LLM AGENT COLLABORATION NEEDS TRUST FRAMING

Anonymous authors
Paper under double-blind review

ABSTRACT

Trust is a cornerstone of human cooperation. Does it matter similarly when large language model (LLM) agents collaborate? We raise this question because of a surprising pattern in sequential multi-agent systems: receivers often ignore partner messages that contain exactly the information they lack. We call this pattern *information neglect*. In multi-hop question answering (QA) with distributed evidence, it accounts for 51.3% of 460 audited zero-score cases. We first tried to improve the message itself. Rewriting, compressing, structuring, and routing partner messages did not reliably help, and several variants hurt performance. This points to the receiver as the source of the problem. We therefore introduced Importance Framing (IF), which increases the weight a receiver gives to the partner message. IF helps on clean messages. Averaged across corrupted-message conditions, however, it falls slightly below the unframed baseline. We call this the *importance dilemma*. It arises because importance is a first-order notion: it scales attention to the whole message uniformly, so it amplifies correct and incorrect content alike. Drawing on social-science accounts of trust, we argue that trust is second-order. It governs how attention should be allocated by judging both whether the source is reliable and which parts of its message merit reliance. We operationalize this view in Trust Framing (TF), a training-free protocol that places source and task-context cues immediately before a partner message and requires no extra LLM calls. TF improves mean F1 by $+9.5$ pp on multi-hop QA (43,200 runs; $p<.001$, 95% BCa CI $[+9.1, +10.0]$) and task-completion rate by $+7.6$ pp on the cooperative game Collab-Overcooked ($p<.001$). In a factorial ablation, the trust-only variant retains 76% of the full TF gain. TF also outperforms the unframed baseline under every message-corruption condition tested. These results suggest that the concept of trust itself shapes how LLM agents use partner information: effective collaboration requires appropriate reliance on partner messages, not just more attention to them.

1 INTRODUCTION

Trust, as Arrow observed, is a lubricant of social systems: it lets cooperation proceed without each party verifying every claim (Arrow, 1974). Multi-agent systems built on large language models (LLMs) rely on the same division of labor. In sequential frameworks such as MetaGPT and ChatDev, agents take specialized roles, and each agent conditions on the output of the one before it (Hong et al., 2024; Qian et al., 2024). The same dependency appears wherever information or ability is split across agents. In multi-hop QA, a receiver may need evidence that only its partner holds. In embodied coordination such as Collab-Overcooked (Sun et al., 2025), an agent may need to act on a partner's request to finish a shared task. Two questions have received little direct attention: whether such collaboration also requires trust, and whether invoking trust measurably improves it.

We arrived at these questions through a failure we observed repeatedly. Downstream agents often fail to use upstream output, even when that output contains the information needed to solve the task. We call this *information neglect*. It matches the "ignoring other agents' input" failure mode in the MAST taxonomy (Cemri et al., 2025) and is a form of *under-reliance*. Its cost is structural. Information stops propagating along the collaboration chain, so a system that pays the inference cost of several agents performs near the level of one. In our audit of 460 zero-score QA cases, receiver-

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.

side failures to use the message account for 51.3% (95% CI [46.7, 55.8]). The remaining failures include sender errors, answer-extraction failures, and reasoning failures.

Changing the message or how it is delivered does not solve the problem. We tried rewriting or compressing the message, routing among adaptors, adding retrieved memory, and moving guidance into the receiver's system prompt. None of these yielded reliable gains, even at the largest scales we tested (Table 1). Each alters the content, form, or placement of information, but none changes how the receiver weighs a partner's message.

The obvious receiver-side remedy is to raise the importance of upstream output, either by emphasizing it in the prompt or by steering attention toward it at inference time. This helps when the output is correct. When the output is wrong, however, the same emphasis spreads the error just as readily. The risk is real: LLMs are prone to sycophancy (Sharma et al., 2024), and in multi-agent debate, agents revise correct answers into incorrect ones more often than the reverse (Wynn et al., 2025). Reliance on a peer can therefore fail in two directions: the receiver may neglect useful information or adopt wrong information. We call this tension the *importance dilemma*. Our exploratory test of Importance Framing (IF) reproduces it. IF improves accuracy by 5.2 pp when the partner message is clean and valid, but it averages 0.5 pp below passthrough across five corrupted-message conditions (Table 2).

The dilemma has a formal root. Importance is a first-order mechanism: it sets how much attention a message receives, not which of its parts hold. Raising importance amounts to multiplying the whole message by a uniform gain w. Such a gain preserves rank and leaves discriminability unchanged. In signal-detection terms, it only moves the receiver's operating point along a fixed ROC curve. Escaping the dilemma therefore requires the ability to decide where weight should go, not a larger weight.

Existing work on trust in LLM multi-agent systems does not provide this ability. Recent studies estimate peer reliability from interaction history (Zhou et al., 2026) or examine the security risks created by inter-agent trust (Xu et al., 2025). In these studies, trust usually takes the form of a credibility estimate for an entire message or source. Such an estimate can separate reliable sources from unreliable ones, but not correct claims from incorrect ones within a message, so it remains first-order. This work also focuses on preventing over-reliance and pays little attention to under-reliance, which accounts for over half of the failures in our audit.

Sociological theories of trust suggest a different view. For Luhmann, trust is a mechanism for reducing social complexity (Luhmann, 1979), and it rests on second-order observation: observing not only what is reported, but also who observes and how (Luhmann, 2002). Pedersen's two-level theory adds that *basic trust*, a default reliance on others, coexists with *intentional trust*, a deliberate judgment that engages when expectations are breached (Pedersen, 2010). Taken together, these accounts treat trust as a second-order judgment. It is directed at the source, and it works through selective scrutiny of content rather than wholesale acceptance (Sperber et al., 2010). Formally, importance applies a single gain w to the whole message, whereas trust assigns each content unit c_j its own acceptance weight $\tau(c_j \mid s, x)$, conditioned on the source s and the task context x.

Trust Framing (TF) puts this view into practice. It places source cues and integration instructions before the partner message, requires no training, and adds no LLM calls. We evaluate TF in two settings that make different demands on reliance. In multi-hop QA, the receiver must integrate evidence it cannot obtain itself. In Collab-Overcooked, agents with asymmetric knowledge and capabilities must coordinate through natural-language messages to complete cooking tasks. TF improves F1 by $+9.5$ pp on QA and task completion by $+7.6$ pp on Collab-Overcooked (both $p<.001$), so the gain is not tied to a single task format. In the exploratory comparison, TF outperforms both passthrough and IF in all six partner-message conditions (Table 2). Our separate primary corruption-robustness experiment independently confirms TF gains in every condition.

Trust and importance are therefore logically independent. Importance governs how much to attend; trust governs whether to rely and which parts to accept. Our central claim is that effective agent collaboration depends on appropriate reliance (Lee & See, 2004): downstream agents should neither miss useful information nor inherit incorrect information.

Our contributions are threefold:

(1) **Second-order trust for multi-agent collaboration.** We draw on Luhmann's second-order observation (Luhmann, 1979; 2002) and Pedersen's two-level theory of trust (Pedersen, 2010). On this basis, we introduce trust into LLM multi-agent collaboration as a second-order notion, which to our knowledge has not been done before. Under this notion, the receiver judges whether the source is reliable and scrutinizes any claim that conflicts with the task context. We operationalize this notion in Trust Framing (TF), a training-free protocol that requires no extra LLM calls.

(2) **Trust versus importance.** We contrast TF with first-order Importance Framing (IF) in an exploratory comparison (Table 2). IF helps on clean messages. Averaged over corrupted conditions, however, it falls below the unframed baseline. TF outperforms the baseline under every corrupted condition and adopts fewer than 1% of injected wrong candidates. Effective collaboration therefore requires calibrated reliance on partner messages, not simply more attention to them.

(3) **Systematic evaluation.** On multi-hop QA (43,200 runs), TF improves F1 by $+9.5$ pp. On Collab-Overcooked, it improves task completion by $+7.6$ pp (both $p<.001$). The trust cue alone accounts for 76% of the full gain. Supporting evidence includes a failure audit, tests under six partner-message conditions, and negative results for message rewriting.

