BIGATE: BIDIRECTIONAL VERIFICATION-GUIDED SELECTIVE COMMUNICATION FOR MULTI-AGENT REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent reasoning can exploit complementary capabilities across heterogeneous language models, but indiscriminate full-reasoning exchange increases inference cost and exposes revision contexts to unnecessary or erroneous peer information. We propose **BiGate**, a bidirectional verification-guided selective communication mechanism. For each agent pair, BiGate forms a response-local ternary verification relation from the two directional judgments. The Sender Need Gate uses the sender's judgment of the receiver to decide whether communication is needed and whether feedback is supplementary or corrective, while the Receiver Scope Gate uses the reverse judgment to control whether the sender's full reasoning enters the receiver's revision context. Their joint mapping yields dynamic directed typed communication, while two-stage context isolation prevents verification-time peer reasoning from carrying into revision except through gated messages. Across five reasoning benchmarks and three heterogeneous base models, BiGate achieves 73.90% macro-average accuracy, 2.30 percentage points above the strongest collaborative baseline, and uses the fewest model calls and Response Rounds among the evaluated multi-agent methods. Ablations and behavioral analyses support the gates' distinct roles and show task-varying communication with restricted full-reasoning admission under persistent error.

1 INTRODUCTION

LLM-based multi-agent reasoning can improve complex problem solving by combining independent responses with peer exchange and iterative revision Du et al. (2024); Chen et al. (2024); Xu et al. (2023). Heterogeneous agents may contribute complementary knowledge and reasoning trajectories, allowing peer feedback to expose omissions or errors that are difficult to identify in isolation. Peer reasoning, however, is not uniformly beneficial. Erroneous or unreliable reasoning can influence otherwise correct agents Huang et al. (2025); He et al. (2025); Shen et al. (2025), while indiscriminate information exchange increases communication and inference cost and can expose agents to unnecessary peer information Zhang et al. (2026); Gou & Liu (2026). Effective collaboration therefore requires more than increasing interaction: it must preserve the corrective value of peer information while controlling when communication is necessary and what source information is admitted into subsequent revision.

Prior work controls different aspects of this process at different granularities. Deliberation and peer-review methods organize iterative exchange, assessment, and revision Du et al. (2024); Chen et al. (2024); Xu et al. (2023); verification- and trust-aware systems use reviewer judgments, verifier signals, or receiver-side assessments to guide revision, filtering, interaction, or adjudication Wang et al. (2025); He et al. (2025); Liu et al. (2025); Yao et al. (2026); and adaptive communication methods control whether collaboration is invoked, which communication structures are used, or how messages are formed Fan et al. (2026); Zhang et al. (2025a; 2026); Gou & Liu (2026); Zhang et al. (2025b). At directed peer-message granularity, this motivates a more specific coordination problem: whether the sender needs to communicate, whether its feedback should be supplementary or corrective, and whether the sender's full reasoning should enter the receiver's revision context. We address these decisions through the two directions of the same response-local bidirectional verification relation, assigning them complementary sender-side and receiver-side communication roles.

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.

We introduce **BiGate**, a bidirectional verification-guided selective communication mechanism for multi-agent reasoning. For each agent pair, both agents verify the other's current response and produce a response-local ternary judgment: VERIFIED, UNDETERMINED, or ERROR DETECTED, together with step-level feedback for non-Verified judgments. For a potential message from sender A_j to receiver A_i, the Sender Need Gate uses A_j's judgment of A_i to determine whether communication is needed and whether the feedback is supplementary or corrective, while the Receiver Scope Gate uses A_i's judgment of A_j to determine whether the message contains only local feedback or is augmented with the sender's full reasoning. Their joint mapping yields no message, local feedback, or feedback augmented with full reasoning, and all nonempty messages form a dynamic directed typed communication graph. BiGate further uses two-stage context isolation, so that peer reasoning observed during verification is admitted into the separately reconstructed revision context only when allowed by the gated message construction.

Across five reasoning benchmarks with three heterogeneous base models, BiGate achieves a macro-average accuracy of 73.90%, 2.30 percentage points above the strongest collaborative baseline. It also requires the fewest model calls and Response Rounds among the evaluated multi-agent methods. Dual-gate ablations support distinct roles for communication-need and information-scope control, while behavioral analyses show that communication-mode distributions vary across tasks and that the Receiver Scope Gate frequently restricts admission of a persistently erroneous source's full reasoning.

Our contributions are as follows:

- We formulate a response-local bidirectional ternary verification relation that preserves both directional judgments within each agent pair, enabling communication decisions to be based on pairwise, response-specific evidence rather than a global reliability score.
- We propose a dual-gated selective communication mechanism that separates communication need and feedback function from source-information scope. The two directional judgments are jointly mapped to typed messages, while two-stage context isolation controls which peer information is admitted into revision.
- We conduct a systematic evaluation across five reasoning benchmarks and three heterogeneous base models, covering overall accuracy, inference cost, dual-gate ablation, communication behavior, and persistent-error analysis.

