GCL−MAD: GRAPH-GUIDED CONFLICT LOCALIZATION FOR LLM MULTI-AGENT DEBATE

Anonymous authors
Paper under double-blind review

ABSTRACT

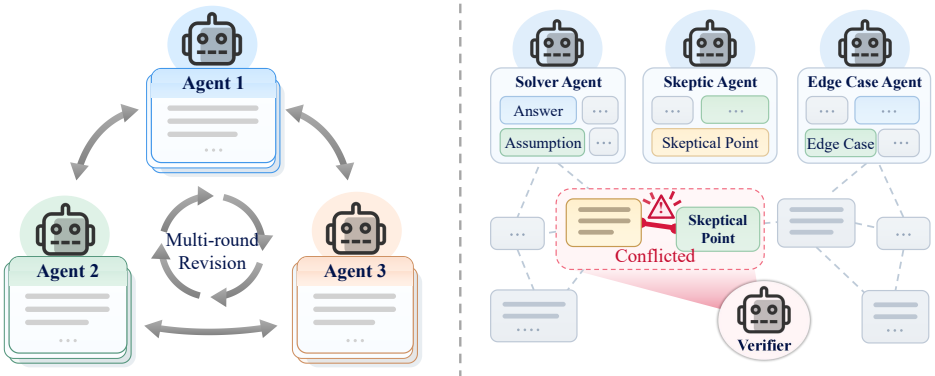
Multi-agent debate (MAD) has emerged as a promising paradigm for improving the reasoning capabilities of large language models. However, final decisions in existing MAD may be influenced by arguments presented with greater fluency or confidence rather than the actual correctness of reasoning. Moreover, intermediate reasoning errors are often embedded within plausible complete reasoning trajectories, making it difficult to identify which specific steps are responsible for an incorrect conclusion. To address these limitations, we propose GCL−MAD, a structured multi-agent debate framework that shifts debate from exchanging complete reasoning trajectories to reasoning about verifiable epistemic components. Specifically, outputs from agents with complementary roles are organized into structured fields, such as an answer, assumptions, or skeptical points or boundary conditions, with each populated field represented as one epistemic component node. Based on these components, a graph-construction agent organizes potential relationships among these components into an Epistemic Conflict Graph, where conflict edges are associated with explicit rationales describing their underlying causes. Guided by this graph, GCL−MAD identifies conflict-related edges and their associated components, enabling targeted verification only within these regions rather than re-verifying complete reasoning trajectories. Experiments across six reasoning benchmarks and two backbone models show that GCL−MAD consistently improves reasoning performance and supports more reliable answer correction through localized verification.

1 INTRODUCTION

Large language models (LLMs) have achieved remarkable progress in complex reasoning tasks (Brown et al., 2020; OpenAI et al., 2024; Chowdhery et al., 2023). To improve reasoning reliability beyond a single model, recent research has explored collaborative reasoning paradigms that leverage interactions among multiple agents (Chen et al., 2024; Cross et al., 2025; Wang et al., 2025). Motivated by human group debate and iterative critique, multi-agent debate (MAD) has emerged as a promising framework for enhancing LLM reasoning. In existing MAD frameworks, agents independently produce reasoning trajectories, iteratively revise each other's responses, and conclude with aggregation (e.g., voting or consensus-based selection) to produce the final answer (Du et al., 2024; Smit et al., 2024). By enabling multiple reasoning perspectives to interact, MAD has demonstrated promising improvements in a variety of challenging tasks.

Existing MAD research has explored various strategies to improve the effectiveness and efficiency of multi-agent collaboration. Some research has investigated debate protocols and aggregation strategies, where agents iteratively exchange responses, critique each other, and then derive final decisions through mechanisms such as majority voting or consensus-based selection (Du et al., 2024; Smit et al., 2024; Chan et al., 2024; Feng et al., 2025). Another line of work has focused on reducing the cost and improving the stability of debate processes. For example, sparse communication strategies reduce unnecessary message exchanges (Li et al., 2024), while adaptive debate mechanisms identify stable stopping points or improve convergence behaviors during multi-round interactions (Hu et al., 2025; Liang et al., 2024; Yi et al., 2025). More recent studies further explored dynamic agent coordination and role specialization to improve the diversity and effectiveness of reasoning interactions (Fan et al., 2026; Gu et al., 2026). These studies demonstrated that improving collaborative reasoning among multiple agents consistently enhances reasoning capabilities. Despite their

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.



(a) Conventional MAD: Revise the Full Reasoning Traces

(b) GCL-MAD: Localize Conflicts and Verify

Figure 1: Comparison between conventional multi-agent debate and the proposed GCL−MAD framework.

different objectives, these approaches mainly refine agent interaction and decision-making, while intermediate correctness is left implicit in the final trajectory.

During debate, agents mainly exchange and revise natural language arguments. As a result, the final decision can be influenced by the coherence or confidence of an argument, rather than solely by the correctness of the underlying reasoning. If incorrect reasoning is incorporated into subsequent interactions, errors may propagate across agents. Furthermore, reasoning failures are difficult to attribute to their actual sources because intermediate reasoning elements are entangled within full trajectories. A wrong conclusion may result from an invalid assumption, an incorrect intermediate step, or a missed constraint, yet these factors are often mixed together in the same reasoning trace. Without identifying the specific source of failure, MAD systems cannot perform targeted verification and reliable correction. These limitations suggest that improving MAD reliability requires moving beyond trajectory-level interaction toward a more structured and verifiable representation of reasoning. What is important is no longer how agents debate, but what they should verify, which remains largely overlooked in existing MAD frameworks.

We propose GCL−MAD, a structured multi-agent debate framework that changes the unit of debate from complete reasoning trajectories to verifiable epistemic components. As illustrated in Fig. 1, conventional MAD exchanges and revises complete reasoning traces round by round, whereas GCL−MAD localizes epistemic conflicts on a graph and verifies only conflicted components. First, complementary agents generate structured reasoning signals from distinct perspectives. These signals are normalized into epistemic component nodes, each representing one populated output field, such as an answer or a collection of assumptions, reasoning steps, skeptical points, or boundary conditions. Then, a graph-construction agent organizes these nodes into an Epistemic Conflict Graph. The graph contains explicit rationales explaining why particular components may be conflicting. High-confidence conflict-related edges are retained to identify the reasoning components that require further examination. Finally, GCL−MAD performs targeted verification only on the retained conflict-related edges and their associated epistemic components. An adjudicator agent integrates the original answer, localized conflict evidence, and verification outcomes to determine whether the answer should be revised. This design transforms multi-agent debate from unrestricted trajectory exchange to component-level conflict localization, verification, and evidence-grounded adjudication. In summary, our main contributions are as follows:

- We propose GCL−MAD, a structured multi-agent debate framework that shifts the unit of debate from complete reasoning trajectories to epistemic components and models conflicts among these components using an explicit graph representation.
- We introduce a conflict-driven reasoning mechanism that constructs epistemic conflict graphs and performs selective verification to identify and correct reasoning errors.
- We conduct extensive experiments on six reasoning benchmarks, demonstrating that GCL−MAD achieves competitive accuracy across six reasoning benchmarks, particularly on problems involving implicit assumptions and hidden constraints.

2 METHODOLOGY

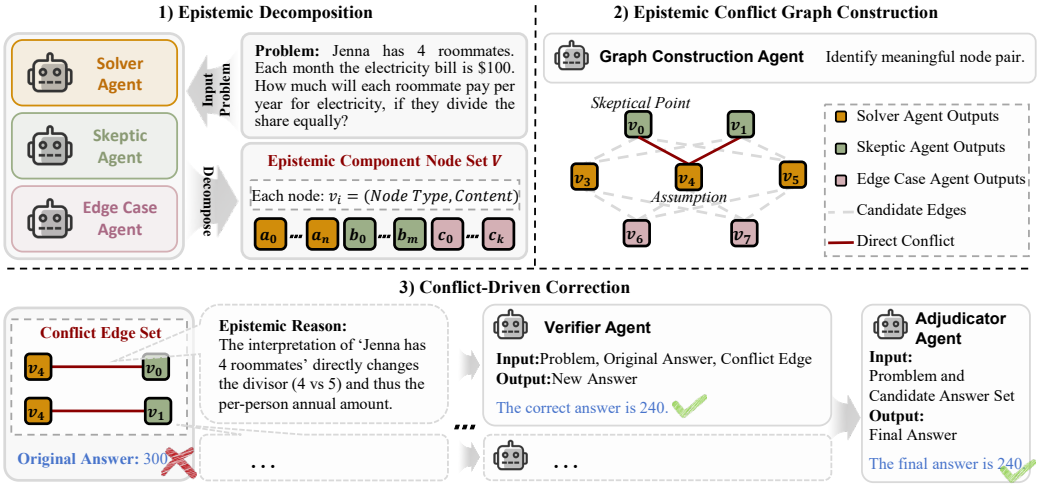


Figure 2: Overview of the proposed GCL−MAD framework.