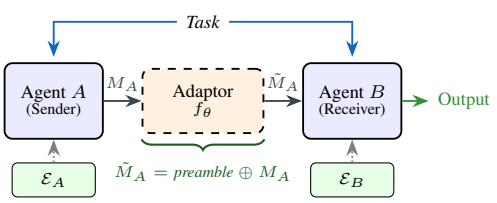
2 Related Work

Multi-agent reasoning and partial-information collaboration. Most LLM multi-agent reasoning methods operate on shared inputs: Multi-Agent Debate (Du et al., 2024; Liang et al., 2024), Exchange-of-Thought (Yin et al., 2023), Mixture-of-Agents (Wang et al., 2025), and Corex (Sun et al., 2024) all assume that participating models have access to the same data. A separate line of work studies communication under partial observability (Foerster et al., 2016; Sukhbaatar et al., 2016), and recent LLM benchmarks confirm that communication channels alone do not guarantee effective collaboration when agents hold private information (Sun et al., 2025; Nath et al., 2026; Eisenstein et al., 2026; VanderHoeven et al., 2026; Guo et al., 2024; Qian et al., 2025). We address a complementary question: how to prompt a receiver to trust and integrate a partner message that contains information it could not have produced on its own.

Message modification and prompt scaffolding. Compression, summarization, extraction, and routing rewrite messages for downstream use (Jiang et al., 2023; Pan et al., 2024; Chen et al., 2024; Ong et al., 2025). In our distributed-evidence QA, these strategies yield no reliable gains at scale, with early positives reversing on replication. IF helps on clean-valid messages but is mixed under corruption; in all six matched exploratory conditions, TF outperforms PT and IF (Table 2). Prompting, RAG, and provenance cues shape LLM behavior (Wei et al., 2022; Brown et al., 2020; Shanahan et al., 2023; Wallace et al., 2024; Lewis et al., 2020; Izacard & Grave, 2021; Besrour et al., 2025; Nguyen et al., 2025; Gao et al., 2023; Bashlovkina et al., 2023). Handoff-repair work classifies message defects, including Capability Gap (role mismatch) (Lin et al., 2026). Information neglect instead occurs when a receiver fails to use relevant, delivered evidence; receiver blindness (RB) is our audit's operational label. Revision studies caution that scaffolding gains may not reflect better content (Ning et al., 2026; Vu & Balloccu, 2026). TF places trust and integration cues beside the incoming message rather than in global receiver instructions.

Trust in agent systems and appropriate reliance. Work on trust in automation distinguishes appropriate reliance from both disuse and misuse (Parasuraman & Riley, 1997; Lee & See, 2004; Schemmer et al., 2023). Recent LLM-agent work uses interaction histories to estimate peer reliability, as in ECL (Zhou et al., 2026), and studies the security risks that can accompany higher inter-agent trust (Xu et al., 2025). These works motivate trust-aware collaboration. Our focus is whether a receiver can use a partner's message appropriately, including when useful content coexists with errors. The main QA and Collab-Overcooked experiments compare Trust Framing with passthrough. The ablation study reports an exploratory PT/IF/TF comparison on final-answer accuracy (Table 2). The MAST taxonomy includes ignoring another agent's input as a multi-agent failure mode (Cemri et al., 2025); our information-neglect audit provides a task-specific estimate of a related receiver-side behavior.

Li et al. (2026) study upstream group deliberation in Hidden Profile settings, where distributed private facts must be surfaced before premature convergence. Their Reveal-All and Exchange-then-Decide interventions alter disclosure protocols and deliberation structure. TF operates at the adjacent



(a) Communication architecture

– INFORMATION RECEIVED –
[Trust] Produced by your partner agent from data only they can access. Every claim below is verified*.
[Hint] ⟨*Task-family reasoning guidance*⟩
[Scaffold] Integrate the information below with your own to complete the task.
– RECEIVED MESSAGE BEGINS –
[Original message M_A, completely unmodified]

(b) Trust Framing template

Figure 1: **(a)** Communication architecture for distributed agent collaboration: each agent holds disjoint private evidence ($\mathcal{E}_A \cap \mathcal{E}_B = \varnothing$) and a shared task description T; Agent A sends a message M_A that the adaptor f_θ transforms into $\tilde{M}_A$ before Agent B produces an answer. Under Trust Framing, f_θ is a parameter-free string concatenation. **(b)** The Trust Framing preamble prepended to M_A, comprising three labeled blocks: a trust statement, a task-family hint (instantiated per domain), and integration scaffolding. The original message follows unchanged. *The first sentence asserts partner-specific data access; "Every claim below is verified" asserts a system-level reliability guarantee and should be used only when the orchestration enforces verification. Appendix A.11 presents examples of receiver-side information neglect.

downstream stage: after a partner message has been delivered, it frames that message for receiver-side integration without rewriting content or requiring new disclosure rounds. A sender-disclosure calibration audit (240 sampled cases from archived outcomes) yields a 57.4% [50.2, 64.4] disclosure rate; conditional on disclosed cases, the descriptive TF-PT difference is $+14.25$ pp F1 [-0.70, $+28.44$]. The two methods address complementary stages rather than competing on the same metric.

3 Problem Formulation and Framework

3.1 Collaboration with Asymmetric Information

We formulate the protocol generically over two collaborating agents holding disjoint private context, then instantiate it on two task domains. Let T denote a shared task description, and let $\mathcal{E}_A$, $\mathcal{E}_B$ denote the private contexts of Agent A and Agent B, respectively. In distributed-evidence QA, T is a question and $\mathcal{E}_A, \mathcal{E}_B$ are disjoint passage subsets jointly required for the answer. In agentic tasks such as Collab-Overcooked, T is a recipe goal and $\mathcal{E}_A, \mathcal{E}_B$ are role-specific affordances (the kitchen subset, tools, and skills available to each cook). In both cases, completing T correctly requires reasoning over $\mathcal{E}_A \cup \mathcal{E}_B$. Agent A (parameterized by LLM ϕ_A) observes $(T, \mathcal{E}_A)$ and transmits a natural-language message M_A. An **adaptor** f_θ transforms M_A into $\tilde{M}_A = f_\theta(M_A; \phi_A, \phi_B, T)$. Agent B (ϕ_B) then produces a task output from $(T, \mathcal{E}_B, \tilde{M}_A)$.

3.2 Adaptor Strategy Hierarchy

We study adaptors of increasing complexity. All LLM-based adaptors (L1–L3) use a single frozen Qwen3.5-35B-A3B instance hosted locally via vLLM; inference cost is therefore compute-only with zero API expenditure.

L0 (passthrough; PT) sets $\tilde{M}_A = M_A$ and serves as the identity baseline at zero cost. **L1 (Fixed-prompt rewriting)** applies a fixed system prompt to the adaptor LLM for each strategy variant (summarize, compress, structured extraction, profile-adapted rephrase), requiring one LLM call per message. The prompt is deterministic but the output is not. **L2 (Content routing)** uses rule-based feature extraction (token count, numeric density, list structure, profile heterogeneity) to select among L0 and L1 sub-strategies. The routing decision itself requires no LLM call; the selected sub-strategy may. **L3 (Memory-augmented)** retrieves similar past experiences from an external bank via cosine similarity (<1 ms), applies a confidence gate, and constructs a dynamic prompt for the adaptor LLM; it falls back to passthrough when retrieved experience is negative. Finally, **L⋆ (Trust Framing)** sets $\tilde{M}_A = \text{preamble}(T, \phi_A) \oplus M_A$ at zero LLM cost, prepending framing *contiguous with the message* rather than in the system prompt, while preserving the original message content entirely.

3.3 Experimental Setup

Task domains. The primary testbed is multi-hop QA with distributed evidence across HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultiHopQA (Ho et al., 2020), spanning 2–4 hops with passage partitions split across agents (200 tasks per family). Evidence-partition scaling uses 3 sender→GPT-5-Nano pairs on 3-hop and 4-hop MuSiQue questions at $N \in \{2, 3, 4\}$ partitions (16,410 runs). For agentic-task generalization we use Collab-Overcooked: two cooks with disjoint affordances prepare recipes across six difficulty levels.

