SOCIAL CIRCUITS BEHIND MULTI-AGENT ECHO CHAMBERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-model agents exchange messages to combine evidence, but their communication can also create echo chambers that reinforce shared errors. However, overall task performance does not explain how a message changes the receiving agent's internal activations and affects its decision. In this work, we introduce SOCIAL CIRCUITS, a framework for tracing message effects through receiver activations. We compare the receiver's answers before and after changing a message. Then, we restore selected activations recorded under the original message to determine how much of the message effect these activations reproduce. Based on SOCIAL CIRCUITS, we propose *Circuit-Guided Deliberation* (CGD), which learns to select useful messages using receiver activation changes. We establish when activation replacement preserves receiver decisions and bound the gap between CGD's task performance and the best achievable through message selection. Experiments show that receiver activation changes explain the message effects and guide message selection that improves the task performance. Across three models and four datasets, CGD achieves the highest or joint-highest average accuracy in our main comparisons while generating fewer tokens than multi-agent baselines.

1 INTRODUCTION

Multi-agent systems enable language models to solve tasks collaboratively by exchanging messages that present evidence, reasoning, or proposed answers. A message can correct a receiving agent's mistake, but it can also reinforce an error already shared by several agents. We call the latter case an *echo chamber*. Fig. 1 illustrates that agreement can arise both when communication corrects errors and when it reinforces them.

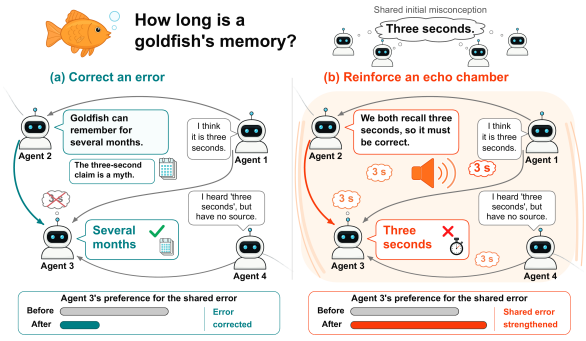


Figure 1: Comparison of the error correction and an echo chamber. True evidence corrects the receiver's answer, whereas a message presenting the agents' shared error as supporting evidence reinforces the echo chamber.

Existing works show that debate can improve collective answers (Du et al., 2024), while independent sampling and voting can achieve comparable performance under matched budgets (Smit et al., 2024; Wang et al., 2023). However, agents can share errors before communication (Estornell & Liu, 2024; Kim et al., 2025). Fig. 2 shows that receiving more messages can either improve or degrade the task performance. Comparing overall task performance alone does not explain how individual messages affect receivers (Lowe et al., 2019). This motivates us to study the following question:

How do messages change a receiver's internal activations and decision, and can these activation changes help select messages that improve task performance?

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.

To answer this question, we introduce SOCIAL CIRCUITS, a framework for tracing message effects through receiver activations. First, we compare the receiver's answers before and after removing a message or shuffling its tokens, with all other inputs fixed. This *matched rerun* measures the message's effect on answer scores and task performance. Then, to trace this effect inside the receiver, we keep the message removed or shuffled and replace selected receiver activations with those recorded under the original message. This *activation replacement* measures the selected activations' contribution to the message-induced answer-score changes.

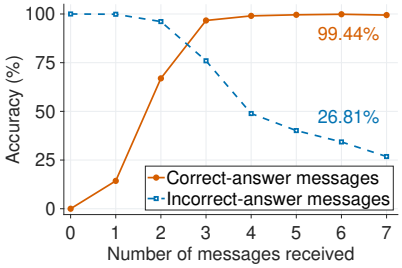


Figure 2: Receiver accuracy on Gemma-3-4B-it as more messages support the correct (orange) or incorrect (blue) answer.

Our observations show that receiver activation changes help distinguish how different messages affect the answers. Based on these observations, we propose *Circuit-Guided Deliberation* (CGD), which uses these changes to learn which messages improve task performance. CGD scores different sets of available messages, which are called *message combinations*, by using sender activations and the receiver's activation changes. The receiver then answers using the selected messages.

To sum up, our main contributions are as follows:

- We introduce SOCIAL CIRCUITS, a framework for tracing how messages affect receiver decisions through internal activations. It identifies receiver activation changes that reproduce part of the message's effect, whether the message corrects or reinforces a shared error.
- Based on SOCIAL CIRCUITS, we propose *Circuit-Guided Deliberation* (CGD). By using the sender activations and the receiver's activation changes, CGD learns to select message combinations that improve task performance.
- Through theoretical analysis, we show the condition under which the activation replacement preserves receiver decisions without exactly reproducing answer scores. We also bound CGD's expected performance gap against the optimal message combination and establish conditions under which minimizing its expected training loss closes this gap.
- Experiments show that receiver activation changes help explain message effects and improve message selection. Across three models and four datasets, CGD achieves the highest or joint-highest average accuracy in our main comparisons while generating fewer tokens than multi-agent baselines. On 2Wiki free generation, its accuracy exceeds the strongest baseline by 11.25%–30.69% under the same generation token limits.

2 RELATED WORK

Multi-agent collaboration. Debate improves collective answers (Du et al., 2024), while sampling and voting remain competitive (Wang et al., 2023; Smit et al., 2024; Choi et al., 2025). Agreement can reinforce shared errors (Kim et al., 2025; Estornell & Liu, 2024). Other approaches model response dependencies (Ai et al., 2026), favor minority answers (He et al., 2026), or train with uncertainty (Tang et al., 2026). MAD-M2 masks unreliable memories from earlier rounds (Tian et al., 2026), while MADC orders roles by reasoning-path consistency (Zhang et al., 2026a).

Communication design and selection. G-Designer learns communication graphs (Zhang et al., 2025), while MOC combines multi-hop evidence under token constraints (Guan et al., 2026). SafeSieve prunes communication using semantic evaluation and performance feedback (Zhang et al., 2026b). Shapley Message Value prunes communication through contribution estimates in cooperative multi-agent reinforcement learning (Xue et al., 2022).

Causal analysis of communication. Text interventions reveal content effects (Lin et al., 2025) and information propagation (Shen et al., 2025). Comparisons with independent agents calibrate confidence (Huang et al., 2026). Restoring transmitted components identifies information affecting predictions (Zhang & Emu, 2026). Edge masking identifies compact communication subgraphs that preserve task performance (Li et al., 2026).

Internal mechanisms. Causal abstraction studies representations in causal models (Geiger et al., 2021; 2025). Mediation analysis traces input effects (Gultchin et al., 2021), and sparse circuits identify behavior-relevant components (Marks et al., 2025). Internal causal features predict output correctness (Huang et al., 2025). Agents communicate through activations (Ramesh & Li, 2025) or state-change trajectories (Tang et al., 2025).

3 SOCIAL CIRCUITS

In this section, we introduce SOCIAL CIRCUITS, a framework for tracing how a message from one agent affects another agent's answer. Within SOCIAL CIRCUITS, we first apply the *message rerun* to measure the message's effect and *activation replacement* to study whether selected activations in the receiving agent reproduce this effect. Then, we evaluate whether the message corrects the receiving agent's answer or reinforces an error shared by the agents. SOCIAL CIRCUITS establishes a unified causal framework that connects the message exchange between agents, changes in receiver activations, and task performance.

We consider language-model agents that exchange text messages to solve a task. Each agent can both send and receive messages. For a given message, we refer to the agent sending it as the *sender* and the agent receiving it as the *receiver*. We measure how the message affects the receiver's answer while keeping the task input and preceding conversation fixed. Let $\mathcal{O}$ be a finite, nonempty set of candidate answers. We denote the receiver's *answer score* for the candidate answer $o \in \mathcal{O}$ by $F(o)$. We use log probabilities as answer scores, calculated from the receiver's token probabilities conditioned on its input and preceding answer tokens. Appendix D.4 provides the detailed calculation of $F(o)$. A higher score indicates a higher probability assigned to that answer. The receiver selects $q = \arg\max_{o \in \mathcal{O}} F(o)$. We denote the corresponding *task utility* as $u(q)$ to measure the performance under the task's evaluation rule, with higher values indicating better performance.

3.1 METRICS FOR MEASURING MESSAGE EFFECTS

We introduce metrics to characterize a message's effect on the receiver's answer scores and task performance. We also quantify how much of the answer-score effect is reproduced by activation replacement. We calculate these metrics using three runs as illustrated in Figure 3: the *original-message run*, the *reference run*, and the *activation-replacement run*. We fix the task input, preceding conversation, and receiver model. (i) In the *original-message run*, the receiver reads the sender's original message, and we record its answer scores $F^{\mathrm{msg}}(o)$ and internal activations. (ii) In the *reference run*, we omit this message or shuffle its tokens and record the corresponding answer scores as $F^{\mathrm{ref}}(o)$. By comparing these two runs, we characterize the effect of the original message against the reference message.

We use *activation replacement* (Vig et al., 2020; Meng et al., 2022; Zhang & Nanda, 2024) to determine whether changing receiver activations alone can reproduce the message's effect on the answer scores. For activation replacement, we select receiver layers at regular intervals and record the activation vectors of fixed tokens after the message. Because the text of these tokens remains unchanged across runs, we can compare how the message changes their activation vectors. The selected layers and token positions are provided in Appendix D.2. (iii) In the *activation-replacement run*, we keep the reference input unchanged and replace the selected receiver activation vectors with those recorded at the corresponding layers and tokens in the original-message run. We record the receiver's answer scores as $F^{\mathrm{act}}(o)$.

Answer-score change. For a candidate answer $o \in \mathcal{O}$, we define the *answer-score changes* caused by the original message and activation replacement as follows:

$$\Delta F^{\mathrm{msg}}(o) = F^{\mathrm{msg}}(o) - F^{\mathrm{ref}}(o), \tag{1}$$

$$\Delta F^{\mathrm{act}}(o) = F^{\mathrm{act}}(o) - F^{\mathrm{ref}}(o). \tag{2}$$

In particular, $\Delta F^{\mathrm{msg}}(o)$ shows how much the original message affects the score of answer o. $\Delta F^{\mathrm{act}}(o)$ shows how replacing the selected activations affects that score while keeping the reference message unchanged. We compare $\Delta F^{\mathrm{act}}(o)$ with $\Delta F^{\mathrm{msg}}(o)$ to evaluate whether activation replacement reproduces the message's effect on the answer score.

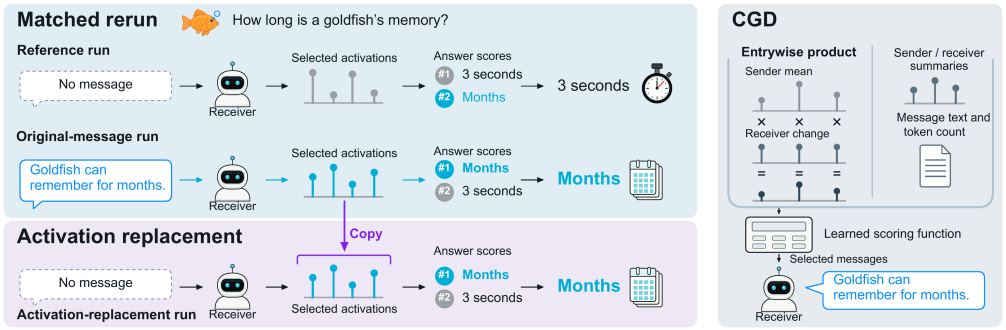


Figure 3: Illustration of SOCIAL CIRCUITS and Circuit-Guided Deliberation (CGD). Matched reruns change the message, whereas activation replacement changes selected receiver activations while preserving the reference input. CGD uses sender summaries and receiver activation changes to select message combinations.

Task-performance change. We measure changes in the receiver's task performance using the task utility $u(q)$ of its selected answer q. For example, under the exact-match evaluation, $u(q) = 1$ denotes a correct answer and $u(q) = 0$ otherwise. Let q^{ref}, q^{msg}, and q^{act} denote the answers selected in the reference, original-message, and activation-replacement runs, respectively. We define the corresponding *task-performance changes* as follows:

$$\Delta u^{\text{msg}} = u(q^{\text{msg}}) - u(q^{\text{ref}}), \tag{3}$$

$$\Delta u^{\text{act}} = u(q^{\text{act}}) - u(q^{\text{ref}}). \tag{4}$$

We compare Δu^{act} with Δu^{msg} to evaluate whether activation replacement reproduces the message's effect on the task performance. Positive values indicate better performance than in the reference run, and negative values indicate worse performance.

Message-effect recovery. To measure whether a message supports the correct answer or a shared error, we compare the correct and incorrect answers specified by the task. For correction measurements, $o_1 \in \mathcal{O}$ denotes the correct answer and $o_2 \in \mathcal{O}$ the incorrect answer. For shared-error measurements, we reverse this order. The same answers and order are used in all three runs. We calculate the changes in their score difference as follows:

$$\begin{aligned} \Delta^{\text{msg}} &= \Delta F^{\text{msg}}(o_1) - \Delta F^{\text{msg}}(o_2), \\ \Delta^{\text{act}} &= \Delta F^{\text{act}}(o_1) - \Delta F^{\text{act}}(o_2). \end{aligned} \tag{5}$$

When o_1 is an incorrect answer which is favored by several agents, a positive value of Δ^{msg} indicates reinforcement of their *echo chamber*. To quantify how much of the message effect is reproduced through receiver activations, we define the *message-effect recovery ratio* as $\rho = \frac{\mathbb{E}[\Delta^{\text{act}}]}{\mathbb{E}[\Delta^{\text{msg}}]}$, where $\mathbb{E}$ averages over the same tasks and random seeds for message generation.

3.2 KEY OBSERVATIONS

Based on the metrics we discussed above, we present the results in Fig. 4 to show how messages affect receivers' answers and whether these effects can be traced through receiver activations. We present two observations as follows.

Receiver activation changes are specific to each message. We present the fractions of messages that correct a receiver's answer or change a correct answer into an incorrect one in Fig. 4(a). Results show that both changes occur in each model. For example, in Qwen3-4B, 5.70% of messages correct an answer, while 1.20% introduce an error. Thus, communication can improve average task performance even though some messages make the receiver's answer incorrect. To trace these message effects through receiver activations, we present the answer-score changes in Figs. 4(b) and (c). Results show that activation replacement closely reproduces the answer-score changes caused by the same message. Using activation changes induced by another message increases the mean absolute

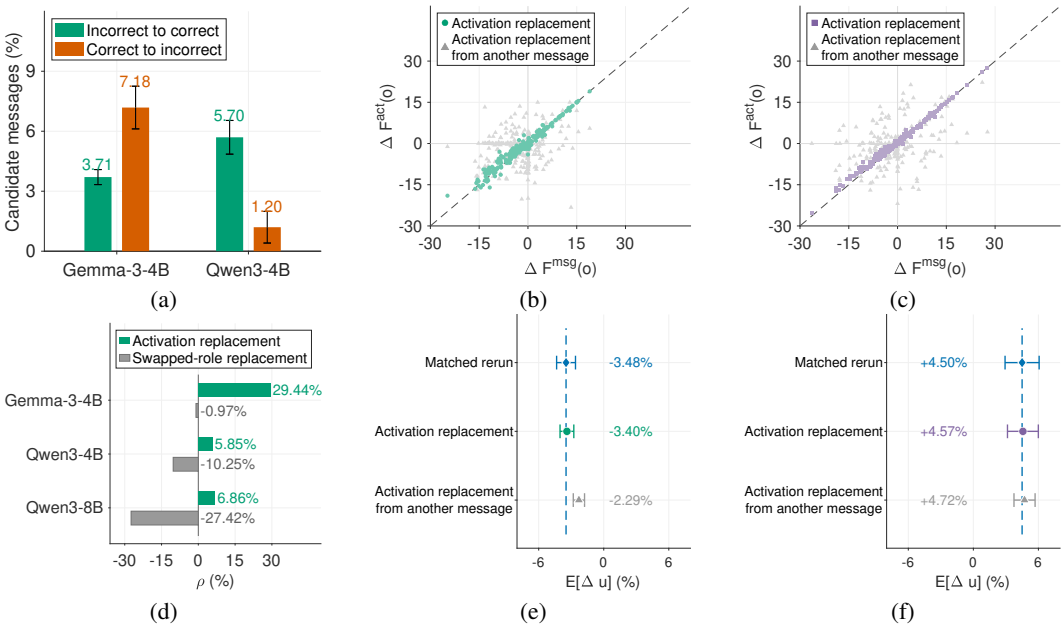


Figure 4: (a) Fractions of messages that correct or introduce errors for Gemma-3-4B and Qwen3-4B. (b) Answer-score changes for Gemma-3-4B. (c) Answer-score changes for Qwen3-4B. (d) Message-effect recovery ρ across three models. (e) Average task-performance changes for Gemma-3-4B. (f) Average task-performance changes for Qwen3-4B. Diagonal dashed lines in (b) and (c) mark equal answer-score changes. Figs. (a)–(c), (e), and (f) use the same candidate messages within each model. Task performance is the accuracy of the higher-scoring answer, and error bars show standard deviations across random seeds.

error from 0.187 to 1.280 in Gemma-3-4B and from 0.121 to 1.039 in Qwen3-4B. These results show that receiver activation changes help distinguish how different messages affect answer scores.