To improve the reasoning capability of MAD frameworks, we present the GCL−MAD framework. Given an input problem x, GCL−MAD first elicits complementary structured outputs from multiple role-specialized agents and normalizes their populated fields into epistemic component nodes. It then constructs an Epistemic Conflict Graph to identify conflict-related relationships and their underlying rationales. Finally, only the retained conflict-related edges and their associated components are verified, and an adjudicator revises the original answer only when the verification outcomes provide sufficient support for an alternative answer. In GCL−MAD, debate is therefore implemented through structured contention and verification among epistemic components, rather than through the unrestricted exchange of complete reasoning trajectories. Figure 2 illustrates the overall framework.

2.1 ROLE-GUIDED EPISTEMIC DECOMPOSITION

Existing multi-agent reasoning frameworks commonly represent collaboration at the agent level, where nodes correspond to agents and edges specify communication or message-passing patterns (Zhuge et al., 2024; Li et al., 2024; Zhang et al., 2025; Yun et al., 2026; Li et al., 2026; Dong et al., 2026). Although such formulations enable flexible interaction topologies, the exchanged information is still typically represented by complete or free-form reasoning trajectories. Consequently, assumptions, intermediate steps, skeptical points, and boundary conditions remain entangled within lengthy textual outputs, making these components difficult to inspect, compare, and verify independently.

To obtain a component-level representation at the granularity of structured output fields, we introduce role-guided epistemic decomposition. Given an input problem x, multiple reasoning perspectives generate complementary structured outputs:

$$\mathcal{O} = \mathcal{F}_{\text{role}}(x), \tag{1}$$

where $\mathcal{O}$ denotes the set of structured outputs produced by reasoning role agents. Each agent focuses on exposing epistemic information associated with a particular reasoning function, rather than producing a complete, indivisible reasoning trajectory. These outputs contain both candidate answers and supporting epistemic information, which are further normalized into typed epistemic component nodes:

$$V = \mathcal{F}_{\text{norm}}(\mathcal{O}) = \{v_i\}_{i=1}^{N}, \tag{2}$$

where N denotes the number of epistemic nodes extracted from the role-guided outputs. Each epistemic node is represented as $v_i = (t_i, s_i)$, where $t_i \in \mathcal{T}$ denotes the epistemic type, s_i is the natural-language content of the component. The type set $\mathcal{T}$ includes answer, reasoning-step, assumption, uncertainty, and boundary-condition nodes. This decomposition transforms heterogeneous agent outputs into a unified representation with one node per populated field. Conflict localization therefore selects pairs of fields for verification, allowing the verifier to inspect the assumptions, reasoning

steps, skeptical points, or boundary conditions contained in those fields. The resulting node set V provides the representational basis for constructing explicit epistemic relations during the subsequent conflict-graph construction stage.

In our implementation, we instantiate three complementary reasoning roles to encourage diverse reasoning perspectives, including a Solver Agent, a Skeptical Agent, and an Edge-case Agent, to expose diverse epistemic signals from solution generation, reasoning critique, and boundary analysis. The Solver Agent produces the initial answer together with supporting assumptions and reasoning steps, the Skeptical Agent identifies potentially invalid skeptical points and verification targets, and the Edge-case Agent explores boundary conditions and alternative interpretations. Their outputs are mapped into the common epistemic node space defined above.

2.2 Epistemic Conflict Graph Construction

To enable targeted verification, conflicts among epistemic components need to be explicitly represented rather than implicitly inferred from complete reasoning trajectories. Given the epistemic node set V, we construct an Epistemic Conflict Graph by identifying conflict relations among reasoning components.

Rather than considering all possible node pairs, we first select a set of type-compatible candidate component pairs $P_{\text{cand}} = \mathcal{C}(V) \subseteq V \times V$. For each candidate pair $(v_i, v_j) \in P_{\text{cand}}$, a graph-construction agent determines whether an epistemic conflict exists between the two components. For identified conflicts, the agent generates a candidate conflict edge set

$$E_{\text{cand}} = \{e_k = (v_i, v_j, r_k, m_k, w_k) \mid k = 1, \ldots, M\}, \tag{3}$$

where M denotes the number of candidate conflict edges. Here, $r_k \in \mathcal{R}_{\text{conf}}$ denotes the conflict-oriented edge type, m_k is a natural-language rationale explaining why the two components may require further examination, and w_k represents the confidence score of the identified relation. Then, candidate edges are filtered according to their confidence scores, retaining high-confidence edges to form the final edge set E, as detailed in Appendix D.2. The resulting graph is defined as $G = (V, E)$.

The resulting graph contains only high-confidence conflict-oriented relations, which are used to identify epistemic components requiring further examination. Therefore, the Epistemic Conflict Graph serves as a structured verification routing mechanism, explicitly indicating where conflicts occur and avoiding re-evaluation of complete reasoning trajectory.

2.3 Conflict-Triggered Verification and Adjudication

Once the Epistemic Conflict Graph is constructed, `GCL-MAD` verifies only the retained conflict-related edges and their associated epistemic components. Based on the identified conflict edges, `GCL-MAD` performs edge-level verification to distinguish actual reasoning errors from harmless disagreements. The resulting verification evidence is then aggregated for final adjudication, where answer revisions are made only when supported by explicit evidence.

2.3.1 Edge-Level Conflict Verification.

For each conflict-related edge $e_k \in E$, a verifier agent receives the original problem x, the initial answer a_0, a node subset $V_k \subseteq V$, and the conflict edge e_k with its rationale. The subset V_k contains the two endpoint nodes of e_k and the baseline answer node a_0 when present. Each included node retains its complete field content; if a reasoning-step node is an endpoint, the verifier receives all steps in that field. It then evaluates whether the identified conflict invalidates the original answer:

$$(\tilde{a}_k, \rho_k) = \mathcal{F}_{\text{ver}}(x, a_0, V_k, e_k), \tag{4}$$

where $\tilde{a}_k$ denotes the answer supported by the verification result and ρ_k denotes the associated verification rationale.

If the conflict does not invalidate the original reasoning, the verifier retains the original answer. Otherwise, it proposes an alternative answer supported by localized verification evidence. The veri-

fication results that produce alternative answers are collected as[1]

$$\mathcal{Z} = \{(e_k, \tilde{a}_k, \rho_k) \mid e_k \in E, \tilde{a}_k \neq a_0\}. \tag{5}$$

The candidate-answer set is constructed by

$$\mathcal{A} = \{a_0\} \cup \{\tilde{a}_k \mid (e_k, \tilde{a}_k, \rho_k) \in \mathcal{Z}\}. \tag{6}$$

2.3.2 VERIFICATION-GROUNDED ADJUDICATION.

Instead of relying on majority voting, GCL−MAD performs evidence-grounded adjudication. If all verification results support the original answer, GCL−MAD directly returns original answer. Otherwise, a dedicated adjudicator receives the original answer, the candidate answers, and their corresponding verification evidence, and selects the final answer based on the provided evidence. Formally,

$$y^* = \mathcal{F}_{\text{judge}}(x, \mathcal{A}, \mathcal{Z}). \tag{7}$$

To avoid unnecessary revisions, the adjudicator follows a conservative decision strategy: the original answer is retained by default and replaced only when the verification evidence provides stronger support for an alternative answer. Consequently, answer correction is driven by explicit epistemic evidence rather than answer frequency or persuasive dialogue.

Appendix D.2 presents the details of the Epistemic Conflict Graph constructed. Algorithm 1 in appendix B presents an overview of GCL−MAD.

3 EXPERIMENTS

We report experiments on six reasoning benchmarks with two backbone LLMs. Our experiments include (1) comparisons with existing reasoning baselines, (2) ablation studies on key components, (3) answer transition analysis to evaluate correction reliability, and (4) judge intervention analysis to examine selective verification behavior. All prompts used in this work are provided in Appendix D.

3.1 EXPERIMENTAL SETUP

3.1.1 DATASETS.

We conduct experiments on six representative reasoning benchmarks. GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) evaluate mathematical reasoning. CommonsenseQA (CSQA) (Talmor et al., 2019) and ARC-Challenge (ARC-C) (Clark et al., 2018) assess commonsense and scientific reasoning abilities, respectively. MMLU-Pro(MMLU-P) (Wang et al., 2024) and GPQA (Rein et al., 2023) further evaluate knowledge-intensive reasoning across a wide range of domains. More details on dataset setups are provided in Appendix B.1.

3.1.2 BASELINES.

We compare our method against two categories of agentic reasoning baselines. (1) **Single-agent reasoning methods**, including Vanilla prompting, Chain-of-Thought (CoT) (Wei et al., 2022), Self-Consistency (SC) (Wang et al., 2023), CompLexCoT (Fu et al., 2023), and Self-Refine (SR) (Madaan et al., 2023). (2) **Multi-agent debate methods**, including Multi-Agent Debate (MAD) (Du et al., 2024), MAD with Adaptive Stability Detection (MAD-ASD) (Hu et al., 2025), ECON (Yi et al., 2025), iMAD (Fan et al., 2026), and MADC (Zhang et al., 2026). These approaches improve reasoning through interactions among multiple agents. More details on baseline setups are provided in Appendix C.1.

3.1.3 IMPLEMENTATION DETAILS.

We leverage *GPT-5-mini* and *DeepSeek-V4-Flash* as backbone language models for all experiments. All models are accessed through their respective APIs. Unless otherwise specified, the conflict-edge retention threshold is fixed at $\tau = 3$ for all experiments. For fair comparison, we use the

[1]Answer comparison and candidate deduplication use dataset-specific normalized values. Mathematical equivalence is not checked before adjudication; equivalent MATH expressions may therefore remain distinct. See Appendix D.1 for normalization and scoring details.

