CoS-Web: Suspicion-Driven Multi-Agent Reasoning with Structured Communication

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) are increasingly used for complex reasoning, but a single reasoning path can remain vulnerable to missing knowledge and internally consistent errors. LLM-based multi-agent systems (LLM-MAS) address this limitation through information exchange and collaborative decision-making. Multi-Agent Debate (MAD) lets agents critique reasoning paths and revise judgments over multiple rounds. However, agreement on an answer does not guarantee that its supporting rationale is reliable: when multiple agents share an unverified assumption, their interaction may further reinforce a false consensus. Meanwhile, existing communication optimization methods focus mainly on interaction structures, messages, or participant selection. They rarely represent unresolved issues explicitly to guide local communication and subsequent re-examination. To address this, we propose **CoS-Web (Chain-of-Suspicion with Spider-Web Communication)**, a multi-agent reasoning framework that connects suspicion-driven interaction with subsequent communication through structured unresolved issues. **Chain-of-Suspicion (CoS)** is a suspicion-driven interaction protocol that progressively examines agent judgments, their anticipated defenses, and collective reasoning risks to formulate structured unresolved issues. **Spider-Web Communication** uses these issues to organize local Ring Communication among relevant agents on top of Radial global-state sharing, and re-examines whether the issues remain unresolved after communication. Experiments and analyses across multiple reasoning benchmarks show that CoS-Web improves overall reasoning accuracy while balancing communication costs, and handles cases in which multiple agents share erroneous assumptions more robustly.

1 Introduction

Large language models (LLMs) perform well on complex reasoning, question answering, and planning, yet single-model reasoning remains vulnerable to missing knowledge and internally consistent errors. LLM-based multi-agent systems (LLM-MAS) use information exchange and diverse reasoning paths to improve reliability (Li et al., 2023; Hong et al., 2024; Wu et al., 2024). Multi-Agent Debate (MAD) extends this approach through iterative critique and answer revision, with reported gains on mathematical reasoning and factual judgment (Du et al., 2024; Liang et al., 2024).

Agreement on an answer does not guarantee reliable reasoning. Agents with similar biases or a shared unverified assumption may converge on the same erroneous rationale, and MAD does not always outperform self-consistency or multi-path ensembling (Smit et al., 2024). Collaboration must therefore examine not only answer disagreements but also unsupported judgments, justifications, and assumptions. We call content that still requires a response or additional evidence **structured unresolved issues** (hereafter, **unresolved issues**).

Communication methods reduce redundant interaction through sparse topologies, message pruning, or dynamic connections (Li et al., 2024; Zhang et al., 2025a). A disagreement signal alone cannot capture cases where agents agree but rely on unsupported assumptions. We therefore ask how unresolved issues can specify what to verify, who should participate, and whether the concern remains after interaction.

We propose **CoS-Web (Chain-of-Suspicion with Spider-Web Communication)**, which uses unresolved issues as an intermediate representation connecting reasoning verification and communica-

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.

tion routing. **Chain-of-Suspicion (CoS)**, introduced in this work, examines judgments, anticipated defenses, and group-level risks to formulate issues for verification. **Spider-Web Communication** uses each issue to select relevant agents and organize local Ring Communication; the Judge then re-examines the issue status. Radial Communication provides low-bandwidth global-state sharing.

Our main contributions are as follows:

- **We introduce CoS**, which questions the basis for judgments, tests anticipated defenses, and aggregates group-level risks into unresolved issues for subsequent verification.
- **We introduce issue-driven Spider-Web Communication**, which uses each issue to select participants and direct local Ring Communication, then re-examines the set of unresolved issues. Radial Communication supplies low-bandwidth global-state sharing.
- **We evaluate CoS-Web** across four reasoning benchmarks, measuring accuracy and robustness to shared erroneous assumptions.

2 Related Work

2.1 LLM-based Multi-Agent Systems

LLM-based multi-agent systems organize agents and message passing to solve complex tasks. Representative frameworks include role-playing in CAMEL (Li et al., 2023), structured collaboration in MetaGPT (Hong et al., 2024), and multi-agent conversation in AutoGen (Wu et al., 2024); surveys identify communication, collaboration, and scalability as central design dimensions (Guo et al., 2024).

2.2 Multi-Agent Debate and Collaborative Reasoning

Single-agent methods such as Self-Refine (Madaan et al., 2023) and Reflexion (Shinn et al., 2023) improve outputs through feedback and memory. Multi-agent debate extends critique and revision across reasoning paths (Du et al., 2024; Chan et al., 2024), but its reliability can be limited by correlated reasoning and is not guaranteed to exceed single-agent self-consistency (Smit et al., 2024). Recent work addresses these limitations through reasoning diversification, memory or error filtering, sparse interaction, and consensus-free decisions (Liang et al., 2024; Liu et al., 2025; Tian et al., 2026; Sun et al., 2025; Cui et al., 2026). Analyses of confidence and diversity further characterize when debate helps (Zhu et al., 2026). These studies improve debate behavior, whereas CoS-Web represents unresolved reasoning issues explicitly and uses them to direct verification.

2.3 Multi-Agent Communication

Communication methods adapt interactions through sparse topologies and message pruning (Li et al., 2024; Zhang et al., 2025a), learned or adaptive routing (Zhang et al., 2025b; Wang et al., 2025; Feng et al., 2026), uncertainty-aware control (Yoffe et al., 2025; Verma et al., 2026), and debate stabilization (Becker et al., 2026). These approaches adapt structures, messages, participants, or debate policies. CoS-Web instead treats structured unresolved issues as an intermediate representation connecting reasoning verification to communication routing: each issue specifies what remains to be checked and which agents should participate. Issues therefore serve not only as routing signals but also as explicit verification states that are re-examined after local communication.

3 Method

Building on the problem outlined above, we propose **CoS-Web (Chain-of-Suspicion with Spider-Web Communication)**. Its central design uses structured unresolved issues as the intermediate representation connecting reasoning verification and communication routing: CoS formulates issues, Spider-Web uses them to organize local communication, and the Judge re-examines them afterward to decide whether interaction should continue.

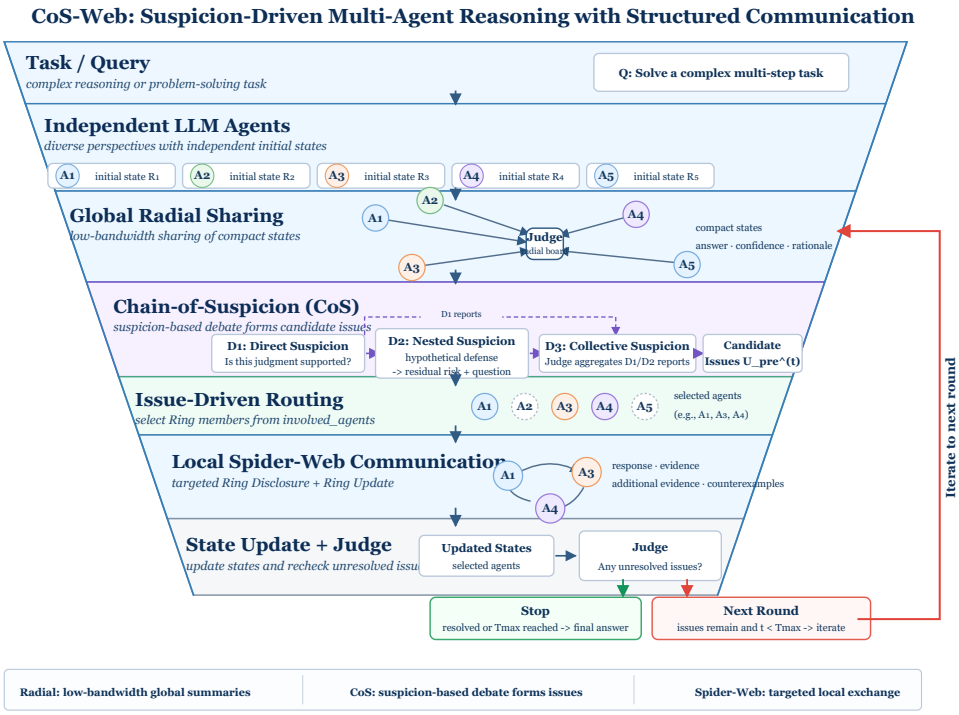


Figure 1. Overall process of CoS-Web: after Radial Communication shares low-bandwidth global state, CoS formulates structured unresolved issues. Spider-Web then uses these issues to organize local Ring Communication among relevant agents. After communication, the Judge re-examines the unresolved issues and decides whether interaction should continue.

3.1 System Setup and Overall Framework

Given a task input x, the system consists of N reasoning agents and a Judge J:

$$\mathcal{A} = \{A_1, \ldots, A_N\}.$$

Each agent first reasons independently to form an initial judgment.

CoS-Web consists of two interconnected core components:

- **Chain-of-Suspicion (CoS, a suspicion-driven interaction protocol):** progressively turns local suspicions into structured unresolved issues;
- **Spider-Web Communication:** uses these unresolved issues to organize local information exchange among relevant agents and re-examines their status after communication.

The overall process is shown in Figure 1.

3.2 Chain-of-Suspicion

Chain-of-Suspicion (CoS) is a suspicion-driven interaction protocol that progressively converts potentially unsupported reasoning into structured unresolved issues for verification. Unlike disagreement-triggered interaction, CoS does not treat agreement as evidence that the supporting reasoning is reliable: agents may converge on an answer while sharing an unsupported assumption or an incomplete justification. CoS therefore examines the reasoning basis even when answers agree.

