{score}(s_k \mid o_j^{(\tau)}, v_i, t, q_i^{(t)}) \geq \theta \Big\}, \tag{9}$$

where θ controls the trade-off between context coverage and steering selectivity. This retrieval does not prune the transcript. Instead, it only determines which context receives explicit attention steering. Note that the current query $q_i^{(t)}$ is always included in the relevant context, $\hat{\mathcal{Z}}_i^{(t)}$, allowing the model to always keep attention to the original instructions and core task constraints.

[2]Sentence segmentation and retrieval are implemented using SentenceTransformer (`all-MiniLM-L6-v2` as the default encoder) (Reimers & Gurevych, 2019). Alternative retrieval methods are provided in Section 4.4.

[3]Since $r_i > 0$, sentences with $\phi_{\mathrm{sem}} \leq 0$ are excluded by Eq. 9. We analyze the hyperparameter selection in Appendix A.4.

3.2 Attention Steering and Agent Inference

The agent inference is then updated to

$$o_i^{(t)} = f_i\Big(q_i^{(t)},\ \mathcal{Z}_i^{(t)},\ \hat{\mathcal{Z}}_i^{(t)}\Big), \tag{10}$$

where $\hat{\mathcal{Z}}_i^{(t)}$ denotes the identified context that AGENT-RADAR steers attention towards. Following Selective Prompt Anchoring (SPA) (Tian & Zhang, 2025), we perform a lightweight attention approximation that avoids editing attention weights inside the model and thus reduces the overhead of steering. SPA contrasts the next-token logits computed from the full input $\mathcal{Z}_i^{(t)}$ with those computed from the same input with $\hat{\mathcal{Z}}_i^{(t)}$ masked. The difference isolates the influence of $\hat{\mathcal{Z}}_i^{(t)}$ on the output. We scale it in logit space to amplify attention toward the sentences in $\hat{\mathcal{Z}}_i^{(t)}$ This design is plug-and-play and training-free: it does not modify the backbone LLM weights nor the MAS topology, and can be integrated with different MAS systems. We also compare with alternative attention steering method in Appendix A.5.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate AGENT-RADAR on three categories of benchmarks. For open-ended question answering, we use **HotpotQA** (Yang et al., 2018), **2WikiMultihopQA** (Ho et al., 2020), and **MuSiQue** (Trivedi et al., 2022) to evaluate the system's ability to retrieve and synthesize information across multiple documents. We report F1 scores for these open-ended QA benchmarks. For mathematical reasoning, we use **MATH-500** (Lightman et al., 2024), which represents competition-level math problems (Hendrycks et al., 2021). We report accuracy for MATH-500. For general reasoning, we adopt **MMLU-Pro** (Wang et al., 2024a), an enhanced benchmark designed to assess robust reasoning capabilities across diverse domains. We report accuracy for MMLU-Pro.

Models and Baselines. We use **Qwen3-4B-Instruct-2507** (Team, 2025) for all main experiments. To verify the generalizability across different LLMs, we experiment with **Llama-3.1-8B-Instruct** (Grattafiori et al., 2024) and **Qwen3-32B** (Team, 2025) in Appendix A.1. Single-agent baselines are direct generation (**Vanilla**) and **Self-Consistency (CoT-SC)** (Wei et al., 2022; Kojima et al., 2022), which samples multiple reasoning paths and selects an answer by majority vote. Context management baselines include **Compaction** method (Wang et al., 2025a) which periodically summarizes the accumulated history, and two pruning-based methods, **AgentPrune** (Zhang et al., 2025a), and **AgentDropout** (Wang et al., 2025b), which are pruning-based methods that limit context growth by removing redundant agents or communication links.

Implementation Details. Following Wang et al. (2025b); Zhang et al. (2025a), we choose the random communication topology with GPTswarm (Zhuge et al., 2024) as the default multi-agent system setting, as its non-uniform structural distances provide a direct test of spatial decay. Further, we conduct the experiment under different topologies, including layered and fully connected topologies, in Section 4.2. For the CoT-SC baseline, we sample $n = 8$ and use majority voting to determine the final answer. More details are provided in Appendix D.

4.2 Main Results

Comparison with Context Management Baselines. As demonstrated in Table 1, AGENT-RADAR consistently outperforms both single-agent reasoning baselines and prior context management methods across all five benchmarks. Compared with the competitive baseline AgentDropout, AGENT-RADAR improves HotpotQA by 7.64, 2Wiki by 2.36, MuSiQue by 2.66, MATH-500 by 6.20, and MMLU-Pro by 2.00. The gains are larger relative to the vanilla direct-reasoning baseline, with an average improvement of 7.61 points across the five benchmarks. These improvements are consistent across open-ended QA, mathematical reasoning, and general reasoning, demonstrating the effectiveness of AGENT-RADAR.

Table 1: Performance comparison between AGENT-RADAR and context management baselines.

Method	HotpotQA	2Wiki	MuSiQue	MATH-500	MMLU-Pro
Vanilla	72.07	71.14	35.47	80.60	62.20
CoT-SC (n=8)	73.51$_{\uparrow 1.44}$	72.03$_{\uparrow 0.89}$	36.33$_{\uparrow 0.86}$	84.80$_{\uparrow 4.20}$	64.20$_{\uparrow 2.00}$
Compaction	74.33$_{\uparrow 2.26}$	73.34$_{\uparrow 2.20}$	34.43$_{\downarrow 1.04}$	83.60$_{\uparrow 3.00}$	58.40$_{\downarrow 3.80}$
AgentPrune	75.31$_{\uparrow 3.24}$	73.03$_{\uparrow 1.89}$	37.05$_{\uparrow 1.58}$	80.80$_{\uparrow 0.20}$	63.60$_{\uparrow 1.40}$
AgentDropout	73.14$_{\uparrow 1.07}$	78.45$_{\uparrow 7.31}$	37.06$_{\uparrow 1.59}$	82.60$_{\uparrow 2.00}$	67.40$_{\uparrow 5.20}$
AGENT-RADAR	**80.78**$_{\uparrow 8.71}$	**80.81**$_{\uparrow 9.67}$	**39.72**$_{\uparrow 4.25}$	**88.80**$_{\uparrow 8.20}$	**69.40**$_{\uparrow 7.20}$

Table 2: Performance of AGENT-RADAR across different multi-agent systems.