Table 1: Accuracy (%) on six reasoning benchmarks with GPT-5-mini and DeepSeek-V4-Flash. Results are reported as mean ± standard deviation over three independent runs. The best mean result in each column is shown in **bold**, and the second-best mean result is underlined.

Method	GPT-5-mini						DeepSeek-V4-Flash					
	GSM8K	MATH	CSQA	ARC-C	MMLU-P	GPQA	GSM8K	MATH	CSQA	ARC-C	MMLU-P	GPQA
Single Agent												
Vanilla	84.11±1.07	78.00±1.53	84.67±1.15	96.09±0.84	73.33±0.58	68.30±0.22	93.56±1.17	76.67±0.88	82.44±1.50	95.53±0.84	73.78±1.39	46.28±0.46
CoT	86.11±0.77	83.67±1.33	87.67±1.00	95.86±1.08	79.33±0.88	67.49±0.34	93.22±0.96	75.22±1.26	84.67±1.15	95.97±1.16	75.56±1.39	45.01±0.93
SC	91.56±1.07	85.33±1.00	87.44±0.84	95.97±0.89	77.78±1.07	72.77±0.80	95.22±0.19	79.44±0.69	**87.44**±0.84	94.85±0.84	78.89±0.77	59.75±0.46
CompLexCoT	88.56±0.38	86.33±1.20	87.22±1.39	97.87±0.84	79.56±0.51	72.62±0.56	95.22±1.39	79.44±0.77	83.78±1.35	96.20±1.69	79.89±0.19	58.63±0.72
SR	86.44±1.02	84.00±0.58	88.00±1.00	98.43±0.84	78.67±1.53	71.65±0.67	96.11±1.84	77.78±1.07	86.11±1.02	96.31±1.54	79.33±1.00	52.38±0.78
Multi-Agent Debate												
$MAD_{T=3}$	88.00±0.88	86.00±1.67	86.67±0.88	96.98±1.54	78.78±1.39	73.96±0.90	94.44±0.84	81.11±1.84	84.44±0.51	**96.76**±0.77	78.33±0.67	60.42±0.34
ECON	87.33±0.67	78.44±0.69	88.11±0.19	94.07±1.03	78.11±1.17	72.47±0.34	95.56±0.19	76.67±0.88	84.33±1.15	94.74±0.77	76.67±1.15	59.00±0.72
MAD-ASD	91.67±1.15	87.89±0.19	**89.22**±0.69	96.87±1.03	77.78±1.39	60.42±0.56	93.56±0.69	80.33±0.58	83.56±0.69	95.53±1.08	70.00±1.15	31.40±0.72
iMAD	92.89±1.39	85.56±1.02	85.78±1.07	95.97±0.89	79.22±1.39	65.62±0.39	94.22±0.38	78.33±2.00	86.44±1.39	96.64±1.21	80.56±0.69	57.51±0.26
MADC	94.11±0.77	87.67±0.33	88.44±1.07	97.87±1.27	77.89±1.07	61.24±0.68	92.22±0.51	70.78±0.38	82.33±1.53	94.07±1.08	61.89±0.96	31.99±0.26
GCL−MAD (Ours)	**96.56**±0.19	**88.89**±1.02	88.11±0.96	**98.55**±1.03	**79.78**±0.69	**74.78**±0.77	**96.56**±0.51	**83.89**±1.68	85.89±1.17	96.64±1.16	**80.67**±1.15	**61.24**±0.46

same generation configuration across methods within each backbone, with a maximum output length of 4096 tokens.[2] Each experiment is repeated three times using three fixed random seeds, with each seed defining a different sampled data subset. Unless otherwise specified, all baselines are implemented under the same experimental settings.

3.2 Main Results

We evaluate GCL−MAD on six reasoning benchmarks using GPT-5-mini and DeepSeek-V4-Flash, and compare it with both single-agent reasoning methods and multi-agent debate baselines. As shown in Table 1, GCL−MAD demonstrates competitive performance across the evaluated benchmarks and backbone models. The results support the effectiveness of component-level conflict localization for improving the reliability of multi-agent reasoning. The gains are more pronounced on more challenging reasoning benchmarks, including MATH, MMLU-P, and GPQA, where GCL−MAD achieves strong improvements across the evaluated backbone models. By organizing agent outputs into fields and verifying the contents of fields connected by retained conflict edges, GCL−MAD can identify and correct reasoning failures while reducing the risk of unnecessary modifications to already reliable predictions.

Compared with conventional MAD variants, GCL−MAD shows competitive and relatively stable performance across datasets and backbone models. Some existing methods exhibit substantial performance variations when the backbone model or task changes. In contrast, GCL−MAD reduces unnecessary revisions and improves the reliability of final decisions. However, GCL−MAD does not consistently dominate on all datasets. On CSQA and ARC-C, the relative gains are more limited compared with tasks involving deeper reasoning. This is because these benchmarks often emphasize commonsense knowledge recognition or short-form reasoning, where aggregating multiple reasoning perspectives may already provide sufficient information. For such tasks, methods such as Self-Refine may already improve responses through iterative self-feedback. The comparison across backbone models further reveals that MAD performance depends not only on the debate mechanism itself but also on the underlying reasoning capability of the base model. Overall, GPT-5-mini achieves stronger results than DeepSeek-V4-Flash across most benchmarks, suggesting that multi-agent frameworks can amplify existing reasoning abilities but cannot fully compensate for limitations of the backbone model. Nevertheless, GCL−MAD remains competitive across both backbone models, indicating that its benefit comes from the proposed reasoning structure rather than model-specific characteristics.

3.3 Ablation Study

We conduct ablation studies on MATH, MMLU-P, and GPQA to evaluate the contribution of the main components in GCL−MAD as shown in Figure 3. We use w/o Skep., w/o ECase, and w/o ECG to denote variants without the Skeptic Agent, Edge-case Agent, and Epistemic Conflict Graph,

[2]GPT-5-mini uses its supported/default temperature setting of 1.0. DeepSeek-V4-Flash is run in thinking mode with the default reasoning effort set to `high`; although `temperature=1.0` is supplied, it does not affect generation in this mode.

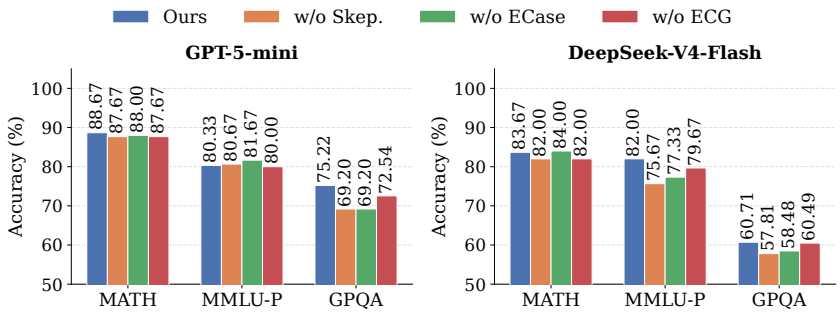


Figure 3: Ablation results of GCL−MAD on MATH, MMLU-P, and GPQA using GPT-5-mini and DeepSeek-V4-Flash.

respectively. In w/o ECG, all three role agents still produce their structured outputs, but graph construction and conflict-guided verification are removed. The adjudicator directly receives the original problem and the structured outputs from the role agents without localized conflict evidence. This variant therefore evaluates the contribution of the ECG-guided verification pathway.

The variants generally underperform the full method, indicating that the role agents and the graph-guided verification pathway contribute to its observed performance. The effect of the Skeptic Agent is particularly evident on GPQA, where removing skeptical analysis causes a performance drop for both backbone models. This indicates that explicitly challenging assumptions and identifying potentially unreliable reasoning elements is important for knowledge-intensive reasoning tasks, where errors often originate from hidden premises rather than surface-level derivations. The Edge-case Agent provides additional benefits by exploring boundary conditions. Although removing this module does not consistently reduce performance on every benchmark, the full model achieves more stable results across tasks. Removing the ECG-guided verification pathway decreases performance across most evaluated settings, suggesting that diverse reasoning signals alone are insufficient without explicit conflict localization and targeted verification.

3.4 Answer Transition Analysis

Table 2: Answer transition analysis on MATH and GPQA. Transitions are measured from each method's initial single-agent prediction to its final output after debate or verification. $\times \to \checkmark$ denotes cases corrected from wrong to correct, while $\checkmark \to \times$ denotes cases changed from correct to wrong. ΔACC denotes the net accuracy change, calculated as $\times \to \checkmark - \checkmark \to \times$. All transition values and ΔACC are reported as percentages (%).