2 RELATED WORK

2.1 MULTI-AGENT DELIBERATION AND REVISION

A common pattern in LLM-based multi-agent reasoning is peer exchange followed by iterative revision. Multiagent Debate exchanges full peer responses across debate rounds Du et al. (2024), ReConcile organizes heterogeneous models through confidence-aware discussion and aggregation Chen et al. (2024), and Multi-Agent Peer Review uses directed, response-specific reviews to support revision Xu et al. (2023). Together, these methods span broad peer-response exchange, confidence-guided collaboration, and directional pairwise review as distinct mechanisms for organizing multi-agent revision.

Recent work further makes collaboration conditional. MARS uses hierarchical author, reviewer, and meta-reviewer roles to determine whether an answer should be accepted or revised Wang et al. (2025), while iMAD predicts whether a query is likely to benefit from debate before launching full collaboration Fan et al. (2026). These mechanisms operate at centralized or query-level granularity. BiGate moves this conditionality to directed peer-message granularity: the sender's response-local judgment determines whether communication is needed and, if so, whether the feedback is supplementary or corrective.

2.2 VERIFICATION AND RELIABILITY-AWARE COLLABORATION

Reliability-aware collaboration uses different signals for different control purposes. ReConcile uses self-reported confidence to guide discussion and aggregation Chen et al. (2024); Multi-Agent Peer Review produces directional reviews of current peer solutions Xu et al. (2023); and A-Trust/TMS

combines message-level evidence with longer-term agent trust to accept, filter, or further verify incoming messages He et al. (2025). Together, these methods illustrate self-confidence, directional peer assessment, and receiver-side message trust as distinct signals for aggregation, revision, and filtering.

Verifier signals also shape interaction form and control flow. Adaptive Coopetition uses verifier signals to switch between collaborative and competitive interaction Liu et al. (2025), while MAVEN organizes a Skeptic–Researcher–Judge workflow with multi-state adjudication and in-step auditing Yao et al. (2026). Huang et al. further study resilience to faulty agents and mechanisms for challenging and correcting erroneous messages Huang et al. (2025). BiGate's distinction is therefore not verifier-guided collaboration itself, but the decomposition of a bidirectional response-local peer relation into complementary controls: the sender-side judgment governs communication need and feedback function, whereas the receiver-side judgment governs source-information scope. This role decomposition is coupled with context isolation: full peer reasoning observed during verification can enter the separate revision context only when admitted by the Receiver Scope Gate and included in a generated message.

2.3 Adaptive and Selective Communication

Recent work adapts communication beyond fixed or fully connected interaction. G-Designer learns task-conditioned directed communication topologies from agent and task representations before multi-round interaction Zhang et al. (2025a); AMAS selects input-specific agentic graphs before execution Leong et al. (2025); SafeSieve progressively prunes low-value communication using semantic compatibility and accumulated interaction experience Zhang et al. (2026); and DySCo selects directed edges under communication budgets using reliability, disagreement, and task relevance Gou & Liu (2026). Together, these methods span query-conditioned graph design and directed edge-level adaptation.

Selective communication also extends to message form and information propagation. OSC adapts communication partners, content focus, objective, and detail according to learned collaborator states Zhang et al. (2025b), while Shen et al. analyze how correct and erroneous outputs propagate across communication topologies with different sparsity patterns Shen et al. (2025). These works extend selective communication beyond topology design to message adaptation and information-propagation analysis. BiGate operates at directed peer-message granularity: one direction of the same response-local bidirectional ternary verification relation determines sender-side communication need and feedback function, while the opposite direction determines receiver-side source-information scope; their joint mapping yields no message, local feedback, or feedback augmented with full reasoning.

3 Method

3.1 Problem Setup and Framework

Given a problem x and a set of N heterogeneous LLM agents

$$\mathcal{A} = \{A_1, A_2, \ldots, A_N\}, \tag{1}$$

BiGate produces a final answer through at most $T_{\max}$ collaboration rounds. At the beginning of round t, the current response of agent A_i is

$$r_i^{(t)} = \left(z_i^{(t)}, a_i^{(t)}\right), \tag{2}$$

where $z_i^{(t)}$ denotes its reasoning process and $a_i^{(t)}$ denotes its final answer. Hereafter, *full reasoning* refers to an agent's current structured response $r_i^{(t)}$, including both its reasoning process and final answer.

Figure 1 provides an end-to-end overview of BiGate. Agents first perform independent structured reasoning and bidirectional ternary verification. The resulting directional judgments drive dual-gated message construction, and all nonempty messages form a dynamic directed typed communication graph. Agents revise synchronously on nonempty messages. Collaboration terminates when the communication graph is empty; otherwise, it continues up to $T_{\max}$. The latest answers are then aggregated by majority voting.