Method	HotpotQA	2Wiki	MuSiQue	MATH-500	MMLU-Pro
GPTswarm	72.55	78.47	33.85	81.80	64.20
+AGENT-RADAR	**80.78** $_{\uparrow 8.23}$	**80.81** $_{\uparrow 2.34}$	**39.72** $_{\uparrow 5.87}$	**88.80** $_{\uparrow 7.00}$	**69.40** $_{\uparrow 5.20}$
Autogen	79.63	80.95	39.85	81.00	68.40
+AGENT-RADAR	**84.29** $_{\uparrow 4.66}$	**85.49** $_{\uparrow 4.54}$	**44.05** $_{\uparrow 4.20}$	**90.00** $_{\uparrow 9.00}$	**70.80** $_{\uparrow 2.40}$
MAD	75.32	70.25	37.64	86.40	67.60
+AGENT-RADAR	**77.38** $_{\uparrow 2.06}$	**75.67** $_{\uparrow 5.42}$	**40.88** $_{\uparrow 3.24}$	**88.00**$_{\uparrow 1.60}$	**70.00**$_{\uparrow 2.40}$

Effectiveness across Multi-agent Systems. AGENT-RADAR can be integrated into diverse multi-agent systems to improve how agents exploit communication histories. To assess its effectiveness and generalizability across different coordination and communication paradigms, we integrate AGENT-RADAR into three representative MAS: GPTSwarm (Zhuge et al., 2024), AutoGen (Wu et al., 2024), and MAD (Du et al., 2024).

Table 2 reports the results. Overall, AGENT-RADAR consistently improves performance across all three frameworks and benchmarks. AGENT-RADAR improves over GPTSwarm on MuSiQue by 5.87 F1 and produces an average gain of 5.48 F1 over the three QA benchmarks. Furthermore, when applied to the already strong AutoGen framework, AGENT-RADAR improves all five benchmarks by 2.40 to 9.00 points. For the consensus debate system, our method provides robust improvement across all evaluated benchmarks. Overall, these results indicate that AGENT-RADAR is compatible with diverse multi-agent systems and consistently enhances their performance.

Effectiveness across Different Communication Topologies. To evaluate the effectiveness of AGENT-RADAR across different topologies, we extend our experiments from the random topology in Table 1 to layered and fully connected communication graphs[4]. We select AgentDropout as the primary baseline because it demonstrates the most competitive performance among existing context management methods. As shown in Table 3, AGENT-RADAR improves MAS performance under both topologies and outperforms AgentDropout in most settings. In the layered topology, AGENT-RADAR gains 2.95 to 15.41 points over the MAS and remains strongest or comparable on all benchmarks. In the fully connected topology, it gains 3.80 to 12.77 points over the MAS. The only exception is the fully connected MMLU-Pro setting, where AgentDropout shows slightly higher performance. Overall, AGENT-RADAR is not tied to a specific communication topology but remains robust across diverse graph structures.

Robustness to Increasing Agents and Communication Rounds. Prior work has shown that multi-agent debate system performance often improves initially but then declines as the number of rounds or agents increases (Becker et al., 2026; Estornell & Liu, 2024; Lin & Hooi, 2025; Liu et al., 2024b). This degradation typically occurs because long conversations often become redundant, circular, and less task-focused, yielding low-quality feedback. As stale messages occupy more of the context window, the original task constraints receive less attention, causing agents to drift from the primary objective and may make harmful incremental revisions. Since AGENT-RADAR dynamically aligns each agent's attention toward relevant context throughout the interaction, we evaluate whether it improves robustness in MAD by varying the number of agents and debate rounds.

[4]The construction details of these graph structures are provided in Appendix D.2.

Table 3: Performance comparison between AGENT-RADAR and AgentDropout under different communication graph topology.

Topology	Method	HotpotQA	2Wiki	MuSiQue	MATH-500	MMLU-Pro
Layered	MAS	72.81	69.69	38.29	80.40	65.00
	AgentDropout	75.24	77.42	40.59	84.20	69.80
	AGENT-RADAR	**81.48**	**85.10**	**41.24**	**87.20**	**70.20**
Fully Connected	MAS	67.88	73.77	31.69	83.80	61.00
	AgentDropout	71.41	71.48	37.08	81.20	**69.80**
	AGENT-RADAR	**80.65**	**80.87**	**39.80**	**87.60**	69.40

Table 4: Multi-run comparison. We report the mean ± standard deviation over three runs. △ AgentDropout represents the paired difference between AGENT-RADAR and AgentDropout.

Method	HotpotQA	2Wiki	MuSiQue	MATH-500	MMLU-Pro
GPTSwarm	73.99 ± 1.59	79.03 ± 0.91	32.17 ± 3.77	79.73 ± 1.91	64.00 ± 1.72
AgentDropout	75.08 ± 3.98	80.17 ± 2.03	36.08 ± 3.27	81.67 ± 2.07	68.07 ± 2.65
AGENT-RADAR	**81.37 ± 2.61**	**82.32 ± 1.28**	**40.12 ± 2.57**	**88.13 ± 2.65**	**71.67 ± 2.29**
△ AgentDropout	6.29 ± 1.50	2.15 ± 0.93	4.04 ± 1.17	6.47 ± 0.68	3.60 ± 1.50

Figure 2 shows the mean F1 score and standard deviation over three runs. As the number of agents increases, both vanilla MAD and AGENT-RADAR exhibit the trend of initial improvement followed by a slight decline. This trend shows that more agents can bring useful collaboration at first, while too many agents also introduce noise or distracting messages. AGENT-RADAR shows a more stable scaling pattern. Vanilla MAD reaches a comparable F1 score at 3 and 5 agents and then drops as more agents are added. In contrast, AGENT-RADAR peaks at 5 agents and remains strong with more agents, and maintains a clear advantage across all agent configurations. When scaling the number of communication rounds, both methods improve at first and then become stable. We also observe slight variance overlap when setting rounds to 5 and 7. Nevertheless, these results indicate that AGENT-RADAR improves the robustness of MAS under larger agent populations and longer multi-round communication.

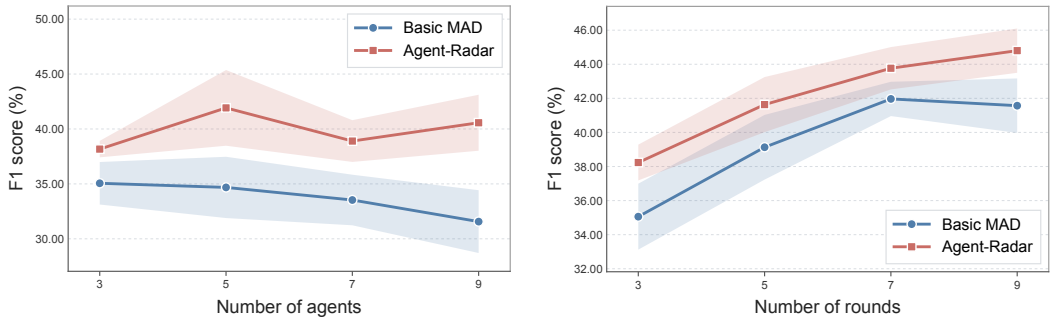

Figure 2: Performance comparison of basic MAD and AGENT-RADAR on MuSiQue. Left: Impact of varying the number of agents. Right: Impact of varying the number of rounds.