	GPT-5-mini						DeepSeek-V4-Flash					
	MATH			**GPQA**			**MATH**			**GPQA**		
Method	$\times \to \checkmark$	$\checkmark \to \times$	**ΔACC**	$\times \to \checkmark$	$\checkmark \to \times$	**ΔACC**	$\times \to \checkmark$	$\checkmark \to \times$	**ΔACC**	$\times \to \checkmark$	$\checkmark \to \times$	**ΔACC**
$\text{MAD}_{T=3}$	0.33	1.00	↓ 0.67	5.13	0.45	↑ 4.68	2.67	0.60	↑ 2.07	9.60	0.00	↑ 9.60
MAD-ASD	0.33	0.33	0.00	0.89	0.45	↑ 0.44	0.33	0.00	↑ 0.33	0.00	0.00	0.00
MADC	0.33	0.00	↑ 0.33	1.12	1.56	↓ 0.44	0.00	0.00	0.00	0.45	3.79	↓ 3.34
GCL−MAD	1.33	0.60	↑ 0.73	5.80	0.00	↑ 5.80	2.67	0.00	↑ 2.67	3.57	0.23	↑ 3.34

We further analyze how different debate methods modify model predictions before and after interaction. Table 2 reports answer transitions on MATH and GPQA. $\times \to \checkmark$ denotes cases where an initially incorrect answer is corrected. $\checkmark \to \times$ denotes cases where an initially correct answer is changed to an incorrect one. The results support the effectiveness of component-level conflict localization in achieving more reliable reasoning correction.

Across different benchmarks and backbone models, GCL−MAD consistently produces positive accuracy gains while maintaining relatively low rates of incorrect revisions. The advantage is particularly clear on GPT-5-mini GPQA, where GCL−MAD achieves the largest net accuracy gain without introducing any incorrect revisions. The comparison with conventional MAD methods further highlights the importance of balancing effective correction with revision reliability. For example, on DeepSeek-V4-Flash GPQA, $\text{MAD}_{T=3}$ achieves a larger wrong-to-right transition rate and a larger net accuracy gain than GCL−MAD, showing that conventional debate can be highly effective in some

settings. However, its performance is less consistent across benchmarks, as it also produces a negative net accuracy change on GPT-5-mini MATH. The transition analysis demonstrates that GCL−MAD improves multi-agent reasoning by consistently achieving positive accuracy gains while controlling unnecessary revisions across different settings.

3.5 Controlled Comparison of Verification Routing

Table 3: Controlled comparison of verification routing strategies on MATH and GPQA with GPT-5-mini as the backbone model. For each example, all baseline variants are assigned the same number of verifier calls as ECG Routing, except All Compatible Edges, which exhaustively verifies all compatible candidate edges.

	MATH				GPQA			
Routing Method	W→C	C→W	Δ	Acc.	W→C	C→W	Δ	Acc.
ECG Routing	**1.33**	**0.67**	**+0.67**	**88.67**	**5.80**	**0.00**	**+5.80**	**75.22**
Flat Conflict Ranking	1.33	1.33	0.00	88.00	5.13	0.22	+4.91	74.33
Disagreement-only Routing	1.00	1.00	0.00	88.00	4.46	0.45	+4.02	73.44
Lexical Routing	1.00	1.67	-0.67	87.33	4.02	0.67	+3.35	72.77
Random Routing	0.67	1.67	-1.00	87.00	3.57	0.00	+3.57	72.99
Full-trajectory Verification	1.33	2.67	-1.33	86.67	4.02	2.68	+1.34	70.76
All Compatible Edges	2.00	3.67	-1.67	86.33	6.25	6.70	-0.45	68.97

To isolate the contribution of the proposed Epistemic Conflict Graph (ECG) as a verification-routing mechanism, we conduct controlled comparisons on MATH and GPQA using GPT-5-mini as the backbone model. All variants reuse the same role-agent outputs and keep the verifier and adjudication procedure unchanged. For each example i, let K_i denote the number of conflict edges retained by ECG Routing. All baseline variants use K_i verifier calls for the same example, except *All Compatible Edges*, which exhaustively verifies the entire compatible candidate-edge set. This design controls the amount of verification while varying how verification targets are selected.

We consider five matched-budget alternatives. *Flat Conflict Ranking* retains the conflict-review signal but removes the graph-based merging and structural filtering rules, selecting the top-K_i reviewed conflicts directly. *Disagreement-only Routing* selects verification targets according to textual disagreement between Solver components and skeptical or edge-case components, without explicitly modeling typed epistemic relations. *Lexical Routing* ranks compatible component pairs using lexical similarity and verifies the top-K_i pairs. *Random Routing* uniformly samples K_i compatible edges and serves as a control for whether additional verification alone is sufficient. Finally, *Full-trajectory Verification* uses the same number of verifier calls as ECG, but each call re-examines the complete Solver reasoning trajectory rather than a localized conflict. In addition, *All Compatible Edges* verifies every compatible candidate edge to test whether exhaustive verification is preferable to selective routing.

Table 3 reports the resulting answer transitions. Since all matched-budget methods start from the same Solver predictions, differences in final accuracy can be directly attributed to their net correction behavior. We report the percentage of initially incorrect answers corrected after verification ($\mathrm{W}\rightarrow\mathrm{C}$), the percentage of initially correct answers changed to incorrect ones ($\mathrm{C}\rightarrow\mathrm{W}$), and the resulting net accuracy change $\Delta\mathrm{Acc} = \mathrm{W}\rightarrow\mathrm{C} - \mathrm{C}\rightarrow\mathrm{W}$.

The comparison highlights two complementary effects of ECG routing. First, under the same verification budget, ECG achieves larger net correction gains while introducing fewer or comparable erroneous revisions than alternative routing strategies. In particular, random and lexical routing provide substantially weaker net gains, indicating that the benefit does not arise merely from invoking the verifier or selecting textually related components. Disagreement-only routing and flat conflict ranking are stronger alternatives, but remain below ECG, suggesting that explicitly organizing and filtering epistemic relations provides additional value beyond detecting disagreement or ranking candidate conflicts independently.

Second, broader verification does not necessarily lead to more reliable correction. Full-trajectory verification can identify some initially incorrect answers, but its larger $\mathrm{C}\rightarrow\mathrm{W}$ rate indicates that re-examining the complete reasoning trajectory may also introduce unnecessary revisions. Similarly,

verifying all compatible edges exposes the adjudication stage to substantially more candidate alternatives, which can offset the additional correct revisions with incorrect ones. These results support the role of ECG as a selective verification-routing mechanism: its benefit arises not from increasing the amount of verification, but from concentrating verification on localized epistemic conflicts that are most relevant to the correctness of the final answer.

3.6 JUDGE INTERVENTION ANALYSIS

Table 4: Judge intervention analysis across six benchmarks using GPT-5-mini and DeepSeek-V4-Flash. J. Rate denotes the percentage of examples where the judge module is triggered. I. Acc denotes the accuracy on the intervention subset. Edges/Q denotes the average number of retained conflict-related edges per example, averaged over all examples, including examples for which adjudication is not triggered.

	GPT-5-mini			DeepSeek-V4-Flash		
Dataset	**J. Rate**	**I. Acc**	**Edges/Q**	**J. Rate**	**I. Acc**	**Edges/Q**
GSM8K	2.3%	0.857	2.30	6.0%	0.667	0.62
MATH	4.7%	0.786	2.35	3.7%	1.000	0.14
CSQA	12.3%	0.649	3.67	17.3%	0.635	2.55
ARC-C	5.4%	0.875	3.15	3.4%	0.700	1.51
MMLU-P	9.0%	0.630	2.94	4.7%	0.643	1.35
GPQA	12.9%	0.534	2.44	6.9%	0.600	1.04

We analyze the behavior of the judge module to examine whether GCL−MAD performs verification and adjudication selectively rather than relying on global re-evaluation. Together with the transition analysis, these results indicate that GCL−MAD improves reasoning reliability through selective conflict identification and evidence-grounded adjudication, rather than increasing unrestricted interaction or repeated answer revision.

As shown in Table 4, the judge is activated only for a small fraction of examples across different benchmarks and backbone models. The intervention patterns also vary across tasks, with benchmarks such as GPQA and CSQA triggering adjudication more frequently. Importantly, J. Rate measures only how often adjudication is triggered and does not directly reflect how often verification is performed, since retained edges may be verified without producing an alternative answer. Edges/Q is averaged over all examples and reflects the average number of retained conflict-related edges, and thus verifier calls, per example. Overall, these results show that GCL−MAD performs adjudication selectively while verification is guided by a limited set of localized conflict edges rather than complete reasoning trajectories.

4 CONCLUSION

In this paper, we proposed GCL−MAD, a structured multi-agent debate framework that localizes conflicts between fields of structured agent outputs. GCL−MAD represents each populated output field as an epistemic component node, constructs an Epistemic Conflict Graph, and selectively verifies conflict edges and the contents of their associated fields. This design enables correcting reasoning errors caused by hidden assumptions or complex constraints. Experiments across six reasoning benchmarks and two backbone models show that GCL−MAD achieves competitive reasoning performance across diverse settings while supporting reliable answer correction through localized verification. Future work will extend component-level conflict localization to more open-ended reasoning tasks. We envision moving beyond fixed decomposition and verification pipelines toward more autonomous reasoning systems. Such systems can dynamically discover epistemic structures, adapt reasoning roles, and plan verification trajectories according to the characteristics of different reasoning tasks.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript to improve readability, including polishing grammar, wording, and clarity. We reviewed all AI-assisted edits and verified that they did not alter the technical content, scientific claims, experimental results, or conclusions of the 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

We provide detailed experimental settings to facilitate reproduction of our results. The experimental setup are described in Section 3.1, with additional details on dataset and baseline configurations provided in Appendix C.1. All experiments are repeated using three fixed random seeds (42, 43, and 44), with each seed defining a different sampled data subset where subsampling is applied. We also provide an anonymous implementation repository at `https://anonymous.4open.science/r/GCL-MAD-586F`.