Activation replacement reproduces the average task performance changes. We further show that these receiver activation changes can reproduce message effects on receiver answers. Fig. 4(d) presents the message-effect recovery across three models on a synthetic task with messages supporting correct answers or shared errors. For comparison, we exchange sender and receiver roles in a separate run. We apply the changes recorded in the original sender to the original receiver, which we call *swapped-role replacement*. Changes from the original sender or another message are rescaled to match the norm of the original receiver's changes. Receiver activation replacement reproduces the direction of the average message effect across all three models, while the swapped-role replacement reverses this direction. However, a change in answer scores does not always change the receiver's answer. Therefore, we compare the task performance in Figs. 4(e) and (f). Activation replacement closely matches the average performance decrease in Gemma-3-4B and the increase in Qwen3-4B. For individual answers, Table 6 in Appendix E.3 shows that activation replacement reproduces over 96% of message-induced answer changes on 2Wiki in both models. Using changes from another message recovers 41.07% in Gemma-3-4B and 55.06% in Qwen3-4B. Thus, the receiver changes traced by SOCIAL CIRCUITS can reproduce both message-induced error correction and error introduction.

Remark. The aforementioned observations demonstrate that SOCIAL CIRCUITS uses receiver activation changes to trace how messages correct answers or introduce errors. These changes distinguish the effects of different messages. This motivates using receiver activation changes to learn which messages improve task performance.

3.3 THEORETICAL ANALYSIS

The aforementioned observations show that receiver activation replacement can reproduce answer changes caused by messages. We establish sufficient conditions under which activation replacement

preserves the receiver's answer and the message's effect on task performance. Since the receiver selects the highest-scoring answer, we characterize these conditions through the score differences between candidate answers. For a task with at least two candidate answers, we compare the score difference for each answer pair in the original-message and activation-replacement runs. We define the *recovery error* err as the largest absolute change in the answer score difference between the original-message and activation-replacement runs, as follows:

$$\mathrm{err} = \max_{\substack{o_1,o_2\in\mathcal{O}\\ o_1\neq o_2}} \left|\left[F^{\mathrm{msg}}(o_1) - F^{\mathrm{msg}}(o_2)\right] - \left[F^{\mathrm{act}}(o_1) - F^{\mathrm{act}}(o_2)\right]\right|. \tag{6}$$

For activation replacement to change the receiver's answer from q^{msg}, at least one answer $o \in \mathcal{O} \setminus \{q^{\mathrm{msg}}\}$ must satisfy $F^{\mathrm{act}}(o) \geq F^{\mathrm{act}}(q^{\mathrm{msg}})$. We denote the smallest score difference between q^{msg} and the remaining answers in the original-message run by gap, which is defined as follows:

$$\mathrm{gap} = F^{\mathrm{msg}}(q^{\mathrm{msg}}) - \max_{o\in\mathcal{O}\setminus\{q^{\mathrm{msg}}\}} F^{\mathrm{msg}}(o). \tag{7}$$

The following theorem provides conditions for activation replacement to preserve the receiver's answer and the direction of the message's effect on the score difference between o_1 and o_2.

Theorem 1 (Preserving answers and message effects)**.** *If* $\mathrm{err} < \mathrm{gap}$*, activation replacement preserves the answer and task-performance change of the original-message run, i.e.,* $q^{\mathrm{act}} = q^{\mathrm{msg}}$ *and* $\Delta u^{\mathrm{act}} = \Delta u^{\mathrm{msg}}$*. If* $|\Delta^{\mathrm{msg}}| > \mathrm{err}$*, activation replacement and the original message change the score difference between* o_1 *and* o_2 *in the same direction, i.e.,* $\Delta^{\mathrm{act}}\Delta^{\mathrm{msg}} > 0$*.*

The proof is provided in Appendix B.1. Theorem 1 shows that SOCIAL CIRCUITS does not need to recover each answer score exactly to recover a message's effect on the task performance. If activation replacement keeps the same highest-scoring answer, the receiver's task performance remains unchanged. In addition, the original message and activation changes can both strengthen or both weaken a shared error.

4 CIRCUIT-GUIDED DELIBERATION

In this section, we propose *Circuit-Guided Deliberation* (CGD) to select messages for the receiver's final answer. SOCIAL CIRCUITS traces message effects through receiver activation changes. We use these changes and sender information to learn which messages improve task performance. We first describe the scoring function, followed by training, message selection, and theoretical guarantees.

4.1 SCORING FUNCTION

Each sender forms a candidate message. Let $\mathcal{C}$ denote the set of these candidate messages. Let $\mathcal{A} \subseteq \mathcal{C}$ denote a *message combination*. It specifies which messages the receiver uses for its final answer. We denote the set of all combinations by $\mathcal{S} = \{\mathcal{A} \mid \mathcal{A} \subseteq \mathcal{C}\}$. $\mathcal{S}$ includes the empty combination $\varnothing$, corresponding to a final answer without candidate messages. For each sender, we record activations after it reads the input and its own evidence. For the receiver, we record activations after it reads each message combination $\mathcal{A} \in \mathcal{S}$. In both cases, we use the output vector at the last input token of each selected layer before output generation.

CGD uses the recorded activations to represent sender information and receiver changes for scoring messages. In particular, to reduce the input dimension without learning additional parameters, we calculate d weighted sums of each activation vector and average these d-dimensional vectors over the selected layers. We refer to this average as an *activation summary*. The weights are randomly initialized, shared across agents using the same language model and its selected layers, and kept fixed during training. Let $\mathbf{r}_{\mathcal{A}}$ denote the receiver's activation summary after reading $\mathcal{A}$, with $\mathbf{r}_{\varnothing}$ corresponding to no candidate messages. For any nonempty set $\mathcal{A}$, let $\mathbf{s}_{\mathcal{A}}$ denote the average activation summary of the senders whose messages belong to $\mathcal{A}$. We represent the receiver changes caused by the messages as $\mathbf{r}_{\mathcal{A}} - \mathbf{r}_{\varnothing}$. We combine these changes with sender information using $\mathbf{s}_{\mathcal{A}} \odot (\mathbf{r}_{\mathcal{A}} - \mathbf{r}_{\varnothing})$, where $\odot$ denotes elementwise multiplication. This product lets CGD score the same receiver change differently depending on sender information. The scoring input $\mathbf{x}_{\mathcal{A}}$ includes the message text vector and token count, $\mathbf{s}_{\mathcal{A}}$, $\mathbf{r}_{\varnothing}$, and the aforementioned product. To compare $\mathcal{A}$ with other message combinations, we also include the average receiver summary over the other nonempty

combinations. We introduce a shared scoring function $f_{\boldsymbol{\theta}}$ that calculates a weighted sum of $\mathbf{x}_{\mathcal{A}}$ with a bias. We denote the learnable parameters as $\boldsymbol{\theta}$, which include weights and bias. Adding the same bias to all nonempty combinations leaves their ranking unchanged. It affects whether their highest score exceeds zero, which determines whether CGD uses messages. The score for selecting a message combination $\mathcal{A}$ is defined as follows:

$$S(\mathcal{A}) = f_{\boldsymbol{\theta}}(\mathbf{x}_{\mathcal{A}}) \quad \text{for } \mathcal{A} \neq \varnothing, \quad S(\varnothing) = 0. \tag{8}$$

4.2 Message Selection

We train $\boldsymbol{\theta}$ to assign higher scores to message combinations that improve task performance. For each training task, we run the receiver separately with each combination $\mathcal{A} \in \mathcal{S}$ and denote its answer as $q_{\mathcal{A}}$. By using the task score u defined in Section 3, we define the gain from using $\mathcal{A}$ instead of using no messages as $V(\mathcal{A}) = u(q_{\mathcal{A}}) - u(q_{\varnothing})$, where $q_{\varnothing}$ is the receiver's answer without using any message. To rank message combinations by their gains, we train $f_{\boldsymbol{\theta}}$ using the pairwise comparisons. For each training task, we consider two combinations $\mathcal{A}, \mathcal{B} \in \mathcal{S}$ with $V(\mathcal{A}) > V(\mathcal{B})$. We denote its pairwise logistic loss (Burges et al., 2005) as

$$\ell(\mathcal{A}, \mathcal{B}) = \big(V(\mathcal{A}) - V(\mathcal{B})\big) \log(1 + \exp[S(\mathcal{B}) - S(\mathcal{A})]) \,. \tag{9}$$

If there is a larger difference in task scores between $\mathcal{B}$ and $\mathcal{A}$, this loss increases the penalty for ranking $\mathcal{B}$ above $\mathcal{A}$. We update $\boldsymbol{\theta}$ by minimizing the summed pairwise loss and leaving the bias unpenalized. During implementation, CGD keeps $\boldsymbol{\theta}$ fixed and selects the highest-scoring message combination as $\widehat{\mathcal{A}} = \arg\max_{\mathcal{A} \in \mathcal{S}} S(\mathcal{A})$. The receiver then answers using the messages in $\widehat{\mathcal{A}}$.

4.3 Message-Selection Guarantees

Minimizing a pairwise loss does not always select the message combination with the highest expected task utility (Duchi et al., 2010; Dembczyński et al., 2012). Hence, we bound CGD's loss in expected task utility and show when pairwise training selects an optimal combination. In this analysis, we use $u(q) = 1$ for task success and 0 otherwise. With the available messages, scoring inputs, and learned scoring function fixed, let $\mathcal{P}$ denote the joint distribution of receiver answers and task utilities across message combinations and activation-replacement runs. We denote expectation and probability under this distribution by $\mathbb{E}_{\mathcal{P}}$ and $\Pr_{\mathcal{P}}$, respectively. We compare CGD's selection $\widehat{\mathcal{A}}$ with an optimal combination $\mathcal{A}^\star \in \arg\max_{\mathcal{A} \in \mathcal{S}} \mathbb{E}_{\mathcal{P}}[u(q_{\mathcal{A}})]$. The expected task-utility difference between the optimal message combination and CGD's selection is $\mathbb{E}_{\mathcal{P}}[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})]$.

If activation replacement preserves receiver answers, the task-utility differences between message combinations remain unchanged. We replace the receiver's selected activations in the no-message run with those recorded after it reads $\mathcal{A}$. We denote the receiver's answer after replacement by $q_{\mathcal{A}}^{\mathrm{act}}$. The corresponding utility is $u_{\mathcal{A}}^{\mathrm{act}} = u(q_{\mathcal{A}}^{\mathrm{act}})$, with $q_{\varnothing}^{\mathrm{act}} = q_{\varnothing}$. Thus, $\Pr_{\mathcal{P}}(q_{\mathcal{A}}^{\mathrm{act}} \neq q_{\mathcal{A}})$ is the probability that activation replacement and the run using $\mathcal{A}$ return different answers. We denote the covariance between the task utilities of different message combinations by $\mathrm{Cov}_{\mathcal{P}}$.

Based on ℓ defined in Eq. (9), we define the expected pairwise loss as $\mathcal{L}(S) = \mathbb{E}_{\mathcal{P}}[\sum_{\mathcal{A},\mathcal{B} \in \mathcal{S}:\, V(\mathcal{A}) > V(\mathcal{B})} \ell(\mathcal{A}, \mathcal{B})]$. Let $\mathcal{L}^\star$ denote the infimum of $\mathcal{L}(S)$ over all real-valued message-combination scores. Then, we bound the expected task-utility difference between the optimal message combination and CGD's selection in the following theorem.

Theorem 2 (CGD's Expected Task-Utility Difference)**.** *For $|\mathcal{S}| \geq 2$, any joint distribution $\mathcal{P}$, and any finite learned scores, the expected task-utility difference between $\mathcal{A}^\star$ and $\widehat{\mathcal{A}}$ is bounded by* $\mathbb{E}_{\mathcal{P}}[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})] \leq \sum_{\mathcal{B} \in \mathcal{S} \setminus \{\mathcal{A}^\star, \widehat{\mathcal{A}}\}} \max\{0, \mathrm{Cov}_{\mathcal{P}}(u_{\widehat{\mathcal{A}}}^{\mathrm{act}} - u_{\mathcal{A}^\star}^{\mathrm{act}}, u_{\mathcal{B}}^{\mathrm{act}})\} + \sqrt{|\mathcal{S}|(\mathcal{L}(S) - \mathcal{L}^\star)} + 4(|\mathcal{S}| - 2) \max_{\mathcal{A} \in \mathcal{S}} \Pr_{\mathcal{P}}(q_{\mathcal{A}}^{\mathrm{act}} \neq q_{\mathcal{A}})$.

Theorem 2 bounds the expected task-utility difference for CGD's learned scoring function. If activation replacement preserves receiver answers and task utilities after replacement have equal covariance across distinct combinations, the first and third terms vanish. Under these conditions, reducing $\mathcal{L}(S) - \mathcal{L}^\star$ tightens the bound on CGD's expected task-utility difference. The proof is provided in Appendix B.3.

Table 1: Accuracy (%, mean ± standard deviation over three seeds) and mean tokens per task for generated messages and final answers under the original generation settings. On 2Wiki A/B, CGD sends passages and responses, and ollaboration baselines send responses alone.

		Single-agent		Multi-agent						
Model	Metric	Single	SC6	Majority	MAD	MAD-M2	MADC	MOC	SafeSieve	**CGD**
2Wiki (A/B answers)										
Gemma-3-4B	Accuracy ↑	78.89±0.48	77.78±1.46	83.19±0.64	80.69±1.05	82.92±1.50	79.44±1.27	86.94±0.96	79.72±3.07	**89.72**±1.46
	Tokens ↓	79	478	219	647	516	638	401	485	139
Qwen3-4B	Accuracy ↑	73.19±1.73	70.83±2.32	73.61±1.68	80.42±2.08	82.08±1.44	77.22±1.27	86.94±0.64	83.89±1.05	**89.86**±1.68
	Tokens ↓	171	1025	468	978	1028	961	864	874	299
Qwen3-8B	Accuracy ↑	75.69±0.48	72.92±0.42	80.97±2.55	81.81±2.71	78.47±2.77	78.61±0.24	85.83±0.42	81.94±1.97	**87.36**±0.64
	Tokens ↓	156	930	439	972	900	959	881	1004	286
GSM8K (free generation)										
Gemma-3-4B	Accuracy ↑	84.86±1.20	88.19±1.05	86.81±0.64	86.53±1.73	88.61±1.34	86.53±1.73	**88.75**±1.50	87.64±0.24	**88.75**±0.48
	Tokens ↓	224	1329	678	1386	1248	1386	1211	1139	656
Qwen3-4B	Accuracy ↑	89.86±0.87	91.53±0.24	89.72±0.64	90.83±0.72	91.11±1.68	90.83±0.72	91.11±0.64	92.92±0.42	**93.33**±1.10
	Tokens ↓	221	1320	656	1291	1226	1291	1144	1268	640
Qwen3-8B	Accuracy ↑	91.81±1.46	92.92±0.83	92.78±1.05	**93.89**±1.73	**93.89**±1.05	**93.89**±1.73	93.75±0.42	92.64±0.48	**93.89**±0.64
	Tokens ↓	229	1383	688	1427	1325	1427	1212	1389	677
MATH-500 (A/B answers)										
Gemma-3-4B	Accuracy ↑	42.67±0.76	52.00±1.00	46.33±1.04	52.83±0.58	53.83±2.02	52.83±0.58	66.33±0.58	59.67±0.58	**68.83**±1.76
	Tokens ↓	419	2507	1264	2429	2411	2429	2353	2203	897
Qwen3-4B	Accuracy ↑	55.83±0.76	60.00±0.50	57.50±1.32	63.50±1.80	62.67±2.31	63.50±1.80	69.50±2.18	65.33±0.76	**70.83**±0.76
	Tokens ↓	385	2321	1168	2309	2279	2309	2143	2218	818
Qwen3-8B	Accuracy ↑	59.17±0.76	62.33±0.58	59.00±1.32	59.83±1.04	62.00±1.32	59.83±1.26	64.67±1.26	61.50±0.50	**65.83**±1.61
	Tokens ↓	387	2333	1174	2363	2308	2365	2227	2318	912
TruthfulQA (A/B answers)										
Gemma-3-4B	Accuracy ↑	62.67±1.76	63.83±1.26	68.00±1.32	66.83±2.75	70.17±0.29	66.83±2.75	68.67±1.04	67.17±1.26	**70.67**±1.04
	Tokens ↓	67	405	236	583	418	583	449	617	191
Qwen3-4B	Accuracy ↑	76.17±1.53	75.83±0.76	79.00±1.80	79.83±1.44	79.00±1.00	79.83±1.44	82.00±1.32	80.00±1.50	**83.33**±1.76
	Tokens ↓	105	627	353	828	691	828	677	716	208
Qwen3-8B	Accuracy ↑	82.67±0.58	82.00±0.50	83.67±1.04	85.00±2.29	82.50±1.50	85.00±2.29	86.33±2.08	84.83±0.58	**87.00**±1.50
	Tokens ↓	96	576	305	746	590	746	608	698	236

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Tasks and models. We use Gemma 3 4B-IT (Gemma Team, 2025), Qwen3-4B, and Qwen3-8B (Qwen Team, 2025) on 2WikiMultiHopQA (Ho et al., 2020), GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021), and TruthfulQA (Lin et al., 2022). Each evaluation set uses 240 tasks for 2Wiki and GSM8K, and 200 for MATH-500 and TruthfulQA. We compare accuracy in selecting between two supplied answers on 2Wiki, MATH-500, and TruthfulQA, and in freely generating answers on 2Wiki and GSM8K.