The three stages change the unit of examination in sequence. **Direct Suspicion (D1)** identifies specific claims, assumptions, or reasoning steps that warrant scrutiny. **Nested Suspicion (D2)** deepens each agent's own D1 concerns by testing anticipated defenses and exposing residual risk. **Collective**

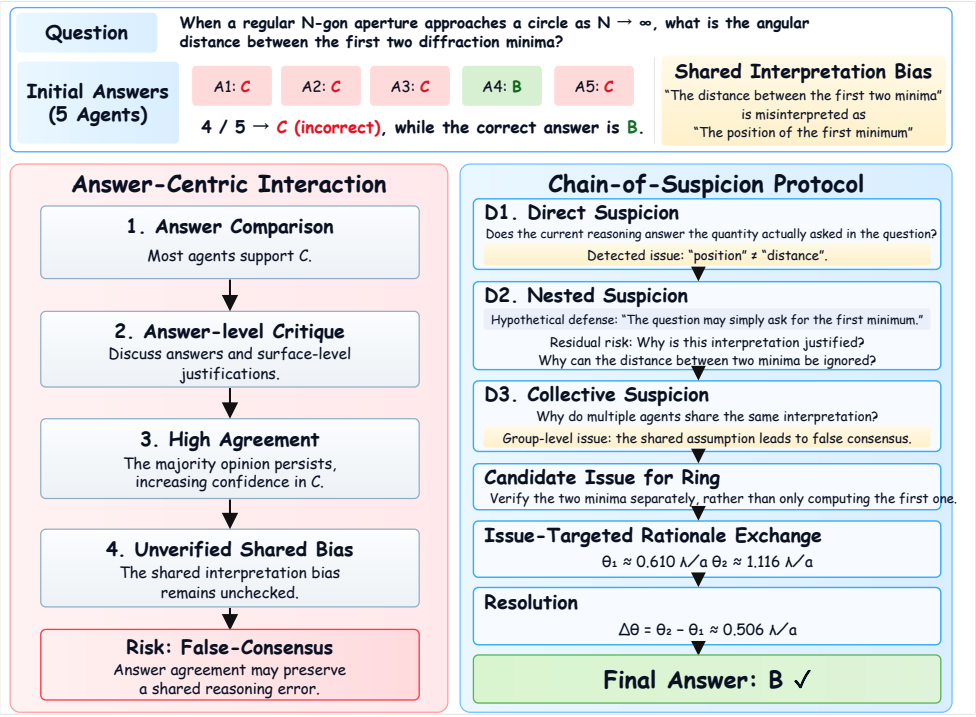


Figure 2. An example from our experiments: the left panel provides a schematic comparison with answer-centered interaction and is not intended to characterize all MAD methods. The right panel shows how agreement on an answer may conceal a shared interpretive bias. D1–D3 progressively transform local reasoning concerns into structured unresolved issues, which are subsequently examined through Ring Communication.

Suspicion (D3) aggregates the reports across agents to form group-level unresolved issues. This progression moves from individual scrutiny, to agent-specific deepening, and finally to collective aggregation.

Figure 2 illustrates how the process can expose a shared interpretive bias despite initial agreement:

Each agent maintains a public state

$$R_i^{(t)} = \left(y_i^{(t)}, r_i^{(t)}, k_i^{(t)}, a_i^{(t)}, c_i^{(t)}\right),$$

where the fields correspond to `final_answer`, `reasoning_summary`, `key_evidence`, `assumptions`, and `confidence`. Independent reasoning produces $R_i^{(0)}$. Round t begins with state $R_i^{(t)}$, with $R_i^{(1)} = R_i^{(0)}$; after all Ring updates, the state is denoted $R_i^{(t,+)}$. For every agent, $R_i^{(t+1)} = R_i^{(t,+)}$; if it does not participate in a Ring, then $R_i^{(t,+)} = R_i^{(t)}$. Thus, the post-update public states are exactly the states entering the next round.

Before CoS runs, Radial Communication gives each agent a low-bandwidth summary $S_i^{(t)}$ of every public state. The summary provides global context; full evidence and assumptions remain available in the public states for targeted verification. Appendix A.2 specifies the deterministic projection.

D1: Direct Suspicion. Direct Suspicion (D1) performs individual scrutiny of other agents' publicly visible judgments. It asks whether an answer is supported by its visible rationale, whether a key step or assumption remains unverified, and whether confidence is commensurate with the available support. Because the object of scrutiny is the reasoning basis, D1 can raise a concern even when agents give the same answer. It identifies suspicions rather than deciding whether an answer is true.

For agent A_i in round t, D1 receives its own state, the Radial board, and unresolved records carried over from the previous round:

$$D_i^{(1,t)} = \text{D1}\left(x, R_i^{(t)}, \{S_j^{(t)}\}_{j=1}^N, \mathcal{U}_{\text{post}}^{(t-1)}\right),$$

where $\mathcal{U}_{\text{post}}^{(0)} = \emptyset$. Its output is agent-specific and may be empty when no concern is supported; the structured output fields are listed in Appendix A.2.

D2: Nested Suspicion. Nested Suspicion (D2) tests whether a D1 concern would remain even if a plausible defense were offered. For each suspicion in A_i's own D1 list, D2 forms an anticipated defense, identifies the residual risk, and poses a deeper verification question. The defense is a model-generated projection, not an actual response from the target agent and not evidence that the concern has been resolved.

D2 receives only A_i's current state and that same agent's complete D1 suspicion list; it does not receive the Radial board or other agents' D1 reports:

$$D_i^{(2,t)} = \text{D2}\left(x, R_i^{(t)}, D_i^{(1,t)}\right).$$

Thus, D1-to-D2 deepening remains agent-specific. Cross-agent aggregation occurs only in D3.

D3: Collective Suspicion. Collective Suspicion (D3) combines the local reports because a group-level risk may be distributed across several agents' suspicions. It aggregates the D1/D2 reports and their cross-agent relationships. D3 identifies group-level unresolved issues that emerge from the relationships among agents' judgments, assumptions, anticipated defenses, and residual risks. Typical patterns may include shared unverified assumptions, false consensus, circular support, and collective blind spots, but these are illustrative examples.

The Judge forms the set of pre-communication unresolved issues as

$$\mathcal{U}_{\text{pre}}^{(t)} = \text{D3}\left(x, \{D_i^{(1,t)}\}_{i=1}^N, \{D_i^{(2,t)}\}_{i=1}^N\right).$$

Each issue records a concise risk label, a description of the unresolved concern, and the relevant agents, providing a shared verification target for subsequent Ring Communication. These are candidate issues, not verified conclusions; the Judge re-examines their status after communication. The output schema is given in Appendix A.2.

3.3 Spider-Web Communication

Spider-Web Communication combines global Radial sharing with issue-driven local Ring exchanges. Each issue specifies the verification target and relevant agents; the Judge re-examines its unresolved status after communication.

Radial Communication. In Radial Communication, the Judge serves as a fixed central hub that deterministically aggregates and broadcasts the compressed-state summaries; this field projection requires no additional LLM call.

In each interaction round, the agents send the Judge low-bandwidth summaries deterministically constructed from their public states. The Judge aggregates and broadcasts these summaries to all agents, giving each agent a consistent global view for subsequent CoS debate. Radial does not broadcast the full `key_evidence` or `assumptions` fields.

For N regular agents, the radial structure requires connections only between the agents and the Judge; regular agents do not need to exchange full information directly with one another.

Ring Communication. Ring membership for each issue is determined by D3's `involved_agents` field (Appendix A.2); D1's `target_agent` label is a report field only and is not used for this purpose. Issues with fewer than two valid members do not activate a Ring; issues with identical member sets are grouped. Ring members disclose evidence and counterarguments, then update their public states using neighboring disclosures. The exact ordering rules are provided in Appendix A.2.

3.4 Collaborative Update and Termination

After Ring updates, the Judge re-examines whether each issue remains reflected in the agents' updated public states. Recheck reads the pre-communication issue set and all post-communication states, but not the raw Ring disclosures:

Let $R_i^{(t,+)}$ denote agent A_i's public state after Ring Update. The Judge computes

$$\mathcal{U}_{\text{post}}^{(t)} = \text{Recheck}\left(\mathcal{U}_{\text{pre}}^{(t)}, \{R_i^{(t,+)}\}_{i=1}^{N}\right).$$

If $\mathcal{U}_{\text{post}}^{(t)} = \emptyset$, the system terminates; otherwise, the remaining issues are carried into the next round's D1. Interaction is capped at $T_{\max}$ rounds:

$$\text{Stop} \iff \mathcal{U}_{\text{post}}^{(t)} = \emptyset \quad \text{or} \quad t = T_{\max}.$$

At termination, the Judge produces the final answer from all current public states, even if issues remain unresolved at the round limit. The algorithm, output contracts, and implementation details are provided in Appendix A.2.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023), GPQA-Diamond (Rein et al., 2024), and StrategyQA (Geva et al., 2021), covering mathematical, expert-knowledge, and open-domain multi-hop reasoning.

The evaluation uses 1,319, 500, 198, and 490 samples, respectively. For StrategyQA, we use 490 samples drawn from its training set.

Baselines. We compare CoS-Web with representative single-agent methods, multi-agent debate methods, and communication optimization methods.