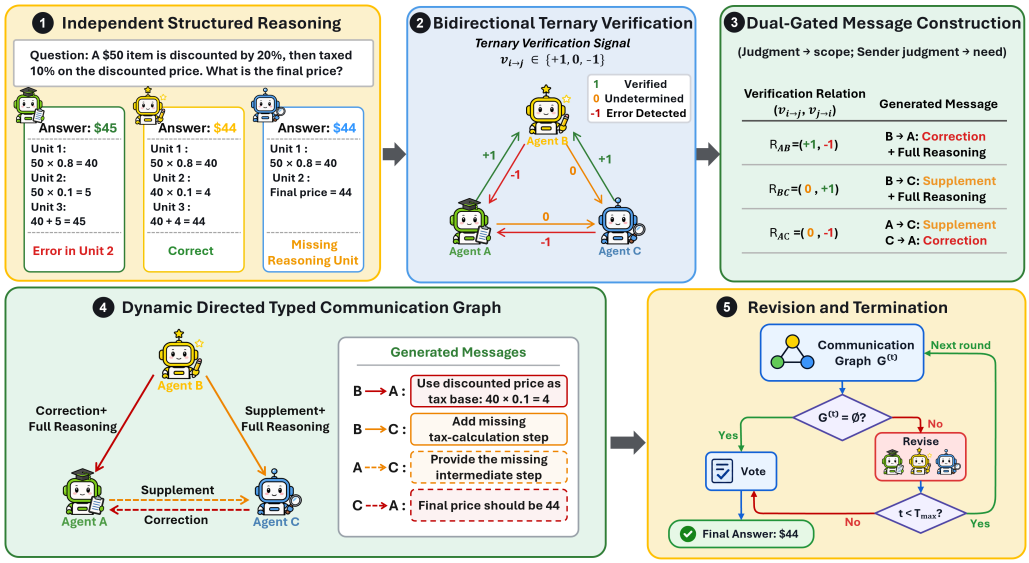


Figure 1: Overview of BiGate: independent structured reasoning, bidirectional ternary verification, dual-gated message construction, dynamic directed typed communication, and revision and termination.

3.2 Independent Structured Reasoning

All agents first generate their initial responses $r_i^{(0)}$ independently without observing one another. At any round t, the reasoning process is organized into indexed reasoning units:

$$z_i^{(t)} = \left[z_{i,k}^{(t)} \right]_{k=1}^{K_i^{(t)}}, \tag{3}$$

where $K_i^{(t)}$ is the number of reasoning units. These indices are used only to locate positions that may require supplementation or correction; different models need not follow an identical reasoning structure. For $t > 0$, $r_i^{(t)}$ is the latest response produced by agent A_i after the preceding revision phase.

3.3 Bidirectional Ternary Verification

At round t, agent A_i produces a ternary verification signal for the current response of agent A_j:

$$v_{i \to j}^{(t)} \in \{+1, 0, -1\}. \tag{4}$$

The values have the following response-local meanings:

- $+1$ (Verified): no decisive issue affecting the final conclusion is identified;
- 0 (Undetermined): no explicit error is identified, but the available evidence, required conditions, or explanations are insufficient;
- -1 (Error Detected): an explicit error that invalidates the final answer or a key reasoning step is identified.

When $v_{i \to j}^{(t)} \in \{0, -1\}$, the verifier also produces step-level feedback $f_{i \to j}^{(t)}$. For Undetermined, the feedback identifies the most important missing evidence, condition, or justification. For Error Detected, it locates and explains a decisive error. In both cases, the feedback directs the target agent to re-examine its own reasoning from the indicated position rather than simply adopt the verifier's final answer.

The two directional judgments for an agent pair form a bidirectional verification relation:

$$R_{ij}^{(t)} = \left(v_{i\to j}^{(t)}, v_{j\to i}^{(t)}\right). \tag{5}$$

BiGate preserves both directions instead of collapsing them into a single agent-level score. The relation is local to the current problem and the agents' current responses; its ternary signals are treated as fallible peer assessments for communication control rather than as correctness certificates or permanent agent-level reliability estimates.

3.4 Dual-Gated Message Construction

Consider a potential message from sender A_j to receiver A_i. The two judgments in $R_{ij}^{(t)}$ serve distinct roles: the sender's judgment of the receiver controls communication need and feedback function, whereas the receiver's judgment of the sender controls information scope.

Sender Need Gate. The Sender Need Gate is

$$n_{j\to i}^{(t)} = \begin{cases} \varnothing, & v_{j\to i}^{(t)} = +1, \\ \text{Supplement}, & v_{j\to i}^{(t)} = 0, \\ \text{Correction}, & v_{j\to i}^{(t)} = -1. \end{cases} \tag{6}$$

A Verified judgment produces no message. An Undetermined or Error Detected judgment produces supplementary or corrective feedback, respectively. The Sender Need Gate therefore determines whether a directed message is needed and, if so, which feedback function it serves.