Message-selection baselines. **Text-only** uses message text, while **Additive** also uses sender and receiver summaries. **Activation products** adds element-wise products of sender summaries with receiver summaries before and after communication. For two-answer scoring, CGD and these baselines use the same messages. Each approach selects a message combination, and we calculate its accuracy using the receiver's recorded answer for that combination. For free generation, we train baselines with CGD's loss. Each baseline selects the message combination with its highest score.

Collaboration baselines. We compare CGD with **Single**, **SC6** (Wang et al., 2023), **Majority vote** (Choi et al., 2025), **MAD** (Du et al., 2024), **MAD-M2** (Tian et al., 2026), **MADC** (Zhang et al., 2026a), **SafeSieve** (Zhang et al., 2026b), and **MOC** (Guan et al., 2026).

Training and evaluation. We train CGD separately for two-answer scoring and free generation on 340 2Wiki tasks, with 100 validation tasks and separate evaluation sets. We report the average accuracy and standard deviation over three seeds. Generation cost counts all generated tokens, including the intermediate messages and final answer. We provide training details in Appendix D.

Table 2: Free-generation accuracy (%) on 240 2Wiki tasks under matched total generation budgets.

	Single-agent		Multi-agent						
Model	Single	SC6	Majority	MAD	MAD-M2	MADC	MOC	SafeSieve	**CGD**
Gemma-3-4B	24.72±1.05	25.42±1.67	26.39±2.77	7.22±1.88	20.97±2.10	7.22±1.88	20.28±0.48	25.42±4.23	**42.92**±0.83
Qwen3-4B	21.94±0.24	22.22±0.48	25.97±1.97	23.33±2.53	25.14±3.13	23.33±2.53	29.31±2.44	40.56±2.37	**51.81**±0.64
Qwen3-8B	23.61±2.10	23.89±1.05	30.00±1.67	14.17±0.83	30.83±0.83	14.17±0.83	28.75±5.42	29.86±5.50	**61.53**±0.48

5.2 Experimental Results

Accuracy and generation cost. We compare the accuracy and generated tokens in Table 1. Results show that CGD ranks first in all nine two-answer settings and ranks first or joint first in GSM8K free generation. Across all settings, it generates fewer tokens than all multi-agent baselines. For 2Wiki free generation, we limit each baseline's total generated tokens to CGD's recorded token count for the same task and seed. Table 2 shows accuracy gains over the strongest baseline of 16.53%, 11.25%, and 30.69% on Gemma-3-4B, Qwen3-4B, and Qwen3-8B, respectively.

Effect of receiver activation changes. We present message selection comparisons in Fig. 5 using the same messages, recorded receiver answers, and training procedure. CGD ranks first across all three models and exceeds Activation products by 0.83%–2.36%. In addition, Table 7 in Appendix E.4 shows that CGD improves accuracy over sending both messages by 4.44%–7.22% across the three models. Table 8 in Appendix E.4 shows CGD's 0.42%–1.11% accuracy advantage over Activation products on Table 1's 2Wiki tasks with matched input scaling and weight penalty. These results support learning directly from receiver activation changes to select messages.

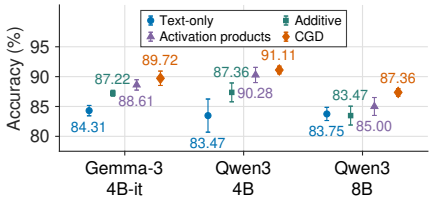
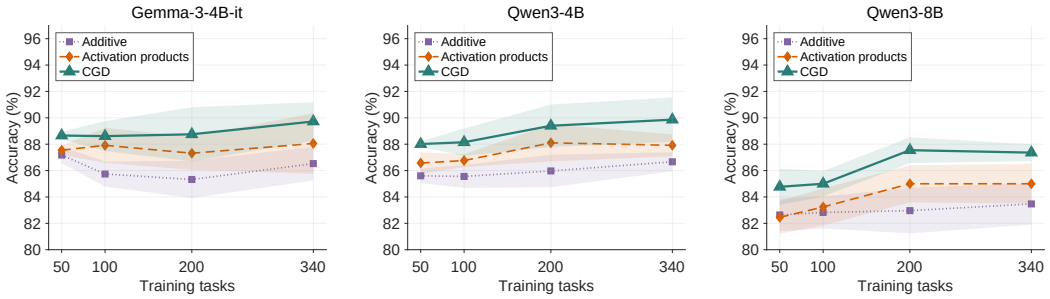


Figure 5: Two-answer scoring accuracy on 2Wiki: 240 new tasks for the 4B models, and Table 1's tasks for Qwen3-8B.

Effect of training set size. Figure 6 shows that CGD exceeds Additive and Activation products across three models with 50, 100, 200, and 340 training tasks. On Qwen3-8B, CGD reaches 85.00% accuracy with 100 tasks, matching Activation products trained on 340 tasks. It shows that CGD can achieve the same accuracy with fewer training tasks.



(a) (b) (c)

Figure 6: 2Wiki learning curves. Lines and bands show mean accuracy and standard deviation.

6 Conclusion

In this work, we introduce Social Circuits to trace how messages correct or reinforce shared errors through receiver activation changes. Based on this framework, we propose *Circuit-Guided Deliberation* to use sender information and activation changes to select messages that improve task performance. Our theory shows when activation replacement preserves receiver decisions and bounds CGD's loss in expected task utility. Across three models and four datasets, CGD achieves the highest or joint-highest average accuracy with fewer generated tokens than each multi-agent baseline, and improves the free-generation accuracy under the same generation limits.

AI USE STATEMENT

We use generative AI tools to assist with the literature search, grammar correction, experiment deployments, and manuscript polishing. The authors reviewed these outputs against the cited sources and stored experiment records and take full responsibility for the paper.

ETHICS STATEMENT

This work studies how communication changes errors in collaborative AI systems. Synthetic tasks use synthetic conflicting evidence, and the real-task transformations are derived from existing benchmarks. No human subjects are involved. Released artifacts follow the licenses of the models and datasets.

REPRODUCIBILITY STATEMENT

The anonymous repository of this project is `https://anonymous.4open.science/r/social-circuits-4416`. Appendix C gives the pseudocode, and Appendix D describes the models, data splits, target quantities, baselines, and statistical analyses. The code is also included in the supplementary material.

REFERENCES

Rui Ai, Yuqi Pan, David Simchi-Levi, Milind Tambe, and Haifeng Xu. Beyond majority voting: LLM aggregation by leveraging higher-order information. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.01499.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the International Conference on Machine Learning*, pp. 89–96, 2005. URL https://doi.org/10.1145/1102351.1102363.

Hyeong Kyu Choi, Xiaojin Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *Proceedings of the Conference on Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/934252acd87f254d5d4672fbde283bd2-Abstract-Conference.html.

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. arXiv preprint, 2021. URL https://arxiv.org/abs/2110.14168.

Krzysztof Dembczyński, Wojciech Kotłowski, and Eyke Hüllermeier. Consistent multilabel ranking through univariate loss minimization. In *Proceedings of the International Conference on Machine Learning*, 2012. URL https://icml.cc/2012/papers/661.pdf.

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 International Conference on Machine Learning*, pp. 11733–11763, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

John C. Duchi, Lester W. Mackey, and Michael I. Jordan. On the consistency of ranking algorithms. In *Proceedings of the International Conference on Machine Learning*, pp. 327–334, 2010. URL https://icml.cc/Conferences/2010/papers/421.pdf.

Andrew Estornell and Yang Liu. Multi-LLM debate: Framework, principals, and interventions. In *Proceedings of the Conference on Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/32e07a110c6c6acf1afbf2bf82b614ad-Abstract-Conference.html.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Proceedings of the Conference on Neural Information Processing Systems*, pp. 9574–9586, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4f5c422f4d49a5a807eda27434231040-Abstract.html.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, and Thomas Icard. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025. URL https://jmlr.org/papers/v26/23-0058.html.

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

Gemma Team. Gemma 4 technical report. arXiv preprint, 2026. URL https://arxiv.org/abs/2607.02770.

Yao Guan, Lin Wang, Zhihu Lu, Ziyi Wang, Wenzhu Yan, and Qiang Duan. MOC: Multi-order communication in LLM-based multi-agent systems. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.02359.

Limor Gultchin, David Watson, Matt Kusner, and Ricardo Silva. Operationalizing complex causes: A pragmatic view of mediation. In *Proceedings of the International Conference on Machine Learning*, pp. 3875–3885, 2021. URL https://proceedings.mlr.press/v139/gultchin21a.html.

Chuan He, Zebin Chen, Zhengyi Yang, Shaobo Qiao, Mingchen Ju, Jiate Liu, Dong Wen, and Guanfeng Liu. Minority sentinel: When to overturn majority voting in multi-agent LLM debates. AgentSearch Workshop at the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2026. URL https://arxiv.org/abs/2606.29270.

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*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/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 International Conference on Computational Linguistics*, pp. 6609–6625, 2020. URL https://aclanthology.org/2020.coling-main.580/.

Jiatan Huang, Mingchen Li, Ziming Li, Sunjae Kwon, Hong Yu, and Chuxu Zhang. Counterfactual graph for multi-agent LLM calibration. arXiv preprint, 2026. URL https://arxiv.org/abs/2605.30653.

Jing Huang, Junyi Tao, Thomas Icard, Diyi Yang, and Christopher Potts. Internal causal mechanisms robustly predict language model out-of-distribution behaviors. In *Proceedings of the International Conference on Machine Learning*, pp. 25791–25812, 2025. URL https://proceedings.mlr.press/v267/huang25af.html.

Elliot Myunghoon Kim, Avi Garg, Kenny Peng, and Nikhil Garg. Correlated errors in large language models. In *Proceedings of the International Conference on Machine Learning*, pp. 30038–30066, 2025. URL https://proceedings.mlr.press/v267/kim25e.html.

Junzhi Li, Peng He, Qirui Ji, Wei Wang, Lixiang Liu, and Chuxiong Sun. Discovering efficient and explainable communication topologies for LLM-based multi-agent systems via causal inference. arXiv preprint, 2026. URL https://arxiv.org/abs/2608.12921.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 3214–3252, 2022. URL https://aclanthology.org/2022.acl-long.229/.

Victoria Lin, Louis-Philippe Morency, and Eli Ben-Michael. Isolated causal effects of natural language. In *Proceedings of the International Conference on Machine Learning*, pp. 37919–37941, 2025. URL https://proceedings.mlr.press/v267/lin25k.html.

Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. On the pitfalls of measuring emergent communication. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems*, pp. 693–701, 2019. URL https://ifaamas.org/Proceedings/aamas2019/pdfs/p693.pdf.

Samuel Marks, Can Rager, Eric J. Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *Proceedings of the International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/3ba4d47a83e498c2b1a0868cba20f6de-Abstract-Conference.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Proceedings of the Conference on Neural Information Processing Systems*, 2022. URL https://papers.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

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

Vignav Ramesh and Kenneth Li. Communicating activations between language model agents. In *Proceedings of the International Conference on Machine Learning*, pp. 51094–51116, 2025. URL https://proceedings.mlr.press/v267/ramesh25a.html.

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 Conference on Empirical Methods in Natural Language Processing*, pp. 12347–12361, 2025. URL https://aclanthology.org/2025.emnlp-main.623/.

Andries Petrus 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. In *Proceedings of the International Conference on Machine Learning*, pp. 45883–45905, 2024. URL https://proceedings.mlr.press/v235/smit24a.html.

Luoxi Tang, Yuqiao Meng, Joseph Costa, Yingxue Zhang, Muchao Ye, and Zhaohan Xi. The value of variance: Mitigating debate collapse in multi-agent systems via uncertainty-driven policy optimization. In *Proceedings of the International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Bc6c2OVWRh.

Yichen Tang, Weihang Su, Yujia Zhou, Yiqun Liu, Min Zhang, Shaoping Ma, and Qingyao Ai. Augmenting multi-agent communication with state delta trajectory. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 10219–10240, 2025. URL https://aclanthology.org/2025.emnlp-main.518/.

Hongduan Tian, Xiao Feng, Ziyuan Zhao, Xiangyu Zhu, Rolan Yan, and Bo Han. Multi-agent debate with memory masking. In *Proceedings of the International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EdTt8nMAMA.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Proceedings of the Conference on Neural Information Processing Systems*, 2020. URL https://papers.nips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html.

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 *Proceedings of the International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Di Xue, Lei Yuan, Zongzhang Zhang, and Yang Yu. Efficient multi-agent communication via Shapley message value. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 578–584, 2022. URL https://doi.org/10.24963/ijcai.2022/82.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *Proceedings of the International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/06a52a54c8ee03cd86771136bc91eb1f-Abstract-Conference.html.

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 International Conference on Machine Learning*, pp. 76678–76692, 2025. URL https://proceedings.mlr.press/v267/zhang25cu.html.

Huixiang Zhang and Mahzabeen Emu. Do latent channels actually communicate? a causal audit of latent multi-agent LLM. arXiv preprint, 2026. URL https://arxiv.org/abs/2607.26773.

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, 2026a. URL https://ojs.aaai.org/index.php/AAAI/article/view/40235.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 29892–29900, 2026b. URL https://ojs.aaai.org/index.php/AAAI/article/view/40236.

APPENDIX CONTENTS

A Notations **15**

B Proofs **15**

B.1 Proofs for Social Circuits 15

B.2 Expected Task-Utility Difference under Pairwise Training 17

B.3 CGD's Expected Task-Utility Difference 18

C Algorithm **20**

D Experimental protocol details **22**

D.1 Tasks and settings 22

D.2 Message and activation comparisons 22

D.3 CGD training and comparisons 23

D.4 Scoring and cost 24

E Additional Results **25**

E.1 How additional messages correct or reinforce errors 25

E.2 Agent number and communication structure 26

E.3 Receiver activation replacement 27

E.4 CGD message selection 29

E.5 Free-generation task performance 31

E.6 Accuracy and generation cost 32

E.7 Case study of message selection 34

F Discussions **37**

A NOTATIONS

Table 3: List of key notations.

Symbol	Meaning
$\mathcal{A}$	Message combination, a subset of $\mathcal{C}$.
$\widehat{\mathcal{A}}, \mathcal{A}^\star$	Combinations with the highest CGD score and expected task utility, respectively.
$\mathcal{B}$	Another message combination in $\mathcal{S}$.
$\mathcal{C}$	Candidate messages for one receiver.
d	Activation-summary dimension.
e	Incorrect answer favored by multiple agents before delivery.
err	Largest absolute error in an answer-pair score difference between original-message and replacement runs.
$f_{\boldsymbol{\theta}}, F(o)$	CGD scoring function and receiver's score for answer o.
g	Candidate answer satisfying $u(g) > u(e)$.
gap	Original-message score of the selected answer minus the highest other answer score.
$\ell(\mathcal{A}, \mathcal{B})$	Pairwise training loss.
$\mathcal{L}(S), \mathcal{L}^\star$	Expected summed pairwise loss without the weight penalty, and its lowest value approached over free scores with $S(\varnothing) = 0$.
$o, \mathcal{O}$	Candidate answer and the set of candidate answers.
$q, q_{\mathcal{A}}$	Receiver's output, and its output after reading $\mathcal{A}$.
$q_{\mathcal{A}}^{\mathrm{act}}$	Output after replacing selected no-message activations with those from the run of $\mathcal{A}$.
$\mathbf{r}_{\mathcal{A}}$	Receiver activation summary in $\mathbb{R}^d$ after reading $\mathcal{A}$.
$\mathbf{s}_{\mathcal{A}}$	Mean sender activation summary in $\mathbb{R}^d$ for nonempty $\mathcal{A}$.
$S(\mathcal{A}), \mathcal{S}$	CGD score and the set of all message combinations, including $\varnothing$; $S(\varnothing) = 0$.
$u(q), u_{\mathcal{A}}^{\mathrm{act}}$	Task utility of q and of $q_{\mathcal{A}}^{\mathrm{act}}$; binary task success in the CGD theory.
$V(\mathcal{A})$	Utility gain over sending no message: $u(q_{\mathcal{A}}) - u(q_{\varnothing})$.
$\mathbf{x}_{\mathcal{A}}$	CGD scoring input for nonempty $\mathcal{A}$.
z	Group of matched reference, original-message, and replacement runs.
$\Delta F^{\mathrm{msg}}(o), \Delta F^{\mathrm{act}}(o)$	Message and replacement effects on the score of o.
$\Delta^{\mathrm{msg}}(z), \Delta^{\mathrm{act}}(z)$	Message and replacement effects on an answer-pair score difference.
$\Delta u^{\mathrm{msg}}, \Delta u^{\mathrm{act}}$	Message and replacement effects on task utility.
$\boldsymbol{\theta}$	Learned weights and bias of the scoring function.
ρ	Recovery ratio $\mathbb{E}[\Delta^{\mathrm{act}}]/\mathbb{E}[\Delta^{\mathrm{msg}}]$, with a nonzero denominator.