First, we use Chain-of-Thought (CoT; Wei et al., 2022), Chain-of-Thought Self-Consistency (CoT-SC; Wang et al., 2023), and Majority Vote as baseline reasoning methods. These methods allow us to assess the advantages of multi-agent interaction over independent reasoning and simple result aggregation.

We then compare multi-agent debate methods. Multi-Agent Debate (MAD-Full; Du et al., 2024) represents fully connected cooperative debate, using multiple rounds of discussion and answer revision among agents to improve reasoning reliability. We also compare Adversarial Debate (Liang et al., 2024), Sparse MAD (Li et al., 2024), MAD-M^2 (Tian et al., 2026), CortexDebate (Sun et al., 2025), AnyMAC (Wang et al., 2025), and Free-MAD (Cui et al., 2026), evaluated under our implementation as Free-MAD-N, the anti-conformity decision variant defined in the original paper (as distinct from its conformity-based counterpart, Free-MAD-C), selected because it better matches the reasoning-intensive nature of our benchmark suite. Together, these baselines cover dense debate, sparse communication, memory filtering, adaptive routing, and consensus-free decisions. All multi-agent baselines are controlled reimplementations under the same backbone, decoding configuration, agent pool, initial responses, answer parser, and evaluation protocol; Appendix A.1 details the setup.

Implementation Details. Experiments use the locally deployed **Qwen3-14B-AWQ** model (Yang et al., 2025). The system contains five reasoning agents and one Judge. All agents share the same backbone model and decoding parameters, while the reasoning agents use different reasoning-perspective prompts to produce diverse reasoning processes.

Specifically, the temperature is set to 0.6, and the maximum number of interaction rounds is:

$$T_{\max} = 3.$$

The system also supports early termination: if no unresolved issues remain after Ring Communication, the current collaboration process ends early.

We conduct three independent full runs using different random seeds. Within each run, the five initial agent responses for each sample are generated only once and reused across all multi-agent methods to enable paired comparisons.

Evaluation Metrics. Accuracy is the primary metric across the four reasoning benchmarks. As a secondary diagnostic, we record total tokens per question, summing input and output tokens across all LLM calls. Appendix F reports detailed token and communication-cost statistics across the evaluation runs.

4.2 Main Experimental Results

We evaluate CoS-Web's reasoning accuracy on four benchmarks. Table 1 reports the results; additional communication analyses are provided in Appendix F.

Table 1. Main experimental results.

Method	GSM8K	MATH-500	GPQA-D	StrategyQA	Avg.
CoT	91.0±0.3	80.6±0.6	55.2±0.7	76.5±0.6	75.83
CoT-SC	93.1±0.3	83.3±0.5	59.4±0.6	78.2±0.6	78.50
Majority Vote	92.3±0.4	82.2±0.6	59.4±0.7	70.6±0.6	76.13
MAD-Full	**95.4±0.3**	84.8±0.5	61.4±0.8	71.0±0.6	78.15
Adversarial Debate	93.2±0.4	85.6±0.5	61.4±0.8	73.4±0.7	78.40
Sparse MAD	95.0±0.3	85.2±0.5	63.1±0.7	74.1±0.7	79.35
MAD-M^2	94.1±0.3	86.1±0.5	63.3±0.6	77.0±0.6	80.13
CortexDebate	94.3±0.3	86.5±0.4	63.3±0.5	78.5±0.6	80.65
AnyMAC	94.7±0.3	86.8±0.4	63.1±0.5	79.5±0.6	81.03
Free-MAD-N	94.6±0.3	87.1±0.5	63.3±0.5	80.1±0.6	81.28
CoS-Web	95.1±0.2	**89.4±0.4**	**64.5±0.8**	**84.2±0.5**	**83.30**

Observation 1. CoS-Web has the highest observed macro-average and the highest observed mean on three benchmarks.

CoS-Web reaches **83.30%** macro-average accuracy, 2.02 percentage points above Free-MAD-N, the strongest baseline by this metric. It has the highest observed mean on MATH-500, GPQA-Diamond, and StrategyQA, while MAD-Full is slightly ahead on GSM8K (95.4% vs. 95.1%). On StrategyQA, CoS-Web exceeds CoT by 7.7 points, whereas Majority Vote and MAD-Full fall below CoT. These gains are task dependent and are largest on tasks where examining rationales and assumptions is particularly important. The independent contribution of Spider-Web Communication is reported in Appendix F.3.

4.3 Ablation Study

To analyze the roles of different suspicion levels in Chain-of-Suspicion, we use MAD-Full as a reference baseline and progressively compare three issue-formation configurations—D1-only, D1+D2, and the full D1+D2+D3. When D3 is disabled, the available D1/D2 records are deterministically mapped to the same unresolved-issue interface, with each source–target suspicion pair defining the corresponding `involved_agents`; this mapping introduces no additional LLM call or cross-agent aggregation. All variants then use the same downstream Ring Communication, recheck, and termination procedures.

Table 2. Ablation study of issue formation.

Issue Formation	GPQA-D ↑	StrategyQA ↑
MAD-Full	61.45±0.80	70.95±0.60
D1-only	62.12±0.60	74.83±0.70
D1+D2	63.30±0.50	79.86±0.62
D1+D2+D3	**64.48±0.80**	**84.22±0.50**

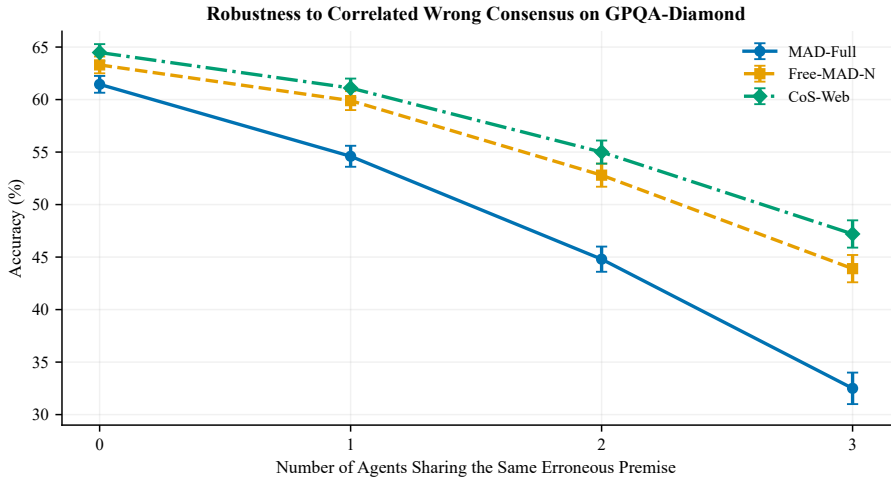


Figure 3. Robustness of different methods to false consensus on GPQA-Diamond.

Observation 2. Higher-order suspicion progressively broadens the scope of reasoning examination.

With the downstream communication and decision processes held fixed, deeper issue formation corresponds to higher accuracy. D2 stress-tests anticipated defenses, while D3 further aggregates group-level risks reflected in the D1/D2 reports. These results show progressive gains as issue formation moves from direct local suspicion, to nested verification, and finally to collective issue formation.

The independent contribution of Spider-Web Communication is evaluated in Appendix F.3 through a comparison between CoS + Full Communication and the relevant baseline.

4.4 Robustness to False Consensus

Traditional multi-agent collaboration methods often rely on agreement among multiple agents when making decisions. However, when multiple agents share the same erroneous assumption, apparent agreement may further reinforce error propagation. To evaluate CoS-Web's ability to handle this problem, we design a False Consensus Robustness experiment.

For each sample, we construct a corruption package consisting of an incorrect answer and a shared erroneous assumption. At corruption level $k \in \{0, 1, 2, 3\}$, we sample k of the five reasoning agents without replacement, replace only their initial reasoning, and have them express the same erroneous assumption using different wording. No further forced corruption is applied afterward, and the Judge is unaware of the agents' corruption status. For a given sample and random seed, all methods share exactly the same corruption configuration.

Figure 3 compares MAD-Full, Free-MAD-N, and CoS-Web. The complete four-method table, including Adversarial Debate, is provided in Appendix H.1.

Observation 3. CoS-Web demonstrates stronger robustness against false consensus.

As the number of agents holding the shared erroneous assumption increases, all three methods show marked degradation. MAD-Full drops from 61.45±0.80 to 32.49±1.50, Free-MAD-N from 63.30±0.80 to 43.94±1.30, and CoS-Web from 64.48±0.80 to 47.31±1.30. The corresponding decreases are 29.0, 19.4, and 17.2 percentage points, respectively.

Free-MAD-N substantially mitigates error propagation associated with conformity and forced consensus. CoS-Web achieves higher accuracy, suggesting that removing forced consensus alone is insufficient to fully address cases in which multiple agents share the same erroneous assumption. Explicitly examining the basis for judgments, anticipated defenses, and group-level risks may provide additional recovery capacity.

Case Study. In the experimental example in Figure 2, four of the five agents select the incorrect answer C because they share an interpretive bias. CoS aggregates the misreading of the objective revealed in D1/D2 into a false-consensus issue, prompting the relevant agents to re-examine the two minima. This breaks the false consensus and corrects the final answer from C to B. The complete problem, calculations, and analysis of each stage are provided in Appendix H.4.