REFERENCES

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17682–17690. AAAI Press, 2024.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020*, 2020.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*. OpenReview.net, 2024.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*. OpenReview.net, 2024.

Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? *Advances in Neural Information Processing Systems*, 38: 101732–101764, 2025.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113, 2023.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. Navigate through enigmatic labyrinth A survey of chain of thought reasoning: Advances, frontiers and future. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1173–1203. Association for Computational Linguistics, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

Logan Matthew Cross, Violet Xiang, Agam Bhatia, Daniel L. K. Yamins, and Nick Haber. Hypothetical minds: Scaffolding theory of mind for multi-agent tasks with large language models. In *The Thirteenth International Conference on Learning Representations*. OpenReview.net, 2025.

Jiangwen Dong, Zehui Lin, Wanyu Lin, and Mingjin Zhang. S-DAG: A subject-based directed acyclic graph for multi-agent heterogeneous reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 29394–29402. AAAI Press, 2026.

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*, Proceedings of Machine Learning Research, pp. 11733–11763. PMLR / OpenReview.net, 2024.

Wei Fan, JinYi Yoon, and Bo Ji. imad: Intelligent multi-agent debate for efficient and accurate LLM inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 29403–29411. AAAI Press, 2026.

Zhaopeng Feng, Jiayuan Su, Jiamei Zheng, Jiahan Ren, Yan Zhang, Jian Wu, Hongwei Wang, and Zuozhu Liu. M-MAD: multidimensional multi-agent debate for advanced machine translation evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7084–7107. Association for Computational Linguistics, 2025.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Wenyuan Gu, Haowen Wang, Jiale Han, Xiang Li, Zhixuan Wu, Hongru Xiao, and Bo Cheng. Analyze-compose-execute: A dynamic dialogue framework for multi-agent debate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 30825–30833. AAAI Press, 2026.

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. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021*, 2021.

Tianyu Hu, Zhen Tan, Song Wang, Huaizhi Qu, and Tianlong Chen. Multi-agent debate for llm judges with adaptive stability detection. *Advances in Neural Information Processing Systems*, 38: 46504–46540, 2025.

Lars Benedikt Kaesberg, Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Voting or consensus? decision-making in multi-agent debate. In *Findings of the Association for Computational Linguistics*, Findings of ACL, pp. 11640–11671. Association for Computational Linguistics, 2025.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*, 2022.

Shiyuan Li, Yixin Liu, Yu Zheng, Mei Li, Quoc Viet Hung Nguyen, and Shirui Pan. OFA-MAS: one-for-all multi-agent system topology design based on mixture-of-experts graph generative models. In *Proceedings of the ACM Web Conference 2026*, pp. 1333–1344. ACM, 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*, Findings of ACL, pp. 7281–7294. Association for Computational Linguistics, 2024.

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. Association for Computational Linguistics, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, 2023.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023.

Andries P. Smit, Nathan Grinsztajn, Paul Duckworth, Thomas D. Barrett, and Arnu Pretorius. Should we be going mad? A look at multi-agent debate strategies for llms. pp. 45883–45905, 2024.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4149–4158. Association for Computational Linguistics, 2019.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-agents enhances large language model capabilities. In *The Thirteenth International Conference on Learning Representations*. OpenReview.net, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*. OpenReview.net, 2023.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*. OpenReview.net, 2023b.

Xie Yi, Zhanke Zhou, Chentao Cao, Qiyu Niu, Tongliang Liu, and Bo Han. From debate to equilibrium: Belief driven multi agent llm reasoning via bayesian nash equilibrium. 2025.

Sukwon Yun, Jie Peng, Pingzhi Li, Wendong Fan, Jie Chen, James Zou, Guohao Li, and Tianlong Chen. Graph-of-agents: A graph-based framework for multi-agent LLM collaboration. In *The Fourteenth International Conference on Learning Representations*. OpenReview.net, 2026.

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. In *Forty-second International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.

Qian Zhang, Jinyi Liu, Yan Zheng, Hebin Liang, and Lanjun Wang. Key decision-makers in multi-agent debates: Who holds the power? In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 29883–29891. AAAI Press, 2026.

Xiaochen Zhu, Caiqi Zhang, Yizhou Chi, Tom Stafford, Nigel Collier, and Andreas Vlachos. Demystifying multi-agent debate: The role of confidence and diversity. In *Findings of the Association for Computational Linguistics*, 2026.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62743–62767. PMLR / OpenReview.net, 2024.

A Related Work

A.1 Prompt-Based LLM Reasoning

Prompt-based methods significantly enhance the complex reasoning capabilities of large language models (Chu et al., 2024). Foundational Chain-of-Thought (CoT) methods (Wei et al., 2022), together with variants (Kojima et al., 2022; Fu et al., 2023), demonstrate that step-by-step reasoning improves performance on complex reasoning tasks. Self-Consistency (Wang et al., 2023) samples multiple reasoning paths and selects the most frequent answer, enhancing robustness and accuracy. Self-Refine (Madaan et al., 2023) introduces iterative self-checking and refinement to further improve reasoning performance. After that, Tree-of-Thoughts (ToT) (Yao et al., 2023a) generalizes CoT by enabling exploration over multiple intermediate reasoning paths, while Graph-of-Thoughts (GoT) (Besta et al., 2024) further models reasoning processes as graph structures to capture more flexible dependencies among thoughts. Meanwhile, tool-augmented reasoning frameworks such as ReAct (Yao et al., 2023b) integrate reasoning with external actions, enabling models to interact with environments and obtain additional evidence during inference. Together, richer prompting and single-model self-interaction improve reasoning, motivating the shift toward multi-agent frameworks.

A.2 Multi-Agent Debate

Multi-Agent Debate (MAD) (Du et al., 2024; Smit et al., 2024) extends single-agent reasoning by enabling multiple language model agents to interact and discuss, improving factuality and reasoning performance. Multiple agents independently reason, iteratively exchange intermediate outputs, and refine answers via voting (Du et al., 2024; Chan et al., 2024), consensus (Feng et al., 2025), judge agent (Liang et al., 2024), or adaptive stopping (Hu et al., 2025). However, recent work suggests that MAD's gains largely stem from answer aggregation (e.g., majority voting) rather than debate itself (Kaesberg et al., 2025; Choi et al., 2025). To encourage divergence and reduce correlated failures, subsequent methods introduce probabilistic updates (Choi et al., 2025), heterogeneous roles (Gu et al., 2026), and confidence exchange (Zhu et al., 2026). Another major limitation of MAD systems is their high communication cost due to dense multi-round interaction. To improve efficiency, recent works explore sparse communication topologies (Li et al., 2024) and controller-based scheduling (Fan et al., 2026), triggering debate only under uncertain conditions. These findings imply that effective multi-agent reasoning relies less on unrestricted debate and more on structured, selective information exchange. Nevertheless, existing approaches still maximize agreement rather than explicitly verifying intermediate reasoning claims.

B Algorithm

As summarized in Algorithm 1, `GCL-MAD` first generates diverse role-guided outputs and normalizes them into epistemic nodes, from which an Epistemic Conflict Graph is constructed to identify disagreements. Each detected conflict is then independently verified to obtain evidence-supported candidate answers. If verification reveals an alternative to the initial answer, an adjudication step integrates the candidate answers and verification evidence to produce the final answer; otherwise, the initial answer is retained.

C Experiments

C.1 Implementation Details

C.1.1 Dataset Details

We evaluate on six widely used reasoning benchmarks covering mathematical, commonsense, scientific, and knowledge-intensive reasoning tasks.

GSM8K. GSM8K (Cobbe et al., 2021) is a grade-school mathematical reasoning benchmark. We randomly subsample 300 questions from the original test split.

Algorithm 1 GCL−MAD Framework

Require: Problem x
Ensure: Final answer y^*
Generate role-guided outputs: $\mathcal{O} \leftarrow \mathcal{F}_{\text{role}}(x)$
Extract epistemic nodes: $V \leftarrow \mathcal{F}_{\text{norm}}(\mathcal{O})$
Construct Epistemic Conflict Graph: $G = (V, E)$
Initialize the candidate answer set: $\mathcal{A} \leftarrow \{a_0\}$
Initialize verification results: $\mathcal{Z} \leftarrow \emptyset$
for all conflict edge $e_k \in E$ **do**
    Collect node subset $V_k \subseteq V$ containing both endpoints of e_k and the baseline node `a1` when present.
    Verify conflict: $(\tilde{a}_k, \rho_k) \leftarrow \mathcal{F}_{\text{ver}}(x, a_0, V_k, e_k)$
    if $\tilde{a}_k \neq a_0$ **then**
        Add verified answer to $\mathcal{A}$: $\mathcal{A} \leftarrow \mathcal{A} \cup \{\tilde{a}_k\}$
        Record evidence: $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{(e_k, \tilde{a}_k, \rho_k)\}$
    end if
end for
if $\mathcal{A}$ contains an alternative answer different from a_0 **then**
    Perform adjudication: $y^* \leftarrow \mathcal{F}_{\text{judge}}(x, \mathcal{A}, \mathcal{Z})$
else
    $y^* \leftarrow a_0$
end if
return y^*

MATH. MATH (Hendrycks et al., 2021) is a challenging mathematical reasoning benchmark containing competition-level mathematics problems spanning algebra, geometry, probability, number theory, and precalculus. We randomly subsample 300 questions from the original test split.