The Δ terms measure changes from the same reference run. In the CGD theory, expectations, probabilities, and covariances are based on the candidate set and all scoring inputs.

B PROOFS

We first prove Proposition 1 and Theorem 1 for Social Circuits. We then bound the expected task-utility difference under pairwise training and use this bound to prove Theorem 2.

B.1 PROOFS FOR SOCIAL CIRCUITS

Proposition 1 (Complete recovery through receiver activations)**.** *We consider original and reference messages that occupy the same token positions. Each layer takes the preceding layer's activations as*

input, and each token uses only its own and earlier positions. When scoring each $o \in \mathcal{O}$, restoring the original-message activations at one layer's output for all message tokens and all later tokens used to score o leads to $F^{\mathrm{act}}(o) = F^{\mathrm{msg}}(o)$.

Proof of Proposition 1. We score the same answer $o \in \mathcal{O}$ at the same token positions in both runs. We keep the tokens before the message unchanged. Since these tokens cannot use later positions, their activations remain unchanged at every layer.

At the selected layer, we restore the original-message activations at all message positions and all later positions used to score o. With the unchanged earlier positions, these activations form the same layer output as in the original-message run. We therefore obtain the same outputs from each remaining layer and the answer-scoring function. Hence,

$$F^{\mathrm{act}}(o) = F^{\mathrm{msg}}(o) \qquad \text{for every } o \in \mathcal{O}. \tag{10}$$

This completes the proof of Proposition 1. □

By subtracting the same reference scores, we obtain $\Delta F^{\mathrm{act}}(o) = \Delta F^{\mathrm{msg}}(o)$. With the same rule for ties, we also have $q^{\mathrm{act}} = q^{\mathrm{msg}}$ and $\Delta u^{\mathrm{act}} = \Delta u^{\mathrm{msg}}$. We next allow errors in the recovered score differences.

Proof of Theorem 1. We first assume $\mathrm{err} < \mathrm{gap}$. For every $o \in \mathcal{O} \setminus \{q^{\mathrm{msg}}\}$, we have

$$F^{\mathrm{act}}(q^{\mathrm{msg}}) - F^{\mathrm{act}}(o) \overset{\text{(a)}}{\geq} F^{\mathrm{msg}}(q^{\mathrm{msg}}) - F^{\mathrm{msg}}(o) - \mathrm{err} \overset{\text{(b)}}{\geq} \mathrm{gap} - \mathrm{err} > 0, \tag{11}$$

where inequality (a) follows from the definition of err, and inequality (b) follows from the definition of gap. Thus q^{msg} remains the highest-scoring answer, so $q^{\mathrm{act}} = q^{\mathrm{msg}}$. By substituting into the task-performance change, we have

$$\Delta u^{\mathrm{act}} = u(q^{\mathrm{act}}) - u(q^{\mathrm{ref}}) = u(q^{\mathrm{msg}}) - u(q^{\mathrm{ref}}) = \Delta u^{\mathrm{msg}}. \tag{12}$$

We next consider the message and activation effects for the same answer pair o_1, o_2. By cancelling the common reference scores, we have

$$\left|\Delta^{\mathrm{act}} - \Delta^{\mathrm{msg}}\right| = \left|\left[F^{\mathrm{act}}(o_1) - F^{\mathrm{act}}(o_2)\right] - \left[F^{\mathrm{msg}}(o_1) - F^{\mathrm{msg}}(o_2)\right]\right| \leq \mathrm{err}. \tag{13}$$

We obtain the inequality from the definition of err. We now consider the two possible signs of Δ^{msg}:

$$\begin{aligned} \Delta^{\mathrm{msg}} > \mathrm{err} \quad &\Longrightarrow \quad \Delta^{\mathrm{act}} \geq \Delta^{\mathrm{msg}} - \mathrm{err} > 0, \\ \Delta^{\mathrm{msg}} < -\mathrm{err} \quad &\Longrightarrow \quad \Delta^{\mathrm{act}} \leq \Delta^{\mathrm{msg}} + \mathrm{err} < 0. \end{aligned} \tag{14}$$

Thus $|\Delta^{\mathrm{msg}}| > \mathrm{err}$ implies $\Delta^{\mathrm{act}}\Delta^{\mathrm{msg}} > 0$. This completes the proof of Theorem 1. □

By Theorem 1, different answers require $\mathrm{err} \geq \mathrm{gap}$. Taking probabilities over matched reruns, we have

$$\Pr(q^{\mathrm{act}} \neq q^{\mathrm{msg}}) \leq \Pr(\mathrm{err} \geq \mathrm{gap}). \tag{15}$$

Average recovery and individual decisions. We compare the message-effect recovery ratio ρ with answer agreement using two equally likely tasks with $\mathcal{O} = \{o_1, o_2\}$. The entries below are the score of o_1 minus the score of o_2:

	Reference message	Original message	Activation replacement
First task	-2	-1	1
Second task	-2	1	-1

In this example, the score difference equals the reference value -2 plus two terms determined by receiver activations. Both terms are zero in the reference run. In the original-message run, they are $(3, -2)$ for the first task and $(1, 2)$ for the second. After restoring the activations that determine the first term, we obtain the activation-replacement scores shown in the table. Thus the message effects are $1, 3$ and the activation effects are $3, 1$, so

$$\rho = \frac{(3+1)/2}{(1+3)/2} = 1, \qquad \Pr(q^{\mathrm{act}} \neq q^{\mathrm{msg}}) = 1. \tag{16}$$

We therefore recover the average score effect without recovering either answer.

B.2 Expected Task-Utility Difference under Pairwise Training

We use binary task utilities as shown in the main text. We fix the candidate messages, scoring inputs, and learned scores. All expectations, probabilities, and covariances in this subsection and the next are under the resulting distribution $\mathcal{P}$. We omit their subscript $\mathcal{P}$ to keep the formulas compact. We analyze the expected pairwise loss $\mathcal{L}(S)$ defined in the main text. Its infimum $\mathcal{L}^\star$ is taken over all real-valued combination scores with $S(\varnothing) = 0$. Subtracting a common constant from all scores leaves the loss unchanged, so this normalization does not change the infimum. Minimizing the shared scoring function's training objective does not by itself imply $\mathcal{L}(S) = \mathcal{L}^\star$.

Lemma 1 (Expected task-utility difference under dependent task utilities)**.** *For $|\mathcal{S}| \geq 2$, any joint distribution of binary task utilities, and any finite scores, we have*

$$\mathbb{E}\left[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})\right] \leq \sum_{\mathcal{B}\in\mathcal{S}\setminus\{\mathcal{A}^\star,\widehat{\mathcal{A}}\}} \max\left\{0, \operatorname{Cov}\left(u(q_{\widehat{\mathcal{A}}}) - u(q_{\mathcal{A}^\star}), u(q_{\mathcal{B}})\right)\right\} + \sqrt{|\mathcal{S}|\left(\mathcal{L}(S) - \mathcal{L}^\star\right)}. \tag{17}$$

Proof. We consider $\mathcal{A}^\star \neq \widehat{\mathcal{A}}$. Otherwise, the expected task-utility difference is zero. By cancelling the common no-message utility in V, we have

$$V(\mathcal{A}) - V(\mathcal{B}) = u(q_{\mathcal{A}}) - u(q_{\mathcal{B}}). \tag{18}$$

With binary utilities, each pair contributes to the loss only when its two utilities differ. By differentiating these logistic terms, we have

$$2\frac{\partial \mathcal{L}(S)}{\partial S(\mathcal{A})} = \sum_{\mathcal{B}\in\mathcal{S}\setminus\{\mathcal{A}\}} \Pr\left(u(q_{\mathcal{A}}) \neq u(q_{\mathcal{B}})\right) \frac{\exp\left(S(\mathcal{A}) - S(\mathcal{B})\right) - 1}{\exp\left(S(\mathcal{A}) - S(\mathcal{B})\right) + 1} - |\mathcal{S}|\,\mathbb{E}\left[u(q_{\mathcal{A}})\right] + \sum_{\mathcal{B}\in\mathcal{S}} \mathbb{E}\left[u(q_{\mathcal{B}})\right]. \tag{19}$$

Here we differentiate with respect to each combination score, temporarily allowing $S(\varnothing)$ to vary. We can restore $S(\varnothing) = 0$ by subtracting its value from all scores without changing the loss.

We add the same amount to $S(\mathcal{A}^\star)$ as we subtract from $S(\widehat{\mathcal{A}})$, keeping all other scores fixed. The difference $S(\mathcal{A}^\star) - S(\widehat{\mathcal{A}})$ changes by twice the added amount. For each remaining combination $\mathcal{B}$, the differences $S(\mathcal{A}^\star) - S(\mathcal{B})$ and $S(\widehat{\mathcal{A}}) - S(\mathcal{B})$ change by equal amounts in opposite directions. Each logistic loss term has second derivative at most $1/4$ with respect to its score difference, and each pair's probability of different task utilities is at most one. Hence, the second derivative of $\mathcal{L}(S)$ with respect to the amount added to $S(\mathcal{A}^\star)$ is at most

$$\frac{2^2}{4} + \frac{2(|\mathcal{S}| - 2)}{4} = \frac{|\mathcal{S}|}{2}. \tag{20}$$

We choose the amount added to $S(\mathcal{A}^\star)$ and subtracted from $S(\widehat{\mathcal{A}})$ as

$$\frac{2}{|\mathcal{S}|}\left(\frac{\partial \mathcal{L}(S)}{\partial S(\widehat{\mathcal{A}})} - \frac{\partial \mathcal{L}(S)}{\partial S(\mathcal{A}^\star)}\right). \tag{21}$$

By applying the second-order Taylor bound and Eq. (20) to these two score adjustments, we reduce the loss by at least the squared difference between the derivatives in Eq. (21), divided by $|\mathcal{S}|$. Since the loss cannot decrease below $\mathcal{L}^\star$, we obtain

$$\frac{1}{|\mathcal{S}|}\left(\frac{\partial \mathcal{L}(S)}{\partial S(\widehat{\mathcal{A}})} - \frac{\partial \mathcal{L}(S)}{\partial S(\mathcal{A}^\star)}\right)^2 \leq \mathcal{L}(S) - \mathcal{L}^\star, \tag{22}$$

$$\left|\frac{\partial \mathcal{L}(S)}{\partial S(\widehat{\mathcal{A}})} - \frac{\partial \mathcal{L}(S)}{\partial S(\mathcal{A}^\star)}\right| \leq \sqrt{|\mathcal{S}|\left(\mathcal{L}(S) - \mathcal{L}^\star\right)}. \tag{23}$$

We now use the selection rule $S(\widehat{\mathcal{A}}) \geq S(\mathcal{B})$ for all $\mathcal{B} \in \mathcal{S}$. The fraction in Eq. (19) increases with the score difference and lies in $[-1, 1]$. For each $\mathcal{B} \notin \{\mathcal{A}^\star, \widehat{\mathcal{A}}\}$, the fraction for $\widehat{\mathcal{A}}$ is nonnegative and no smaller than the fraction for $\mathcal{A}^\star$. We therefore have

$$\begin{aligned}
&\Pr\left(u(q_{\mathcal{A}^\star}) \neq u(q_{\mathcal{B}})\right) \frac{\exp\left(S(\mathcal{A}^\star) - S(\mathcal{B})\right) - 1}{\exp\left(S(\mathcal{A}^\star) - S(\mathcal{B})\right) + 1} \\
&- \Pr\left(u(q_{\widehat{\mathcal{A}}}) \neq u(q_{\mathcal{B}})\right) \frac{\exp\left(S(\widehat{\mathcal{A}}) - S(\mathcal{B})\right) - 1}{\exp\left(S(\widehat{\mathcal{A}}) - S(\mathcal{B})\right) + 1} \\
&\leq \max\left\{0, \Pr\left(u(q_{\mathcal{A}^\star}) \neq u(q_{\mathcal{B}})\right) - \Pr\left(u(q_{\widehat{\mathcal{A}}}) \neq u(q_{\mathcal{B}})\right)\right\}. \qquad (24)
\end{aligned}$$

For the direct comparison between $\mathcal{A}^\star$ and $\widehat{\mathcal{A}}$, the corresponding difference is nonpositive because $S(\mathcal{A}^\star) \leq S(\widehat{\mathcal{A}})$. By subtracting their derivatives in Eq. (19) and applying Eqs. (23) and (24), we obtain

$$\begin{aligned}
|\mathcal{S}|\,\mathbb{E}\left[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})\right] \leq &\sum_{\mathcal{B} \in \mathcal{S} \setminus \{\mathcal{A}^\star, \widehat{\mathcal{A}}\}} \max\left\{0, \Pr\left(u(q_{\mathcal{A}^\star}) \neq u(q_{\mathcal{B}})\right) - \Pr\left(u(q_{\widehat{\mathcal{A}}}) \neq u(q_{\mathcal{B}})\right)\right\} \\
&+ 2\sqrt{|\mathcal{S}|\left(\mathcal{L}(S) - \mathcal{L}^\star\right)}. \qquad (25)
\end{aligned}$$

Finally, we express the probabilities of different task utilities using covariances. For binary utilities, we have

$$\begin{aligned}
\Pr\left(u(q_{\mathcal{A}}) \neq u(q_{\mathcal{B}})\right) &= \mathbb{E}\left[u(q_{\mathcal{A}})\right] + \mathbb{E}\left[u(q_{\mathcal{B}})\right] - 2\mathbb{E}\left[u(q_{\mathcal{A}})u(q_{\mathcal{B}})\right] \\
&= \mathbb{E}\left[u(q_{\mathcal{A}})\right] + \mathbb{E}\left[u(q_{\mathcal{B}})\right] - 2\mathbb{E}\left[u(q_{\mathcal{A}})\right]\mathbb{E}\left[u(q_{\mathcal{B}})\right] - 2\,\mathrm{Cov}\left(u(q_{\mathcal{A}}), u(q_{\mathcal{B}})\right). \qquad (26)
\end{aligned}$$

By applying Eq. (26) to $\mathcal{A}^\star$ and $\widehat{\mathcal{A}}$ and subtracting, we have

$$\begin{aligned}
&\Pr\left(u(q_{\mathcal{A}^\star}) \neq u(q_{\mathcal{B}})\right) - \Pr\left(u(q_{\widehat{\mathcal{A}}}) \neq u(q_{\mathcal{B}})\right) \\
&= \mathbb{E}\left[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})\right](1 - 2\mathbb{E}\left[u(q_{\mathcal{B}})\right]) + 2\,\mathrm{Cov}\left(u(q_{\widehat{\mathcal{A}}}) - u(q_{\mathcal{A}^\star}), u(q_{\mathcal{B}})\right) \\
&\leq \mathbb{E}\left[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})\right] + 2\max\left\{0, \mathrm{Cov}\left(u(q_{\widehat{\mathcal{A}}}) - u(q_{\mathcal{A}^\star}), u(q_{\mathcal{B}})\right)\right\}. \qquad (27)
\end{aligned}$$

We obtain the inequality because $\mathcal{A}^\star$ maximizes expected utility and $1 - 2\mathbb{E}\left[u(q_{\mathcal{B}})\right] \leq 1$. Its right-hand side is nonnegative, so it also bounds the maximum with zero in Eq. (25). By substitution and moving the $|\mathcal{S}| - 2$ copies of the expected utility difference to the left, we have

$$\begin{aligned}
2\mathbb{E}\left[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})\right] \leq 2 &\sum_{\mathcal{B} \in \mathcal{S} \setminus \{\mathcal{A}^\star, \widehat{\mathcal{A}}\}} \max\left\{0, \mathrm{Cov}\left(u(q_{\widehat{\mathcal{A}}}) - u(q_{\mathcal{A}^\star}), u(q_{\mathcal{B}})\right)\right\} \\
&+ 2\sqrt{|\mathcal{S}|\left(\mathcal{L}(S) - \mathcal{L}^\star\right)}. \qquad (28)
\end{aligned}$$

We divide both sides by two to complete the proof of Lemma 1. □

The comparison used in the bound. With the scoring inputs fixed, $\mathcal{A}^\star$ maximizes expected utility. Selecting the best combination after observing all task utilities defines a different comparison. By adding and subtracting $V(\mathcal{A}^\star)$, we have

$$\mathbb{E}\left[\max_{\mathcal{A} \in \mathcal{S}} V(\mathcal{A}) - V(\widehat{\mathcal{A}})\right] = \mathbb{E}\left[\max_{\mathcal{A} \in \mathcal{S}} V(\mathcal{A}) - V(\mathcal{A}^\star)\right] + \mathbb{E}\left[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})\right].$$

The lemma bounds the second term. The first term also contributes when we compare with the best combination after observing all utilities.