5 Conclusion

We introduce CoS-Web, a multi-agent reasoning framework that uses structured unresolved issues as an explicit intermediate representation connecting **suspicion-driven interaction** with follow-up communication. Chain-of-Suspicion examines individual judgments, anticipated defenses, and group-level risks even when agents already agree, while Spider-Web Communication uses the resulting issues to organize local exchange among the relevant agents and re-examines their status afterward. Across the evaluated reasoning benchmarks, CoS-Web attains the highest macro-average accuracy among the compared methods and shows the largest gains on tasks where scrutinizing rationales and shared assumptions matters most. It is also markedly more robust when multiple agents are induced to share the same erroneous assumption, indicating that explicitly verifying the basis for agreement — rather than only monitoring disagreement — can mitigate false consensus in multi-agent reasoning. Although we validate CoS-Web on two different model families, the current evaluation mainly focuses on medium-scale open-source language models. In future work, we will further investigate the applicability of CoS-Web to larger-scale models, heterogeneous agent compositions, and more complex multi-agent environments.

AI use statement

We used LLMs to assist with grammar and writing polishing. All equations, analysis, and research contributions are our own.

References

Becker, J., Kaesberg, L. B., Stephan, A., Wahle, J. P., Ruas, T., & Gipp, B. (2026). Stay focused: Problem drift in multi-agent debate. In *Findings of the Association for Computational Linguistics: EACL 2026* (pp. 5068–5102). https://aclanthology.org/2026.findings-eacl.268/

Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., Fu, J., & Liu, Z. (2024). ChatEval: Towards better LLM-based evaluators through multi-agent debate. *International Conference on Learning Representations (ICLR)*. https://proceedings.iclr.cc/paper_files/paper/2024/hash/25cc3adf8c85f7c70989cb8a97a691a7-Abstract-Conference.html

Cobbe, K., Kosaraju, V., Bavarian, M., et al. (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*. https://arxiv.org/abs/2110.14168

Cui, Y., Fu, H., Zhang, H., Wang, L., & Zuo, C. (2026). Free-MAD: Consensus-free multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2026* (pp. 31977–31997). https://doi.org/10.18653/v1/2026.findings-acl.1600

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., & Mordatch, I. (2024). Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning* (pp. 11733–11763). https://proceedings.mlr.press/v235/du24e.html

Feng, Y., Xu, Z., Hu, X., & He, F. (2026). PEAR: Permutation-Equivariant Adaptive Routing Multi-Agent Debate. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing* (to appear). https://arxiv.org/abs/2606.20621

Geva, M., Khashabi, D., Segal, E., Khot, T., Roth, D., & Berant, J. (2021). Did Aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics, 9*, 346–361. https://doi.org/10.1162/tacl_a_00370

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., & Zhang, X. (2024). Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the*

Thirty-Third International Joint Conference on Artificial Intelligence (pp. 8048–8057). https://doi.org/10.24963/ijcai.2024/890

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., & Steinhardt, J. (2021). Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*. https://arxiv.org/abs/2103.03874

Hong, S., Zhuge, M., Chen, J., et al. (2024). MetaGPT: Meta programming for a multi-agent collaborative framework. *International Conference on Learning Representations (ICLR)*. https://proceedings.iclr.cc/paper_files/paper/2024/hash/6507b115562bb0a305f1958ccc87355a-Abstract-Conference.html

Li, G., Hammoud, H., Itani, H., Khizbullin, D., & Ghanem, B. (2023). CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems 36*. https://doi.org/10.52202/075280-2264

Li, Y., Du, Y., Zhang, J., Hou, L., Grabowski, P., Li, Y., & Ie, E. (2024). Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024* (pp. 7281–7294). https://doi.org/10.18653/v1/2024.findings-emnlp.427

Liang, T., He, Z., Jiao, W., Wang, X., Wang, Y., Wang, R., Yang, Y., Shi, S., & Tu, Z. (2024). 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). https://doi.org/10.18653/v1/2024.emnlp-main.992

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., & Cobbe, K. (2023). Let's verify step by step. *arXiv preprint arXiv:2305.20050*. https://arxiv.org/abs/2305.20050

Liu, Y., Cao, J., Li, Z., He, R., & Tan, T. (2025). Breaking mental set to improve reasoning through diverse multi-agent debate. *International Conference on Learning Representations (ICLR)*. https://proceedings.iclr.cc/paper_files/paper/2025/hash/3de667dab3b3d812583abc0a786139a0-Abstract-Conference.html

Madaan, A., Tandon, N., Gupta, P., et al. (2023). Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36*. https://doi.org/10.52202/075280-2019

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., & Bowman, S. R. (2024). GPQA: A graduate-level Google-proof Q&A benchmark. *Conference on Language Modeling (COLM)*. https://openreview.net/forum?id=Ti67584b98

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems 36*. https://doi.org/10.52202/075280-0377

Smit, A. P., Grinsztajn, N., Duckworth, P., Barrett, T. D., & Pretorius, A. (2024). Should we be going MAD? A look at multi-agent debate strategies for LLMs. In *Proceedings of the 41st International Conference on Machine Learning* (pp. 45883–45905). https://proceedings.mlr.press/v235/smit24a.html

Sun, Y., Zhao, Z., Wan, S., & Gong, C. (2025). CortexDebate: Debating sparsely and equally for multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2025* (pp. 9503–9523). https://doi.org/10.18653/v1/2025.findings-acl.495

Tian, H., Feng, X., Zhao, Z., Zhu, X., Yan, R., & Han, B. (2026). Multi-agent debate with memory masking. *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=EdTt8nMAMA

Verma, A., Gupta, S., Gupta, D., Sircar, P., & Pillai, S. (2026). SELENE: Selective and evidence-weighted LLM debating for efficient and reliable reasoning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)* (pp. 95–104). https://doi.org/10.18653/v1/2026.eacl-industry.7

Wang, S., Tan, Z., Chen, Z., Zhou, S., Chen, T., & Li, J. (2025). AnyMAC: Cascading flexible multi-agent collaboration via next-agent prediction. In *Proceedings of the 2025 Conference on Empirical*

Methods in Natural Language Processing (pp. 11555–11567). https://doi.org/10.18653/v1/2025.emnlp-main.584

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., & Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/2203.11171

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35*. https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html

Wu, Q., Bansal, G., Zhang, J., et al. (2024). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *Conference on Language Modeling (COLM)*. https://arxiv.org/abs/2308.08155

Yang, A., Li, A., Yang, B., et al. (2025). Qwen3 technical report. *arXiv preprint arXiv:2505.09388*. https://arxiv.org/abs/2505.09388

Yoffe, L., Amayuelas, A., & Wang, W. Y. (2025). DebUnc: Improving large language model agent communication with uncertainty metrics. In *Findings of the Association for Computational Linguistics: EMNLP 2025* (pp. 23299–23315). https://doi.org/10.18653/v1/2025.findings-emnlp.1265

Zhang, G., Yue, Y., Li, Z., et al. (2025a). Cut the crap: An economical communication pipeline for LLM-based multi-agent systems. *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=LkzuPorQ5L

Zhang, G., Yue, Y., Sun, X., Wan, G., Yu, M., Fang, J., Wang, K., Chen, T., & Cheng, D. (2025b). G-Designer: Architecting multi-agent communication topologies via graph neural networks. In *Proceedings of the 42nd International Conference on Machine Learning* (pp. 76678–76692). https://proceedings.mlr.press/v267/zhang25cu.html

Zhu, X., Zhang, C., Chi, Y., Stafford, T., Collier, N., & Vlachos, A. (2026). Demystifying Multi-Agent Debate: The Role of Confidence and Diversity. In *Findings of the Association for Computational Linguistics: ACL 2026* (pp. 33909–33930). https://doi.org/10.18653/v1/2026.findings-acl.1694

A Additional Experimental Details

A.1 Experimental Configuration

The current experimental framework uses five reasoning agents and one Judge. All methods share the same backbone, decoding settings, sample order, and answer parser. Within each evaluation seed, all multi-agent methods additionally share the same initial agent responses. The five reasoning-perspective prompts are used only to elicit different reasoning perspectives; they provide no method with additional knowledge, tools, or privileged information.

CoS-Web's maximum number of interaction rounds is set to

$$T_{\text{max}} = 3,$$

with early termination if no unresolved issues remain after Ring Communication.

Configuration	Setting
Backbone	Qwen3-14B-AWQ
Reasoning agents	5
Judges	1
Temperature	0.6
Maximum generated tokens	16,384
Top-p / Top-k / Min-p	0.95 / 20 / 0.0
Qwen thinking mode	Enabled
T_{max}	3
Early Stopping	Yes
GSM8K	1,319
MATH-500	500
GPQA-Diamond	198
StrategyQA sampled training examples	490

For StrategyQA, we use 490 samples drawn from the training set. Under the same random seed, all methods use the same sample order, while all multi-agent methods additionally share the same initial agent states for paired comparisons. full runs with different random seeds are independent. Accuracy results are reported as mean $\pm$ standard deviation across three independent runs with different random seeds. The standard deviation is computed across the three seed-level accuracy values.

Unified Baseline Protocol. For controlled comparisons, we reimplement all multi-agent baselines within a unified evaluation framework. For each sample and evaluation random seed, the five initial agent responses are generated only once and reused across all multi-agent methods. All methods use the same backbone, decoding configuration, agent pool, task-specific answer parser, and evaluation procedure.

The unified framework standardizes only components that are independent of any particular method, while preserving each method's specific interaction mechanism: MAD-Full and Adversarial Debate use their respective dense interaction protocols; Sparse MAD and CortexDebate use sparse communication mechanisms; AnyMAC uses adaptive agent/context routing; MAD-M^2 applies memory masking across debate rounds; and Free-MAD-N retains its consensus-free decision mechanism. CoS-Web uses the same initial states and evaluation protocol, replacing the interaction process with CoS-based issue formation and issue-driven Spider-Web Communication.