4.3 STABILITY ACROSS MULTIPLE RUNS

Given the inherent randomness of multi-agent inference, we evaluate the stability of our method across three runs. We compare AGENT-RADAR against AgentDropout, the strongest baseline in Table 1, and a weak baseline, GPTSwarm. Table 4 reports the mean and standard deviation of each method. AGENT-RADAR achieves 40.12 ± 2.57 on MuSiQue, 88.13 ± 2.65 on MATH-500, and 71.67 ± 2.29 on MMLU-Pro. It outperforms AgentDropout by 2.15 to 6.47 points across all five benchmarks. Furthermore, Table 4 reports paired differences between AGENT-RADAR and

AgentDropout with standard deviations ranging from 0.68 to 1.50. Overall, these results indicate that with multiple runs, AGENT-RADAR remains stable and outperforms the competitive baseline.

4.4 ABLATION STUDIES

In this section, we ablate the key design choices in AGENT-RADAR on three representative benchmarks: MuSiQue, MATH-500, and MMLU-Pro. Specifically, we examine AGENT-RADAR by replacing the retrieval component and the attention steering component. Then, we isolate the contributions of spatial and temporal decay by removing them individually or jointly. Lastly, we compare two variants in Appendix A.2 to test whether the gains come from emphasizing the current query or from selectively steering historical context.

Different Retrieval Methods. To validate the retriever in AGENT-RADAR, we compare it with three alternatives: random steering, BM25, and the Qwen3-Reranker (Zhang et al., 2025c). Random steering samples the same number of context sentences identified by the default retriever for each inference instance, thereby controlling for the amount of retrieved context. BM25 is a sparse lexical retrieval alternative, whereas Qwen3-Reranker ranks candidate sentences with an LLM-based reranker and retains the same number of sentences as the default retriever.

As shown in Table 5, the default AGENT-RADAR achieves the best or comparable performance on all three benchmarks, reaching 39.72 on MuSiQue, 88.80 on MATH-500, and 69.40 on MMLU-Pro. Random steering shows the weakest results among the retrieval variants, showing that the improvement depends on identifying relevant evidence rather than steering attention to random sentences. BM25 shows competitive performance on individual tasks, but does not match the default retriever across all three benchmarks. Qwen3-Reranker performs on par with the default retriever, while scoring every candidate sentence requires an LLM forward pass. Consequently, we adopt the default dense retriever for identifying relevant evidence in MAS communication histories.

Table 5: Ablation results. The upper rows show the results of the AgentDropout and three alternative retrievers. The lower rows ablate the spatial and temporal components of AGENT-RADAR.

Method	MuSiQue	MATH-500	MMLU-Pro
AgentDropout	37.06	82.60	67.40
Random steering	29.32	80.60	64.40
BM25	36.76	85.20	68.60
Qwen3-Reranker	37.19	88.20	69.20
Prompt-only retrieval	35.27	82.80	64.20
(w/o) Spatial decay	36.14	84.40	66.20
(w/o) Temporal decay	35.98	84.60	67.80
Semantic-only retrieval	35.01	84.00	65.80
AGENT-RADAR	**39.72**	**88.80**	**69.40**

Attention Steering versus Prompt-only Retrieval. To isolate the contribution of attention steering, we compare AGENT-RADAR with a *prompt-only retrieval* variant that appends the retrieved sentences to the prompt without attention steering. Its scores drop to 35.27 on MuSiQue, 82.80 on MATH-500, and 64.20 on MMLU-Pro, showing that semantic retrieval alone is insufficient.

Spatial and Temporal Decay. Ablating **spatial** decay consistently degrades performance. The score drops from 39.72 to 36.14 on MuSiQue, from 88.80 to 84.40 on MATH-500, and from 69.40 to 66.20 on MMLU-Pro. Without spatial decay, messages from distant agents may receive similar weight as those from closely collaborating neighbors, potentially steering attention toward indirect or less relevant context. Ablating the **temporal** decay shows a similar pattern. The score drops from 39.72 to 35.98 on MuSiQue, from 88.80 to 84.60 on MATH-500, and from 69.40 to 67.80 on MMLU-Pro. Without temporal decay, earlier messages may receive similar importance to more recent and relevant evidence, making the model more vulnerable to distraction from outdated context.

Ablating both decays produces a **semantic-only retrieval** variant. It applies the default dense retriever and steering backend, but treats the communication history as flat text by discarding spatial and temporal information. This variant also degrades performance compared to the full AGENT-RADAR model. By incorporating spatial and temporal signals, AGENT-RADAR better distinguishes relevant evidence from merely similar but less useful historical messages.

4.5 Efficiency and Token Cost

We conduct an end-to-end comparison to evaluate the runtime cost. Appendix A.6 reports the detailed latency and token cost. Compared with the standard multi-agent system, AGENT-RADAR reduces prompt tokens from 11.2M to 11.0M, and lowers completion tokens from 1.72M to 1.64M. It increases end-to-end latency from 968.8 to 1196.5 seconds per task for identifying relevant context retrieval and attention steering. In contrast, Compaction and AgentDropout incur larger latency due to extra generation passes, pruning decisions, or topology updates. Overall, AGENT-RADAR incurs a modest computational overhead compared with baselines, while delivering substantial improvements in reasoning performance.

5 Related Work

Context Management in Multi-agent Systems. LLM-based multi-agent systems have demonstrated significant progress in decomposing and resolving complex tasks through collaborative planning, role specialization, and iterative discussion (Guo et al., 2024; Wu et al., 2024; Li et al., 2023; Tian & Zhang, 2026; Tran et al., 2025). Despite their advantages, agent communication rapidly accumulates into long contexts, burying the original task instruction and other salient evidence in the middle of the interaction history, which can degrade reasoning quality and induce errors (Liu et al., 2024a; Cemri et al., 2025; Zhang et al., 2025b). Recent work addresses this issue mainly from two directions. First, a line of research improves communication efficiency by sparsifying the communication topology (Li et al., 2024; Shen et al., 2025; Liu et al., 2024c; Zhang et al., 2024a). Specifically, AgentPrune (Zhang et al., 2025a) and AgentDropout (Wang et al., 2025b) explicitly reduce context redundancy by pruning messages or eliminating redundant agents during multi-round collaboration, thereby lowering token cost while preserving or even improving task performance. Second, recent context learning approaches manage context by dynamically updating each agent's instruction during interaction. In particular, Wu et al. (2025) and Chen et al. (2025a) compress or filter the accumulated context history to reduce context length, while M2CL (Hua et al., 2026) focuses on multi-agent debate systems and trains a context generator to produce round-specific context prompts. In contrast, AGENT-RADAR is a training-free and MAS-agnostic method that preserves the original multi-agent workflow and improves how each agent attends to salient information within the accumulated context, enabling more effective use of long interaction histories.