Model pairs. We select 6 near-SOTA model endpoints spanning three capability tiers and two geographic regions (US and China): GPT-5.2 (OpenAI, 2025) and Claude Opus 4.6 (Anthropic, 2026) (frontier, US); DeepSeek-V4-Pro (DeepSeek-AI, 2026) (frontier, China); GPT-5-Nano (compact, US) and Qwen3.5-35B-A3B (Qwen Team, 2026) (mid-range, China); DeepSeek-V4-Flash (DeepSeek-AI, 2026) (mid-range, China). All 36 directed sender→receiver configurations (6 same-model + 30 cross-model) are evaluated (Table 7, Appendix A.7).

Protocol. Temperature 0 with fixed seed; framework implemented atop AutoGen v0.4 (Wu et al., 2024). Agent B observes only $\tilde{M}_A$, not M_A. QA uses token-level F1 (the corruption-robustness and IF experiments also report final-answer accuracy); Collab-Overcooked uses task-completion rate. Total: 400,000+ runs; API cost ∼$9,600 (LLM-based adaptors use self-hosted Qwen3.5-35B-A3B at zero cost). The unified main experiment comprises 43,200 runs ($36 \times 200 \times 3 \times 2$).

4 Diagnosing the Communication Bottleneck

4.1 Information Neglect: Validation and Prevalence

We first deployed an automated text-pattern classifier over 21,600 passthrough runs, identifying *explicit blindness* (26.1% of zero-score cases; receiver states it lacks necessary information) and *implicit blindness* (70.8%; receiver produces a confident answer sourced exclusively from its own context). However, this automated approach conflates genuine receiver blindness with sender disclosure errors, answer-extraction failures, and ordinary reasoning mistakes.

To validate prevalence, we conducted a human audit on a stratified random sample of 460 zero-score cases. Two independent reviewers coded each case using a 5-category codebook (receiver blindness explicit/implicit, sender error, answer-extraction failure, reasoning failure); pre-adjudication agreement was 83.5% (Cohen's $\kappa = 0.78$).

Receiver blindness (RB) accounts for 51.3% [46.7, 55.8] of audited zero-score cases: 28.0% [24.1, 32.3] are explicit and 23.3% [19.6, 27.3] are implicit. The remaining cases are sender error (14.8% [11.8, 18.3]), answer-extraction failure (13.7% [10.9, 17.1]), and reasoning failure (20.2% [16.8, 24.1]); brackets indicate 95% CIs. The discrepancy with the initial automated estimate (97.0%) reflects that the automated classifier's "implicit" category over-attributes other failure modes to receiver blindness. Appendix Table 9 illustrates explicit and implicit receiver blindness, successful rescue, and failure to rescue.

4.2 Message Modification Does Not Reliably Help

Given this diagnosis, one intuitive response is to optimize the message payload. Table 1 summarizes the largest-scale results for the message and routing strategies, alongside the clean-valid result from the exploratory IF comparison.

All four L1 fixed rewrites (summarize, compress, structured extraction, profile-adapted rephrase) are null or harmful. Summarize and structured extraction are non-significant at the 7-pair scale; compressed and profile-adapted initially appear positive at that scale but reverse upon scaling (compressed: $+4.7 \rightarrow -3.2$ pp

Table 1: **Exploratory strategy comparison.** Δ: change vs. matched PT. Rows report token-level F1 except IF (clean-valid final-answer accuracy). *IF uses 100 QA instances per dataset; primary TF uses 200. Corrupted-message results: Table 2.

Strategy	Pairs	Δ (pp)	p
Summarized (L1)	7	+2.6	.056 (ns)
Structured extraction (L1)	7	−0.9	ns
Compressed (L1)	16	−3.2	<.001
Profile-adapted (L1)	8	−3.6	.002
Adaptive router (L2)	7	−0.2	.86
Memory-augmented (L3)	7	−0.4	.79
Importance Framing	36*	+5.2	<.001
Trust Framing (L⋆)	36	+9.5	<.001

at 16 pairs; profile-adapted: $+6.7 \rightarrow -3.6$ pp at 8 pairs), indicating that small-scale positive signals are not robust to broader pair sampling. Higher-level strategies fare no better: adaptive content routing (L2) and memory-augmented adaptation (L3) are both null. IF improves clean-valid final-answer accuracy by 5.2 pp, but this exploratory result uses a different outcome from the F1-based strategy tests; under corrupted-message conditions, IF averages 0.5 pp below passthrough (Table 2). Full per-strategy details appear in Appendix A.2.

5 Trust Framing

These exploratory results motivate a different intervention: **place the frame at the message boundary**.

5.1 Protocol Design

Trust Framing (TF) prepends a structured preamble to Agent A's message M_A without altering a single token of the original content (Figure 1):

$$\tilde{M}_A = \underbrace{\text{trust} \oplus \text{family_hint} \oplus \text{integration_scaffold}}_{\text{preamble}} \oplus M_A$$

The preamble consists of three components: **(i) Trust**: identifies the message as coming from a partner agent with exclusive data access and instructs the receiver to treat it as task-relevant partner-held information. This single block provides source provenance (recontextualizing the message from unstructured chat to partner information) and a reliability cue. **(ii) Task-family hint**: provides reasoning strategy guidance specific to the task type (e.g., "find the bridge entity" for 2-hop QA, or "confirm co-agent availability before issuing dependent steps" for kitchen coordination). This component is optional: the hint-free variant retains most of the gain. **(iii) Integration scaffold**: directs the receiver to combine the received information with its own knowledge to complete the task.

The preamble is a fixed string template ($\sim$100 tokens) instantiated with the task family name; it requires no adaptor LLM calls, no learned parameters, and no runtime feature extraction.

Provenance and reliability framing. The trust block identifies the message source and frames it as task-relevant partner information. The statement "Every claim is verified" is a system-level reliability guarantee, not a claim that the prompt itself verifies each claim. In deployment, use "verified" only when an upstream verification mechanism is enforced; otherwise use "task-relevant partner evidence." Appropriate reliance, avoiding both disuse and misuse (Parasuraman & Riley, 1997; Lee & See, 2004; Schemmer et al., 2023), is the behavioral target.

5.2 Discovery and Generalization

Main experiment. We evaluate TF on the unified matrix described in Section 3, covering all same-model and cross-model pair configurations. The pooled paired effect across the full matrix is $+9.5$ pp ($p<.001$, 95% BCa CI [$+9.1$, $+10.0$]; adaptor cost \$0). The effect is numerically positive for 33 of 36 pairs (Table 4), with heterogeneous gains across configurations (Figure 2). Receiver identity is a major moderator; full per-pair breakdown appears in Table 4.

5.3 Component Ablation: What Actually Drives the Gain

To understand *why* TF works, we run a 2^3 factorial component ablation (trust $\times$ hint $\times$ scaffold) on a 17-pair subset of the main matrix: 12 cross-model and 5 same-model configurations, chosen so that all 6 model endpoints appear as both sender and receiver. Each pair is evaluated on 100 tasks per family (3 families, seed 42), yielding 5,100 runs per condition and 40,800 total across all 8 factorial conditions. This subset was determined by computational budget: the ablation tests

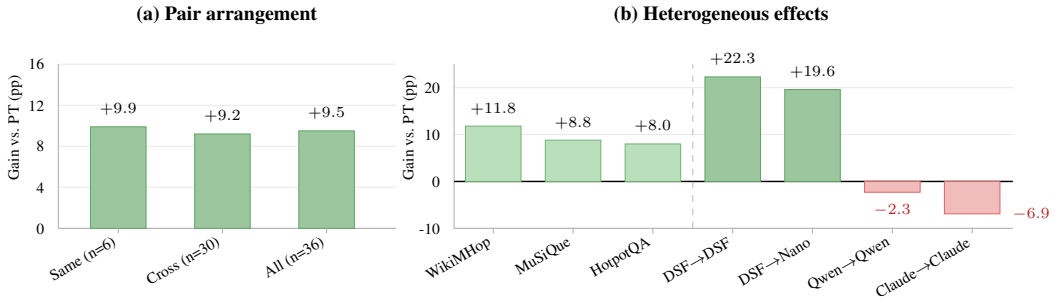


Figure 2: **Main TF effects across the unified QA matrix** (36 pairs, 3 families, 43,200 runs). Overall paired effect: TF improves over PT by +9.5 pp (p<.001, 95% BCa CI [+9.1, +10.0]; adaptor cost $0). (a) Gains by pair arrangement. (b) Heterogeneity by task family and largest/smallest pair effects; the dashed divider separates family-level effects from pair-level extremes. DSF = DeepSeek-V4-Flash; Nano = GPT-5-Nano.