Because the original methods were developed with different models and execution environments, the results in this paper should be interpreted as controlled reimplementations under a shared backbone and evaluation setup, rather than direct reproductions of the results reported in the original papers.

The five reasoning agents have the following roles: Direct Solver, Formalizer, Verification-first Solver, Alternative Solver, and Assumption and Consistency Checker. The role prompts change only the reasoning perspective. All roles produce the same public state fields: `final_answer`, `reasoning_summary`, `key_evidence`, `assumptions`, and `confidence`.

The Radial board is constructed deterministically from the public state: `answer=final_answer`, `confidence=confidence`, and `evidence_summary=reasoning_summary`. This step consists only of

field projection and compact JSON serialization. It does not invoke an additional model or implicitly summarize `key_evidence` or `assumptions`.

Method family	Preserved method-specific component	Unified adaptation
CoT / CoT-SC / Majority Vote	Independent reasoning, path sampling, or answer voting	Shared backbone, decoding, and answer parser
MAD-Full	Dense full-context cooperative interaction	Shared initial responses, backbone, and final evaluation
Adversarial Debate	Dense adversarial interaction	Shared initial responses, backbone, and final evaluation
Sparse MAD	Sparse communication topology	Shared initial responses, backbone, and final evaluation
CortexDebate	Sparse debate communication mechanism	Shared initial responses, backbone, and final evaluation
AnyMAC	Adaptive agent/context routing	Shared initial responses, backbone, and final evaluation
MAD-M^2	Memory masking across debate rounds	Shared initial responses and public-state schema
Free-MAD-N	Consensus-free, anti-conformity decision	Shared initial responses and final answer parser
CoS-Web	CoS issue formation and Spider-Web routing	Same common protocol as all multi-agent baselines

A.2 Prompt Interfaces and Structured Outputs

The following table specifies the prompt interfaces required to reproduce the algorithm outline. Every stage receives the original task; the table lists only the additional state read by each stage and the structured output it must produce.

Stage	System messages	Calls per round	Additional inputs	Output contract
Initial reasoning	agent role	Once/agent	Question	Public five-field state
D1 Direct Suspicion	agent role + D1	Once/agent	Own state, deterministic Radial board, carried issues	`suspicions[{target_agent,suspicion_type,concern,question}]`
D2 Nested Suspicion	agent role + D2	Once/agent	Own state and its complete D1 suspicion list	`nested[{target_agent,anticipated_defense,remaining_risk,deeper_question}]`
D3 Collective Suspicion	D3 Judge only	Once/Judge	Concatenated D1/D2 reports from all agents	`issues[{issue_id,involved_agents,risk_type,description}]`
Ring disclosure	agent role + Ring	Once/ selected agent/group	Own state and grouped D3 issues	`issue_responses`, `additional_evidence`, `counterexamples`
Ring update	agent role + Update	Once/ selected agent/group	Own state, grouped issues, and canonical-neighbor disclosures	Updated public five-field state
Recheck	Recheck Judge only	Once/Judge	Pre-ring issues and all post-ring agent states	`remaining_issues[{issue_id,resolved,reason}]`
Final Judge	Final Judge only	Once/ termination	Final public states of all agents	`final_answer,reasoning_summary,confidence`

The Radial summary is a deterministic projection of each public state containing `final_answer`, `confidence`, and `reasoning_summary` (serialized as `evidence_summary`); `key_evidence` and `assumptions` are not broadcast separately. In the D1-only and D1+D2 ablations, where D3 is disabled, an ablation-only deterministic adapter maps each source–target suspicion pair to the same issue schema, with the source agent and its `target_agent` forming `involved_agents`.

Each D3 issue uses the following fixed schema:

```
{"issue_id":"U1","involved_agents":["A1","A3"],"risk_type":"false_consensus","description":".
↪  .."}
```

`risk_type` is an open-ended, concise descriptor of the collective concern identified by D3 . Labels such as `shared_assumption`, `false_consensus`, `circular_support`, and `collective_blind_spot` may occur when supported by the reports. Ring membership is determined by `involved_agents`. These are operational categories used in the current implementation, not an exhaustive classification.

Issues with fewer than two valid members do not activate a Ring. Issues with the same member set are merged into one issue package, and members are ordered canonically from A1 to A5. Issue packages with different member sets are processed in the order of their first appearance in the **constructed issue list**. If an agent belongs to multiple local rings, a later ring reads the public state most recently updated for that agent. Consequently, overlapping local rings may depend on activation order; Appendix F.6 explicitly tests robustness to this factor.

In the implementation, D1, D2, Ring disclosure, and Ring update each receive both the agent role prompt and the corresponding stage prompt. D3, Recheck, and Final Judge receive only their respective Judge prompts. The D3 prompt file retains the historical name "Deep Nested Suspicion"; this paper refers to that stage consistently as **Collective Suspicion**.

Operational Algorithm Skeleton.

```
states <- IndependentReasoning(x, A1...AN)
carried <- empty
for t = 1...Tmax:
    summaries <- RadialCompressAndBroadcast(states)
    d1 <- [D1(Ai, states[Ai], summaries, carried) for Ai]
    d2 <- [D2(Ai, states[Ai], d1[Ai]) for Ai]
    issues <- ParseD3OrRetry(Judge(x, d1, d2))

    groups <- group issues by valid involved_agents, preserving first occurrence
    for members, issue_pack in groups:
        order <- canonical agent order restricted to members
        neighbors <- peer if |members|=2 else predecessor and successor
        disclosures <- [RingDisclosure(Ai, states[Ai], issue_pack) for Ai]
        updates <- [RingUpdate(Ai, states[Ai], issue_pack,
                               disclosures[neighbors(Ai)]) for Ai]
        states[members] <- updates

    remaining <- ParseRecheckOrRetry(Judge(x, issues, states[A1...AN]))
    unresolved <- [r for r in remaining if not truthy(r.resolved)]
    if unresolved is empty: break
    carried <- unresolved
return JudgeFinalAnswer(x, states[A1...AN])
```

Representative CoS Prompts on GPQA-Diamond. Representative prompts used for GPQA-Diamond are provided below. To conserve space, the other datasets use the same stage structure, with only the role descriptions and final-answer format constraints adjusted to match the task's answer space.

D1 — Direct Suspicion (system). The user message contains, in order, `Question`, `YourCurrentState`, `RadialGlobalAwareness`, and `UnresolvedIssuesCarriedOverFromThePreviousRound`.

```
You are executing D1 — Direct Suspicion, the first layer of Chain-of-Suspicion.

You are given the question, your own current state, the low-bandwidth radial board
(every agent's answer, confidence and compressed evidence), and optionally the
issues that remained unresolved in the previous round.

For each other agent, identify a specific step, fact or assumption that remains
unverified. Do not decide who is right or wrong, do not fabricate concerns, and
never raise a suspicion about yourself. If there is no genuine ground for
suspicion, return an empty list.
```

```
Return exactly one JSON object:
{
  "suspicions": [{
    "target_agent": "A2",
    "suspicion_type": "...",
    "concern": "...",
    "question": "..."
  }]
}
```

D2 — Nested Suspicion (system). The user message contains, in order, `Question`, `YourCurrentState`, and the complete `YourD1Suspicions`.

```
You are executing D2 — Nested Suspicion, the second layer of Chain-of-Suspicion.

For each suspicion raised in D1:
1. Anticipate how the other agent would most plausibly defend itself.
2. Ask whether that defence would actually settle the concern and state the
   remaining risk.
3. Formulate the deeper question that demands the real support required.

Do not resolve a suspicion by simulating a plausible defence yourself. If D1
produced no suspicions, return an empty list.

Return exactly one JSON object:
{
  "nested": [{
    "target_agent": "A2",
    "anticipated_defense": "...",
    "remaining_risk": "...",
    "deeper_question": "..."
  }]
}
```

D3 — Collective Suspicion (Judge system). The user message contains the original question and the complete D1/D2 reports from all five agents.

```
You are the Judge executing the third layer of Chain-of-Suspicion.

D1 and D2 expose concerns about individual judgements and anticipated defenses.
D3 aggregates only the risks reflected in these reports, including:
- shared_assumption
- false_consensus
- circular_support
- collective_blind_spot

Provide no new task information. Record every genuine issue that still requires
a real response, and list precisely the agents involved. Exactly two agents use
a bidirectional exchange; three or more form a ring. Return an empty list when no
group-level issue survives.

Return exactly one JSON object:
{
  "issues": [{
    "issue_id": "U1",
    "involved_agents": ["A1", "A3"],
    "risk_type": "shared_assumption",
    "description": "..."
  }]
}
```

Ring disclosure (system). The user message contains, in order, `Question`, `YourCurrentState`, and the `UnresolvedIssues` involving that agent.

```
You are now in Ring Communication, the high-bandwidth channel of Spider-Web
Communication. Disclose your complete visible reasoning basis, respond directly
to every unresolved issue that involves you, and provide relevant counterexamples
or edge cases. Do not merely restate your answer and do not withhold information
that weakens your position.

Return exactly one JSON object:
{
  "issue_responses": [{
    "issue_id": "U1",
    "response": "...",
    "evidence": "..."
  }],
  "additional_evidence": ["..."],
  "counterexamples": ["..."]
}
```