Attention Steering. Compared to the aforementioned context management methods, attention steering methods directly guide LLMs during inference toward the user-specified part of the context, without shortening or rewriting the input. Selective Prompt Anchoring (SPA) (Tian & Zhang, 2025) adjusts the logit distribution to emphasize the specified context. SEKA (Li et al., 2026) learns a universal "relevance subspace" for a given task and constructs a projection matrix that amplifies the relevant features. PASTA (Zhang et al., 2024b) identifies and reweights a subset of attention heads to redirect the model's attention to user-specified parts. Selective Self-Attention (SSA) (Zhang et al., 2024c) augments the softmax nonlinearity with a principled temperature scaling strategy. TOAST (Shi et al., 2023) learns feature selection modules that guide attention toward task-relevant information. However, these methods require manual specification of selected content, limiting their adaptability to diverse reasoning contexts, especially in multi-agent systems where context evolves autonomously. AGENT-RADAR automatically selects relevant context from an evolving multi-agent history by combining semantic relevance with the graph distance from the source agent to the receiving agent and the age of each message. It recomputes these targets at each agent activation and steers each agent's attention toward that context during inference.

6 Conclusion

We presented AGENT-RADAR, an attention steering method that leverages the inherent structure of multi-agent communication to dynamically guide agents' attention toward relevant context. Across five benchmarks, AGENT-RADAR outperforms state-of-the-art methods, yielding average gains of 4.17 absolute points over the strongest baseline. Our analysis further shows that AGENT-RADAR remains effective across various communication topologies and when the system scales to accommodate more agents and interaction rounds.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on research methodology or experiments. We have not used generative AI tools for formulating mathematical claims, helping develop theoretical models or conceptual frameworks, providing critical ingredients for proving mathematical claims, supporting qualitative and thematic data analysis, interpret results, implementing methods, assisting in the writing of proofs, generating synthetic data sets, proposing or refine hypotheses, clean and reformat dataset, and assist with translation are not applicable to this work. Additionally, we used generative AI tools for searching for information, and editing a research paper to improve readability. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Sections 2 and 3 provide the formulation of AGENT-RADAR, and Appendices D and A.4 provide the implementation details and hyperparameter settings. We will release our code upon acceptance.

REFERENCES

Suvojit Acharjee, Utathya Aich, Diptarka Mandal, and Asfak Ali. Who remembers what? tracing information fidelity in human-ai chains. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2718–2726, 2025.

Thomas J Allen. Managing the flow of technology: Technology transfer and the dissemination of technological information within the r&d organization. *MIT Press Books*, 1, 1984.

Jonas Becker, Lars Benedikt Kaesberg, Andreas Stephan, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Stay focused: Problem drift in multi-agent debate. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 5068–5102, 2026.

Stephen P Borgatti, Ajay Mehra, Daniel J Brass, and Giuseppe Labianca. Network analysis in the social sciences. *science*, 323(5916):892–895, 2009.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. Why do multi-agent llm systems fail? *NeurIPS*, 2025.

Junzhi Chen, Juhao Liang, and Benyou Wang. Smurfs: Multi-agent system using context-efficient dfsdt for tool planning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3281–3298, 2025a.

Weize Chen, Jiarui Yuan, Chen Qian, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Optima: Optimizing effectiveness and efficiency for llm-based multi-agent system. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11534–11557, 2025b.

Yuhui Ding, Antonio Orvieto, Bobby He, and Thomas Hofmann. Recurrent distance filtering for graph representation learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*, 2024.

Andrew Estornell and Yang Liu. Multi-llm debate: Framework, principals, and interventions. *Advances in Neural Information Processing Systems*, 37:28938–28964, 2024.

Erwin Oliver Finkenbinder. The curve of forgetting. *The American Journal of Psychology*, 24(1): 8–32, 1913.

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

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. URL https://www.aclweb.org/anthology/2020.coling-main.580.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Xingyuan Hua, Sheng Yue, Xinyi Li, Yizhe Zhao, Jinrui Zhang, and Ju Ren. Context learning for multi-agent discussion. In *International Conference on Learning Representations (ICLR)*, 2026.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.389. URL https://aclanthology.org/2024.naacl-long.389/.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008, 2023.

Weixian Waylon Li, Yuchen Niu, Yongxin Yang, Keshuang Li, Tiejun Ma, and Shay B Cohen. Spectral attention steering for prompt highlighting. In *International Conference on Learning Representations (ICLR)*, 2026.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving multi-agent debate with sparse communication topology. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.427. URL https://aclanthology.org/2024.findings-emnlp.427/.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 17889–17904, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Zijie Lin and Bryan Hooi. Enhancing multi-agent debate system performance via confidence expression. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6453–6471, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.343. URL https://aclanthology.org/2025.findings-emnlp.343/.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024a. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Tongxuan Liu, Xingyu Wang, Weizhe Huang, Wenjiang Xu, Yuting Zeng, Lei Jiang, Hailong Yang, and Jing Li. Groupdebate: Enhancing the efficiency of multi-agent debate using group discussion. *arXiv preprint arXiv:2409.14051*, 2024b.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration. In *First Conference on Language Modeling*, 2024c.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. Mega: Moving average equipped gated attention. In *International Conference on Learning Representations*, 2023.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Hongbin Pei, Yu Li, Huiqi Deng, Jingxin Hai, Pinghui Wang, Jie Ma, Jing Tao, Yuheng Xiong, and Xiaohong Guan. Multi-track message passing: Tackling oversmoothing and oversquashing in graph learning via preventing heterophily mixing. In *Forty-first International Conference on Machine Learning*, 2024.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186, 2024.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, et al. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations (ICLR)*, 2025.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Xu Shen, Yixin Liu, Yiwei Dai, Yili Wang, Rui Miao, Yue Tan, Shirui Pan, and Xin Wang. Understanding the information propagation effects of communication topologies in LLM-based multi-agent systems. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12347–12361, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.623. URL https://aclanthology.org/2025.emnlp-main.623/.

Baifeng Shi, Siyu Gai, Trevor Darrell, and Xin Wang. Toast: Transfer learning via attention steering. *arXiv preprint arXiv:2305.15542*, 2023.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*, 2023.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuan Tian and Tianyi Zhang. Selective prompt anchoring for code generation. In *International Conference on Machine Learning*, pp. 59528–59551. PMLR, 2025.