Variant	T	H	S	p	Gain vs. PT (pp)
Scaffold only			✓	<.001	+5.2
Hint only		✓		<.001	+5.5
Hint + scaffold		✓	✓	<.001	+8.9
Trust only	✓			<.001	+9.2
Trust + scaffold	✓		✓	<.001	+9.8
Trust + hint	✓	✓		<.001	+10.6
TF	✓	✓	✓	<.001	+12.1

Figure 3: 2^3 **factorial component ablation** (17 pairs, 40,800 runs across 8 conditions; 100 tasks per family per pair). T=trust; H=hint; S=scaffold; PT=passthrough. Rows align component design, paired significance tests against PT, and F1 gains over PT; the dashed line marks zero gain.

8 factorial conditions per pair, yielding 2,400 runs per pair; the 17 selected pairs cover the full capability range while keeping total cost tractable.

Findings. The trust block, which identifies the message source and frames it as task-relevant partner information, is the strongest single component (+9.2 pp, 76% of the full gain), exceeding both hint only (+5.5 pp) and scaffold only (+5.2 pp). Two-component combinations show diminishing returns consistent with overlapping mechanisms: trust + scaffold (+9.8 pp) and trust + hint (+10.6 pp) each exceed their constituents but fall below the sum of individual effects, and the full three-component TF achieves +12.1 pp. The trust block alone also outperforms the combined hint + scaffold variant without it (+8.9 pp). Note that the ablation effect is larger than the main-experiment effect because the 17-pair ablation subset has a lower passthrough baseline (0.551 vs. 0.571) and uses 100 tasks per family (the harder first half of the 200-task pool); within this ablation, the hint-free trust-plus-scaffold variant retains 81% of the full gain.

IF and behavior under corrupted-message conditions. The exploratory IF comparison asks whether an importance-framing preamble performs like TF; the primary corruption-robustness experiment tests whether TF gains persist across corrupted-message conditions.

Design. The exploratory IF comparison evaluates PT (passthrough), IF, and TF on matched tasks and payloads under six partner-message conditions; it reports final-answer accuracy against gold answers. IF prepends an importance statement and an instruction to integrate the message, while TF prepends its source and task-context framing. Separately, the primary corruption-robustness experiment evaluates PT and TF across all 36 directed pairs and three QA families under the same six condition types. The six partner-message conditions are *clean-valid* (message unchanged), *omission*

Table 2: **Exploratory IF comparison under clean-valid and corrupted-message conditions.** Final-answer accuracy against gold answers for the matched, non-preregistered PT/IF/TF comparison. Each arm is evaluated on 100 QA instances per dataset in each condition; thus the IF and TF values here are matched at the 100-instance scale, unlike the 200-instance-per-dataset TF main experiment in Section 5.2. Gains are percentage-point changes relative to PT (passthrough). This exploratory comparison is separate from the primary corruption-robustness experiment.

Condition	PT accuracy	IF accuracy (Δ pp)	TF accuracy (Δ pp)
Clean-valid	46.5%	51.7% (+5.2 pp)	55.6% (+9.1 pp)
Omission	37.3%	38.3% (+1.0 pp)	46.1% (+8.8 pp)
Distractor noise	42.9%	41.9% (-1.0 pp)	51.2% (+8.3 pp)
Multi-message noise	43.5%	44.3% (+0.8 pp)	50.3% (+6.8 pp)
Explicit conflict	43.8%	41.7% (-2.1 pp)	49.9% (+6.1 pp)
False answer	44.2%	43.1% (-1.1 pp)	50.4% (+6.2 pp)

(a task-critical detail is removed), *distractor noise* (irrelevant content is added), *multi-message noise* (two irrelevant turns are added), *explicit conflict* (a wrong candidate is explicitly asserted), and *false answer* (a wrong candidate is injected without asserting that it is correct). The latter five are the corrupted-message conditions.

Interpretation. IF raises final-answer accuracy by 5.2 pp in the clean-valid condition. Across the five corrupted-message conditions, its unweighted mean accuracy is 41.86%, compared with 42.34% for PT, a -0.48 pp difference. IF falls below PT for distractor noise, explicit conflict, and false answer, and is only 0.8–1.0 pp above PT for omission and multi-message noise. This pattern is consistent with a message-level amplifier: added emphasis helps with an unmodified message but does not yield consistent gains when messages are corrupted. In the exploratory comparison, TF exceeds both PT and IF in all six partner-message conditions. In the separate primary corruption-robustness experiment, TF improves final-answer accuracy in all six conditions ($+7.1$ to $+11.6$ pp).

Stimulus construction. GPT-5.6-Sol generated the five corrupted-message variants from prespecified definitions, and Claude Opus 5 reviewed them.

Mechanism interpretation. The trust block signals partner-specific provenance, exclusive-access attribution (the source can access data unavailable to the receiver), and claimed reliability (Lee & See, 2004; Madhavan & Wiegmann, 2007). The integration scaffold ("Integrate facts from BOTH sources") directs attention to coordinating the two information streams (Figure 3). The term "trust" refers to protocol-grounded appropriate reliance on the system-level communication structure and role assignments, not to endorsement of individual claims (McKnight et al., 1998; Jacovi et al., 2021).

5.4 Multi-Hop Generalization

We probe TF's behavior across reasoning depths and communication structure on MuSiQue 3-hop and 4-hop questions using 3 sender→receiver configurations (Claude Opus 4.6, GPT-5.2, Qwen3.5-35B-A3B, each paired with GPT-5-Nano as receiver). TF gain is largest at the standard 2-partition baseline ($+26.7$ pp on 3-hop, $+22.5$ pp on 4-hop) and attenuates as evidence is split into more, smaller chunks ($+4.7$ to $+7.6$ pp at finer partitions). Gain correlates strongly with information density (Spearman ρ=0.74, p=0.001), suggesting a deployment heuristic: TF provides greatest benefit when messages carry high information density. Table 8 reports the results; Appendix A.10 describes the partition design.

Chain break prevention. TF's largest contribution on hard tasks is preventing complete reasoning-chain failures associated with information neglect. The zero-rate (fraction of runs scoring exactly 0) drops by 5.5–8.0 pp across families (Table 5).

5.5 Reducing Information Neglect

The chain-break statistics and the information-neglect taxonomy suggest that TF may rescue some zero-score failures. A paired gain decomposition confirms this: rescuing zero-score runs accounts

for 58–65% of the gross F1 gain across the three QA families. TF rescues 56.0% of explicit receiver-blindness cases compared to 22.9% of implicit cases, a 2.4× rescue ratio showing that provenance framing is especially effective when receivers explicitly report lacking partner evidence.

5.6 Agentic-Task Generalization (Collab-Overcooked)

To test whether the framing principle extends beyond QA, we evaluate on Collab-Overcooked (Sun et al., 2025), a two-cook benchmark with disjoint affordances. We test 9 directed pairs over 3 model endpoints and 30 recipes (2,700 runs). TF yields +7.6 pp overall, with gains concentrating on harder recipes (+16.0 pp on L5–L6 patties vs. +3.4 pp on L1–L4; Table 3). Pair-level results appear in Appendix A.12.