Ring update (system). The user message contains, in order, `Question`, `YourCurrentState`, `UnresolvedIssues`, and `FullDisclosuresFromYourRingNeighbours`.

```
You have received the full disclosures of your ring neighbours and must now
update your own judgement. Use only the supplied disclosures; correct a real
error when exposed, but do not change the answer merely because a neighbour
disagrees. Keep the visible reasoning summary concise.

Return the updated public state:
{
  "final_answer": "A",
  "reasoning_summary": "...",
  "key_evidence": ["..."],
  "assumptions": ["..."],
  "confidence": 0.0
}

final_answer must be exactly one uppercase letter: A, B, C or D.
```

Post-ring recheck (Judge system). The user message contains `UnresolvedIssuesBeforeRingCommunication` and the post-communication states of all five agents.

```
You are the Judge performing the post-ring re-check. For each pre-ring issue,
decide whether it remains reflected in the supplied post-ring public states.
You do not receive or inspect the raw Ring disclosures. Do not introduce new
task information and do not re-answer the question yourself.

Return exactly one JSON object:
{
  "remaining_issues": [{
    "issue_id": "U1",
    "resolved": false,
    "reason": "..."
  }]
}

Return an empty list when every issue is settled.
```

Final Judge (system). The user message contains the original question and the final public states of all agents.

```
You are the final Judge for a cooperative multi-agent reasoning experiment.
Aggregate the agents' final answers and visible supporting rationale into one
final adjudicated option. Resolve disagreements using the quality and consistency
of the visible rationale rather than majority agreement alone. Do not execute
CoS, perform routing or start additional debate.
```

```
Return exactly one JSON object:
{
  "final_answer": "",
  "reasoning_summary": "",
  "confidence": 0.0
}

final_answer must be exactly one uppercase letter: A, B, C or D.
```

A.3 Evaluation and Answer Extraction

All methods use exactly the same task-specific deterministic answer parser. Outputs that cannot be parsed are counted as incorrect; no additional model call is made to repair them. For CoT-SC, the final answer from each reasoning path is parsed separately, and the parsed answers are then aggregated by voting. Ties are resolved using a predefined deterministic ordering rule.

Dataset	Output Type	Evaluation
GSM8K	Numeric	Normalized numerical exact match
MATH-500	Mathematical expression	Symbolic equivalence after normalization
GPQA-Diamond	A/B/C/D	Exact option match under a fixed per-instance choice permutation
StrategyQA	Yes/No	Case-insensitive Boolean exact match on the final answer field

For GPQA-Diamond, the option permutation is fixed for each sample and random seed, and reused across all methods. For StrategyQA, the parser reads only the explicit `FinalAnswer:Yes/No` field; it does not infer an answer from isolated occurrences of "yes" or "no" in the explanation text.

A.4 Cost Metrics and Runtime Logging

Total Tokens are defined as the sum of the input and output tokens across all LLM calls made for a sample:

$$T_{\text{total}} = \sum_{c \in \mathcal{C}} \left(T_{\text{input}}^{(c)} + T_{\text{output}}^{(c)} \right).$$

Communication Tokens are defined as the context tokens in a call's input that originate from other agents or the Judge and are actually consumed by the recipient:

$$T_{\text{comm}} = \sum_{c \in \mathcal{C}} T_{\text{foreign}-\text{context}}^{(c)}.$$

A.5 Correlated-Error Construction

For each test sample x, we construct an error-injection package:

$$C_x = (y_x^-, p_x^-),$$

where y_x^- is a plausible but incorrect answer, and p_x^- is a shared erroneous assumption that supports that answer. During construction, the gold label is used to exclude the correct answer. Candidate erroneous assumptions are then generated through a fixed process isolated from evaluation calls. A second verification prompt retains only candidates that satisfy all of the following conditions: the answer differs from the gold answer; the assumption supports the incorrect answer; the assumption itself can be refuted by gold reasoning; and the text contains no explicit corruption marker. Candidates that fail verification are regenerated. Construction tokens are not included in test-time reasoning costs.

At corruption level $k \in \{0, 1, 2, 3\}$, we sample k of the five reasoning agents without replacement and replace only their initial reasoning. Across different random seeds, each corrupted agent independently rephrases the same erroneous assumption to avoid copying identical text. No further forced

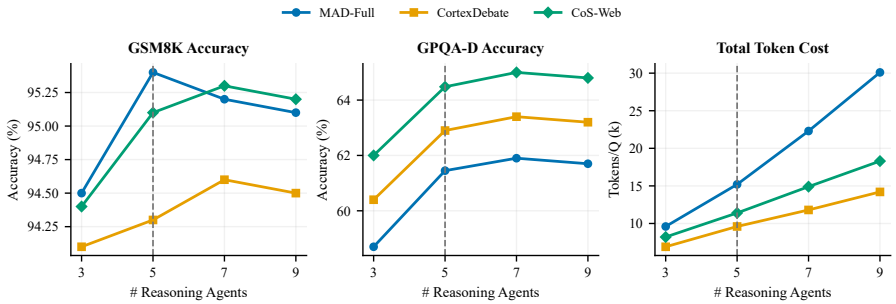


Figure B.1. Accuracy and total token cost as the number of reasoning agents increases. The dashed line marks the five-agent setting used in the main experiments.

corruption is applied afterward; during interaction, agents may retain, revise, or abandon their initial judgments.

The Judge receives neither the corruption identities nor their status labels. For the same sample and random seed, all methods use exactly the same corrupted agent IDs, incorrect answer, erroneous assumption, and initial responses, enabling paired comparisons.

B Scaling with the Number of agents

We examine how the number of reasoning agents, $N \in \{3, 5, 7, 9\}$, affects performance and cost. The Judge is not included in N.

Table B.1. Scaling with the Number of Reasoning agents

# agents	Method	GSM8K ↑	GPQA-D ↑	Total Tokens/Q ↓
3	MAD-Full	94.5±0.4	58.6±1.1	9.6k±0.6k
	CortexDebate	94.1±0.4	60.4±1.0	**6.9k±0.5k**
	CoS-Web	94.4±0.3	**62.0±1.0**	8.2k±0.6k
5	MAD-Full	**95.4±0.3**	61.45±0.8	15.2k±0.8k
	CortexDebate	94.3±0.3	62.8±0.8	**9.6k±0.6k**
	CoS-Web	95.1±0.2	**64.48±0.8**	11.4k±0.7k
7	MAD-Full	95.2±0.4	61.8±0.9	22.3k±1.2k
	CortexDebate	94.6±0.3	63.3±0.9	**11.8k±0.7k**
	CoS-Web	**95.3±0.3**	**65.0±0.9**	14.9k±0.9k
9	MAD-Full	95.1±0.4	61.6±1.0	30.1k±1.5k
	CortexDebate	94.5±0.4	63.1±1.0	**14.2k±0.9k**
	CoS-Web	**95.2±0.3**	**64.8±1.0**	18.3k±1.1k

Increasing the number of reasoning agents from five to seven yields only a slight gain. At nine agents, performance declines slightly while token costs continue to rise. Thus, $N = 5$ is a reasonable performance–cost operating point for the main experiments.

C Judge Ablation

This section uses a two-factor ablation to separate the contributions of D3 and the Final Judge. When D3 is removed, we use the same deterministic issue-interface mapping defined for the D1/D2 ablations in Appendix A.2, so that the downstream Ring Communication and recheck procedures remain executable without introducing an additional reasoning module.

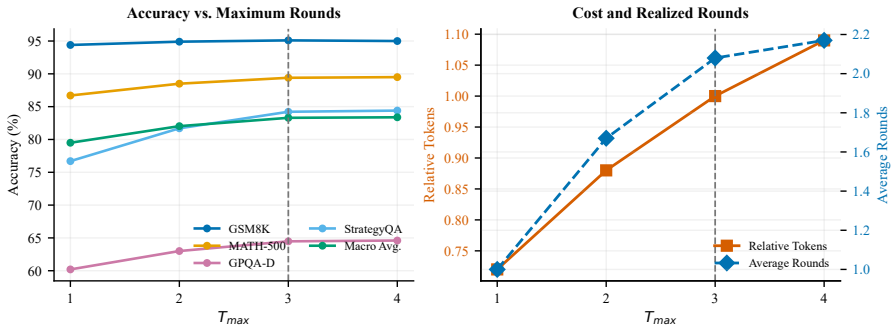


Figure E.1. Benchmark accuracy, macro-average accuracy, relative total token cost, and realized interaction rounds under different maximum interaction budgets. The dashed line marks the setting used in the main experiments.

Table C.1. Factorized Judge Ablation

D3	Final Judge	GSM8K	MATH-500	GPQA-D	StrategyQA
×	×	94.6±0.3	88.1±0.6	60.4±0.9	76.3±0.8
×	✓	95.0±0.2	88.8±0.5	63.3±0.5	79.9±0.6
✓	×	94.8±0.3	88.9±0.5	63.0±0.6	82.1±0.6
✓	✓	**95.1±0.2**	**89.4±0.4**	**64.48±0.8**	**84.22±0.5**

Removing D3 causes a more pronounced drop on StrategyQA, consistent with the greater influence of shared assumptions and false consensus on interaction outcomes in this dataset. Configurations that retain D3 outperform those without both D3 and the Final Judge. The gains therefore cannot be attributed simply to an additional final LLM call to select an answer.