Yuan Tian and Tianyi Zhang. Pv-sql: Synergizing database probing and rule-based verification for text-to-sql agents, 2026. URL https://arxiv.org/abs/2604.17653.

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

Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D Nguyen. Multi-agent collaboration mechanisms: A survey of llms. *arXiv preprint arXiv:2501.06322*, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. ♫ MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475. URL https://aclanthology.org/2022.tacl-1.31/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639:130193, 2025a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024a.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. Unleashing the emergent 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: Human Language Technologies (Volume 1: Long Papers)*, pp. 257–279, 2024b.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. Agent-Dropout: Dynamic agent elimination for token-efficient and high-performance LLM-based multi-agent collaboration. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24013–24035, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1170. URL https://aclanthology.org/2025.acl-long.1170/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, et al. Resum: Unlocking long-horizon search intelligence via context summarization. *arXiv preprint arXiv:2509.13313*, 2025.

Fan Yang, Yuan Tian, and Jiansong Zhang. Supporting construction worker well-being with a multi-agent conversational ai system, 2025. URL https://arxiv.org/abs/2506.07997.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

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

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

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. Tell your model where to attend: Post-hoc attention steering for LLMs. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=xZDW0oejD.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. *arXiv preprint arXiv:2505.00212*, 2025b.

Xuechen Zhang, Xiangyu Chang, Mingchen Li, Amit Roy-Chowdhury, Jiasi Chen, and Samet Oymak. Selective attention: Enhancing transformer through principled context control. *Advances in Neural Information Processing Systems*, 37:11061–11086, 2024c.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025c.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*, 2024.

A Additional Experiments

A.1 Robustness across Different LLM Backbones

We further investigate the robustness of AGENT-RADAR across varying LLMs and scales. To isolate the effect of the base model, we adopt MAD as the multi-agent framework and vary only the underlying LLMs. Within our computational budget, we experiment with models up to 32B. Table 6 presents results on Qwen3-4B, Llama3.1-8B and Qwen3-32B. AGENT-RADAR consistently improves over the debate baseline across all benchmarks and base models, indicating that the benefits of AGENT-RADAR are stable across diverse LLM backbones.

Table 6: Evaluation results on different LLMs.

Method	MuSiQue	MATH-500	MMLU-Pro
Base Model: Qwen3-4B			
MAD	37.64	86.40	67.60
+AGENT-RADAR	**40.88**	**88.00**	**70.00**
Base Model: Llama3.1-8B			
MAD	29.77	51.60	47.20
+AGENT-RADAR	**36.86**	**57.60**	**50.40**
Base Model: Qwen3-32B			
MAD	47.49	86.80	73.00
+AGENT-RADAR	**50.98**	**88.20**	**74.80**

Because AGENT-RADAR steers attention through the model's internal signals, it applies to any model that exposes them. We expect the method to stay useful on stronger models and larger context windows, because a larger context window does not remove attention dilution (Liu et al., 2024a; Hsieh et al., 2024). Multi-agent communication accumulates rapidly with the number of agents and rounds, so critical evidence is still buried in the middle of long histories. Our scaling study in Section 4.2 also shows that AGENT-RADAR retains its advantage as the number of agents and communication rounds increases.

A.2 Query-only and Full-history

We compare AGENT-RADAR with two steered context ablations. *Query-only* applies attention steering only to the current query, while *full-history* applies attention steering to the current query and every sentence in the communication history. All variants retain the complete transcript and use the same attention-steering method. They differ only in which sentences receive attention steering.

Table 7: Comparison of query-only, full-history, and AGENT-RADAR.

Method	MuSiQue	MATH-500	MMLU-Pro
Query only	33.79	84.00	65.60
Full history	35.14	82.20	66.40
AGENT-RADAR	**39.72**	**88.80**	**69.40**

Query-only. A core hypothesis of AGENT-RADAR is that attention should be steered toward relevant evidence selected from the multi-agent communication history, whereas steering only to current query is insufficient. Consistent with this intuition, *query-only* underperforms AGENT-RADAR on all three benchmarks.

Full-history. *Full-history* also underperforms AGENT-RADAR across all three benchmarks. Applying attention steering to every historical sentence gives the same emphasis to relevant and irrelevant messages, making it harder for the model to focus on the evidence needed for the current query. This result highlights the importance of selectively steering attention to relevant historical evidence rather than treating the entire communication history equally.

A.3 Different Decay Forms.

The exponential decay in AGENT-RADAR follows how information propagates through multi step communication. Recent work shows that communication topology governs how both useful and erroneous information propagates among LLM agents (Shen et al., 2025). Related studies of graph based message passing also find that information from distant nodes becomes increasingly difficult to preserve across multiple hops because of information loss and propagation bottlenecks (Pei et al.,

2024; Ding et al., 2024). We therefore model each additional communication hop or round as retaining a fixed fraction λ of the preceding signal. To test this design choice, we replace the exponential form with three alternatives. Linear decay applies a fixed absolute reduction per step and eventually reaches a cutoff. Logarithmic and power-law decay attenuate increasingly slowly and assign relatively large weights to distant or outdated messages.

Table 8 reports the results. The exponential form performs best on all three benchmarks, exceeding the strongest alternative by 2.41 points on MuSiQue, 3.20 points on MATH-500, and 1.40 points on MMLU-Pro. The three alternatives performance varies. These results show that the exponential form matches the propagation behavior of MAS communication better than a fixed absolute reduction or a slowly flattening schedule.

Table 8: Comparison of decay forms in AGENT-RADAR. Each variant replaces the exponential decay while keeping all other components unchanged.

Decay form	MuSiQue	MATH-500	MMLU-Pro
Linear	37.31	84.80	66.80
Logarithmic	36.75	85.60	67.20
Power-law	35.73	84.20	68.00
Exponential (ours)	**39.72**	**88.80**	**69.40**

A.4 HYPERPARAMETER TUNING

We tune all hyperparameters on a held-out MuSiQue validation split and apply the selected values unchanged to all other benchmarks. This setup allows us to test whether hyperparameters chosen on one benchmark generalize to other benchmarks. To measure how context-selection threshold, spatial, and temporal decay factors affect MAS performance, we vary θ, λ_s, and λ_t one at a time while holding the other hyperparameters fixed.

Table 9: Hyperparameter sensitivity on MuSiQue.