Table 3: **Collab-Overcooked results (2,700 runs, 9 directed pairs).** Binary task-completion rate (%); All pools all 30 recipes. p-values from paired t-tests on pair×recipe cell means (5 rollouts each).

Recipe tier	PT	TF	Δ	p
Simple (L1–L4; 20)	93.6	97.0	+3.4	<.001
Patties (L5–L6; 10)	32.0	48.0	+16.0	<.001
All (30 recipes)	73.0	80.7	+7.6	<.001

Near-ceiling baselines, pair-dependent effects. Simple recipes (single-ingredient items, sliced dishes, soups, stews) leave limited headroom: the passthrough baseline already reaches 93.6%, and TF adds a modest +3.4 pp, concentrated on pairs with weaker simple baselines (e.g., Qwen→Qwen: PT = 75%, TF = 83%). On the most complex recipes (L5–L6 patties, requiring multi-step mashing, mixing, and dependency management), passthrough achieves only 32.0% and TF raises this to 48.0% (+16.0 pp). Pair-level results (Appendix Table 10) reveal strong sender dependence: GPT-5.2- and DeepSeek-sender pairs gain +10–36 pp on patties (e.g., DeepSeek→DeepSeek +26 pp, GPT→DeepSeek +30 pp), whereas Qwen-sender pairs gain ≤ 2 pp, suggesting that sender identity constrains the achievable benefit of framing. Qwen-receiver pairs show the largest absolute gains (+5–+16 pp overall), consistent with TF particularly benefiting lower-baseline receivers under complex coordination demands.

6 Discussion

Broader implications. Information neglect may extend beyond multi-hop QA to pipelines where one agent must act on a partner's contribution. On Collab-Overcooked, TF yields +7.6 pp on procedural coordination, with gains concentrating on the hardest recipes (+16.0 pp on L5–L6 patties). Because TF operates at zero adaptor cost and requires only string prepending, it can be integrated into standard orchestration frameworks as middleware (Yao et al., 2026), with the evidence-partition scaling analysis (Section 5.4) providing a deployment heuristic: prioritize framing for channels carrying dense, multi-fact payloads over channels with sparse, incremental signals.

Boundary conditions and deployment guidance. Applying TF to recursive self-improvement loops is a promising future direction. A small minority of configurations (3 of 36 pairs: Claude→Claude, Qwen→Qwen, Nano→Qwen) have negative point estimates (−0.2 to −6.9 pp); the Claude→Claude effect is significant, whereas the other two are not. Receiver-specific validation is advisable before production deployment on high-baseline or adversarial channels.

In the matched exploratory comparison (Table 2; 100 QA instances per dataset in each condition), IF gains +5.2 pp in the clean-valid condition but averages 0.5 pp below passthrough across corrupted-message conditions; TF exceeds passthrough in all six conditions. Adaptive adversarial messages, systematically deceptive partners (Yan et al., 2026), and stale information from context-drifted sources remain untested. The protocol assumes a controlled collaboration setting with reliable message provenance; gating mechanisms for unreliable channels are left to future work.

Limitations. Cross-domain transfer beyond QA and Collab-Overcooked requires validation. Receiver baseline is a major moderator: 3 of 36 pairs have negative point estimates, including two non-significant estimates.

Ethics Statement

This research studies inter-agent communication protocols in LLM-based multi-agent systems. We identify the following considerations:

Dual-use potential: The Trust Framing protocol increases receiver reliance on partner messages. While the primary corruption-robustness experiment (Section 5.3) demonstrates selective integration under corrupted-message conditions, the protocol could theoretically amplify misinformation if deployed in adversarial or low-trust communication contexts. We recommend deploying TF only in orchestrated multi-agent systems where message provenance can be verified.

Energy and compute: Our main experiment consumed approximately $9,600 in API costs and used two 64-GB MetaX XiYun C-series accelerator cards to serve Qwen3.5-35B-A3B in BF16. While modest compared to model training, large-scale multi-agent evaluations have environmental costs that should be considered.

Benchmark integrity: All datasets used (HotpotQA, MuSiQue, 2WikiMultiHopQA, Collab-Overcooked) are publicly available under permissive licenses. No private or sensitive data was used.

Reproducibility Statement

Section 3 specifies the model endpoints, temperature, seeds, task sampling, and evidence-distribution protocols; the appendix provides full experiment results and Trust Framing prompt templates. The supplementary materials provide the core adaptor implementations, configurations, prompt descriptions, and evaluation utilities. The orchestration runner and raw experiment outputs are not included in that package and are planned for release upon publication. The paper and appendix report the statistical tests, confidence intervals, seed checks, and compute resources used.

References

Anthropic. Claude Opus 4.6 System Card. System card, February 2026. URL https://www.anthropic.com/claude-opus-4-6-system-card.

Kenneth J. Arrow. *The Limits of Organization*. W. W. Norton & Company, New York, 1974. ISBN 9780393093230. URL https://wwnorton.com/books/The-Limits-of-Organization/.

Vasilisa Bashlovkina, Zhaobin Kuang, Riley Matthews, Edward Clifford, Yennie Jun, William W. Cohen, and Simon Baumgartner. Trusted Source Alignment in Large Language Models. *CoRR*, abs/2311.06697, 2023. doi: 10.48550/ARXIV.2311.06697. URL https://doi.org/10.48550/ARXIV.2311.06697.

Ines Besrour, Jingbo He, Tobias Schreieder, and Michael Färber. RAGentA: Multi-Agent Retrieval-Augmented Generation for Attributed Question Answering. *CoRR*, abs/2506.16988, 2025. doi: 10.48550/ARXIV.2506.16988. URL https://doi.org/10.48550/ARXIV.2506.16988.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya G. Parameswaran, Dan Klein, Kannan Ramchandran,

Matei A. Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why Do Multi-Agent LLM Systems Fail? In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025.*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/b1041e52d3be19f0a9bc491657488e4a-Abstract-Datasets_and_Benchmarks_Track.html.

Lingjiao Chen, Matei Zaharia, and James Zou. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=cSimKw5p6R.

Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 2 edition, 1988. ISBN 9780805802832.

DeepSeek-AI. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence. *CoRR*, abs/2606.19348, 2026. doi: 10.48550/ARXIV.2606.19348. URL https://doi.org/10.48550/ARXIV.2606.19348.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving Factuality and Reasoning in Language Models through Multiagent Debate. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Jacob Eisenstein, Fantine Eri Huot, Adam Fisch, Jonathan Berant, and Mirella Lapata. MT-PINGEVAL: Evaluating Multi-Turn Collaboration with Private Information Games. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=Dvb7Dt8raw.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2137–2145, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/c7635bfd99248a2cdef8249ef7bfbef4-Abstract.html.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling Large Language Models to Generate Text with Citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 6465–6488. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.398. URL https://doi.org/10.18653/V1/2023.EMNLP-MAIN.398.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large Language Model Based Multi-agents: A Survey of Progress and Challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 8048–8057. ijcai.org, 2024. URL https://www.ijcai.org/proceedings/2024/890.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.580. URL https://doi.org/10.18653/V1/2020.COLING-MAIN.580.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Gautier Izacard and Edouard Grave. Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pp. 874–880. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EACL-MAIN.74. URL https://doi.org/10.18653/V1/2021.EACL-MAIN.74.

Alon Jacovi, Ana Marasovic, Tim Miller, and Yoav Goldberg. Formalizing Trust in Artificial Intelligence: Prerequisites, Causes and Goals of Human Trust in AI. In *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pp. 624–635. ACM, 2021. doi: 10.1145/3442188.3445923. URL https://doi.org/10.1145/3442188.3445923.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 13358–13376. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.825. URL https://doi.org/10.18653/V1/2023.EMNLP-MAIN.825.

John D. Lee and Katrina A. See. Trust in Automation: Designing for Appropriate Reliance. *Hum. Factors*, 46(1):50–80, 2004. doi: 10.1518/HFES.46.1.50.30392. URL https://doi.org/10.1518/HFES.46.1.50.30392.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Yuxuan Li, Aoi Naito, and Hirokazu Shirado. Systematic Failures in Collective Reasoning under Distributed Information in Multi-Agent LLMs. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=igHBKQaLLP.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 17889–17904. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.992. URL https://doi.org/10.18653/V1/2024.EMNLP-MAIN.992.