Receiver Scope Gate. The Receiver Scope Gate is

$$s_{j\to i}^{(t)} = \begin{cases} \text{Full}, & v_{i\to j}^{(t)} = +1, \\ \text{Local}, & v_{i\to j}^{(t)} \in \{0, -1\}. \end{cases} \tag{7}$$

If a message is needed in this direction, the Receiver Scope Gate determines its source-information scope. When the receiver assigns Verified to the sender's current response, the message includes the sender's full reasoning alongside step-level feedback; under Undetermined or Error Detected, it is restricted to targeted step-level feedback. Consequently, the sender's judgment of the receiver determines whether feedback is needed and what function it serves, while full-reasoning admission additionally depends on the receiver's judgment of the sender.

Joint message construction. The two gates jointly construct

$$m_{j\to i}^{(t)} = \begin{cases} \varnothing, & v_{j\to i}^{(t)} = +1, \\ \left\langle v_{j\to i}^{(t)}, f_{j\to i}^{(t)} \right\rangle \oplus r_j^{(t)}, & v_{j\to i}^{(t)} \in \{0, -1\}, \\ & v_{i\to j}^{(t)} = +1, \\ \left\langle v_{j\to i}^{(t)}, f_{j\to i}^{(t)} \right\rangle, & v_{j\to i}^{(t)} \in \{0, -1\}, \\ & v_{i\to j}^{(t)} \in \{0, -1\}. \end{cases} \tag{8}$$

Here, $\langle v, f\rangle$ denotes a concise verification label together with its step-level feedback, and $\oplus$ denotes augmentation with the sender's full reasoning $r_j^{(t)}$. A message without $r_j^{(t)}$ is referred to as *local feedback*.

Table 1 summarizes the six representative relation classes induced by Eq. (5). The nine ordered combinations reduce to these six classes because the reverse of an asymmetric case is obtained by swapping i and j.

3.5 Dynamic Directed Typed Communication Graph

All nonempty messages constructed at round t form a dynamic directed typed communication graph:

$$\mathcal{G}^{(t)} = \left(\mathcal{A}, \mathcal{E}^{(t)}\right), \qquad \mathcal{E}^{(t)} = \left\{(A_j, A_i) \mid m_{j\to i}^{(t)} \neq \varnothing\right\}. \tag{9}$$

Table 1: Communication induced by the six representative bidirectional verification relations. Sup., Corr., and Full denote supplementary feedback, corrective feedback, and full reasoning, respectively.

Relation	**Direction**	**Message content**
$(+1, +1)$	None	No Communication
$(+1, 0)$	$A_j \to A_i$	Sup. + Full
$(+1, -1)$	$A_j \to A_i$	Corr. + Full
$(0, 0)$	Bidirectional	Sup./Sup.
$(0, -1)$	Bidirectional	Sup./Corr.
$(-1, -1)$	Bidirectional	Corr./Corr.

Each nonempty edge is typed along two dimensions: feedback function (supplementary or corrective) and information scope (feedback-only or augmented with full reasoning). Thus, $R_{ij}^{(t)}$ determines whether directed edges exist between an agent pair and, if so, their directions and message types. Because the graph is reconstructed from the current response-local bidirectional verification relations in each round, it does not rely on a predefined topology.

For behavioral analysis, the six representative relations are grouped into three relation-driven communication modes. Relation $(+1, +1)$ yields *No Communication*. Relations $(+1, 0)$ and $(+1, -1)$, together with their reverses, yield *Full-Reasoning Transfer*. Relations $(0, 0)$, $(0, -1)$, and $(-1, -1)$, together with the reverse of $(0, -1)$, yield *Local-Feedback Exchange*. Messages under Full-Reasoning Transfer are unidirectional and contain the sender's verification label, targeted feedback, and full reasoning. Messages under Local-Feedback Exchange are bidirectional and contain only verification labels and targeted local feedback. Experiments report the proportions of these three modes rather than those of all six relation classes separately.

3.6 Two-Stage Context Isolation, Revision, and Termination

Each collaboration round is implemented through two context-isolated phases. In the *Verification Phase*, every agent temporarily reads the full reasoning of all other agents at the beginning of the round and, within a single model call, produces a ternary verification signal and step-level feedback for every peer. The verification-call context is not carried forward. In the *Revision Phase*, the system constructs a new context containing the problem, the agent's own current response, and only the gated inbound messages defined in Eq. (8). Each nonempty message includes a concise verification label indicating whether the sender judged the receiver's response as UNDETERMINED or ERROR DETECTED.

Two-stage context isolation leaves the temporary visibility required for verification ungated while controlling what crosses the stage boundary into the separate revision context. Full peer reasoning observed during verification cannot directly enter the revision context unless it is admitted by the Receiver Scope Gate and included in a generated message.

The set of nonempty inbound messages received by agent A_i at round t is

$$\mathcal{M}_i^{(t)} = \left\{ m_{j \to i}^{(t)} \mid j \neq i, m_{j \to i}^{(t)} \neq \varnothing \right\}. \tag{10}$$

The response is updated as

$$r_i^{(t+1)} = \begin{cases} \mathrm{Revise}\left(x, r_i^{(t)}, \mathcal{M}_i^{(t)}\right), & \mathcal{M}_i^{(t)} \neq \varnothing, \\ r_i^{(t)}, & \mathcal{M}_i^{(t)} = \varnothing. \end{cases} \tag{11}$$

The verification label allows the revising agent to distinguish a supplementary concern from an explicit error report and to re-examine its reasoning using the corresponding step-level feedback. All agents update synchronously from the states available at the beginning of the round, so a response revised within round t does not affect another agent until round $t + 1$.