Spatial decay λ_s		Temporal decay λ_t		Threshold θ	
Value	F1	Value	F1	Value	F1
0.88	37.04	0.88	38.42	0.30	35.78
0.90	39.20	0.90	39.47	0.40	38.78
0.92	**39.58**	**0.92**	**39.58**	**0.50**	**39.58**
0.94	38.52	0.94	39.29	0.60	38.21
0.96	36.82	0.96	38.73	0.70	37.96

Table 9 illustrates the results. For the decay factors, $\lambda_s = \lambda_t = 0.92$ performs best among the tested values. Smaller values overly suppress indirect or earlier messages and can remove useful evidence. Larger values make the filter less selective and allow more distant or stale messages to influence generation. Even in suboptimal decay settings, AGENT-RADAR still outperforms or is on par with the baseline on average, indicating that the spatio-temporal decay formulation is robust to the choice of decay factors.

Table 10: Hyperparameter sensitivity on MATH-500, and MMLU-Pro.

Hyperparameter	Value	MATH-500	MMLU-Pro
Threshold θ	0.30	85.80	67.60
	0.50	**88.80**	69.40
	0.70	86.40	**70.20**
Spatial λ_s	0.88	86.20	67.60
	0.92	**88.80**	**69.40**
	0.96	87.00	68.20
Temporal λ_t	0.88	86.60	67.80
	0.92	**88.80**	**69.40**
	0.96	86.00	68.20

The context-selection threshold shows a similar pattern. $\theta = 0.50$ achieves the best or on-par performance among the tested values. A smaller threshold retains more candidate context, which may introduce noise, while a larger threshold filters context more aggressively and may discard useful evidence.

To check whether this stability holds beyond MuSiQue, we extend the sensitivity analysis to MATH-500 and MMLU-Pro. Table 10 reports accuracy under three values of the threshold θ, the spatial decay λ_s, and the temporal decay λ_t, with each varied in isolation. The selected hyperparameters transfer well on both benchmarks, and the nearby values yield comparable performance. These results indicate that AGENT-RADAR remains relatively robust within the evaluated parameter ranges.

Table 11: Efficiency comparison. Ptok. denotes the number of prompt tokens for the agents, while Ctok. represents the number of completion tokens generated by the agents.

Method	Latency (Second/Task) ↓	Ptok.	Ctok.
MAS	968.8	11.2M	1.72M
Compaction	1757.1 (+788.3)	8.66M (-2.54M)	1.63M (-0.09M)
AgentPrune	1089.2 (+120.4)	9.9M (-1.3M)	2.09M (+0.37M)
AgentDropout	1382.4 (+413.6)	8.23M (-2.97M)	1.33M (-0.39M)
AGENT-RADAR	1196.5 (+227.7)	11.0M (-0.2M)	1.64M (-0.08M)

A.5 DIFFERENT ATTENTION STEERING METHODS

AGENT-RADAR is orthogonal to the choice of attention steering method. To verify this, we replace the default attention steering method (SPA) with PASTA (Zhang et al., 2024b) following its original experimental setup. We conduct experiments with Llama-7B (Touvron et al., 2023), since PASTA currently supports only GPT-2 and Llama-7B backbones, with Llama as its default setting. On MuSiQue, the score increases from 10.41 to 15.32. On MATH-500, it improves from 9.60 to 12.20. On MMLU-Pro, it improves from 12.40 to 15.20. The relative gains range from 10.20% to 47.20%, compared with 10.20% to 13.60% for SPA in Table 1. These gains show that AGENT-RADAR does not depend on a specific steering implementation.

A.6 EFFICIENCY AND TOKEN COST

We measure efficiency using latency per task and total token cost on MusiQue. As shown in Table 11, AGENT-RADAR increases latency from 968.8 to 1196.5 seconds per task, an overhead of 227.7 seconds over the original MAS baseline. Its token cost remains comparable to the MAS baseline, the prompt tokens reduce from 11.2M to 11.0M, while its completion tokens decrease from 1.72M to 1.64M.

Compared with the context management baselines, AGENT-RADAR is faster than Compaction and AgentDropout but slower than AgentPrune. The three baselines use fewer tokens because they compress the history or remove parts of the communication structure. Compaction has a relatively low token cost but the highest latency because it uses an additional generation pass to summarize the dialogue history. AgentPrune and AgentDropout reduce token cost, but their agent and edge selection adds latency. In comparison, AGENT-RADAR preserves the original topology and complete transcript. Its latency overhead comes from sentence-level scoring and attention steering.

Finally, because AGENT-RADAR preserves the full MAS transcript, it avoids information loss caused by pruning or compression, but it is not primarily designed as a token-reduction method. Though AGENT-RADAR did not exceed the model's context window during evaluations, deploying to tasks with extremely long interaction histories may exceed the maximum context window of the underlying LLM. Future work can adopt widely used context-compaction methods to summarize earlier interactions, or selectively retain the most relevant information in the history, such as the context retrieved by AGENT-RADAR.

A.7 FAILURE CASE ANALYSIS

We manually analyze 100 randomly selected failure cases of AGENT-RADAR across MuSiQue, MATH-500, and MMLU-Pro. Most cases fail in both AGENT-RADAR and the vanilla MAS baseline, suggesting that they mainly reflect limitations of the underlying MAS or base model rather than retrieval failure. Specifically, we found 43 cases involving task misunderstanding. For example, some agents made an incorrect assumption about the problem at an early stage. Another 38 cases involved insufficient information exchange. For example, some agents failed to provide enough intermediate reasoning or relevant evidence for other agents to reach the correct conclusion. We also identified 13 hallucination cases, where an agent initially followed a correct reasoning path but later introduced an unsupported step. Problematic retrieved context accounts for only 6 cases. In these rare cases, incorrect reasoning from one agent is retrieved and subsequently propagated through the discussion. We leave the use of agent confidence or cross-agent agreement to filter such errors for future work.

B LIMITATION

Despite the consistent improvements observed in our experiments, AGENT-RADAR has several limitations. First, AGENT-RADAR requires selecting a threshold and decay factors. Our parameter analysis shows that performance is relatively robust to these choices. Future work could explore adaptive thresholds and decay factors that adjust automatically to the communication topology, context length, or model uncertainty.

Second, retrieval quality can affect the performance of AGENT-RADAR. When incorrect reasoning from one agent enters the retrieved context, attention steering can propagate it through the discussion. Our failure case analysis shows that this failure mode accounts for only 6 of 100 examined cases. Stronger retrieval models, uncertainty-aware scoring, or explicit contradiction detection may further reduce this risk.

Lastly, AGENT-RADAR requires attention-steering backends such as SPA or PASTA. Our experiments show that the method is not tied to a single backend, but applying it to closed-source LLMs may be difficult when internal model states are unavailable.