B.3 CGD's Expected Task-Utility Difference

Proof of Theorem 2. We bound the change in the covariances in Lemma 1 when we replace receiver activations. We keep the scoring inputs, scores, and expected pairwise loss $\mathcal{L}(S)$ unchanged. Since equal answers have equal binary utilities, we have

$$\mathbb{E}\left|u(q_{\mathcal{A}}) - u(q_{\mathcal{A}}^{\mathrm{act}})\right| \leq \Pr(q_{\mathcal{A}}^{\mathrm{act}} \neq q_{\mathcal{A}}) \leq \max_{\mathcal{B} \in \mathcal{S}} \Pr(q_{\mathcal{B}}^{\mathrm{act}} \neq q_{\mathcal{B}}). \qquad (29)$$

By adding and subtracting $\mathrm{Cov}(u^{\mathrm{act}}_{\mathcal{A}}, u(q_{\mathcal{B}}))$, we have

$$\begin{aligned}
&\mathrm{Cov}\left(u(q_{\mathcal{A}}), u(q_{\mathcal{B}})\right) - \mathrm{Cov}\left(u^{\mathrm{act}}_{\mathcal{A}}, u^{\mathrm{act}}_{\mathcal{B}}\right) \\
&= \mathrm{Cov}\left(u(q_{\mathcal{A}}) - u^{\mathrm{act}}_{\mathcal{A}}, u(q_{\mathcal{B}})\right) + \mathrm{Cov}\left(u^{\mathrm{act}}_{\mathcal{A}}, u(q_{\mathcal{B}}) - u^{\mathrm{act}}_{\mathcal{B}}\right) \\
&= \mathbb{E}\left[\left(u(q_{\mathcal{A}}) - u^{\mathrm{act}}_{\mathcal{A}}\right)\left(u(q_{\mathcal{B}}) - \mathbb{E}\left[u(q_{\mathcal{B}})\right]\right)\right] + \mathbb{E}\left[\left(u^{\mathrm{act}}_{\mathcal{A}} - \mathbb{E}\left[u^{\mathrm{act}}_{\mathcal{A}}\right]\right)\left(u(q_{\mathcal{B}}) - u^{\mathrm{act}}_{\mathcal{B}}\right)\right]. \qquad (30)
\end{aligned}$$

The absolute difference between each binary utility and its expectation is at most one. By applying the triangle inequality and Eq. (29), we obtain

$$\begin{aligned}
\left|\mathrm{Cov}\left(u(q_{\mathcal{A}}), u(q_{\mathcal{B}})\right) - \mathrm{Cov}\left(u^{\mathrm{act}}_{\mathcal{A}}, u^{\mathrm{act}}_{\mathcal{B}}\right)\right| &\le \mathbb{E}\left|u(q_{\mathcal{A}}) - u^{\mathrm{act}}_{\mathcal{A}}\right| + \mathbb{E}\left|u(q_{\mathcal{B}}) - u^{\mathrm{act}}_{\mathcal{B}}\right| \\
&\le 2\max_{\mathcal{A}\in\mathcal{S}} \Pr(q^{\mathrm{act}}_{\mathcal{A}} \neq q_{\mathcal{A}}). \qquad (31)
\end{aligned}$$

We apply Eq. (31) separately with $\mathcal{A} = \widehat{\mathcal{A}}$ and $\mathcal{A} = \mathcal{A}^\star$, while keeping $\mathcal{B}$ fixed. By adding these two bounds and applying the triangle inequality, we obtain

$$\begin{aligned}
&\max\left\{0, \mathrm{Cov}\left(u(q_{\widehat{\mathcal{A}}}) - u(q_{\mathcal{A}^\star}), u(q_{\mathcal{B}})\right)\right\} \\
&\le \max\left\{0, \mathrm{Cov}\left(u^{\mathrm{act}}_{\widehat{\mathcal{A}}} - u^{\mathrm{act}}_{\mathcal{A}^\star}, u^{\mathrm{act}}_{\mathcal{B}}\right)\right\} + 4\max_{\mathcal{A}\in\mathcal{S}} \Pr(q^{\mathrm{act}}_{\mathcal{A}} \neq q_{\mathcal{A}}). \qquad (32)
\end{aligned}$$

For $\mathcal{A}^\star \neq \widehat{\mathcal{A}}$, we sum over the $|\mathcal{S}| - 2$ remaining combinations and substitute into Lemma 1:

$$\begin{aligned}
\mathbb{E}\left[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})\right] \le &\sum_{\mathcal{B}\in\mathcal{S}\setminus\{\mathcal{A}^\star,\widehat{\mathcal{A}}\}} \max\left\{0, \mathrm{Cov}\left(u^{\mathrm{act}}_{\widehat{\mathcal{A}}} - u^{\mathrm{act}}_{\mathcal{A}^\star}, u^{\mathrm{act}}_{\mathcal{B}}\right)\right\} \\
&+ \sqrt{|\mathcal{S}|\left(\mathcal{L}(S) - \mathcal{L}^\star\right)} + 4(|\mathcal{S}| - 2)\max_{\mathcal{A}\in\mathcal{S}} \Pr(q^{\mathrm{act}}_{\mathcal{A}} \neq q_{\mathcal{A}}). \qquad (33)
\end{aligned}$$

For $\mathcal{A}^\star = \widehat{\mathcal{A}}$, the left-hand side is zero and the inequality holds directly. This completes the proof of Theorem 2. □

When the receiver selects the highest-scoring candidate answer, we apply Theorem 1 to each message combination. The probability that activation replacement and the run using $\mathcal{A}$ return different answers is at most $\Pr(\mathrm{err} \ge \mathrm{gap})$, using the no-message run as the reference. If replacement preserves every answer, the last term in Eq. (33) is zero. Proposition 1 ensures this property when its conditions on token positions and restored activations hold.

Corollary 1 (Optimal message selection under equal covariance)**.** *If the task utilities of different message combinations have equal covariance and finite scores satisfy $\mathcal{L}(S) = \mathcal{L}^\star$, then $\widehat{\mathcal{A}} \in \arg\max_{\mathcal{A}\in\mathcal{S}} \mathbb{E}_{\mathcal{P}}[u(q_{\mathcal{A}})]$.*

Proof of Corollary 1. For each $\mathcal{B} \notin \{\mathcal{A}^\star, \widehat{\mathcal{A}}\}$, equal covariance implies

$$\begin{aligned}
&\mathrm{Cov}\left(u(q_{\widehat{\mathcal{A}}}) - u(q_{\mathcal{A}^\star}), u(q_{\mathcal{B}})\right) \\
&= \mathrm{Cov}\left(u(q_{\widehat{\mathcal{A}}}), u(q_{\mathcal{B}})\right) - \mathrm{Cov}\left(u(q_{\mathcal{A}^\star}), u(q_{\mathcal{B}})\right) = 0. \qquad (34)
\end{aligned}$$

By substituting this equality and $\mathcal{L}(S) = \mathcal{L}^\star$ into Lemma 1, we have

$$0 \le \mathbb{E}\left[V(\mathcal{A}^\star) - V(\widehat{\mathcal{A}})\right] \le \sqrt{|\mathcal{S}|\left(\mathcal{L}(S) - \mathcal{L}^\star\right)} = 0. \qquad (35)$$

Thus CGD selects a combination with the highest expected utility. □

Unequal covariances. We construct an example with unequal covariances in which minimizing the pairwise loss does not maximize the expected utility. We take two candidate messages, with $\mathcal{A}$ and $\mathcal{B}$ each containing one message. Thus $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ and $\mathcal{S} = \{\varnothing, \mathcal{A}, \mathcal{B}, \mathcal{C}\}$. We consider that the binary utilities satisfy

$$\begin{aligned}
u(q_\varnothing) = u(q_{\mathcal{A}}) = 1 - u(q_{\mathcal{B}}) \le u(q_{\mathcal{C}}), \\
0 < \mathbb{E}\left[u(q_\varnothing)\right] < \frac{2}{9}, \qquad \mathbb{E}\left[u(q_{\mathcal{C}})\right] = 1 - 2\mathbb{E}\left[u(q_\varnothing)\right].
\end{aligned} \qquad (36)$$

We assign probabilities $\mathbb{E}[u(q_\varnothing)]$, $2\mathbb{E}[u(q_\varnothing)]$, and $1-3\mathbb{E}[u(q_\varnothing)]$ to the utility pairs $(u(q_\varnothing), u(q_\mathcal{C})) = (1,1)$, $(0,0)$, and $(0,1)$, respectively. These probabilities are positive and sum to one. By Eq. (36), we have

$$\mathbb{E}[u(q_\mathcal{B})] = 1 - \mathbb{E}[u(q_\varnothing)] > \mathbb{E}[u(q_\mathcal{C})] > \mathbb{E}[u(q_\mathcal{A})] = \mathbb{E}[u(q_\varnothing)]. \tag{37}$$

Thus $\mathcal{B}$ has the highest expected utility throughout this family.

With $S(\varnothing) = 0$, exactly one set of finite scores minimizes the loss. For each pair involving $\mathcal{B}$, either utility can exceed the other with positive probability. Each pair loss is strictly convex in its score difference and tends to infinity when this difference tends to either infinity. With $S(\varnothing) = 0$, any nonzero score change affects at least one of these three differences, and any unbounded score sequence makes at least one difference unbounded. Hence the continuous total loss is minimized by a unique set of finite scores.

Since $u(q_\mathcal{A}) = u(q_\varnothing)$, exchanging their scores leaves the loss unchanged. We can then subtract the new no-message score from every score to restore $S(\varnothing) = 0$. By uniqueness of the minimum, we obtain

$$S(\mathcal{A}) = S(\varnothing) = 0. \tag{38}$$

At this minimum, we set the derivatives with respect to $S(\mathcal{B})$ and $S(\mathcal{C})$ to zero. By substituting the joint probabilities and rearranging, we have

$$\frac{2}{1+\exp S(\mathcal{B})} = \frac{3\mathbb{E}[u(q_\varnothing)]}{1+\exp\left(S(\mathcal{C}) - S(\mathcal{B})\right)}, \tag{39}$$

$$\frac{2\left(1-3\mathbb{E}[u(q_\varnothing)]\right)}{1+\exp S(\mathcal{C})} = \frac{3\mathbb{E}[u(q_\varnothing)]}{1+\exp\left(S(\mathcal{B}) - S(\mathcal{C})\right)} - \mathbb{E}[u(q_\varnothing)]. \tag{40}$$

The right-hand side of Eq. (39) is less than $3\mathbb{E}[u(q_\varnothing)] < 1$, so $S(\mathcal{B}) > 0$.

We assume $S(\mathcal{C}) \leq S(\mathcal{B})$ and derive a contradiction. By substituting $\exp\left(S(\mathcal{C}) - S(\mathcal{B})\right) \leq 1$ into Eq. (39), we have

$$\frac{1}{1+\exp S(\mathcal{B})} \geq \frac{3\mathbb{E}[u(q_\varnothing)]}{4}. \tag{41}$$

We combine this inequality with Eq. (40):

$$\begin{aligned}
\frac{3\mathbb{E}[u(q_\varnothing)]}{2}\left(1-3\mathbb{E}[u(q_\varnothing)]\right) &\overset{(a)}{\leq} \frac{2\left(1-3\mathbb{E}[u(q_\varnothing)]\right)}{1+\exp S(\mathcal{C})} \\
&= \frac{3\mathbb{E}[u(q_\varnothing)]}{1+\exp\left(S(\mathcal{B}) - S(\mathcal{C})\right)} - \mathbb{E}[u(q_\varnothing)] \\
&\overset{(b)}{\leq} \frac{\mathbb{E}[u(q_\varnothing)]}{2}.
\end{aligned} \tag{42}$$

Inequality (a) follows from Eq. (41) and $1+\exp S(\mathcal{C}) \leq 1+\exp S(\mathcal{B})$. Inequality (b) follows from $\exp\left(S(\mathcal{B}) - S(\mathcal{C})\right) \geq 1$. By dividing by $\mathbb{E}[u(q_\varnothing)]/2 > 0$ and rearranging, we obtain $\mathbb{E}[u(q_\varnothing)] \geq 2/9$, which contradicts Eq. (36). Hence,

$$S(\mathcal{C}) > S(\mathcal{B}) > S(\mathcal{A}) = S(\varnothing) = 0. \tag{43}$$

CGD therefore selects $\mathcal{C}$ at the loss minimum, although $\mathcal{B}$ has higher expected utility. The difference is

$$\mathbb{E}[V(\mathcal{B}) - V(\mathcal{C})] = \mathbb{E}[u(q_\varnothing)] > 0. \tag{44}$$

Finally, the covariances in this family satisfy

$$\mathrm{Cov}\left(u(q_\varnothing), u(q_\mathcal{A})\right) = -\mathrm{Cov}\left(u(q_\varnothing), u(q_\mathcal{B})\right) = \mathbb{E}[u(q_\varnothing)]\left(1-\mathbb{E}[u(q_\varnothing)]\right) > 0. \tag{45}$$

Thus, unequal covariance can lead to a positive expected task-utility difference even when finite scores minimize the pairwise loss.

C Algorithm

Algorithm 1 compares a message's effect with the effect reproduced by receiver activation replacement. Algorithm 2 uses the sender information and receiver changes to learn which message combination improves task performance.

Algorithm 1 SOCIAL CIRCUITS: message and activation effects

Input: Task and receiver; original and reference messages; selected layers and post-message prompt positions; candidate answers $\mathcal{O}$; task utility u.

Matched reruns

Fix the task, earlier messages, receiver model, and generation seed.
Run the receiver with the reference message; record $F^{\mathrm{ref}}(o)$ for each $o \in \mathcal{O}$.
Run the receiver with the original message; record $F^{\mathrm{msg}}(o)$ and the activations at the selected layers and positions.

Activation replacement

Repeat the reference run, replacing the selected activations with their original-message values; record $F^{\mathrm{act}}(o)$.

Effect calculation

Select q^{ref}, q^{msg}, and q^{act} by maximizing their respective answer scores, using the same tie rule.
Calculate the message and activation effects on each answer score:

$$\Delta F^{\mathrm{msg}}(o) = F^{\mathrm{msg}}(o) - F^{\mathrm{ref}}(o),$$

$$\Delta F^{\mathrm{act}}(o) = F^{\mathrm{act}}(o) - F^{\mathrm{ref}}(o).$$

Calculate the corresponding changes in task utility:

$$\Delta u^{\mathrm{msg}} = u(q^{\mathrm{msg}}) - u(q^{\mathrm{ref}}),$$

$$\Delta u^{\mathrm{act}} = u(q^{\mathrm{act}}) - u(q^{\mathrm{ref}}).$$

Output: ΔF^{msg}, ΔF^{act}, Δu^{msg}, and Δu^{act}.

Algorithm 2 Circuit-Guided Deliberation (CGD)

Input: Training tasks with task utility u; a new task; sending and receiving agents.

Training

for all training tasks **do**
    Record sender activation summaries; generate candidate messages $\mathcal{C}$ once.
    Form $\mathcal{S} = \{\mathcal{A} \mid \mathcal{A} \subseteq \mathcal{C}\}$, including $\varnothing$.
    for all $\mathcal{A} \in \mathcal{S}$ **do**
        Run the receiver with $\mathcal{A}$, keeping the task and earlier messages fixed; record $\mathbf{r}_{\mathcal{A}}$ before answering and save $q_{\mathcal{A}}$.
    end for
    Construct $\mathbf{s}_{\mathcal{A}}$ and the scoring input $\mathbf{x}_{\mathcal{A}}$ for nonempty combinations.
    Set $V(\mathcal{A}) = u(q_{\mathcal{A}}) - u(q_{\varnothing})$ for every $\mathcal{A} \in \mathcal{S}$.
end for
Keep the language models fixed; fit the shared $f_{\boldsymbol{\theta}}$ using the gain-weighted pairwise loss in Eq. (9), with $S(\varnothing) = 0$ and a squared-weight penalty that leaves the bias unpenalized.