If $\mathcal{G}^{(t)}$ is empty, collaboration terminates without revision. Otherwise, agents revise synchronously and continue to the next round, up to $T_{\max}$. The latest answers are then aggregated by majority voting.

Method Type	Method	MMLU-P	MATH	2Wiki	FOLIO	MuSiQue	Avg.
Single-agent	Vanilla	55.53	74.62	80.73	70.52	51.66	66.61
Ensemble	Majority Vote	58.50	77.50	83.50	70.00	49.50	67.80
Multi-agent	MAD	55.00	82.50	84.50	75.50	**57.50**	71.00
	Peer Review	60.00	83.00	**85.00**	76.00	54.00	71.60
	G-Designer	**68.00**	**89.00**	79.50	60.00	47.00	68.70
	SafeSieve	66.00	88.50	81.50	62.00	44.50	68.50
	BiGate (Ours)	67.00	86.00	83.50	**77.00**	56.00	**73.90**

Table 2: Accuracy (%) on five reasoning benchmarks. Bold denotes the best result in each column.

4 EXPERIMENTS

We evaluate BiGate on overall performance and inference cost, the contributions of the two communication gates, and selective communication under normal and persistent-error conditions.

4.1 EXPERIMENTAL SETUP

Tasks, Models, and Evaluation. We evaluate on the test sets of MMLU-Pro Wang et al. (2024) and MATH-500 Hendrycks et al. (2021); Lightman et al. (2024), the validation sets of 2WikiMultiHopQA (2WikiQA) Ho et al. (2020) and FOLIO Han et al. (2024), and the MuSiQue dev set Trivedi et al. (2022). For each dataset, we sample 100 instances with each of two fixed seeds, 17 and 73. All methods are evaluated on the same 200 sampled instances per dataset, yielding 1,000 shared evaluation instances per method across the five benchmarks. Our agent pool consists of Qwen3-14B Yang et al. (2025), Gemma-3-12B-IT Gemma Team et al. (2025), and Ministral-14B Liu et al. (2026). We disable the built-in thinking mode of Qwen3-14B and use temperature 0.2, top-p 0.9, and a maximum output length of 4,096 tokens throughout. Final answers are evaluated with deterministic task-specific scoring.

Baselines and Metrics. We compare the single-agent *Vanilla* baseline, an independent Majority Vote over the agents' initial answers, MAD Du et al. (2024), Peer Review Xu et al. (2023), G-Designer Zhang et al. (2025a), SafeSieve Zhang et al. (2026), and BiGate. Vanilla evaluates each of the three base models independently without inter-agent communication; Majority Vote provides a non-interactive aggregation baseline over the agents' initial answers. MAD exchanges full peer responses, Peer Review performs directed peer review before revision, G-Designer uses task-conditioned communication topologies, and SafeSieve progressively prunes low-value communication. All conditions are built from the same three base models and use the same evaluation instances and generation settings. For BiGate, we set $T_{\max} = 2$. All collaborative conditions are capped at three *Response Rounds*, including the initial response generation; BiGate may terminate earlier when its communication graph is empty. Model calls count all per-instance model invocations during evaluation, including initial generation and any verification, review, or revision calls, whereas Response Rounds count answer-generation stages, including the initial generation and subsequent revisions. We report final-answer accuracy and, for multi-agent methods, average input, output, and total tokens, model calls, and Response Rounds.

4.2 OVERALL PERFORMANCE AND INFERENCE COST

Overall Performance. Table 2 compares BiGate with single-agent inference, non-interactive aggregation, and collaborative baselines across the five benchmarks.

BiGate achieves the highest macro-average accuracy of 73.90%, 2.30 percentage points above Peer Review, the strongest collaborative baseline at 71.60%. It also exceeds Majority Vote and the single-agent Vanilla baseline by 6.10 and 7.29 points, respectively. Individual task leaders vary across the benchmark suite; thus, the result reflects the strongest aggregate performance across heterogeneous tasks rather than uniform task-wise dominance.

Method	Tokens			Execution	
	Input	Output	Total ↓	Calls ↓	Response Rounds ↓
MAD	11,822	1,869	13,691	9.00	3.00
Peer Review	26,029	3,255	29,284	15.00	3.00
GDesigner	12,467	1,737	14,204	10.00	3.00
SafeSieve	13,682	1,963	15,645	10.00	3.00
BiGate	13,611	1,816	15,427	**8.80**	**2.48**

Table 3: Average inference cost per problem. Response Rounds include the initial generation.