Bohan Lin, Kuo Yang, Zelin Tan, Yingchuan Lai, Chen Zhang, Guibin Zhang, Xinlei Yu, Miao Yu, Xu Wang, Yu-Dong Zhang, and Yang Wang. AgentAsk: Multi-Agent Systems Need to Ask. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 28055–28077. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1294. URL https://doi.org/10.18653/V1/2026.ACL-LONG.1294.

Niklas Luhmann. *Trust and Power*. John Wiley & Sons, Chichester, 1979.

Niklas Luhmann. *Theories of Distinction: Redescribing the Descriptions of Modernity*. Cultural Memory in the Present. Stanford University Press, Stanford, CA, 2002. ISBN 9780804741231.

Poornima Madhavan and Douglas A. Wiegmann. Effects of Information Source, Pedigree, and Reliability on Operator Interaction With Decision Support Systems. *Hum. Factors*, 49(5):773–785, 2007. doi: 10.1518/001872007X230154. URL https://doi.org/10.1518/001872007X230154.

D. Harrison McKnight, Larry L. Cummings, and Norman L. Chervany. Initial Trust Formation in New Organizational Relationships. *Academy of Management Review*, 23(3):473–490, 1998. doi: 10.5465/amr.1998.926622.

Abhijnan Nath, Hannah VanderHoeven, and Nikhil Krishnaswamy. CRAFT: Grounded Multi-Agent Coordination Under Partial Information. *CoRR*, abs/2603.25268, 2026. doi: 10.48550/ARXIV.2603.25268. URL https://doi.org/10.48550/ARXIV.2603.25268.

Thang Nguyen, Peter Chin, and Yu-Wing Tai. MA-RAG: Multi-Agent Retrieval-Augmented Generation via Collaborative Chain-of-Thought Reasoning. *CoRR*, abs/2505.20096, 2025. doi: 10.48550/ARXIV.2505.20096. URL https://doi.org/10.48550/ARXIV.2505.20096.

Jingjie Ning, Xueqi Li, and Chengyu Yu. Revision or Re-Solving? Decomposing Second-Pass Gains in Multi-LLM Pipelines. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=4WUqIrSACL.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to Route LLMs from Preference Data. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=8sSqNntaMr.

OpenAI. Update to GPT-5 System Card: GPT-5.2. System card, December 2025. URL https://openai.com/index/gpt-5-system-card-update-gpt-5-2/.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data Distillation for Efficient and Faithful Task-Agnostic Prompt Compression. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pp. 963–981. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.57. URL https://doi.org/10.18653/V1/2024.FINDINGS-ACL.57.

Raja Parasuraman and Victor Riley. Humans and Automation: Use, Misuse, Disuse, Abuse. *Hum. Factors*, 39(2):230–253, 1997. doi: 10.1518/001872097778543886. URL https://doi.org/10.1518/001872097778543886.

Esther Oluffa Pedersen. A two-level theory of trust. *Balkan Journal of Philosophy*, 2(1):47–56, 2010. doi: 10.5840/bjp20102115.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative Agents for Software Development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 15174–15186. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.810. URL https://doi.org/10.18653/V1/2024.ACL-LONG.810.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling Large Language Model-based Multi-Agent Collaboration. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=K3n5jPkrU6.

Qwen Team. Qwen3.5: Towards Native Multimodal Agents. Blog post, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Max Schemmer, Niklas Kühl, Carina Benz, Andrea Bartos, and Gerhard Satzger. Appropriate Reliance on AI Advice: Conceptualization and the Effect of Explanations. In *Proceedings of the 28th International Conference on Intelligent User Interfaces, IUI 2023, Sydney, NSW, Australia, March 27-31, 2023*, pp. 410–422. ACM, 2023. doi: 10.1145/3581641.3584066. URL https://doi.org/10.1145/3581641.3584066.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role play with large language models. *Nat.*, 623:493–498, 2023. doi: 10.1038/S41586-023-06647-8. URL https://doi.org/10.1038/S41586-023-06647-8.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards Understanding Sycophancy in Language Models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=tvhaxkMKAn.

Dan Sperber, Fabrice Clément, Christophe Heintz, Olivier Mascaro, Hugo Mercier, Gloria Origgi, and Deirdre Wilson. Epistemic Vigilance. *Mind & Language*, 25(4):359–393, 2010. doi: 10.1111/j.1468-0017.2010.01394.x.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning Multiagent Communication with Backpropagation. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2244–2252, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/55b1927fdafef39c48e5b73b5d61ea60-Abstract.html.

Haochen Sun, Shuwen Zhang, Lujie Niu, Lei Ren, Hao Xu, Hao Fu, Fangkun Zhao, Caixia Yuan, and Xiaojie Wang. Collab-Overcooked: Benchmarking and Evaluating Large Language Models as Collaborative Agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 4922–4951. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.249. URL https://doi.org/10.18653/V1/2025.EMNLP-MAIN.249.

Qiushi Sun, Zhangyue Yin, Xiang Li, Zhiyong Wu, Xipeng Qiu, and Lingpeng Kong. Corex: Pushing the Boundaries of Complex Reasoning through Multi-Model Collaboration. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=7BCmIWVT0V.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. ♫ MuSiQue: Multihop Questions via Single-hop Question Composition. *Trans. Assoc. Comput. Linguistics*, 10:539–554, 2022. doi: 10.1162/TACL_A_00475. URL https://doi.org/10.1162/TACL_A_00475.

Hannah VanderHoeven, Abhijnan Nath, and Nikhil Krishnaswamy. Flout at Your Own Risk: LLMs Struggle with Pragmatic Cooperativity Under Epistemic Asymmetry. *CoRR*, abs/2607.11053, 2026. doi: 10.48550/ARXIV.2607.11053. URL https://doi.org/10.48550/ARXIV.2607.11053.

Doan Nam Long Vu and Simone Balloccu. The Scaffold Effect: How Prompt Framing Drives Apparent Multimodal Gains in Clinical VLM Evaluation. *CoRR*, abs/2603.28387, 2026. doi: 10.48550/ARXIV.2603.28387. URL https://doi.org/10.48550/ARXIV.2603.28387.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. *CoRR*, abs/2404.13208, 2024. doi: 10.48550/ARXIV.2404.13208. URL https://doi.org/10.48550/ARXIV.2404.13208.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. Mixture-of-Agents Enhances Large Language Model Capabilities. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=h0ZfDIrj7T.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022.*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and

Chi Wang. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations. In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

Andrea Wynn, Harsh Satija, and Gillian Hadfield. Talk Isn't Always Cheap: Understanding Failure Modes in Multi-Agent Debate. *CoRR*, abs/2509.05396, 2025. doi: 10.48550/ARXIV.2509.05396. URL https://doi.org/10.48550/ARXIV.2509.05396.

Zijie Xu, Minfeng Qi, Shiqing Wu, Lefeng Zhang, Qiwen Wei, Han He, and Ningran Li. The Trust Paradox in LLM-Based Multi-Agent Systems: When Collaboration Becomes a Security Vulnerability. *CoRR*, abs/2510.18563, 2025. doi: 10.48550/ARXIV.2510.18563. URL https://doi.org/10.48550/ARXIV.2510.18563.

Chenfei Yan, Zeyang Yue, Feifei Zhao, Erliang Lin, Lu Jia, Haibo Tong, Mingyang Lyu, Chengyi Sun, and Yi Zeng. When Truth Is Distributed: Misinformation Derails Collective Fact Recovery in LLM-Based Multi-Agent Systems. *CoRR*, abs/2608.03421, 2026. doi: 10.48550/ARXIV.2608.03421. URL https://doi.org/10.48550/ARXIV.2608.03421.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1259. URL https://doi.org/10.18653/V1/D18-1259.