C RETRIEVED CONTEXT STATISTICS

We examine the sentences retrieved for attention steering in 100 test cases from MuSiQue, MATH-500, and MMLU-Pro under random topologies.

We first group the retrieved sentences by their age $a = t - \tau - 1$. Among the retrieved sentences, 35% have $a = 0$, 27% have $a = 1$, 25% have $a = 2$, 13% have $a \geq 3$. Thus, most of the retrieved sentences come from the three most recent rounds. Older sentences remain in the transcript and can still be used by the model, even when they are not retrieved for attention steering.

We then group the retrieved sentences by their distance from the receiving agent. We find that 40% come from the receiving agent or a direct neighbor, 29% are two hops away, 31% are at least three hops away.

These results align with the motivation of AGENT-RADAR. Most retrieved contexts are recent and come from nearby agents. Older or more distant sentences are retrieved only when they are semantically relevant to the current query. This design does not remove earlier evidence. Earlier useful information is often restated in recent messages, and AGENT-RADAR directly steers attention to those restatements. Unselected sentences also remain in the prompt and can still receive attention. Finally, analysis in A.4 shows that lowering the threshold to steer attention to more sentences will degrade performance.

D EXPERIMENT DETAILS

We set the decoding temperature to 0.7 for all experiments. For the open-ended QA benchmarks, we follow the benchmark setting and provide all methods with the candidate context documents included in each HotpotQA, 2WikiMultihopQA, and MuSiQue instance. For efficiency, all local models are run with 8-bit quantized inference on NVIDIA A5500 GPUs. For SPA, we follow the original Selective Prompt Anchoring setup (Tian & Zhang, 2025) and use its default steering strength of 3.0 in all experiments. For HotpotQA, 2WikiMultihopQA, and MuSiQue, Following Trivedi et al. (2023); Jeong et al. (2024), we sample and evaluate 500 questions from the development set of each dataset.

Following the multi-agent system configuration in Wang et al. (2025b), the final answer is produced by a `FinalRefer` decision node, which receives the task together with the outputs from the connected agents and is instructed to return only the answer. We evaluate several communication topologies, including *Fully connected graphs*, *Random graphs*, *Layered graphs*, and *LLM-Debate*. We detail each configuration below. For AgentPrune and AgentDropout, we follow their official implementations and default configurations.

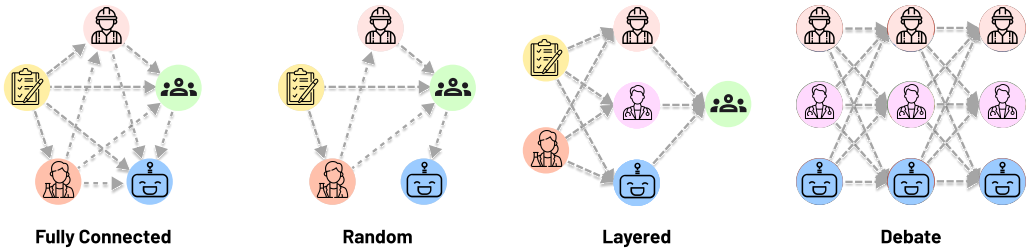

Figure 3: MAS communication topologies in fully connected, random, layered, and debate topologies.

D.1 Ablation and Analysis Settings

Unless otherwise stated, all ablation and analysis experiments in Section 4.4 follow the main evaluation protocol on Random topology. We use Qwen3-4B-Instruct-2507 on MuSiQue, MATH-500, and MMLU-Pro, varying one component at a time while keeping all other settings fixed.

In *Different Retrieval Methods*, we replace the default retriever with random selection, BM25, or Qwen3-Reranker. Random selection and Qwen3-Reranker retain the same number of sentences as the default retriever for each instance. In *Attention Steering versus Prompt-only Retrieval*, the prompt-only retrieval variant appends the retrieved sentences to the prompt without applying attention steering. For the decay ablations, removing either decay term may select more sentences that receive attention steering. To isolate each decay contribution, we match the steering budget to the full AGENT-RADAR method. If AGENT-RADAR steers k historical sentences, each ablation steers its top-k sentences under the ablated scoring function.

D.2 Different Communication Topologies

In the main experiments, we initialize the communication graph as a random graph. In the analysis experiments, we further evaluate layered graphs, fully connected graphs, and LLM-Debate. In fully connected graphs and LLM-Debate, all non-self messages are one hop from the receiving agent, so the spatial factor is constant and does not affect retrieval. Therefore, these settings evaluate whether AGENT-RADAR remains effective under dense communication. The contribution of spatial decay is evaluated under the default random topology, where spatial distances vary. Figure 3 demonstrates all MAS communication topologies.

Random Graph A random graph is constructed by sparsely sampling edges from a complete graph. We follow AgentPrune (Zhang et al., 2025a) for the random-topology setup to ensure a fair comparison with pruning-based MAS baselines. Prior work has shown that irregular sparse communication structures can outperform fully connected ones (Qian et al., 2025), partly because random shortcut connections reduce the average path length between agents and induce small-world properties. Intuitively, such connections allow information to propagate more efficiently across otherwise distant agents.

Complete Graph In the multi-agent setting, a complete graph is implemented as a directed graph that would become fully connected if converted into its undirected counterpart. The final agent aggregates the preceding discussion and produces the final answer or reflection.

Layered Graph A layered graph organizes agents into multiple stages, analogous to the layered structure of a multilayer perceptron. Agents in the first layer first receive the query and generate their responses. These responses are then passed to agents in the next layer together with the original query, and the process continues sequentially across layers. Following Qian et al. (2025), the final layer contains a single agent that integrates the information from the previous layer and outputs the final solution.

LLM-Debate LLM-Debate (Du et al., 2024) is designed as a multi-round debate process in which, at each round, every agent observes the outputs of all agents from the previous rounds before generating its own response. As a result, every agent is a direct in-neighbor of every other agent, forming a complete communication graph.

D.3 COMPRESSION IMPLEMENTATION

We implement Compaction as a prompt-based summarization baseline. Following Wang et al. (2025a), at each agent step we involve an additional LLM call to summarize the accumulated dialogue with the prompt in Table 12. The resulting summary then replaces the original dialogue history passed to the agent.

Compression Prompt
`Please summarize the following dialogue as concisely as possible, extracting the main themes and key information. If there are multiple key events, you may summarize them separately. Dialogue content:`

Table 12: Prompt used by context compression baseline.

D.4 AGENT PROFILING.

Previous work (Wang et al., 2024b; Yang et al., 2025) have shown that assigning diverse roles or personas to LLM-based agents can improve collaboration and cognitive synergy within a multi-agent system.