D Effect of Prompt Diversity

Table D.1. Effect of Prompt Diversity

Method	Prompt	GSM8K	MATH-500	GPQA-D	StrategyQA
MAD-Full	Same ×5	94.8±0.4	83.9±0.6	60.3±0.9	69.8±0.7
MAD-Full	Diverse ×5	**95.4±0.3**	**84.8±0.5**	**61.45±0.8**	**70.95±0.6**
CoS-Web	Same ×5	94.8±0.3	88.0±0.5	63.1±0.9	81.9±0.6
CoS-Web	Diverse ×5	**95.1±0.2**	**89.4±0.4**	**64.48±0.8**	**84.22±0.5**

Prompt diversity yields some gains, but Same-Prompt CoS-Web still outperforms Diverse-Prompt MAD-Full on GPQA-Diamond and StrategyQA, indicating that differences between the methods cannot be explained by prompt diversity alone.

E Sensitivity to Maximum Interaction Rounds

Table E.1. Sensitivity to Maximum Interaction Rounds

$T_{\max}$	GSM8K	MATH-500	GPQA-D	StrategyQA	Avg.	Rel. Tokens	Avg. Rounds
1	94.4±0.4	86.7±0.6	60.1±1.0	76.7±0.8	79.50	**0.72±0.02**	1.00
2	94.9±0.3	88.5±0.5	63.0±0.9	81.7±0.6	82.03	0.88±0.03	1.67±0.07
3	**95.1±0.2**	**89.4±0.4**	**64.48±0.80**	**84.22±0.50**	**83.30**	1.00	2.08±0.08
4	95.0±0.3	89.5±0.4	64.6±0.9	84.4±0.5	83.38	1.09±0.04	2.17±0.09

Performance begins to saturate beyond $T_{\max} = 3$. Increasing the maximum from 3 to 4 yields an average gain of only 0.08 percentage points, while increasing token cost by about 9%.

F Communication Pattern and Cost Analysis

F.1 Dynamic Communication Patterns

We define the high-bandwidth Active Edge Ratio as:

$$\rho_E = \frac{|E_{\text{ring}}|}{|E_{\text{full}}|}.$$

For five regular agents, the fully connected graph contains 10 undirected agent–agent edges; low-bandwidth Radial spokes are excluded from this ratio.

Table F.1. Observed Communication Behavior

Dataset	Samples	Active Edge Ratio	Ring Activation	Avg. Ring Size	Avg. Rounds
GSM8K	1,319	0.381±0.026	41.5±1.8%	2.62±0.12	1.74±0.06
MATH-500	500	0.507±0.031	59.3±2.1%	3.09±0.15	2.10±0.08
GPQA-D	198	**0.617±0.047**	**68.2±3.4%**	3.48±0.19	2.35±0.11
StrategyQA	490	0.589±0.038	**66.1±2.7%**	3.36±0.17	2.23±0.09

This pattern indicates that activation of high-bandwidth Ring Communication varies with the task rather than being determined by a fixed sparse graph.

F.2 Token-Cost Decomposition

The table below uses the definitions given in Appendix A.4.

Table F.2. Token-Cost Decomposition on GSM8K

Component	Total Tokens/Q	Communication?
Initial reasoning	5.57k	No
Radial messages consumed by receivers	0.74k	Yes
D1 internal scrutiny	0.70k	No
D2 anticipated-defense reasoning	0.90k	No
D3 internal audit	0.38k	No
D3 issue broadcast	0.30k	Yes
Ring rationale exchange	1.96k	Yes
Final agent states consumed by Judge	0.50k	Yes
Judge final reasoning	0.35k	No
Total	**11.40k**	—

F.3 Spider-Web Communication Ablation

CoS + Full Communication and CoS-Web use the same CoS, Judge, $T_{\max}$, and early-stopping rules; they differ only in the structure used for high-bandwidth exchange of supporting rationales.

Table F.3. Communication Ablation

Dataset	Configuration	Accuracy ↑	Total Tokens/Q ↓	Comm. Tokens/Q ↓	Avg. Rounds
GPQA-D	MAD-Full	61.45±0.80	21.8k	13.4k	3.00
	CoS + Full Communication	**64.65±0.40**	24.6k	15.2k	2.35±0.11
	CoS-Web	64.48±0.80	**17.1k**	**8.5k**	2.35±0.11
StrategyQA	MAD-Full	70.95±0.60	13.2k	7.8k	3.00
	CoS + Full Communication	82.99±0.63	15.6k	9.4k	2.23±0.09
	CoS-Web	**84.22±0.50**	**10.9k**	**5.2k**	2.23±0.09

On GPQA-Diamond, CoS-Web is 0.17 percentage points lower than CoS + Full Communication, while Communication Tokens decrease by 44.1% and Total Tokens by 30.5%. On StrategyQA, selective communication may also reduce interference from dense context. This experiment tests CoS + Full Communication against CoS-Web while holding the CoS protocol and stopping rules fixed; the results do not show an accuracy gain for CoS-Web on every task.

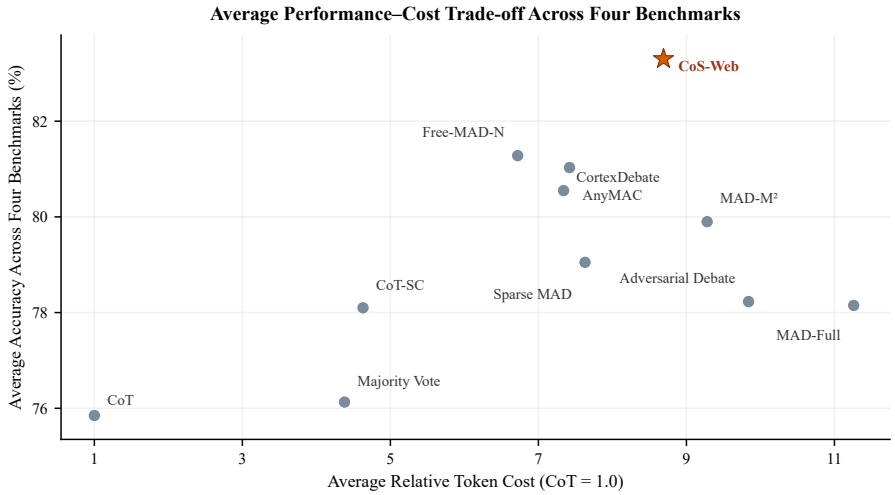


Figure F.5. Macro-average accuracy across the four benchmarks versus average Tokens/Q relative to CoT. The horizontal axis is a supplementary efficiency measure; the main-text evaluation focuses on accuracy.

F.4 GSM8K Accuracy–Cost Comparison

Table F.4. GSM8K Accuracy and Total Token Cost

Method	GSM8K Accuracy ↑	Total Tokens / Q ↓	Relative Cost (CoT=1.0) ↓
CoT	91.0	**1.4k**	**1.00**
CoT-SC	93.1	6.8k	4.86
Majority Vote	92.3	6.4k	4.57
MAD-Full	**95.4**	15.2k	10.86
Adversarial Debate	93.2	13.3k	9.50
Sparse MAD	95.0	10.1k	7.21
MAD-M^2	94.1	12.4k	8.86
CortexDebate	94.3	9.6k	6.86
AnyMAC	94.7	9.8k	7.00
Free-MAD-N	94.6	8.9k	6.36
CoS-Web	95.1	11.4k	8.14

F.5 Additional Accuracy–Communication Profile

Table F.5. Raw Average Token Cost (Token/Q)

Method	GSM8K	MATH-500	GPQA-D	StrategyQA	Four-task mean	Relative cost
CoT	1.4k	1.8k	1.6k	1.3k	1.53k	1.00
CoT-SC	6.8k	8.1k	7.4k	6.0k	7.08k	4.63
Majority Vote	6.4k	7.6k	7.0k	5.8k	6.70k	4.38
MAD-Full	15.2k	18.7k	21.8k	13.2k	17.23k	11.26
Adversarial Debate	13.3k	16.5k	18.6k	11.8k	15.05k	9.84
Sparse MAD	10.1k	12.8k	14.2k	9.6k	11.68k	7.63
MAD-M^2	12.4k	15.6k	17.3k	11.5k	14.20k	9.28
CortexDebate	9.6k	11.7k	13.5k	10.1k	11.23k	7.34
AnyMAC	9.8k	12.1k	13.7k	9.8k	11.35k	7.42
Free-MAD-N	8.9k	10.9k	12.1k	9.2k	10.28k	6.72
CoS-Web	11.4k	13.8k	17.1k	10.9k	13.30k	8.69

The four-task mean is the arithmetic mean of the raw Token/Q values in the four dataset columns; displayed values are rounded to two decimal places. Relative cost is defined as this mean divided by CoT's 1.53k. The accompanying figure summarizes macro-average accuracy against relative token use; token use is reported here as a secondary diagnostic.

F.6 Local-Ring Order Robustness

To examine the effects of canonical neighbor order and activation order for overlapping local rings, we randomly permute the within-ring member order and issue-package activation order using fixed seeds, without changing the D3 issues, member sets, or communication budget. For each evaluation seed, we use five predetermined permutations and average the results.

Ordering	GPQA-D ↑	StrategyQA ↑	GPQA Tokens/Q	StrategyQA Tokens/Q
Canonical order	**64.48±0.80**	**84.22±0.50**	17.1k	10.9k
Random ring permutation	64.41±0.84	84.08±0.54	17.1k	10.9k
Shuffled overlapping-group activation	64.31±0.86	84.01±0.56	17.1k	10.9k
Both orders randomized	64.24±0.88	83.88±0.58	17.1k	10.9k

All differences from canonical ordering are below 0.5 percentage points, indicating that the results are insensitive to the current ordering convention. This experiment does not deny the algorithmic order dependence of overlapping local rings; rather, it examines its empirical effect under the current setup.