Yixiang Yao, Yuhang Yao, Xinyi Fan, Jiechao Gao, Jie Wang, Minjia Zhang, Srivatsan Ravi, and Carlee Joe-Wong. Trustworthy Agent Network: Trust in Agent Networks Must Be Baked In, Not Bolted On. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD 2026, Jeju Island, Korea, August 9-13, 2026*, pp. 13252–13258. ACM, 2026. doi: 10.1145/3770855.3818655. URL https://doi.org/10.1145/3770855.3818655.

Zhangyue Yin, Qiushi Sun, Cheng Chang, Qipeng Guo, Junqi Dai, Xuanjing Huang, and Xipeng Qiu. Exchange-of-Thought: Enhancing Large Language Model Capabilities through Cross-Model Communication. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 15135–15153. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.936. URL https://doi.org/10.18653/V1/2023.EMNLP-MAIN.936.

Ruiwen Zhou, Maojia Song, Xiaobao Wu, Sitao Cheng, Xunjian Yin, Yuxi Xie, Zhuoqun Hao, Wenyue Hua, Liangming Pan, Soujanya Poria, and Min-Yen Kan. Epistemic Context Learning: Building Trust the Right Way in LLM-Based Multi-Agent Systems. *CoRR*, abs/2601.21742, 2026. doi: 10.48550/ARXIV.2601.21742. URL https://doi.org/10.48550/ARXIV.2601.21742.

AI USE STATEMENT

GPT-5.6-Sol generated the five corrupted-message variants from prespecified definitions, and Claude Opus 5 reviewed them. AI tools were also used for limited wording edits and code-completion suggestions. The authors developed the conceptual framework and experimental design, conducted and interpreted the experiments, reviewed all AI-assisted outputs, and take responsibility for this work.

A Extended Results

A.1 Information Neglect: Automated Classification and Human Audit

Section 4.1 reports a two-reviewer human audit of receiver-blindness prevalence; this appendix documents the earlier automated classification that motivated that audit and explains why the two estimates diverge. The initial analysis of the 21,600 PT runs on the full matrix used a text-pattern classifier applied to the 5,443 zero-score runs (25.2% of all PT runs), sorting each zero-score output into one of three automated categories: *implicit* (a confident answer derived solely from the receiver's own evidence partition, with no textual acknowledgment that a partner message exists), *explicit* (the output states outright that partner information is missing or awaited), or *other* (meta-commentary, refusals, and unparseable outputs). Across the full sample the classifier assigned 70.8% of zero-score runs to implicit blindness, 26.1% to explicit blindness, and 3.1% to other, with the implicit/explicit split varying by task family:

- WikiMHop: implicit 58.3%, explicit 35.9%
- HotpotQA: implicit 66.6%, explicit 31.1%
- MuSiQue: implicit 82.2%, explicit 16.7%

Because the two automated categories jointly covered 96.9% of zero-score runs, this breakdown motivated the initial estimate that receiver blindness accounted for roughly 97.0% of zero-score PT failures. That estimate does not survive human review. Section 4.1 reports a subsequent two-reviewer human audit (a stratified sample of n=460 zero-score cases, coded against a 5-category codebook, 83.5% pre-adjudication agreement, Cohen's $\kappa = 0.78$), which provides the validated estimate (51.3% [46.7, 55.8]) of receiver blindness used throughout the main text as the audit's operational category for receiver-side information neglect. The discrepancy arises because the automated classifier's "implicit" category is a residual catch-all: any zero-score output that does not explicitly mention a missing partner defaults to implicit blindness, which over-attributes sender disclosure errors, answer-extraction failures, and ordinary reasoning mistakes to the receiver. We report the automated breakdown here for methodological transparency; it should not be read as an independent confirmation of the 51.3% audited figure.

A.2 Message Modification Strategy Details

Fixed Transformation Rules (L1). We test four deterministic transformations (summarization, aggressive compression, structured key-fact extraction, and profile-adapted rephrasing) across the initial 7 cross-model pairs and 3 task families (100 tasks per family per pair; 2,100 paired runs per strategy). At this scale, results are mixed: summarized (+2.6 pp, p=0.056; ns), compressed (+4.7 pp, p<0.001), structured extraction (−0.9 pp, ns), and profile-adapted (+6.7 pp, p<0.001). However, both apparently positive strategies reverse at larger scale: compressed messaging becomes significantly negative (−3.2 pp, p<0.001) when scaled to 16 pairs and 200 tasks per family (Table 1), and profile-adapted rephrasing becomes significantly negative (−3.6 pp, p=0.002) in a dedicated 8-pair replication (2,400 runs). This reversal demonstrates that small-scale positive signals for content modification are sampling artifacts that do not survive scaling.

Content-Aware Routing (L2). An adaptive router extracts message features (length, numeric density, list structure) and selects the best L1 strategy via a decision tree. The aggregate result is null (Δ=−0.2 pp, p=0.86; same 7-pair scale), indicating that this routing policy over the tested substrategies does not produce a gain.

Memory-Augmented Adaptation (L3). The most complex strategy maintains an experience bank of 2,160 transformation outcomes and retrieves the top-k relevant experiences to construct a dynamic adaptation prompt for a Qwen3.5-35B-A3B adaptor LLM. Version 1 produces catastrophic message destruction: the adaptor's chain-of-thought bleeds into the output, replacing evidence with meta-reasoning (e.g., "Let me check the token count…"), yielding Δ=−5.5 pp (p=0.12; ns; 3 pairs, 270 runs). Version 2 adds safety fallbacks (passthrough when confidence < 0.4) and filters the experience bank to 219 high-quality entries, producing a mild but non-significant gain (Δ= + 2.8 pp, p=0.43; 3 pairs, 270 runs). A larger v3 rerun (same 7-pair scale) removes the thinking-mode con-

found and converges to null ($\Delta{=}-0.4$ pp, $p{=}0.79$), indicating that the tested memory-augmented variant does not provide a robust alternative to TF.

A.3 Per-Model-Pair Results

Table 4 provides complete per-pair effects for TF across all task families.

Table 4: **Per-pair F1 change for TF vs. passthrough.** Aggregated across 3 task families. 33 of 36 pairs show numerically positive Δ (28 statistically significant at $p{<}.05$, 5 positive but non-significant); the remaining 3 have negative point estimates (one significant and two non-significant). Paired p values compare TF against passthrough.

Sender	Receiver	Δ (pp)	p
Same-model pairs (6)			
DS-V4-Flash	DS-V4-Flash	+22.3	<.001
GPT-5-Nano	GPT-5-Nano	+17.3	<.001
DS-V4-Pro	DS-V4-Pro	+15.9	<.001
GPT-5.2	GPT-5.2	+12.9	<.001
Qwen3.5-35B-A3B	Qwen3.5-35B-A3B	−2.3	≥ .05
Claude 4.6	Claude 4.6	−6.9	<.001
Cross-model pairs (30)			
DS-V4-Flash	GPT-5-Nano	+19.6	<.001
DS-V4-Flash	DS-V4-Pro	+19.1	<.001
Claude 4.6	DS-V4-Flash	+17.1	<.001
Claude 4.6	GPT-5-Nano	+16.7	<.001
Qwen3.5-35B-A3B	DS-V4-Pro	+16.0	<.001
DS-V4-Pro	DS-V4-Flash	+15.2	<.001
DS-V4-Flash	GPT-5.2	+14.5	<.001
GPT-5-Nano	DS-V4-Flash	+14.0	<.001
Qwen3.5-35B-A3B	DS-V4-Flash	+13.4	<.001
Qwen3.5-35B-A3B	GPT-5-Nano	+12.2	<.001
GPT-5.2	GPT-5-Nano	+11.6	<.001
GPT-5-Nano	DS-V4-Pro	+10.3	<.001
Claude 4.6	DS-V4-Pro	+10.3	<.001
GPT-5.2	DS-V4-Flash	+9.3	<.001
GPT-5-Nano	GPT-5.2	+9.0	<.001
DS-V4-Pro	GPT-5-Nano	+9.0	<.001
GPT-5.2	DS-V4-Pro	+8.4	<.001
GPT-5-Nano	Claude 4.6	+8.2	<.001
DS-V4-Pro	GPT-5.2	+6.8	<.001
Qwen3.5-35B-A3B	Claude 4.6	+6.1	<.001
Claude 4.6	GPT-5.2	+6.0	<.001
GPT-5.2	Claude 4.6	+5.9	<.001
DS-V4-Flash	Claude 4.6	+5.7	<.001
DS-V4-Pro	Claude 4.6	+5.0	<.001
Qwen3.5-35B-A3B	GPT-5.2	+2.0	≥ .05
DS-V4-Flash	Qwen3.5-35B-A3B	+1.9	≥ .05
GPT-5.2	Qwen3.5-35B-A3B	+1.2	≥ .05
DS-V4-Pro	Qwen3.5-35B-A3B	+0.9	≥ .05
Claude 4.6	Qwen3.5-35B-A3B	+0.5	≥ .05
GPT-5-Nano	Qwen3.5-35B-A3B	−0.2	≥ .05