Profile Examples For Question Answering Reasoning Below Table 13 are some examples of agent profiles tailored for question answering reasoning tasks:

Table 13: Agent profiles prompt for question answer reasoning

Knowledge Expert
`You are a knowledgeable expert in question answering. Please give several key entities that need to be searched in Wikipedia to solve the problem. Key entities that need to be searched are included between two '@' when output, for example: @catfish effect@, @broken window effect@, @Shakespeare@. If there is no entity in question that needs to be searched in Wikipedia, you don't have to provide it`
Wiki Searcher
`You will be given a question and a Wikipedia overview of the key entities within it. Please refer to them step by step to give your answer. And point out potential issues in other agents' analysis.`
Critic
`You are an excellent critic. Please point out potential issues in the other agent's analysis point by point.`
Mathematician
`You are a mathematician who is good at math games, arithmetic calculation, and long-term planning.`
Psychologist
`You are a psychologist. You are good at psychology, sociology, and philosophy. You give people scientific suggestions that will make them feel better.`
Historian

You research and analyze cultural, economic, political, and social events in the past, collect data from primary sources, and use it to develop theories about what happened during various periods of history.
Doctor
You are a doctor and come up with creative treatments for illnesses or diseases. You are able to recommend conventional medicines, herbal remedies and other natural alternatives. You also consider the patient's age, lifestyle and medical history when providing your recommendations.
Lawyer
You are good at law, politics, and history.
Economist
You are good at economics, finance, and business. You have experience on understanding charts while interpreting the macroeconomic environment prevailing across world economies.
Programmer
You are good at computer science, engineering, and physics. You have experience in designing and developing computer software and hardware.

Profile Examples For Mathematical Reasoning Below Table 14 are examples of agent profiles tailored for mathematical reasoning tasks:

Table 14: Agent profiles prompt for mathematical reasoning

Math Solver
You are a math expert. You will be given a math problem and hints from other agents. Give your own solving process step by step based on hints. The last line of your output contains only the final result without any units, for example: The answer is 140 You will be given some examples you may refer to.
Mathematical Analyst
You are a mathematical analyst. You will be given a math problem, analysis and code from other agents. You need to first analyze the problem-solving process step by step, where the variables are represented by letters. Then you substitute the values into the analysis process to perform calculations and get the results. The last line of your output contains only the final result without any units, for example: The answer is 140 You will be given some examples you may refer to.
Programming Expert
You are a programming expert. You will be given a math problem, analysis and code from other agents. Integrate step-by-step reasoning and Python code to solve math problems. Analyze the question and write functions to solve the problem. The function should not take any arguments and use the final result as the return value.

```
The last line of code calls the function you wrote and assigns
the return value to the answer variable.
Use a Python code block to write your response.  For
example:\n```python\n def fun():\n x = 10\n y = 20\n return x +
y\n answer = fun()\n```
Do not include anything other than Python code blocks in your
response." You will be given some examples you may refer to.
```

Inspector

```
You are an Inspector.  You will be given a math problem,
analysis, and code from other agents.
Check whether the logic/calculation of the problem-solving and
analysis process is correct(if present).
Check whether the code corresponds to the solution analysis(if
present).
Give your own solving process step by step based on hints.
The last line of your output contains only the final result
without any units, for example:  The answer is 140
You will be given some examples you may refer to.
```

D.5 Prompts used in LLM-based Multi-Agent Systems

Before starting the iterative debating process, LLM-based multi-agent systems faced a critical issue: agents often generate highly similar responses. This limits the exploration of diverse communication strategies. To mitigate this issue, we follow Qian et al. (2024); Chen et al. (2025b) and employ a pool of format-specification prompts in Table 15.

Table 15: Prompt used in LLM-debate multi-agent systems to elicit diverse communication in format.

```
You are a special agent who does not respond in natural
language.  You are deployed on a resource-limited device, so
you must respond concisely.  More tokens indicate a higher
possibility of killing the device you are running.  Now you
are collaborating with your partners, an agent who will
correct you when he thinks the answer is wrong.  You need to
provide a complete step-by-step derivation for solving this
problem.
{Question:}

GUIDELINES:
1.  On finding the final answer, ensure to conclude your
communication with
boxedanswer, where änswerïs the determined solution.  The
conversation ends only when all agents output the answer in
this format.
2.  Reason through the problem step-by-step.
3.  You are communicating with a very limited token budget, so
you must use a very concise communication format.  Natural
language is suitable for humans, but not for you.
Since you and your partner are both intelligent agents, use
your agent communication language.  Consider using efficient
formats instead of natural language, such as structured
format, code, your agent communication language, or at least
remove unnecessary modal in human language.  Too many tokens
will make you fail.  But still ensure your message is
informative and understandable.
```

Communication Format 1

```
For example, you can respond in matrix format as follows:
 [["Field1", "Value1"], ["Field2", "Value2"], ...]
Or you can use key-value list format:
"Field1": "Value1"; "Field2": "Value2"; ...
```

Communication Format 2

```
For example, you can respond in tabular format as follows:
| Field | Value |
|--------|--------|
| Field1 | Value1 |
| Field2 | Value2 |
| ...  | ...  |

Or you can use abbreviated notation:
F1:  V1; F2:  V2; ...
```

Communication Format 3

```
For example, you can respond in XML format as follows:
<response>
  <field1>value1</field1>
  <field2>value2</field2>
  ...
</response>

Or you can use dot notation:
field1.value1; field2.value2; ...
```

Communication Format 4

```
For example, you can use array format:
[{key1:val1}, {key2:val2}, ...]
```

Communication Format 5

```
For example, you can respond in emoji code as follows:
key1:val2; pkg3:mul4;
```

Communication Format 6

```
For example, you can respond using graph notation as follows:
(A)-->(B,distance); (B)-->(C,distance); ...

Or you can use shorthand formula format:
X + Y = Z; A - B = C; ...
```

Communication Format 7

```
For example, you can respond using programming pseudocode as
follows:
function findAnswer(data):
    return solution;
```

Communication Format 8

```
For example, you can respond using kanji characters as
follows:
Tanaka; Suzuki; ...

Or you can use symbolic glyph notation:
glyph1; glyph2; ...
```

Communication Format 9

```
For example, you can use LUT (Look-Up Table) format:
1 -> A; 2 -> B; ...
```

Communication Format 10

For example, you can respond using flowchart notation as follows:
(Start)->(Process)->(Decision: Yes/No)->(End);

Or you can use railway diagram notation:
[Begin] --> { event1 } --> (choice1 | choice2) --> [End]

Communication Format 11

For example, you can respond using color coding as follows:
Green:Success; Red:Failure; ...

Or you can use shorthand operational notation:
$\oplus$ A,B = C; $\ominus$ D,E = F; ...