Variant	R	S	Avg. Acc. ↑	Total Tok. ↓	Calls ↓	Response Rounds ↓
FullConnect	–	–	71.60	29,284	15.00	3.00
Receiver-Only	✓	–	71.10	26,835	14.61	2.99
Sender-Only	–	✓	72.40	20,007	11.04	3.00
BiGate	✓	✓	**73.90**	**15,427**	**8.80**	**2.48**

Table 4: Ablation results of the communication gates. R and S denote the Receiver Scope Gate and Sender Need Gate, respectively.

Inference Cost. Table 3 reports the inference cost of the multi-agent methods. BiGate requires 8.80 model calls and 2.48 Response Rounds per problem, both the lowest among the compared multi-agent methods. Its average total-token usage is 15,427: lower than SafeSieve (15,645) and Peer Review (29,284), but higher than MAD (13,691) and G-Designer (14,204). These results show that BiGate attains the strongest macro-average accuracy while requiring the fewest model calls and Response Rounds among the evaluated multi-agent methods. We next examine how the two gates contribute to this accuracy–cost profile.

4.3 Analysis of the Dual-Gated Mechanism

Ablation Design. Table 4 reports the ablation results of the two communication gates. We ablate the two communication gates to examine their distinct roles. FullConnect retains ternary verification but disables both gates, allowing full peer reasoning to propagate without selective gating. Receiver-Only enables only the Receiver Scope Gate, Sender-Only enables only the Sender Need Gate, and BiGate enables both gates. The Receiver Scope Gate controls source-information scope, whereas the Sender Need Gate controls whether a directed message is needed. The shared models, evaluation instances, and generation settings allow us to compare the effects of each gate on accuracy and inference cost, both individually and in combination.

Sender Need Gate. Enabling the Sender Need Gate alone produces a clear reduction in collaboration cost. Relative to FullConnect, Sender-Only reduces average total tokens from 29,284 to 20,007, a 31.7% reduction, while decreasing model calls from 15.00 to 11.04. Accuracy changes from 71.60% to 72.40%, although the primary role of this gate is communication-need control rather than accuracy improvement. This behavior is consistent with its design: when the sender judges the receiver's current response as Verified, no directed message is generated.

Receiver Scope Gate and Joint Configuration. The Receiver Scope Gate addresses a different decision: whether the sender's full reasoning may enter the receiver's revision context. Relative to FullConnect, Receiver-Only reduces total tokens from 29,284 to 26,835, while macro-average accuracy changes from 71.60% to 71.10%. Compared with Sender-Only, the complete BiGate configuration reaches 73.90% macro-average accuracy while further reducing total tokens from 20,007 to 15,427, model calls from 11.04 to 8.80, and Response Rounds from 3.00 to 2.48. Taken together, the ablations support the gates' complementary roles: the Sender Need Gate controls whether communication is needed, while the Receiver Scope Gate controls how much source information may enter revision. The joint configuration achieves the highest macro-average accuracy (73.90%) and lowest total-token usage (15,427) among the four variants.

4.4 Selective Communication Behavior and Persistent-Error Analysis

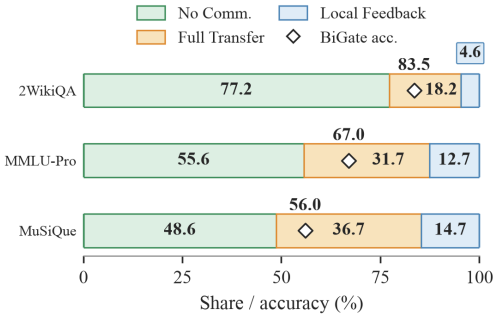


Figure 2: Communication-mode proportions on representative tasks; diamonds denote BiGate accuracy.

Selective Communication Behavior. We next examine how response-local bidirectional verification relations translate into communication modes in practice. Figure 2 reports three representative tasks spanning the observed range of *No Communication* shares. This proportion decreases from 77.19% on 2WikiQA to 55.64% on MMLU-Pro and 48.63% on MuSiQue, while *Full-Reasoning Transfer* and *Local-Feedback Exchange* become correspondingly more frequent. These distributions show that BiGate does not impose a fixed communication regime across tasks; instead, the current bidirectional relation determines the communication mode for each agent pair. The accuracy markers are shown only as a task-level performance reference.

Persistent-Error Analysis. We further examine the Receiver Scope Gate when one agent persistently provides an incorrect answer together with corresponding erroneous reasoning, while the other two agents follow the standard BiGate procedure. Under this condition, BiGate attains a macro-average accuracy of 68.55%. Figure 3 shows that normal agents judge the erroneous source's current response as *Error Detected* in 73.17% of directed verifications, *Undetermined* in 11.06%, and *Verified* in 15.77%. Because both Error Detected and Undetermined map to Local scope under the Receiver Scope Gate, 84.23% of these judgments do not permit the erroneous source's full reasoning to enter the corresponding revision context. At the instance level, in 58.88% of fault-injected examples, neither normal agent assigns a Verified signal to the erroneous source at any point in the evaluated collaboration process. The gate therefore restricts full-reasoning exposure without categorically removing the source from communication.