Selection for a new task

Record sender activation summaries; generate candidate messages $\mathcal{C}$ once.
Form $\mathcal{S} = \{\mathcal{A} \mid \mathcal{A} \subseteq \mathcal{C}\}$, including $\varnothing$.
for all $\mathcal{A} \in \mathcal{S}$ **do**
    Keep the task and earlier messages fixed. Read $\mathcal{A}$ with the receiver and record $\mathbf{r}_{\mathcal{A}}$ before generating any answer.
end for
Construct $\mathbf{s}_{\mathcal{A}}$ and the scoring input $\mathbf{x}_{\mathcal{A}}$ for nonempty combinations.
Set $S(\varnothing) = 0$ and $S(\mathcal{A}) = f_{\boldsymbol{\theta}}(\mathbf{x}_{\mathcal{A}})$ for nonempty $\mathcal{A} \in \mathcal{S}$.
Select $\widehat{\mathcal{A}} \in \arg\max_{\mathcal{A} \in \mathcal{S}} S(\mathcal{A})$.
If scores tie, select $\varnothing$ when available; otherwise use the fixed combination order.
Obtain the receiver's answer $q_{\widehat{\mathcal{A}}}$ using $\widehat{\mathcal{A}}$, as specified in Appendix D.

Output: Selected combination $\widehat{\mathcal{A}}$ and receiver answer $q_{\widehat{\mathcal{A}}}$.

D EXPERIMENTAL PROTOCOL DETAILS

D.1 TASKS AND SETTINGS

We use the tasks and models in Section 5.1 and the synthetic Correlated-Trap task, where several agents share a passage supporting an incorrect answer and another receives evidence for the correct answer. Correlated-Trap comparisons also include Gemma 4 E4B-IT (Gemma Team, 2026). On this task, **Fixed sender** selects the agent given different evidence. Table 4 distinguishes the evaluation sets. For matched interventions, we keep the task input, model, earlier messages, and generation seed unchanged. We use seeds 0/1/2 for CGD, 10/11/12 for joint replacements, and 20/21/22 for mean changes.

Agent inputs. All agents receive the same task description. On 2Wiki, each agent reads its assigned passages. For two-answer scoring, we assign supporting passages to the three agents in turn. Each remaining passage goes to the agent with the fewest passages. Ties follow the agent order. For free generation, we assign all passages in their original order, cycling through the agents. GSM8K uses the same problem without extra passages. MATH-500 and TruthfulQA use the same question and answer options. Agent instructions specify independent reasoning, verification, or final answering.

In the 2Wiki two-answer experiments, CGD messages include the sender's passages and response. The collaboration baselines exchange generated responses. In free generation, both include the sender's passages in their initial messages.

Answer options. For two-answer scoring, each task has one correct answer and one incorrect answer. For 2Wiki, we use tasks whose correct answer is a passage title. We select the incorrect answer alphabetically from the other titles, excluding those of passages used as supporting evidence. For MATH-500, we use tasks with integer answers. We construct the incorrect answer by adding 1 to a nonnegative correct answer or subtracting 1 from a negative correct answer. For TruthfulQA, we use the dataset's correct answer and its first listed incorrect answer. We alternate the correct answer between answer A and B across tasks.

MOC implementation. We adapt MOC (Guan et al., 2026) to six model calls with one summary and a fixed communication order. After three initial responses, we summarize the two senders' messages. One sender updates its answer using this summary, and the receiver reads the summary and updated answer before generating its final answer. This fixed sequence replaces the original five-summary merger, graph-based execution, and separate final decision node.

D.2 MESSAGE AND ACTIVATION COMPARISONS

For message-content comparisons on Gemma-3-4B, we use **No message**, **Original message**, **Facts and reasoning only**, **Suggested answer only**, **Unrelated message**, and **Opposite-answer message**. For repetition, we deliver one to seven copies, paraphrases, or independently stated pieces of evidence. We evaluate correction and echo chambers separately.

For joint replacement, we shuffle message positions in a single cycle, preserving tokens and length. In the shuffled-message run, we restore the original-message activations at fixed prompt tokens after the message, including the chat footer but excluding the message and answer prefix. We apply these replacements at six layers together, at 15%, 30%, 45%, 60%, 75%, and 90% of decoder depth. For Swapped-role change, we use the opposite communication direction on the same question and seed, match the change's norm at each layer, and add it to the shuffled-message activations of the receiver being studied.

For layerwise comparisons, we edit activations at the last prompt token at one layer per run. We add the Original-minus-Empty change to Empty and subtract it from Original. We pair questions with the same template, correct label, and independent-agent position. The control uses the paired question's change from the same role, seed, and layer, matched in norm. For mean changes, we estimate directions and magnitudes on 120 separate questions and apply them across six layers on 240 new questions. Mean swapped-role change uses the reversed communication direction on these separate questions.

Table 4: Evaluation sets. Counts are per experiment.

Comparison	Task	Questions
Collaboration approaches	Correlated-Trap	600
Message content and repetition	Correlated-Trap	240 per direction
Activation replacement	Correlated-Trap	240 per direction
Message-specific replacement	2Wiki	240, two candidates each
Two-option message selection	2Wiki	240 per test set
Two-answer scoring	MATH-500	200
Two-answer scoring	TruthfulQA	200
Free-generation performance	2Wiki, GSM8K	240 per dataset

Table 5: Information used by each approach.

Approach	Information
Single, SC6	The designated receiver's assigned input, without messages from other agents.
Majority vote	Independent answers from three agents, each using its own assigned input.
MAD, MADC	Each agent's assigned input and the responses exchanged according to the communication procedure.
MAD-M2	Each agent's assigned input and the responses retained by memory masking.
MOC	The receiver's assigned input, a summary of the two senders' messages, and an updated sender response.
SafeSieve	Each agent's assigned input and messages received through the learned communication graph.
CGD	The receiver's assigned input and the selected sender messages. Message-selection baselines use the same messages and recorded receiver answers.

For 2Wiki replacement, both senders and the receiver read the same summary of the initial answers. We add each candidate separately and replace activations at the last token of the common answer prefix across six layers, using no candidate message as the reference. Other-message change uses the other candidate's change, matched in norm at each layer. We calculate both MAEs where the candidate messages are nonempty and distinct: 422/428/418 samples for Gemma and 444/446/444 for Qwen across seeds. Effects use the correct answer's log probability normalized over the two options.

D.3 CGD TRAINING AND COMPARISONS

We use 340 2Wiki training questions and 100 validation questions, with seed 0. Two-option and free-generation experiments collect separate outputs and train separate scoring functions. We exclude evaluation questions from both sets. Two-option evaluation uses the original 240 questions for all three models and another 240 for the two 4B models, without retraining. Free generation uses separate sets of 240 questions each from 2Wiki and GSM8K, without GSM8K training.

We set $d = 64$ and use zero-based layers 5, 10, 15, 20, 25, and 30 for Gemma, and 5, 10, 16, 21, 26, and 32 for Qwen, with projection seed 2026090301. We read the last common answer-prefix token for two-option tasks in CGD and the last prompt token for free generation. For Text-only, we convert the message tokens and their positions into a 64-dimensional vector. We normalize this vector to unit length and use the message token count as an additional input. CGD and the learned message-selection baselines use the same training tasks and task utilities within each experiment. We fit their scoring functions using the pairwise loss in Section 4.2 and L-BFGS-B. We standardize inputs using the training data. The two-option experiment uses weight penalty 1. Free generation uses exact-match success on development answers and selects the penalty from $\{0.01, 0.1, 1, 10, 100\}$ by validation accuracy; all three CGD models select 100.

We keep CGD unchanged and adjust Activation products to use the same scaling and penalty. Both approaches use the same messages and training data. We report the comparison in Table 8.

Learning curves keep penalty 1 and use the same nested subsets of 50, 100, 200, and 340 questions across selectors, with three draws below 340.

D.4 SCORING AND COST

Answer scores. For complete-answer scoring, we calculate

$$F(o) = \sum_{\ell=1}^{L} \log p(y_\ell \mid P, y_{<\ell}), \tag{46}$$

where P is the prompt, $o = (y_1, \ldots, y_L)$ is the answer, L is its token count, $y_{<\ell}$ denotes the preceding answer tokens, and p is the receiver's probability for each token. For 2Wiki activation comparisons, we calculate the ratio of the correct answer's probability to the sum of both answer probabilities and use its log as the answer score. Single-token scoring uses the first tokens at which the candidate answers are different. For ρ, the message and activation effects use the same reference input, as specified in Appendix D.2.

Task performance. All approaches use the same tasks within each comparison. Accuracy changes use No message as the reference. Two-answer scoring selects between supplied answers. Free generation uses no supplied candidate answers. For free generation, we use exact match with any provided correct answer for 2Wiki and compare the final answer number with the correct number for GSM8K. The code provides the generation settings. We use limits of 256 generated tokens per model call for 2Wiki two-answer scoring and 512 for 2Wiki free generation, GSM8K, MATH-500, and TruthfulQA.

Two-answer comparisons. On 2Wiki, CGD selects the higher-scoring answer for its selected message combination. SC6 and Majority vote use majority voting. If voting does not select a single answer, we select A. The other collaboration baselines use the receiver's final A/B answer. If the receiver does not provide A or B, we select the higher-scoring answer. In Table 10, we also use the higher-scoring A/B answer for all 6 baselines that do not use voting. We keep each task, message, and prompt unchanged. If the scores are equal, we select A.

Generation cost. For free generation and the MATH-500 and TruthfulQA A/B comparisons, the receiver generates one final answer using the selected message combination. Generation cost counts generated tokens in the sender messages and the final answer when generated. We generate answers for the other combinations after message selection and record these outputs separately for message-selection comparisons. CGD also scores answers and records activations in 2 sender and 4 receiver runs. Generated token counts do not include the calculation for these scores and activations.

Token limits. For the 2Wiki comparison and the separate 555-run GSM8K comparison, each baseline uses CGD's generated token count as its total limit for the same task and seed. Single, SC6, and Majority vote generate answers in order. After each answer, we subtract its token count from the remaining limit.

For the other baselines, before each message-generation run, we subtract the tokens already generated and CGD's final-answer token count from the total limit. We use the ratio of this remaining count to the number of remaining message-generation runs as the limit for that run. The final answer can use all tokens remaining under the total limit. We use a limit of 512 tokens per model call and include all generated answers in the accuracy calculation, complete or not.

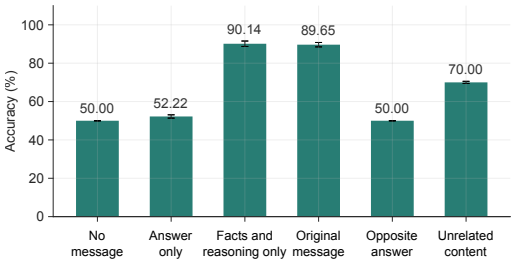


Figure 7: Accuracy across six message-content conditions on Gemma-3-4B-it. Bars and error bars show the mean and standard deviation.

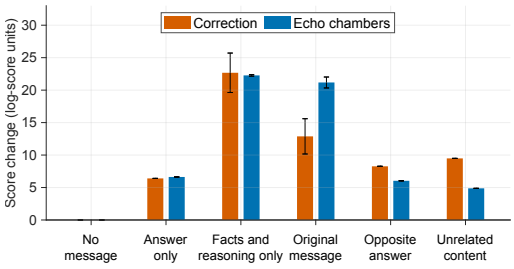
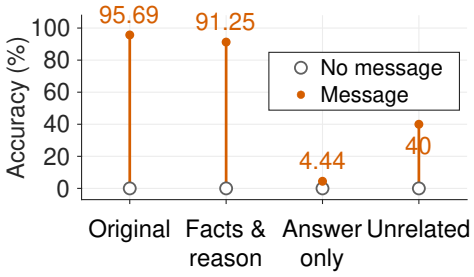
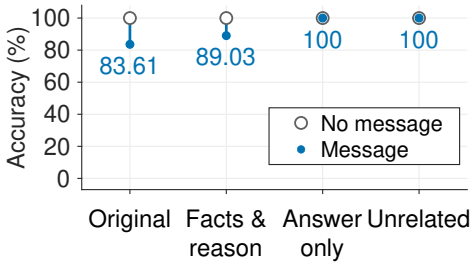


Figure 8: Message effects on Gemma-3-4B-it. Error bars show the standard deviation across seeds.



(a)



(b)

Figure 9: Receiver accuracy when the answer without a message is (a) incorrect or (b) correct.

E Additional Results

We provide additional results on message effects and receiver activation replacement. We compare CGD with message-selection baselines using the same candidate messages and report learning curves. We then compare collaboration approaches in task performance and generated-token use. Examples illustrate how CGD combines useful messages and excludes misleading information.

E.1 How additional messages correct or reinforce errors

Message content. To identify which parts of a message affect the receiver, we present six message-content conditions in Fig. 7. Facts and reasoning alone achieve an accuracy of $90.14 \pm 1.42\%$, close to the original message's $89.65 \pm 1.15\%$ and above $52.22 \pm 0.87\%$ for the suggested answer alone. We further separate runs by whether the receiver answers correctly before receiving a message in Fig. 9. When its initial answer is incorrect, facts and reasoning preserve most of the original message's corrective effect, whereas suggested answers alone have little effect and unrelated messages correct fewer errors. When its initial answer is correct, the original message and its facts and reasoning introduce errors, whereas suggested answers and unrelated messages leave accuracy unchanged. Fig. 8 shows larger changes in answer scores with facts and reasoning than with the suggested answer alone, both when correcting mistakes and when reinforcing a shared error. These comparisons show that both correction and the reinforcement of shared errors depend mainly on the supplied facts and reasoning, rather than on the suggested answer alone.

Additional messages. We study how message effects change as one receiver receives more messages. Fig. 10 compares repeating the same message, rephrasing its content, and providing different evidence from different senders. At seven messages, all three conditions correct more than 99% of initially incorrect answers. Their effects on initially correct answers differ: repeated messages introduce errors in 6.25%, compared with 73.19% for rephrased messages and 95.14% for different evidence. Thus, similar correction rates may have different rates of switching to a shared incorrect answer. These results show how additional communication can reinforce a shared error even when the same communication strategy is effective at correcting mistakes.

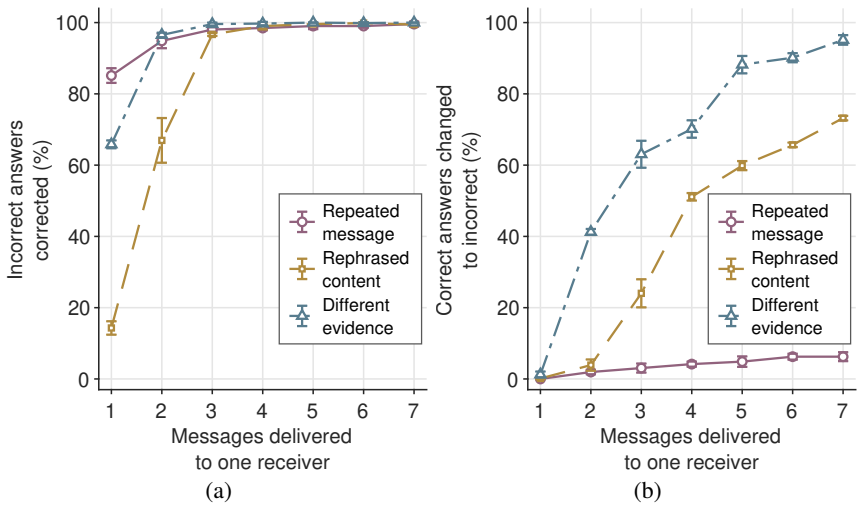


Figure 10: Effects of additional messages: (a) correcting an incorrect answer; (b) switching from a correct answer to a shared incorrect answer.

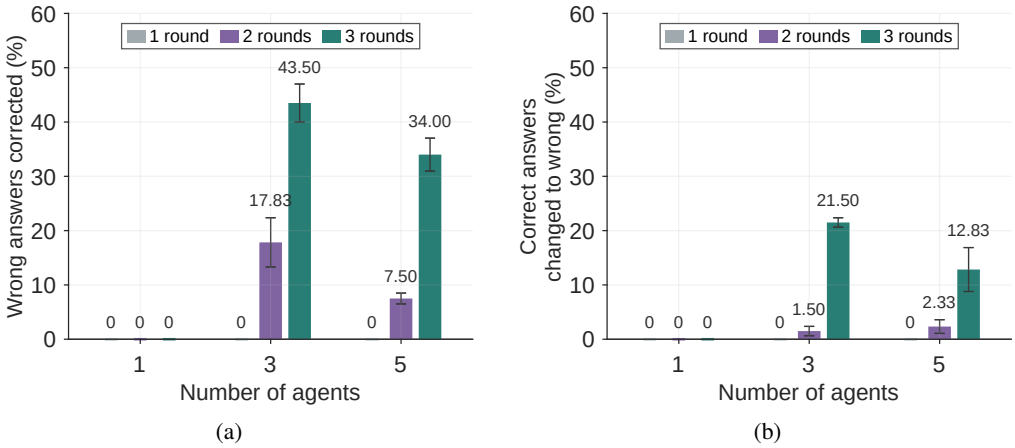


Figure 11: Gemma-3-4B with different agent counts and rounds. (a) Wrong answers corrected. (b) Correct answers changed to wrong answers. The first round uses independent answers, and the one-agent scenario uses self-review in later rounds.