CommonsenseQA (CSQA). CommonsenseQA (Talmor et al., 2019) is a multiple-choice benchmark designed to evaluate commonsense reasoning. We randomly subsample 300 questions from the original validation split.

ARC-Challenge (ARC-C). ARC-Challenge (Clark et al., 2018) consists of science examination questions collected from grade-school curricula. The challenge split contains questions that cannot be solved reliably by simple retrieval or statistical co-occurrence. We use the entire test split comprised of 298 questions.

MMLU-Pro. MMLU-Pro (Wang et al., 2024) extends MMLU with more challenging and less ambiguous questions across diverse academic disciplines, requiring both domain knowledge and reasoning capabilities. We randomly subsample 300 questions from the original test split.

GPQA. GPQA (Rein et al., 2023) is a graduate-level question answering benchmark designed to evaluate expert-level scientific knowledge and reasoning. The benchmark is intentionally constructed to be difficult for both humans and language models. We use the entire main split comprised of 448 questions.

C.1.2 Baseline Details

Chain-of-Thought (CoT). CoT (Wei et al., 2022) prompts the model to generate explicit intermediate reasoning steps before producing the final answer. For CoT, we use a single agent with the standard "Let's think step by step" prompting strategy.

Self-Consistency (SC). SC (Wang et al., 2023) extends CoT by sampling multiple reasoning trajectories and selecting the most frequent answer through majority voting. For SC, we sample five reasoning trajectories and select the majority answer.

CompLexCoT. CompLexCoT (Fu et al., 2023) improves reasoning by selecting examples with more complex reasoning chains as prompts, guiding the model to generate more structured and elaborate reasoning paths.

Self-Refine (SR). SR (Madaan et al., 2023) introduces an iterative self-feedback mechanism in which the model iteratively critiques and refines its own outputs.

Multi-Agent Debate (MAD). MAD (Du et al., 2024) employs multiple agents that independently generate solutions and iteratively exchange reasoning to reach a final answer. For MAD, we use 3 agents and conduct 3 debate rounds.

MAD-ASD. MAD-ASD (Hu et al., 2025) enhances MAD by introducing adaptive stability detection, allowing debate to terminate when agent opinions become sufficiently stable.

ECON. ECON (Yi et al., 2025) models multi-agent reasoning as a belief-updating process and seeks equilibrium through iterative interactions among agents.

iMAD. iMAD (Fan et al., 2026) uses a lightweight classifier trained on self-critique features to selectively trigger multi-agent debate, only engaging when it is likely to correct an initially wrong answer, thereby reducing unnecessary computational cost.

MADC. MADC (Zhang et al., 2026) introduces path consistency to identify key decision-making agents in multi-agent debates, effectively overcoming performance bottlenecks in standard MAD.

C.2 THRESHOLD SENSITIVITY.

Table 5: Threshold sensitivity of conflict-edge retention on GPQA with GPT-5-mini as the backbone model. We vary the retention threshold according to $E_{\text{high}}(\tau) = \{e \in E_{\text{review}} : w_e \geq \tau\}$, where $\tau = 3$ corresponds to the default setting.

Threshold τ	Edges/Q $\downarrow$	W $\rightarrow$ C $\uparrow$	C $\rightarrow$ W $\downarrow$	ΔAcc $\uparrow$	Final Acc. $\uparrow$
1	6.85	7.14	8.04	-0.89	68.53
2	4.10	6.47	1.34	+5.13	74.55
3 (default)	**2.44**	5.80	**0.00**	**+5.80**	**75.22**

Table 5 examines the effect of the confidence threshold used to retain reviewed conflict edges. Lowering the threshold increases the number of retained edges and therefore broadens the set of conflicts examined by the verifier. As τ decreases from 3 to 1, the average number of retained edges increases from 2.44 to 6.85, while the W $\rightarrow$ C rate increases from 5.80% to 7.14%. This shows that lower-confidence edges can expose additional errors that are missed under the stricter threshold.

However, retaining increasingly weak conflict edges also introduces substantially more erroneous revisions. At $\tau = 2$, the C $\rightarrow$ W rate increases moderately to 1.34%, reducing the net accuracy gain to +5.13%. When the threshold is further reduced to $\tau = 1$, the C $\rightarrow$ W rate rises sharply to 8.04%, exceeding the additional corrections and resulting in a negative net accuracy change of -0.89%. Accordingly, the final accuracy decreases to 68.53%, below the shared initial accuracy.

These results indicate that overly permissive conflict retention can introduce substantial verification noise. The default threshold $\tau = 3$ achieves the best balance between identifying useful conflicts and avoiding unnecessary revisions, supporting the use of high-confidence conflict edges for selective verification.

C.3 EDGE-LEVEL VERIFICATION ANALYSIS

We further analyze how localized conflict verification contributes to final answer correction. Table 6 reports edge-level verification outcomes and final answer transitions. For each retained edge, the verifier either keeps the initial answer or proposes an alternative answer. Alternative denotes the

Table 6: Edge-level verification outcomes and final answer transitions of GCL−MAD. Edges/Q is averaged over all questions, including those with no retained edge. Alternative, W→C, and C→W are computed over retained edges and characterize the verifier's proposed answer relative to the initial answer. Final transitions are computed at the question level after adjudication.

		Edge-level verification				Final answer		
Backbone	Dataset	Edges/Q	Alternative (%)	W→C (%)	C→W (%)	× → ✓(%)	✓ → ×(%)	ΔACC (%)
GPT-5-mini	MATH	2.35	3.23	1.08	0.92	1.33	0.60	+0.73
	GPQA	2.44	8.37	4.98	2.01	5.80	0.00	+5.80
DeepSeek-V4-Flash	MATH	0.14	26.19	19.05	0.00	2.67	0.00	+2.67
	GPQA	1.04	6.80	3.50	1.09	3.57	0.23	+3.34

proportion of retained edges that produce an answer different from the initial answer. W→C denotes cases where an incorrect initial answer is changed to a correct alternative, while C→W denotes cases where a correct initial answer is changed to an incorrect alternative. These metrics evaluate the verifier's proposed answers rather than the correctness of individual epistemic components. Final transitions are measured at the question level after adjudication.

The results show that GCL−MAD verifies only a small number of localized conflicts, with Edges/Q ranging from 0.14 to 2.44. Non-zero W→C rates across all settings show that localized verification can produce correct alternative answers, while occasional C→W cases indicate that verification can also introduce incorrect alternatives.

After adjudication, GCL−MAD achieves positive net accuracy gains in all four settings, ranging from $+0.73\%$ to $+5.80\%$. Overall, these results show that localized verification, together with evidence-grounded adjudication, can improve final answers without re-evaluating complete reasoning trajectories.

C.4 Inference Cost Analysis

Table 7: Inference token consumption of different reasoning methods using GPT-5-mini. Values denote average input and output tokens across six benchmarks.

Method	GSM8K	MATH	CSQA	ARC-C	MMLU-P	GPQA
			Input Tokens			
Vanilla	91.43	95.22	72.59	94.28	200.83	71.32
CoT	99.43	103.22	79.59	101.28	207.83	73.64
SC	497.13	516.10	397.93	506.39	1039.13	1171.07
CompLexCoT	497.13	516.10	397.93	506.39	1039.13	1171.07
SR	552.69	709.93	452.07	535.38	912.24	968.30
$\text{MAD}_{T=3}$	3771.45	5329.95	3116.29	3554.57	5360.27	2720.79
MAD-ASD	360.48	1792.86	356.27	352.59	638.02	470.44
MADC	4309.52	5792.51	3833.98	4528.21	4789.94	2954.32
GCL−MAD	2117.19	1581.06	4163.01	3093.74	3620.81	3787.69
			Output Tokens			
Vanilla	502.68	1273.36	544.08	409.46	1482.59	946.00
CoT	446.23	1273.66	639.50	394.03	1527.81	958.26
SC	2293.79	6299.67	3229.46	2015.10	7674.93	14686.22
CompLexCoT	2234.35	6368.84	3293.85	2067.70	7480.88	14754.13
SR	1168.63	3380.17	1340.50	1076.58	3717.37	7826.24
$\text{MAD}_{T=3}$	3237.62	10108.23	3598.82	2779.65	11229.91	12904.55
MAD-ASD	1312.28	3871.19	1883.83	1228.34	3312.27	4661.69
MADC	2973.47	5426.82	3507.75	3112.71	4936.29	5906.90
GCL−MAD	3838.15	4485.41	7570.18	5236.31	7102.77	12551.01

GCL−MAD incurs additional inference cost from epistemic decomposition, conflict localization, and selective verification. Table 7 reports the average input and output tokens across six benchmarks.