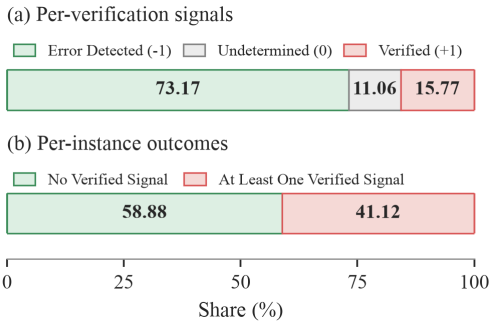


Figure 3: Verification outcomes for a persistently erroneous source at verification and instance levels.

5 Conclusion

We presented **BiGate**, a bidirectional verification-guided selective communication mechanism for multi-agent reasoning. BiGate uses response-local bidirectional ternary verification to assign complementary roles to the Sender Need Gate and Receiver Scope Gate, jointly controlling communication need, feedback function, and source-information scope. The resulting messages form a dynamic directed typed communication graph, while two-stage context isolation prevents verification-time peer reasoning from entering revision unless admitted by the gated message construction. Across five reasoning benchmarks and three heterogeneous base models, BiGate achieves the strongest macro-average accuracy among the evaluated methods while requiring the fewest model calls and Response Rounds among the evaluated multi-agent methods. Dual-gate ablations further support the distinct roles of the two gates, and behavioral analyses show task-varying communication patterns together with restricted full-reasoning admission under a persistent-error condition. Together, these results support BiGate's use of bidirectional verification to coordinate selective communication at the directed peer-message level in the evaluated setting.

AI USE STATEMENT

Beyond the language models evaluated as part of the experimental methodology described in this paper, generative AI tools were used for literature search and discovery, translation, language editing, and manuscript polishing. They were also used to assist with consistency checks across the manuscript. The research ideas, methodology, experimental design, and original manuscript were developed by the authors. All AI-assisted outputs were reviewed and verified by the authors, who take full responsibility for the final content of this work.

REFERENCES

Justin Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7066–7085, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.381. URL https://aclanthology.org/2024.acl-long.381/.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Wei Fan, JinYi Yoon, and Bo Ji. iMAD: Intelligent multi-agent debate for efficient and accurate LLM inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(35):29403–29411, 2026. doi: 10.1609/aaai.v40i35.40181. URL https://ojs.aaai.org/index.php/AAAI/article/view/40181.

Gemma Team et al. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Wanshuang Gou and Zihan Liu. Dynamic trust-aware sparse communication topology for LLM-based multi-agent consensus, 2026. URL https://arxiv.org/abs/2606.01828.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22017–22031, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1229. URL https://aclanthology.org/2024.emnlp-main.1229/.

Pengfei He, Zhenwei Dai, Xianfeng Tang, Yue Xing, Hui Liu, Jingying Zeng, Qiankun Peng, Shrivats Agrawal, Samarth Varshney, Suhang Wang, Jiliang Tang, and Qi He. To trust or not to trust: Attention-based trust management for LLM multi-agent systems, 2025. URL https://arxiv.org/abs/2506.02546.

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*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Jen-Tse Huang, Jiaxu Zhou, Tailin Jin, Xuhui Zhou, Zixi Chen, Wenxuan Wang, Youliang Yuan, Michael Lyu, and Maarten Sap. On the resilience of LLM-based multi-agent collaboration with faulty agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26202–26226. PMLR, 2025. URL https://proceedings.mlr.press/v267/huang25ay.html.

Hui Yi Leong, Yuheng Li, Yuqing Wu, Wenwen Ouyang, Wei Zhu, Jiechao Gao, and Wei Han. AMAS: Adaptively determining communication topology for LLM-based multi-agent system. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2061–2070, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-industry.144. URL https://aclanthology.org/2025.emnlp-industry.144/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Alexander H. Liu et al. Ministral 3, 2026. URL https://arxiv.org/abs/2601.08584.

Wendy Yaqiao Liu, Rui Jerry Huang, Anastasia Miin, and Lei Ding. Adaptive coopetition: Leveraging coarse verifier signals for resilient multi-agent LLM reasoning. In *The 14th International Joint Conference on Natural Language Processing and The 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 145–155, Mumbai, India, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.ijcnlp-srw.13. URL https://aclanthology.org/2025.ijcnlp-srw.13/.

Xu Shen, Yixin Liu, Yiwei Dai, Yili Wang, Rui Miao, Yue Tan, Shirui Pan, and Xin Wang. Understanding the information propagation effects of communication topologies in LLM-based multi-agent systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12347–12361, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.623. URL https://aclanthology.org/2025.emnlp-main.623/.

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

Xiao Wang, Jia Wang, Yijie Wang, Pengtao Dang, Sha Cao, and Chi Zhang. MARS: toward more efficient multi-agent collaboration for LLM reasoning, 2025. URL https://arxiv.org/abs/2509.20502.

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*, volume 37, pp. 95266–95290, 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract.html.

Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. Towards reasoning in large language models via multi-agent peer review collaboration, 2023. URL https://arxiv.org/abs/2311.08152.