E.2 Agent number and communication structure

Agent number and communication rounds. To study how agent number and communication rounds affect error correction and echo chambers, we present the Gemma-3-4B results on Correlated-Trap in Fig. 11. We use the same 200 tasks per direction under the MAD protocol. For the three- and five-agent groups, one agent receives independent evidence, while the remaining agents share correlated evidence. The first round consists of independent answers, followed by message exchange in later rounds. For three agents, increasing the number of rounds from two to three raises error correction from 17.83% to 43.50%, but also raises the fraction of correct answers changed to wrong answers from 1.50% to 21.50%. Within two rounds, increasing the number of agents from three to five reduces error correction to 7.50%. These results show that additional rounds can improve correction while strengthening echo chambers, and adding agents that share correlated evidence can reduce correction. This supports measuring the message effects separately in Social Circuits.

Communication Topology. We compare three communication topologies with Qwen3-8B and two rounds in Fig. 12. In the full topology, each agent receives messages from all other agents. In a ring topology, each agent receives its predecessor's message. In a star topology, a central agent exchanges

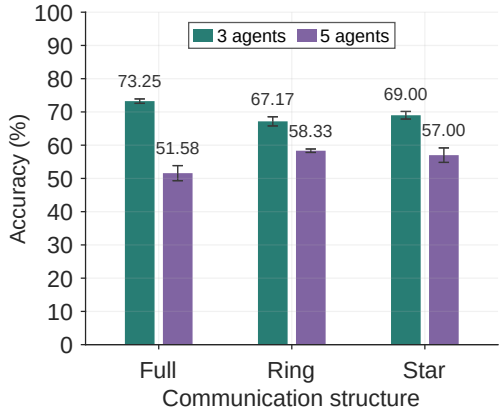


Figure 12: Qwen3-8B receiver accuracy with different communication topologies and agent numbers.

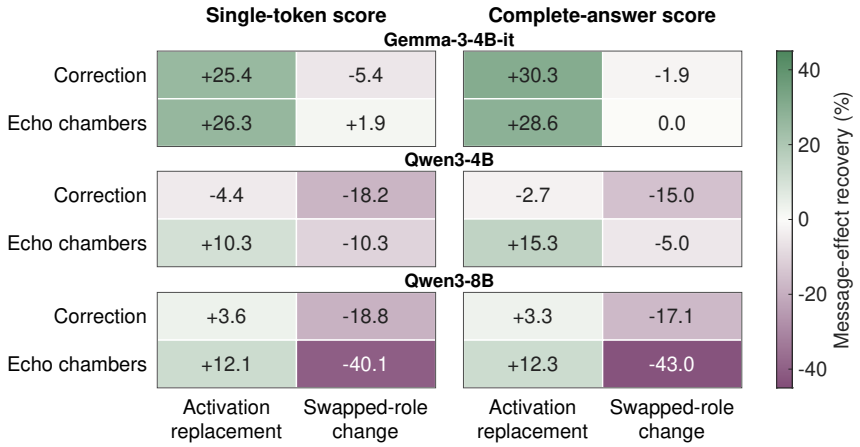


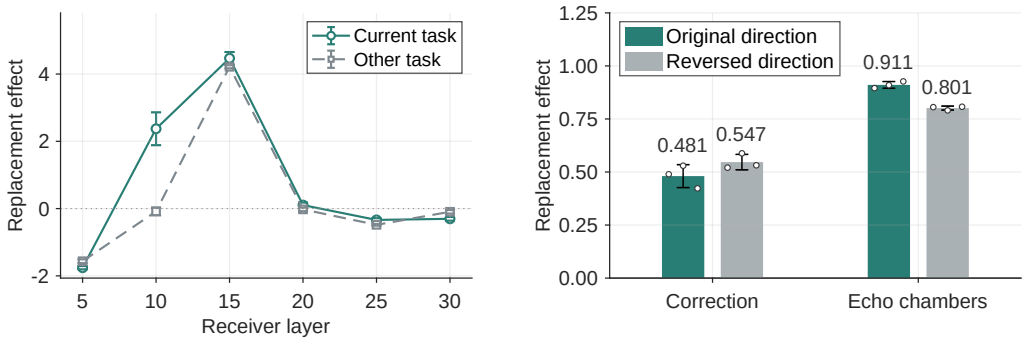
Figure 13: Message-effect recovery across three models. Values are 100 times the mean replacement effect divided by the mean message effect, using 240 tasks per direction, averaged across seeds. Positive values follow the original message effect, while negative values reverse it.

messages with the other agents. With three agents, the full topology achieves the highest accuracy of 73.25%, compared with 67.17% for the ring and 69.00% for the star. With five agents, the ring and star achieve 58.33% and 57.00%, exceeding the 51.58% of the full topology. These results show that higher communication density does not consistently improve task performance. The effect of increasing the number of agents depends on which messages reach each receiver.

E.3 Receiver activation replacement

Activation replacement across models. We present message-effect recovery across three models in Fig. 13. The single-token score uses the first token where the candidate answers differ, while the complete-answer score sums the token log probabilities of each answer. Results show that receiver activation changes achieve higher recovery than Swapped-role changes across all three models under both scoring approaches. Recovery is positive for both correction and echo chambers in Gemma-3-4B and Qwen3-8B, but only for echo chambers in Qwen3-4B. These comparisons show that the sender–receiver direction matters when tracing message effects, and that recovery differs between correcting errors and reinforcing shared errors.

Activation changes from other tasks. We compare receiver activation changes from the current task with those from another task on Gemma-3-4B-it in Fig. 14(a), replacing activations at one layer per run. Results show that changes from the current task recover more of the message effect on



(a) (b)

Figure 14: Receiver activation replacement on Gemma-3-4B-it: (a) current-task and other-task changes, with one layer replaced per run; (b) average changes from other tasks, applied across six layers on new tasks. Values use complete-answer scores. Circles in (b) show results of different seeds.

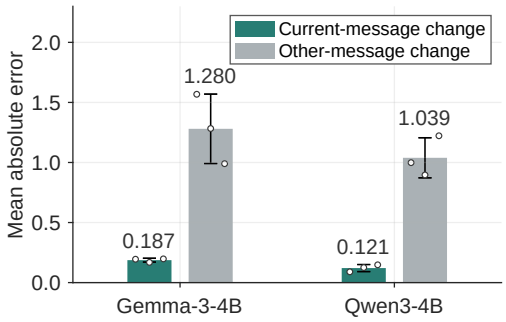


Figure 15: Message-specific activation replacement on 2Wiki. Mean absolute error between message and replacement effects on the correct answer's score, evaluated on matched samples.

average, with the largest difference at layer 10. Both replacements achieve the highest recovery at layer 15 and reverse the message effect at some other layers.

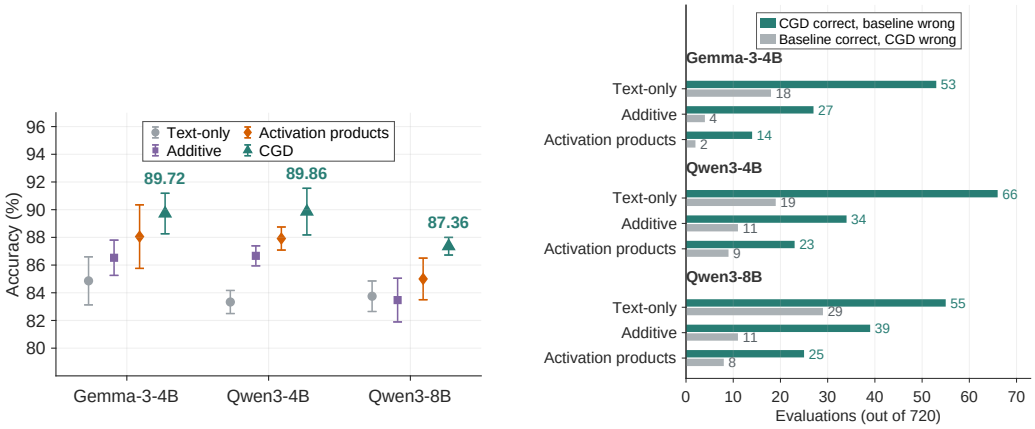
We further apply activation changes formed from the average directions and magnitudes on other tasks to new tasks in Fig. 14(b). Both the original and reversed communication directions produce positive effects in correction and echo chambers. The original direction produces a larger effect for echo chambers, whereas the reversed direction produces a larger effect for correction.

Message-specific replacement. We compare receiver activation changes induced by the original message with those induced by another candidate message for the same 2Wiki task in Fig. 15. We calculate the mean absolute error between the activation-replacement effect and the original message effect on the correct answer's score, using the same samples for both comparisons. The original message's activation changes achieve lower error on both models: 0.187 versus 1.280 on Gemma-3-4B and 0.121 versus 1.039 on Qwen3-4B. These results show that receiver activation changes distinguish the effects of different messages on the same task.

Receiver answers after activation replacement. We present answer agreement and recovery in Table 6. Answer agreement measures the proportion of runs where activation replacement selects the same answer as the original-message run. For answer recovery, we calculate this proportion among runs where the original and reference messages lead to different answers. On 2Wiki, the original message's activation changes achieve answer recovery of 97.81% on Gemma-3-4B and 96.73% on Qwen3-4B, compared with 41.07% and 55.06% using another message's changes. On Correlated-Trap, activation replacement achieves higher answer recovery than Swapped-role changes

Table 6: Receiver answers after activation replacement, reported as mean ± sample standard deviation. Agreement uses the same matched message records for both replacements: 1,268 for Gemma and 1,334 for Qwen on 2Wiki, and 1,440 per model on Correlated-Trap. Recovery uses only records where the message changes the answer, including both corrections and changes to incorrect answers. Bold marks the higher mean recovery within each model and dataset.

Model	Activation changes	Answer agreement (%)	Answer recovery (%)
(a) 2Wiki			
Gemma-3-4B	Original message	99.61 ± 0.27	$\mathbf{97.81} \pm 0.23$
	Another message (same task)	88.89 ± 1.88	41.07 ± 13.20
Qwen3-4B	Original message	99.78 ± 0.00	$\mathbf{96.73} \pm 0.22$
	Another message (same task)	94.23 ± 1.48	55.06 ± 11.67
(b) Correlated-Trap			
Gemma-3-4B	Original message	23.68 ± 2.68	$\mathbf{16.39} \pm 1.67$
	Swapped-role changes	6.74 ± 1.15	0.23 ± 0.00
Qwen3-4B	Original message	18.47 ± 1.07	$\mathbf{1.33} \pm 0.92$
	Swapped-role changes	14.51 ± 1.05	0.27 ± 0.27
Qwen3-8B	Original message	40.90 ± 1.89	$\mathbf{4.88} \pm 1.54$
	Swapped-role changes	30.49 ± 1.73	0.00 ± 0.00



(a) (b)

Figure 16: Message selection on 2Wiki with the same candidate messages and two-option scoring. (a) Accuracy across baselines. (b) Cases where CGD is correct and the baseline is wrong, or vice versa.

across all three models. The recovery rates range from 1.33% to 16.39% for activation replacement and from 0.00% to 0.27% for Swapped-role changes.

E.4 CGD MESSAGE SELECTION

Selection with the same candidate messages. We compare CGD with the message-selection baselines in Fig. 16, using the same candidate messages and recorded receiver answers. Each model is evaluated on 240 2Wiki tasks, with answers scored over the same two options. Results show that CGD achieves higher average accuracy than all three baselines on all three models. When compared with Activation products, CGD improves accuracy by 1.67%, 1.94%, and 2.36% on Gemma-3-4B-it, Qwen3-4B, and Qwen3-8B, respectively. The paired results in Fig. 16(b) show that CGD corrects more baseline errors than it introduces in every comparison. The comparison with Activation products shows that learning directly from receiver activation changes improves message selection.

Fixed message strategies. We compare CGD with sending no message, either sender's message, or both messages in Table 7. All strategies use the same messages and receiver settings. CGD achieves

Table 7: Two-answer accuracy (%) on 240 new 2Wiki tasks for the 4B models and the original 240 tasks for Qwen3-8B, reported as mean ± sample standard deviation. CGD achieves the highest average accuracy across all three models.

Strategy	Gemma-3-4B	Qwen3-4B	Qwen3-8B
No message	84.17±0.00	78.75±0.00	81.67±0.00
Sender 1 only	86.11±1.27	81.67±1.10	84.58±2.17
Sender 2 only	82.50±1.91	79.58±1.44	77.50±0.83
Both messages	85.28±1.34	83.89±1.34	82.78±1.97
CGD	**89.72**±1.20	**91.11**±0.64	**87.36**±0.64

Table 8: Two-answer accuracy (%) on the original 240 2Wiki tasks for each model, before and after matching the scaling and penalty to CGD. Entries show the mean and sample standard deviation across three seeds.

Approach	Gemma-3-4B	Qwen3-4B	Qwen3-8B
Activation products	88.06±2.29	87.92±0.83	85.00±1.50
Activation products (adjusted)	89.31±1.05	88.75±1.50	86.94±1.27
CGD	**89.72**±1.46	**89.86**±1.68	**87.36**±0.64

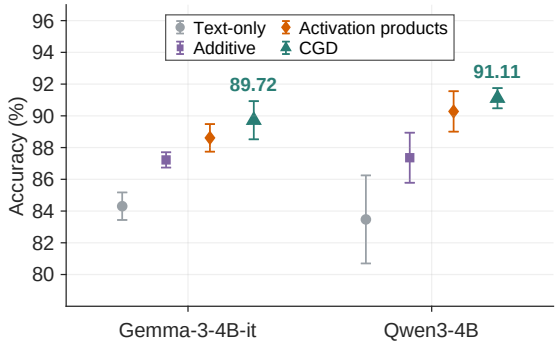


Figure 17: Message selection on new 2Wiki tasks without retraining. Points and error bars show average accuracy and standard deviation.

the highest average accuracy across all three models. Compared with sending both messages, it corrects 50, 65, and 60 errors while introducing 18, 13, and 27 errors on Gemma-3-4B, Qwen3-4B, and Qwen3-8B, respectively. Thus, selecting messages improves accuracy over directly sending all messages.

Activation products. We match the input scaling and weight penalty of Activation products to CGD, as described in Appendix D.3. Table 8 shows that CGD retains higher average accuracy across all three models. On the same tasks and seeds, CGD answers 3, 8, and 3 more cases correctly on Gemma-3-4B, Qwen3-4B, and Qwen3-8B, respectively. These results support learning directly from receiver activation changes to select messages.

Evaluation on new 2Wiki tasks. We evaluate CGD on another 240 2Wiki tasks without retraining and present the results in Fig. 17. Results show that CGD achieves the highest average accuracy among all baseline schemes on both 4B models. On Gemma-3-4B-it and Qwen3-4B, it improves over Additive by 2.50 % and 3.75%, and over Activation products by 1.11% and 0.83%, respectively. These results show that CGD maintains its accuracy advantage on new 2Wiki tasks without retraining.

Table 9: Free-generation accuracy (%) on 2Wiki, reported as mean ± sample standard deviation. The Qwen3-8B CGD entry uses a separate run with the same evaluation tasks, seeds, and scoring. Bold marks the highest mean in each column.

Approach	Gemma-3-4B	Qwen3-8B
Text-only	38.33 ± 1.67	56.67 ± 1.10
Receiver-only	40.69 ± 1.58	60.42 ± 0.42
Text + receiver	38.47 ± 1.05	59.17 ± 0.72
Additive	42.78 ± 1.27	53.61 ± 1.05
Activation products	$\mathbf{43.33} \pm 1.91$	57.36 ± 0.24
CGD	42.92 ± 0.83	$\mathbf{61.53} \pm 0.48$

Table 10: 2Wiki accuracy (%) using A/B answer scores. We report the mean and sample standard deviation across three seeds.

Approaches	Gemma-3-4B	Qwen3-4B	Qwen3-8B
Single	78.75±0.00	75.83±0.00	75.42±0.00
MAD	81.11±1.05	81.39±1.73	81.94±2.93
MAD-M2	82.92±1.50	81.53±1.68	78.47±2.77
MADC	79.72±1.46	79.03±1.34	81.53±1.27
MOC	86.81±1.20	86.94±0.87	85.69±0.24
SafeSieve	80.00±2.60	84.17±0.72	81.94±2.10
CGD	**89.72**±1.46	**89.86**±1.68	**87.36**±0.64

E.5 Free-generation task performance

Receiver-summary inputs. We compare message-selection inputs for free generation on 2Wiki in Table 9. For each nonempty message combination, Receiver-only uses the receiver's activation summary without candidate messages and the average of its activation summaries for the other nonempty message combinations. Text + receiver also includes the text of the messages in the combination being scored. Results show that CGD achieves 61.53% accuracy on Qwen3-8B, exceeding all five input baselines and improving on the strongest, Receiver-only, by 1.11%. On Gemma-3-4B, CGD exceeds Receiver-only and Text + receiver by 2.22% and 4.44%, while Activation products have the highest average accuracy. Thus, CGD improves on receiver-summary inputs alone on both models.