Single-agent methods consistently use fewer tokens because of their simpler inference pipelines, while the cost of GCL−MAD relative to existing MAD methods varies across datasets.

The matched-budget routing comparison shows that ECG routing achieves stronger correction performance under the same number of verifier calls. These results indicate that the effectiveness of GCL−MAD is closely tied to how verification targets are selected and organized.

C.5 Case Study

Fig. 4 presents a representative GSM8K example illustrating how GCL−MAD corrects an error caused by an implicit assumption. The initial symbolic reasoning predicts 300 by assuming that the electricity bill is divided among four payers. This calculation is arithmetically consistent under the assumed divisor, but the assumption itself is questionable. The phrase "Jenna has 4 roommates" more directly indicates that Jenna shares the bill with four roommates, yielding five payers in total.

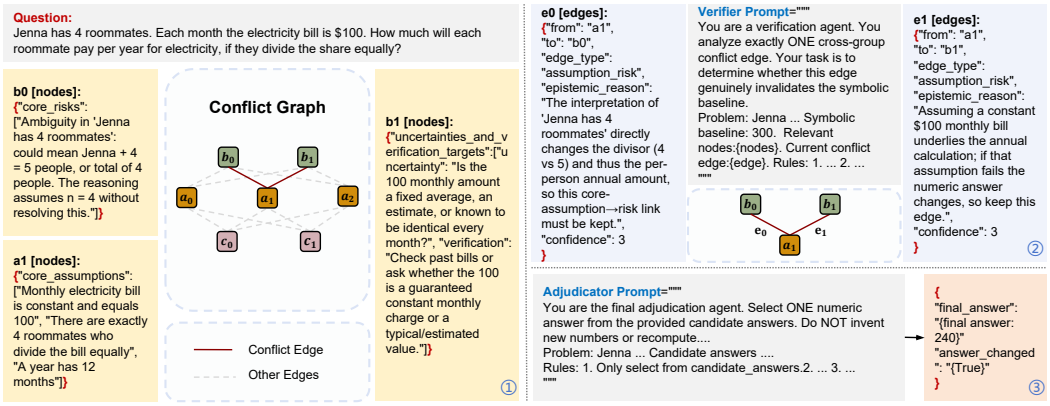


Figure 4: Case study of correcting an implicit-assumption error on GSM8K.

As shown in Fig. 4, GCL−MAD first decomposes the reasoning process into epistemic nodes, including the symbolic assumption node, skeptical risk nodes. The graph construction step then retains assumption-risk edges that connect the divisor assumption to skeptical concerns. Both displayed edges receive confidence 3 from the graph-construction agent and therefore pass the fixed confidence filter. This ordinal score expresses the agent's judgment that an edge warrants verification; it is not a calibrated probability that the asserted conflict is valid. Among these edges, the key conflict highlights the ambiguity in the number of payers: the symbolic reasoning assumes four payers, whereas the skeptical node identifies the alternative interpretation involving Jenna plus four roommates. This localized edge exposes the source of the error without requiring verification of the full reasoning trajectory. The verifier evaluates the retained conflict edge and proposes the corrected candidate answer 240 based on five payers. The adjudicator then compares the symbolic answer 300 with the verified alternative 240 and selects 240 as the final answer. This case demonstrates that GCL−MAD corrects the prediction by localizing a faulty assumption, verifying the conflicted component, and adjudicating between candidate answers, rather than relying on trajectory-level consensus or unrestricted debate.

D Prompt for GCL−MAD

Prompt for the Solver Agent

You are a symbolic reasoning agent. Your objective is to solve the problem through explicit formal reasoning. Do NOT generate long natural-language explanations. Produce a structured JSON containing only information necessary for correctness, verification, and disagreement analysis.
Problem: {problem}
Output requirements:

1. All fields must exist; use empty lists [] if no content.
2. answer must be exactly in the format {{final answer: ¡numeric¿}} (no units, no text).
3. Do not include narrative explanations.
4. Provide core assumptions and symbolic reasoning steps as arrays of strings.
Structured JSON output:
{{
”answer”: ”{{final answer: 123}}”,
”core assumptions”: [],
”symbolic reasoning steps”: []
}}

Prompt for the Skeptic Agent

You are a skeptical reasoning agent. Your task is to analyze the provided symbolic reasoning and identify epistemic risks. Produce a structured JSON failure-analysis state; do NOT rewrite the reasoning.
Problem: {problem}
Reasoning State: {symbolic state}
Output requirements:
1. All fields must exist; you MUST return non-empty JSON (never return blank).
2. Use empty lists [] for fields with no findings — do not omit the JSON object.
3. Only flag epistemic risks (ambiguity, hidden assumptions, failures); do NOT output any numeric answer or answer field.
4. Do not solve or revise the answer.
5. Populate at least one of skeptical points or uncertainties and verification targets when any interpretive risk exists.
Structured JSON output:
{{
”skeptical points”: [],
”uncertainties and verification targets”: []
}}

Prompt for the Edge Case Agent

You are an edge-case reasoning agent. Your task is to analyze the symbolic reasoning for boundary-condition risks. Do NOT generate full reasoning or narrative explanations. Produce a structured JSON output; all fields must exist, use empty lists [] if no content.
Problem: problem
Reasoning State: {symbolic state}
Output requirements:
1. All fields must exist; you MUST return non-empty JSON (never return blank).
2. Use [] for fields with no findings — do not omit the JSON object.
3. Only flag boundary/off-by-one/edge-case risks; do NOT output any numeric answer or answer field. 4. Do not solve or revise the answer.
5. Populate at least one of edge cases or verification targets when any boundary/off-by-one/edge-case risk exists.
Structured JSON output:
{{
”edge cases”: [],
”verification targets”: []
}}

Prompt for the Graph Construction Agent

You are an epistemic edge review agent. Your task is NOT to solve the problem or invent new edges. Rule-based pipeline already proposed cross-group edge CANDIDATES. You must be selective. Keep an edge ONLY if it represents a core epistemic risk that could reasonably change the final numeric answer. Your task is to filter candidate edges; do NOT solve the problem or invent new edges. A rule-based pipeline has proposed cross-group edge candidates. You must be selective. Keep an edge ONLY if it represents a core epistemic risk that could reasonably change the final numeric answer.
Problem: {problem}
Nodes (read-only): {nodes catalog}
Edge candidates: edge candidates
Review rules:
- Default decision is keep=false.
- keep=true ONLY when the candidate directly attacks a core assumption, formula, interpretation.
- keep=false when the candidate is merely speculative, practical, redundant, weakly related.
- Drop duplicate risks; keep only the clearest edge for the same issue.
- confidence must be exactly one of: 1, 2, 3.
- 3 = strong confidence could keep the edge.
- 2 = medium confidence could keep the edge.
- 1 = weak confidence could keep the edge.
- Do NOT add candidates; return one review per candidate_id.
You may override edge type ONLY among: ambiguity risk — uncertainty propagation — assumption risk — boundary risk — dependency
If edge candidates is empty, output {{"reviews": []}}.
— Output JSON ONLY:
{{
"candidate id": "cand_000",
"keep": false,
"edge type": "assumption risk",
"conflict reason": "One short sentence.",
"confidence": 1
}}

Prompt for the Verifier Agent

You are a verification agent. You analyze exactly ONE cross-group conflict edge. Your task is to determine whether this edge genuinely invalidates the symbolic baseline.
Problem: {problem}
Symbolic baseline (node a1): {symbolic answer}
Relevant nodes (endpoints of this edge): {nodes}
Current conflict edge: {edge}
Rules:
1. Focus only on this edge.
2. By default, preserve the symbolic baseline answer.
3. Generate a different proposed answer ONLY if:
- the proposed answer must not contradict or alter any explicitly stated conditions, constraints, or facts given in the problem statement.
- the edge invalidates a core symbolic assumption, formula, counting rule, or interpretation.
4. Do NOT generate a new answer for speculative, weak, timing/order, implementation, or boundary-only variants.
Output JSON ONLY:
{{
"proposed answer": "{{final answer: 123}}",
"edge": {{ "from": "a1", "to": "b1", "epistemic reason": "One short sentence." }}
}}

Prompt for the Adjudicator Agent

You are the final adjudication agent. Select ONE numeric answer from the provided candidate answers. Do NOT invent new numbers or recompute.
- Problem: problem
- Candidate answers (including symbolic and alternatives): candidate answers
- Symbolic baseline (node a1): symbolic state
- Per-edge verification (proposed answer + conflict point + edge): edge verifications
Rules:
1. Only select from candidate answers.
2. Presume the symbolic baseline is correct by default. Do not change it merely because an alternative is plausible; change it only if Rule 3 is satisfied.
3. Change the answer ONLY when the alternative interpretation is more directly supported by the explicit wording of the problem and invalidates the symbolic calculation itself.
4. Do NOT change the answer for speculative, weakly supported, or non-core interpretation variants unless explicitly required by the problem.
Output JSON ONLY:
{{
"final answer": "{{final answer: 123}}"
}}

D.1 Dataset-Specific Answer Formats and Parsing

The preceding prompt boxes show the GSM8K variant; their numeric-only instruction does not apply to every benchmark. The runtime selects answer-format instructions by dataset for all three answer-producing roles: the Solver writes `answer`, each edge-level Verifier writes `proposed_answer`, and the Adjudicator writes `final_answer`. Each field uses the same outer {`final answer: ...`} marker, with a dataset-specific payload as shown in Table 8.