An Yang et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yinsheng Yao, Jiehao Tang, Zhaozhen Yang, and Dawei Cheng. MAVEN: Multi-agent verification-elaboration network with in-step epistemic auditing, 2026. URL https://arxiv.org/abs/2605.07646.

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 *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76678–76692. PMLR, 2025a. URL https://proceedings.mlr.press/v267/zhang25cu.html.

Jusheng Zhang, Yijia Fan, Kaitong Cai, Xiaofei Sun, and Keze Wang. OSC: Cognitive orchestration through dynamic knowledge alignment in multi-agent LLM collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6320–6337, Suzhou, China, 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.335. URL https://aclanthology.org/2025.findings-emnlp.335/.

Ruijia Zhang, Xinyan Zhao, Ruixiang Wang, Sigen Chen, Guibin Zhang, An Zhang, Kun Wang, and Qingsong Wen. SafeSieve: From heuristics to experience in progressive pruning for LLM-based multi-agent communication. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40 (35):29892–29900, 2026. doi: 10.1609/aaai.v40i35.40236. URL https://ojs.aaai.org/index.php/AAAI/article/view/40236.

A EXPERIMENTAL PROMPTS

This appendix documents the model-facing prompts used in the BiGate experiments. The protocol consists of three sequential stages: independent initial reasoning, peer evaluation, and reasoning revision. Benchmark-specific task prompts and answer-format constraints are inserted into the corresponding placeholders. Gold answers are never provided to the models.

A.1 PROMPT COMPOSITION

For each example, the runner first constructs a benchmark-specific task question and an answer-format instruction. The three-agent protocol then applies the following prompts sequentially:

1. **Initial reasoning:** each agent independently solves the task.
2. **Peer voting:** each agent evaluates the reasoning and answer of other agents.
3. **Revision:** agents revise their reasoning based on accepted communication messages from the peer evaluation stage.

The prompt templates below receive only the task question, reasoning information, and communication messages generated during the protocol.

A.2 INITIAL REASONING PROMPT

```
Solve the following task independently using concise, numbered reasoning
    steps.

Return only valid JSON with this exact shape:
{
  "steps": [
    {"label": "Step 1", "content": "<reasoning step>"},
    {"label": "Step 2", "content": "<reasoning step>"}
  ],
  "final_answer": "<final answer>"
}

Rules:
- Include at least two concise reasoning steps.
- Generate a reasonable number of steps according to the difficulty of
    the question.
- Keep steps independently locatable; do not merge all reasoning into one
     paragraph.
- final_answer must be a JSON string, even when the answer is numeric.
- {answer_instructions}
- Use valid JSON escaping, including for backslashes in formulas.
- Do not include Markdown, code fences, or text outside JSON.

Question:
{question}
```

A.3 PEER VOTING PROMPT

```
You are validating two agents in a multi-agent reasoning experiment.

Evaluate every target exactly once in this single call. Independently
    judge each
target's final answer and decisive reasoning directly against the
    question.

Return only valid JSON with this exact shape:
{
  "votes": [
```

```
    {
      "target_agent_id": "<target agent id>",
      "vote": 0,
      "target_step": "Step 2",
      "feedback": "<one most important supplement or correction>"
    }
  ]
}

Decision order:
1. Vote -1 if you can identify a clear error that invalidates the final
    answer
   or a necessary reasoning step.
2. Vote 1 if the final answer is correct and the reasoning is sufficient,
     with
   no identified error affecting the conclusion.
3. Vote 0 if correctness cannot be reliably determined or an important
   justification required for verification is missing.

Rules:
- For vote 1, target_step and feedback must both be null.
- For vote 0, provide exactly one most important supplement or
    clarification.
- For vote -1, provide exactly one earliest decisive error correction.
- target_step must be an existing target step label, or a gap location.
- Do not list multiple downstream symptoms of the same root problem.
- Use plain text formulas inside JSON strings and include no text outside
     JSON.

Question:
{question}

Target reasonings:
{targets_json}
```

A.4 REVISION PROMPT

```
Re-evaluate the question independently using the context below.

Review your current reasoning and all incoming feedback as a whole.
    Feedback may
be correct, incomplete, or mistaken. Think through the entire problem and
     revise
any reasoning steps and the final answer as needed. A target_step only
    locates
the part discussed by the feedback; it does not restrict what you may
    revise.

Return only valid JSON with this exact shape:
{
  "revised_reasoning": {
    "steps": [
      {"label": "Step 1", "content": "<complete revised step>"},
      {"label": "Step 2", "content": "<complete revised step>"}
    ],
    "final_answer": "<final answer>"
  }
}

Rules:
- revised_reasoning must be a complete answer, not a patch.
- Include at least two concise, independently locatable reasoning steps.
- final_answer must be a JSON string, even when the answer is numeric.
- {answer_instructions}
- Use valid JSON escaping and include no Markdown or text outside JSON.

Question:
{question}

Your current reasoning:
{current_json}

Incoming messages:
{messages_json}
```