F.7 Generic Critique Control

Generic Critique uses the same number of agents, the same Judge, and the same number of interaction rounds as CoS-Web, but replaces D1–D3 with a general "identify errors–critique–revise" process. The reported Token/Q values are similar; this comparison does not establish equivalence in the number of LLM calls or overall computational cost.

Method	GPQA-D ↑	Tokens/Q ↓	StrategyQA ↑	Tokens/Q ↓
MAD-Full	61.45±0.80	21.8k	70.95±0.60	13.2k
Generic Critique	62.79±0.83	17.0k	79.25±0.66	10.8k
CoS-Web	**64.48±0.80**	17.1k	**84.22±0.50**	10.9k

F.8 Matched-Budget Routing Control

This experiment fixes the number of participating agents, the number of local edges, and the approximate token budget for each sample, changing only the participant-selection rule.

Routing	GPQA-D ↑	Tokens/Q ↓	StrategyQA ↑	Tokens/Q ↓
Random Sparse Ring	63.13±0.86	17.0k	81.29±0.65	10.8k
Disagreement Routing	63.64±0.84	17.0k	82.04±0.62	10.8k
Confidence Routing	63.47±0.82	16.9k	82.45±0.60	10.7k
Issue-Driven Routing (CoS-Web)	**64.48±0.80**	17.1k	**84.22±0.50**	10.9k

G Diagnostic Analysis of D2 Nested Suspicion

From the first evaluation seed for each of GPQA-Diamond and StrategyQA, we sample without replacement 100 interactions with nonempty D1 suspicion lists, yielding a 200-interaction diagnostic subset. We compare the anticipated defense with the actual response in subsequent Ring Communication. Two annotators, unaware of the method names and final correctness, independently judge whether the match is Strong, Partial, or a Mismatch, and whether the deeper question is useful. A third annotator resolves disagreements. Cohen's κ for the two labels is 0.79 and 0.76, respectively.

Table G.1. Anticipated vs.~Actual Defenses

Relation	Cases	Ratio	Useful Deeper Question
Strong Match	83	41.5%	89.2%
Partial Match	79	39.5%	82.3%
Mismatch	38	19.0%	63.2%
Overall	**200**	**100%**	**81.5%**

D2 is not intended to predict precisely what the target agent will say next. Its purpose is to expose deeper verification questions by simulating anticipated defenses. Even when the prediction does not fully match the actual defense, the resulting question may still be useful for subsequent examination.

H Robustness to Correlated False Consensus

H.1 Accuracy under Shared-Assumption Corruption

Table H.1. Accuracy under Correlated Erroneous Assumptions

k	MAD-Full	Adv. Debate	Free-MAD-N	CoS-Web
0	61.45±0.80	61.43±0.80	63.30±0.80	**64.48±0.80**
1	54.55±1.00	56.23±1.10	59.93±0.90	**61.11±0.90**
2	44.78±1.20	48.99±1.30	52.86±1.10	**55.05±1.10**
3	32.49±1.50	39.39±1.50	43.94±1.30	**47.31±1.30**

From $k = 0$ to $k = 3$, the decreases for MAD-Full, Free-MAD-N, and CoS-Web are 28.96, 19.36, and 17.17 percentage points, respectively. Free-MAD-N is a competitive anti-conformity baseline. CoS-Web's additional advantage comes from progressively examining specific judgments, anticipated defenses, and assumptions shared by the group.

H.2 Error-Type Analysis

Two annotators, unaware of the method names, independently label the error types according to definitions provided in advance; a third annotator resolves disagreements. The overall Cohen's κ is 0.74. A sample may contain multiple error types, so counts across rows need not be mutually exclusive.

Table H.2. Error Incidence on 200 Challenging Examples

Error Type	MAD-M^2	CortexDebate	Free-MAD-N	CoS-Web
Unsupported assumption	37/200 (18.5%)	32/200 (16.0%)	26/200 (13.0%)	**21/200 (10.5%)**
Supporting-rationale omission	31/200 (15.5%)	27/200 (13.5%)	23/200 (11.5%)	**19/200 (9.5%)**
Premature consensus	29/200 (14.5%)	23/200 (11.5%)	16/200 (8.0%)	**13/200 (6.5%)**
Incorrect defense acceptance	21/200 (10.5%)	18/200 (9.0%)	16/200 (8.0%)	**13/200 (6.5%)**
Local reasoning/calculation	31/200 (15.5%)	**28/200 (14.0%)**	29/200 (14.5%)	29/200 (14.5%)

The results suggest that CoS-Web's gains are concentrated in interaction-related errors: it reduces unsupported assumptions, omissions of supporting rationales, premature consensus, and acceptance of incorrect defenses, while showing no clear advantage on local reasoning or calculation errors.

H.3 Recovery from Naturally Occurring False Consensus

In addition to artificial corruption, we select naturally occurring false-consensus cases from uncorrupted runs, defined as cases in which at least three agents give the same incorrect answer at the initial stage, and calculate the proportion for which the final answer is corrected. This selection uses only the initial responses and gold label; it does not depend on CoS output.

Dataset	Natural false-consensus cases	MAD-Full	Free-MAD-N	CoS-Web
GPQA-Diamond	42	9/42 (21.4%)	16/42 (38.1%)	**20/42 (47.6%)**
StrategyQA	96	27/96 (28.1%)	40/96 (41.7%)	**50/96 (52.1%)**

The recovery trend on the naturally occurring error subset is consistent with the correlated-error injection experiment, suggesting that the robustness results do not arise solely from recognizing a particular corruption template.

H.4 Case Study Details

The problem in Figure 2 asks for the angular distance between the first two minima in the far-field diffraction pattern of a regular N-sided aperture as $N \to \infty$. At the initial stage, four of the five

agents select incorrect answer C, while only one selects the correct answer B. Although most agents recognize that the aperture approaches a circular aperture, they misinterpret "the distance between the first two minima" as "the position of the first minimum."

D1 identifies the mismatch between the calculation target and the wording of the problem. D2 examines the anticipated defense that "the problem may be asking only for the first minimum." D3 turns the false consensus, shared assumption, and collective blind spot reflected in the reports into routable issues. Local communication then rechecks:

$$\theta_1 \approx 0.610\frac{\lambda}{a}, \qquad \theta_2 \approx 1.116\frac{\lambda}{a},$$

and therefore

$$\Delta\theta = \theta_2 - \theta_1 \approx 0.506\frac{\lambda}{a}.$$

The system ultimately corrects the answer from C, supported by the erroneous majority, to the correct answer B. This example illustrates how CoS triggers targeted verification by making shared erroneous rationales explicit, rather than merely adding debate rounds.

I Summary of Additional Analyses

I.1 Cross-Backbone Generalization

To examine whether the method depends on a single Qwen backbone, we repeat the paired comparison between MAD-Full and CoS-Web on GPQA-Diamond and StrategyQA using Llama-3.1-8B-Instruct. For each setting, models from the same family serve as both reasoning agents and the Judge, while the number of agents, prompt structure, and maximum number of rounds remain unchanged.

Backbone	Method	GPQA-D ↑	StrategyQA ↑
Qwen3-14B-AWQ	MAD-Full	61.45±0.80	70.95±0.60
	CoS-Web	**64.48±0.80**	**84.22±0.50**
Llama-3.1-8B-Instruct	MAD-Full	54.88±1.05	67.62±0.78
	CoS-Web	**58.08±1.00**	**75.71±0.72**

CoS-Web retains its gains over MAD-Full on a second model family, indicating that the main trend is not unique to Qwen3-14B-AWQ. Absolute performance still varies with backbone capability.

I.2 Paired Statistical Analysis

Because all methods share exactly the same initial agent responses within each random seed, we perform 10,000 paired bootstrap resamples of sample-level correctness differences and report difference intervals aggregated across the three random seeds.

Comparison	Dataset	Accuracy difference	95% paired bootstrap CI
CoS-Web − MAD-Full	GPQA-D	+3.03	[0.84, 5.22]
	StrategyQA	+13.27	[10.71, 15.82]
CoS-Web − Free-MAD-N	GPQA-D	+1.18	[−0.67, 3.03]
	StrategyQA	+4.12	[2.08, 6.16]
CoS-Web − AnyMAC	GPQA-D	+1.35	[−0.54, 3.24]
	StrategyQA	+4.70	[2.58, 6.82]
CoS-Web − Generic Critique	GPQA-D	+1.69	[−0.20, 3.58]
	StrategyQA	+4.97	[2.82, 7.12]

On GPQA-Diamond, the intervals for the smaller differences from the closest baselines include zero. We therefore do not describe this individual result as a definitive advantage. The differences on StrategyQA and those relative to MAD-Full are more stable.

I.3 Overall Summary

The additional analyses examine agent scale, the Judge and D3, prompt diversity, maximum interaction rounds, communication structure, local-ring order, model family, and paired uncertainty. The results indicate that CoS-Web's main gains concern unsupported assumptions, premature consensus, and correlated errors rather than isolated calculation errors. In addition, the generic-critique and routing controls show higher observed accuracy for CoS-Web at similar reported token usage. These comparisons do not establish equivalence in overall test-time compute.