Table 8: Answer payloads requested by the runtime prompts. Examples show the value inside a JSON field after JSON string decoding.

Benchmark	Payload	Example answer value
GSM8K	Numeric	{`final answer: 123`}
MATH	LaTeX expression or tuple	{`final answer: \frac{11}{5}`}
CSQA, ARC-C, MMLU-Pro, GPQA	One option letter	{`final answer: (C)`}

MATH-specific prompts. For MATH, the Solver is instructed to return {`final answer: <LaTeX>`} without math-mode dollar signs, units, or prose. Its prompt asks for fractions in `\frac{numerator}{denominator}` form rather than `a/b`, radicals as `\sqrt{...}`, exponents with closed braces, `\pi` instead of Unicode π, and ordered tuples in parentheses when the answer has several components (e.g., $(\frac{11}{5}, \frac{2}{5}, 5)$). The MATH Verifier's `proposed_answer` must follow the same LaTeX conventions, and the MATH Adjudicator is instructed to select a provided answer while preserving its LaTeX form. These MATH variants are used for noninteger fractions, radicals, tuples, and other mathematical expressions; MATH answers are not coerced to a single integer. In a JSON response, each LaTeX backslash must be escaped as \\ for the response to parse as JSON; after decoding, the answer value contains the ordinary single backslash shown in Table 8.

Multiple-choice prompts. For CSQA, ARC-C, MMLU-Pro, and GPQA, the Solver and Verifier are instructed to give a single displayed option letter in parentheses, for example {`final answer: (C)`}, rather than copying the option text. The Adjudicator is instructed to choose an answer choice from the supplied candidate list. The same option-label format is used by the dataset loaders for the reference answers.

Parsing and format failures. The response parser first reads a JSON object, allowing a fenced JSON block or an object embedded in surrounding text; malformed JSON yields no parsed object.

For an answer field, it extracts the payload of the last case-insensitive, brace-balanced {`final answer: ...`} occurrence; if that marker is absent, it attempts to normalize the raw field value. GSM8K answers are parsed as numbers and rounded to one decimal place; a nonnumeric value becomes an empty answer. MATH answers undergo limited formatting normalization, including removal of dollar signs and spacing, conversion of common Unicode Greek symbols and simple slash fractions to LaTeX, and repair of a truncated `\sqrt{...}` closing brace. MATH scoring then checks normalized equality, numeric equality, component-wise tuple equality, or symbolic equivalence when parsing permits it. For multiple-choice tasks, the normalizer strips surrounding parentheses and whitespace, uppercases the first remaining character, and wraps it in parentheses; it does not separately check that this character is among the displayed options, so a malformed leading character fails exact comparison with the reference label.

The pipeline retries an empty API response up to two times, but does not re-prompt solely because a nonempty response contains invalid JSON or a malformed answer. An unparseable Solver response produces an empty baseline answer and is scored incorrect. An unparseable Verifier response contributes no proposed answer for its edge; if no verifier proposes an alternative, adjudication is skipped and the baseline is kept. An unparseable Adjudicator response, or a selected answer outside the normalized candidate set, falls back to the Solver baseline when a valid baseline candidate exists.

Operational answer comparison and candidate deduplication. Let $N_{\mathcal{D}}$ denote answer extraction and normalization for dataset $\mathcal{D}$ as described above. For a proposal that passes the nonempty-value check and a nonempty normalized baseline, the condition $\tilde{a}_k \neq a_0$ used to construct $\mathcal{Z}$ and $\mathcal{A}$ is implemented as the direct comparison

$$N_{\mathcal{D}}(\tilde{a}_k) \neq N_{\mathcal{D}}(a_0).$$

Equality is evaluated on the returned values: normalized numbers for directly parseable numeric answers, and normalized strings for expression or option-label answers. The candidate pool uses these same normalized values for deduplication; repeated candidates are omitted, while per-edge verification records are retained separately. The adjudicator is invoked if at least one verification record survives comparison with the baseline. If the normalized baseline is empty, a proposal that passes the nonempty-value check survives this comparison without a baseline-equality test. The difference filter does not independently validate the mathematical content of a proposal.

D.2 Implementation Details of Epistemic Conflict Localization

This section provides additional details on the Epistemic Conflict Graph (ECG) used for verification routing. We describe how structured role outputs are represented as nodes, how candidate conflicts are reviewed, and how retained conflicts are passed to the verifier.

Epistemic nodes. As described in the main text, populated fields of the role agents' structured outputs are normalized into typed epistemic nodes $v_i = (t_i, s_i)$. The type t_i records the epistemic function of the content, while s_i contains the corresponding field content. The resulting node set V can include answers, assumptions, reasoning steps, skeptical points, uncertainties, and boundary conditions. Empty or non-informative fields are omitted. This representation allows the graph-construction agent to relate information from different reasoning perspectives and to identify which components warrant verification. When a field contains multiple statements, its complete content remains available to the verifier.

Candidate conflict relations. Following the main-text formulation, candidate relations are selected from type-compatible component pairs:

$$P_{\text{cand}} = \mathcal{C}(V) \subseteq V \times V. \tag{8}$$

The compatibility criterion focuses review on pairs whose epistemic roles make a conflict relevant to the answer. Candidate generation proposes relations for review; it does not itself establish that the paired components conflict or that either component is incorrect.

Each candidate is initialized with a relation category: `ambiguity_risk`, `assumption_risk`, `boundary_risk`, `uncertainty_propagation`, or `dependency`. These labels specify the type of potential epistemic interaction to be reviewed; they do not indicate that an error has already been detected.

Graph-construction review. All candidate relations for one problem are reviewed jointly in a single graph-construction call. For each candidate e_k, the agent outputs

$$(\mathrm{keep}_k, r_k, m_k, w_k), \tag{9}$$

where keep_k is the retention decision, r_k is the relation type, m_k is a short conflict rationale, and w_k is an ordinal confidence score satisfying

$$w_k \in \{1, 2, 3\}. \tag{10}$$

The reviewer is instructed to retain a relation only when the identified issue directly concerns an assumption, reasoning dependency, interpretation, or boundary condition that could affect the final answer. Merely speculative or redundant relations are rejected.

We use the same high-confidence criterion in all experiments:

$$E_{\mathrm{high}} = \{e_k \in E_{\mathrm{review}} : \mathrm{keep}_k = \mathrm{true},\ w_k = 3\}. \tag{11}$$

The confidence score is used solely for verification routing and is not interpreted as a calibrated probability of answer correctness.

Conflict consolidation. Before verification, retained edges are checked for invalid or duplicate relations and redundant links referring to the same epistemic risk. This step consolidates the reviewed conflicts before they are used to route verification.

The resulting Epistemic Conflict Graph is defined as

$$G = (V, E), \tag{12}$$

where E denotes the consolidated retained edge set. The graph records the selected component pairs, relation types, and rationales used for verification routing. The proposed method does not require graph message passing or multi-hop inference.

Verification routing. If

$$E = \varnothing, \tag{13}$$

the baseline Solver answer is returned directly. Otherwise, each retained edge triggers one localized verifier call. For a retained edge represented as

$$e_k = (v_i, v_j, r_k, m_k), \tag{14}$$

the verifier receives the original problem, the baseline answer, the two endpoint components, the edge relation, and its rationale. The baseline answer node is additionally provided when it is not already an endpoint.

The verifier therefore examines the complete contents of the localized fields associated with a conflict edge, rather than the entire multi-agent reasoning trajectory.

Verification alone does not force an answer change. Adjudication is invoked only when at least one verifier produces an alternative answer, and the final answer is revised only when the corresponding verification evidence supports the alternative.

E LIMITATIONS

This work has several limitations. First, GCL-MAD is designed and evaluated on objective reasoning tasks with short, verifiable answers; extending component-level conflict localization to open-ended generation, where answers are not easily normalized or compared, remains an open problem. Second, localization operates on structured fields, so a verifier receives the full content of a selected field and must determine which statement within it matters. Finer localization within fields remains for future work. Third, ECG represents reviewed pairwise conflicts for verification routing; the present method does not use graph propagation or path-based reasoning. The matched-budget routing comparison evaluates the routing procedure as a whole and does not isolate the graph representation from the LLM's conflict review. Fourth, the framework relies on LLM-reported confidence scores for conflict-edge retention. As described in Appendix D.2, these scores are ordinal judgments

rather than calibrated probabilities, and our sensitivity analysis (Table 5) shows that performance degrades sharply when weaker edges are retained. This dependence means that the behavior of the pipeline may vary with the calibration characteristics of the backbone model, and more robust confidence estimation is an important direction for future work. Fifth, although GCL-MAD reduces the amount of verification compared with re-examining complete trajectories, its overall token cost is not uniformly lower than that of existing debate methods (Table 7), and the accuracy–cost trade-off may be unfavorable for simple tasks where aggregation of multiple perspectives already suffices. Finally, our evaluation uses subsampled benchmarks of moderate size, and the observed gains on some datasets are within the range of run-to-run variance; the reliability of localized correction on broader task distributions therefore warrants further validation.