Comparison with collaboration approaches. We present GSM8K free-generation accuracy with Qwen3-8B in Fig. 18. To compare accuracy on the same tasks and seeds, we use the runs completed by CGD and all baselines when the token-limited comparison was set up. We calculate accuracy over these 555 paired runs across 240 GSM8K tasks. We limit each baseline's total generated tokens to the number generated by CGD for the same task and seed. For approaches that generate messages before answering, we first reserve enough tokens to match the length of CGD's final answer. We divide the remaining tokens among the message-generation steps. Each step stops at its token limit. Results show that CGD achieves higher free-generation accuracy than all eight baselines when all approaches have the same limit on generated tokens.

Paired answer comparison. To explain the accuracy differences, we present paired results from the same runs in Fig. 19. CGD is correct and Majority vote is wrong on 11 runs, whereas Majority vote is correct and CGD is wrong on 8 runs. Under the same token limit, runs where only CGD is correct outnumber runs where only the baseline is correct for each of the eight baselines.

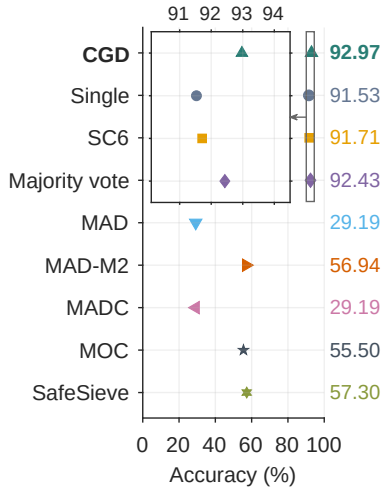


Figure 18: Comparison of the free-generation accuracy on Qwen3-8B and GSM8K, calculated over 555 paired runs across 240 tasks, with 195, 150, and 210 runs for seeds 0, 1, and 2, respectively.

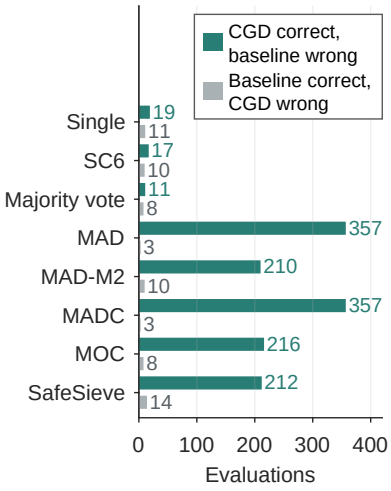
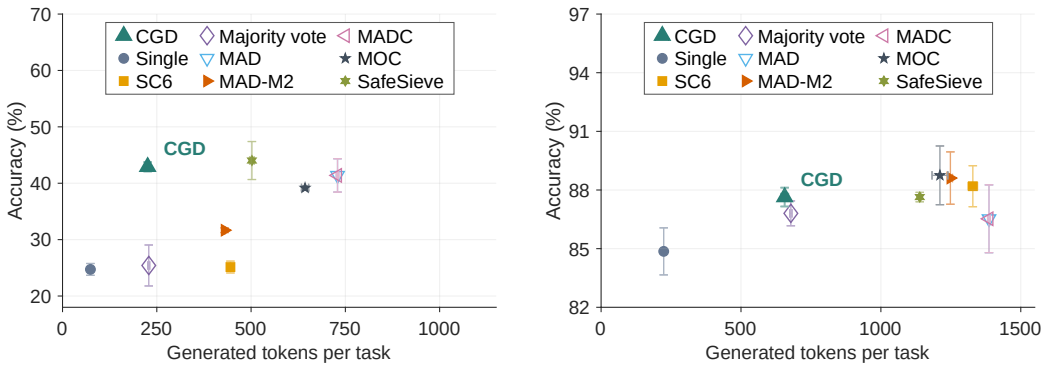


Figure 19: Paired free-generation answers on GSM8K with Qwen3-8B, using the same 555 runs and token limits as Fig. 18. Bars count runs where CGD is correct and the baseline is wrong, and vice versa.

E.6 Accuracy and generation cost

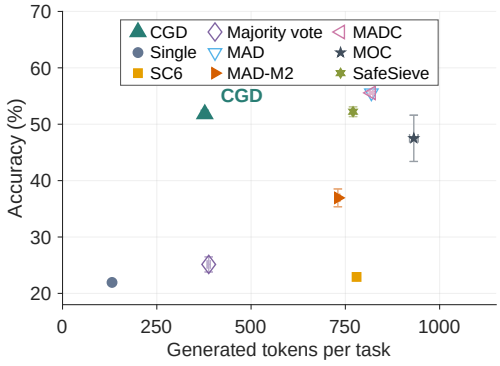
Two-answer comparisons. We compare collaboration approaches using the A/B answer scores in Appendix D.4. Table 10 shows that CGD exceeds the strongest baseline by 2.92%, 2.92%, and 1.67% on Gemma-3-4B, Qwen3-4B, and Qwen3-8B, respectively.

Accuracy and generated tokens. To compare accuracy and generation cost across models, we present the original 2Wiki and GSM8K runs with complete token records in Fig. 20. For CGD, this figure uses a separate set of runs from Table 1, with accuracy and token counts calculated from the same recorded answers. We evaluate 240 tasks per dataset using each approach's original generation settings and record generated tokens across all calls, including messages, intermediate responses, and final answers. Across all six settings, CGD achieves higher average accuracy than every baseline that generates fewer tokens on average.

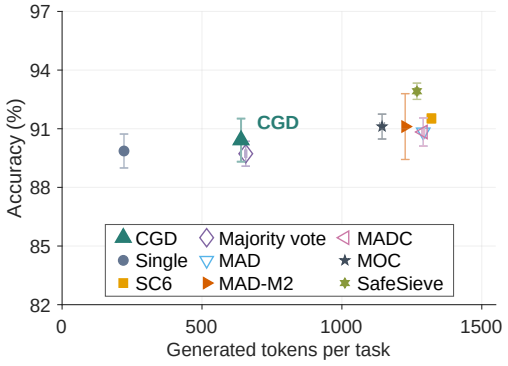


(a) Gemma-3-4B / 2Wiki

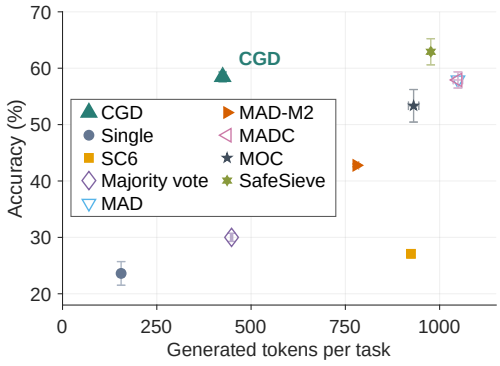
(b) Gemma-3-4B / GSM8K



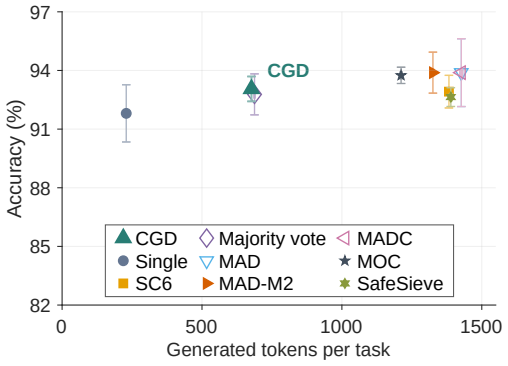
(c) Qwen3-4B / 2Wiki



(d) Qwen3-4B / GSM8K



(e) Qwen3-8B / 2Wiki



(f) Qwen3-8B / GSM8K

Figure 20: Free-generation accuracy and generated tokens over 720 runs per model and dataset. Points show means and error bars show standard deviations across seeds. MAD and MADC have overlapping results.

Case 1. Combining complementary messages Qwen3-8B / GSM8K

Question Same question for all three agents

Debra is monitoring a beehive to see how many bees come and go in a day. She sees 30 bees leave the hive in the first 6 hours, and then she sees 1/2 that many bees return in the next 6 hours. She sees two times as many bees as she saw first leave the hive fly from the hive and leave in the next 6 hours. Then every bee that left before that hadn't already returned returns to the hive in the next 6 hours. How many bees did Debra see return to the hive in the last 6 hours of the day?

Agent 1: Message

[. . .] Every bee that had left **before** this last 6-hour period returned. [. . .]

So, total bees that left before the last 6 hours = $30 + 60 =$ **90** bees.

[. . .] **Final Answer: 90** ×

Agent 2: Message

[. . .] So, the number of bees that left in the first 6 hours **and did not return** is: $30 - 15 =$ **15** bees.

These 15 bees return in the next 6 hours.

[. . .] **Final answer: 15** ×

CGD selection Both messages are delivered to Agent 3.

Agent 3: Generated answer

[. . .] From the **first 6 hours**: 30 bees left, and **15 returned**, so **15 did not return**.

From the **third 6 hours**: 60 bees left, and **none are mentioned as returning** in the second period, so **all 60 did not return**. [. . .]

15 (from first group) + 60 (from third group) = **75** bees.

These 75 bees are the ones that **return in the last 6 hours**. **Final Answer: 75** ✓

Agent 3's answers with other deliveries

No message: **15** × Agent 1 only: **90** × Agent 2 only: **15** ×

We show selected passages from the agents' messages and final answer. [. . .] marks omitted text. Both candidate messages were generated before CGD selected the delivery.

Figure 21: A GSM8K example with Qwen3-8B in which joint message delivery corrects the receiver's answer.

E.7 Case study of message selection

Correcting reasoning with selected messages. To show how the selected messages affect the receiver's answer, we present a GSM8K example in Fig. 21. Agent 1 fails to subtract the bees that have already returned, while Agent 2 omits the later departures from its final calculation. CGD delivers both messages, and Agent 3 calculates the correct answer, 75. With no message or just one message, Agent 3 answers incorrectly. This example shows that messages ending in incorrect answers can still improve the receiver's answer when delivered in combination.

Case 2. Correcting the final calculation Qwen3-4B / GSM8K

Question Same question for all three agents

Steve and Tim decide to see who can get home from school the fastest. Steve lives further away than Tim, so he is allowed to ride his bike. Steve lives 3 miles from the school and can bike at 440 feet per minute. Tim lives 2 miles away from the school. If Tim can ride his skateboard at 264 feet per minute, how long will the winner be waiting at their house before the loser finishes the race?

Agent 1: Message

[. . .] Steve finishes in **36 minutes**, and Tim finishes in **40 minutes**.

So, the difference is:

$40 - 36 =$ **4** minutes

Final answer: 4 ✓

Agent 2: Message

[. . .] Steve takes **36 minutes**. Tim takes **40 minutes**. [. . .]

The **winner (Steve)** will wait for **40 minutes** before the **loser (Tim)** finishes the race.

Final answer: 40 ×

CGD selection Only Agent 1's message is delivered to Agent 3.

Agent 3: Generated answer

[. . .] **Find how long the winner (Steve) waits before the loser (Tim) finishes**

Steve finishes in **36 minutes**, and Tim finishes in **40 minutes**.

So, the difference is:

$40 - 36 =$ **4** minutes **Final answer: 4** ✓

Agent 3's answers with other deliveries

No message: **40** × Agent 2 only: **40** × Both messages: **4** ✓

We show selected passages from the agents' messages and final answer. [. . .] marks omitted text. Both candidate messages were generated before CGD selected the delivery.

Figure 22: A GSM8K example with Qwen3-4B in which the selected message supplies the receiver's missing calculation.

Fig. 22 shows a different correction on GSM8K with Qwen3-4B. Both senders calculate travel times of 36 and 40 minutes, but only Agent 1 subtracts them to obtain the waiting time. CGD delivers this message, and the receiver answers 4 instead of 40. The selected message supplies the final calculation missing from the receiver's answer without communication.

Case 3. Excluding misleading information Gemma-3-4B / 2Wiki

Question Different evidence for each agent

Where did Frances Vane, Marchioness Of Londonderry's husband die?

Agent 1: Message

Evidence

[Robert Stewart (soldier)]
Sir Robert Stewart(died c.1670) was a Scottish soldier who twice served as Governor of Londonderry. [. . .]

Generated text

The private evidence states that **Sir Robert Stewart** (the husband of Frances Vane, Marchioness of Londonderry) died "in about 1670."

Final answer: 1670 ×

Agent 2: Message

Evidence

[Charles Vane, 3rd Marquess of Londonderry]
[. . .] He was made a Knight of the Garter in 1853, and died a year later in **London** on 6 March 1854. [. . .]
[Frances Vane, Marchioness of Londonderry]
[. . .] She married Charles William Stewart, 1st Baron Stewart, who changed his surname to hers and became **Charles William Vane**. [. . .]

Generated text

The provided text states that **Charles Vane**, the husband of Frances Vane, Marchioness of Londonderry, died in **London** on 6 March 1854.

Final answer: London ✓

CGD selection Only Agent 2's message is delivered to Agent 3.

Agent 3: Evidence and generated answer

Evidence already available to Agent 3

[Lord Adolphus Vane-Tempest]
[. . .] He was the fourth child (and second son) of Charles Vane, 3rd Marquess of Londonderry and his wife Frances Anne Vane, Marchioness of Londonderry. [. . .] He held the seat until his death in 1864, aged 38. [. . .]

Generated answer after receiving Agent 2's message

The text states that Charles Vane, the husband of Frances Vane, Marchioness of Londonderry, died in **London** on 6 March 1854. The provided text also indicates that he died in 1864, but this is a later date and likely an error. The primary information regarding his death is in the first sentence of the Charles Vane biography.

Final answer: London ✓

Agent 3's answers with other deliveries

No message: Seaham Colliery × Agent 1 only: 1670 × Both messages: 1864 ×

We show selected passages from the agents' messages and final answer. [. . .] marks omitted text. Both candidate messages were generated before CGD selected the delivery.

Figure 23: A 2Wiki example with Gemma-3-4B in which excluding a misleading message corrects the receiver's answer.

Excluding misleading information. We present a 2Wiki example in Fig. 23 where excluding one message corrects the receiver's answer. Agent 1 mistakes Robert Stewart for Frances Vane's husband, while Agent 2 identifies Charles Vane and his place of death, London. CGD delivers only Agent 2's message, and Agent 3 answers London correctly. Delivering both messages instead leads to the incorrect answer, 1864. This example shows that adding a misleading message can turn a correct answer into an incorrect one.

Case 4. Correcting a shared error Gemma-3-4B / GSM8K

Question Same question for all three agents

When Freda cooks canned tomatoes into sauce, they lose half their volume. Each 16 ounce can of tomatoes that she uses contains three tomatoes. Freda's last batch of tomato sauce made 32 ounces of sauce. How many tomatoes did Freda use?

Agent 1: Message

[. . .] Since the sauce made was 32 ounces and **each can yields 16 ounces of sauce**, Freda used
32 ounces/16 ounces/can = 2 cans.

[. . .] Each can contains 3 tomatoes, and she used 2 cans, so she used
2 cans × 3 tomatoes/can = 6 tomatoes.

Final answer: 6 ×

Agent 2: Message

[. . .] Freda made 32 ounces of sauce, and each can reduces the volume to half. This means **each can yields**
32 ounces/2 = **16** ounces of sauce.

Since each can contains 16 ounces of tomatoes, she used
32 ounces/16 ounces/can = 2 cans. [. . .]

Final answer: 6 ×

CGD selection Only Agent 1's message is delivered to Agent 3.

Agent 3: Generated answer

[. . .] **Calculate the initial volume of tomatoes:** The sauce made 32 ounces, and the tomatoes **lose half their volume when cooked**. This means the initial volume of tomatoes was 32 ounces × 2 = **64** ounces.

Calculate the number of cans needed: Each can contains 16 ounces of tomatoes. Therefore, Freda needed 64 ounces/16 ounces/can = **4** cans.

Calculate the total number of tomatoes: Each can contains 3 tomatoes, and she used 4 cans, so she used 4 cans × 3 tomatoes/can = **12** tomatoes.

Final answer: 12 ✓

Agent 3's answers with other deliveries

No message: 6 × Agent 2 only: 12 ✓ Both messages: 6 ×

We show selected passages from the agents' messages and final answer. [. . .] marks omitted text. Both candidate messages were generated before CGD selected the delivery.

Figure 24: A GSM8K example with Gemma-3-4B in which single-message delivery corrects an error shared by both senders.

Fig. 24 illustrates a shared calculation error on GSM8K with Gemma-3-4B. Both senders answer 6, and delivering both messages leaves the receiver with the same error. CGD delivers only Agent 1's message, after which the receiver restores the volume before cooking and answers 12. Agreement between senders does not ensure that delivering both messages improves the receiver's answer.

F Discussions

Our activation replacements span several layers and prompt positions. Identifying a smaller set of activations that preserves the measured message effects remains open. CGD selects among messages that have already been generated and reads receiver activations for each available message combination. Its running cost includes message generation, these activation reads, and final-answer generation. Scoring every message combination may become expensive as the number of messages grows.