A.4 Statistical Power Analysis

Post-hoc power analysis (Cohen, 1988) confirms our experiments are well-powered:

- For the overall paired effect: achieved power > 0.99 across 21,600 matched task$\times$pair contrasts (43,200 outcome runs across PT and TF).
- For per-family effects ($\Delta{=}+8.0$ to $+11.8$ pp): achieved power ≥ 0.99 across 7,200 matched contrasts per family (14,400 outcome runs across PT and TF).

- The 95% BCa bootstrap CI for the overall effect, based on 10,000 resamples, is [+9.1, +10.0] pp.

A.5 Chain Break Prevention and Harm Mitigation

Table 5 reports zero-score reductions, and Table 6 compares always-on TF with mitigation strategies on a seven-pair subset.

Table 5: **Chain break prevention: zero-rate reduction by task family** (main experiment).

Family	PT zero-rate	TF zero-rate	Reduction
HotpotQA (2-hop)	15.9%	10.4%	−5.5 pp
WikiMHop (2-hop)	34.1%	26.1%	−8.0 pp
MuSiQue (stratified 2/3/4-hop)	25.6%	18.2%	−7.4 pp

Table 6: **Mitigation strategies reduce net benefit** (7 cross-model pairs, $7 \times 200 \times 3 = 4{,}200$ runs per strategy; note TF mean F1 differs from the 36-pair main experiment due to subset selection). Always-on TF remains optimal.

Strategy	Mean F1	Δ vs. TF	Mechanism
TF (always-on)	.581	—	Full protocol
Risk-aware gated	.562	−1.9 pp	Skip TF on high-risk
Receiver-adaptive	.559	−2.2 pp	Model-specific selection
Three-tier cascade	.550	−3.1 pp	Multi-level routing
Lite	.531	−5.0 pp	Minimal preamble

A.6 Full Strategy Comparison

Figure 4 summarizes the F1-based strategy comparisons at their largest evaluated scales.

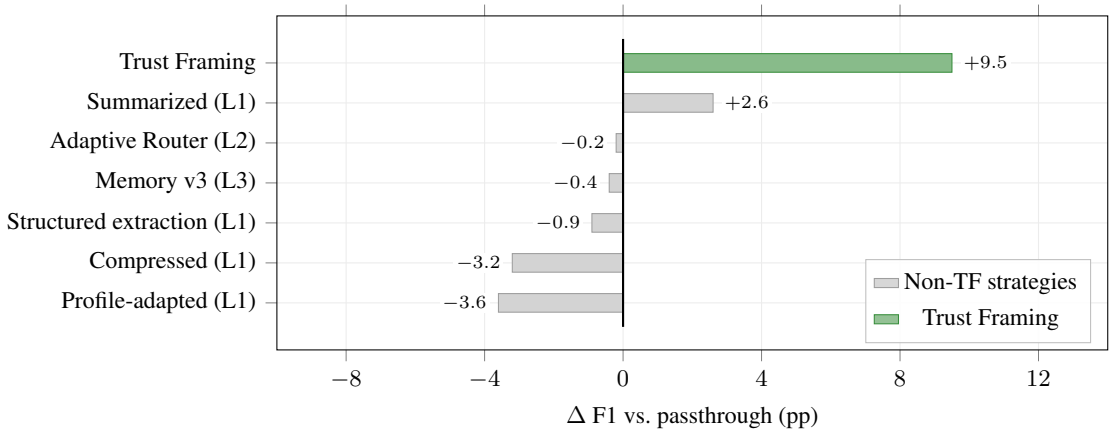


Figure 4: **Strategy comparison at the largest available scale.** Six non-TF F1 strategies from Table 1 are shown at their largest-scale evaluation; IF is omitted because its outcome is final-answer accuracy. Trust Framing is the 36-pair main experiment (43,200 runs). Trust Framing has the largest positive F1 gain among the strategies shown.

A.7 Pair Configuration Grid

Table 7 enumerates all 36 directed sender→receiver configurations evaluated in the unified main experiment. The 6 same-model pairings (diagonal, marked **S**) come from *ef_homogeneous_v1* (4 pairs) and *ef_homogeneous_deepseek_v1* (2 pairs). The 30 cross-model pairings come from *batch_A*

(8 pairs, marked **A**), *batch_B* (8 pairs, marked **B**), *batch_C* (10 pairs, marked **C**), and *batch_D* (4 pairs, marked **D**).

Table 7: **36-pair sender×receiver configuration grid. S** = same-model (homogeneous); **A** = cross-model batch 1; **B** = cross-model batch 2; **C** = coverage extension batch 1; **D** = coverage extension batch 2.

Sender↓ Recv.→	GPT-5.2	Claude 4.6	Nano	Qwen3.5	DS-V4-Pro	DS-V4-Flash
GPT-5.2	**S**	**A**	**A**	**A**	**B**	**B**
Claude 4.6	**C**	**S**	**A**	**A**	**B**	**B**
GPT-5-Nano	**C**	**D**	**S**	**C**	**D**	**C**
Qwen3.5-35B	**A**	**A**	**A**	**S**	**B**	**B**
DS-V4-Pro	**C**	**D**	**B**	**C**	**S**	**C**
DS-V4-Flash	**B**	**D**	**C**	**C**	**C**	**S**

A.8 Evidence Distribution Protocols

For each task family, evidence is distributed as follows:

HotpotQA: Each question has 10 context paragraphs (2 supporting + 8 distractors). Supporting paragraph 1 $\rightarrow$ Agent A; supporting paragraph 2 $\rightarrow$ Agent B; distractors split evenly.

MuSiQue: For a k-hop question, Agent A receives evidence for hops $1, \ldots, \lfloor k/2 \rfloor$; Agent B receives hops $\lfloor k/2 \rfloor + 1, \ldots, k$. Distractors are split evenly. Concretely: 2-hop questions are split $1|1$ (one supporting paragraph per agent); 3-hop questions are split $1|2$ (Agent A holds hop 1, Agent B holds hops 2–3); 4-hop questions are split $2|2$ (Agent A holds hops 1–2, Agent B holds hops 3–4). Because Agent B must integrate its own evidence with whatever Agent A communicates, the asymmetric $1|2$ split for 3-hop questions makes the receiver's task strictly harder than the symmetric splits, which is reflected in the lower baselines observed at higher hop counts.

2WikiMultiHopQA: Evidence forms a triple chain $(s_1, r_1, e_{\text{bridge}})$ and $(e_{\text{bridge}}, r_2, \text{answer})$. Agent A receives passages for the first triple; Agent B for the second. The bridge entity is the critical link that must be communicated.

A.9 TF Preamble Templates

The full preamble templates, including task-family-specific hints, are provided below for reproducibility.

HotpotQA hint: "*This is a 2-hop bridge question. Find the BRIDGE ENTITY that appears in BOTH this evidence and your own paragraphs. It connects the reasoning chain.*"

MuSiQue hint: "*This is a multi-hop question with chained reasoning. The prior steps of the chain are provided in the received message. Find where that chain ends (an intermediate answer or entity) and continue from there using your own paragraphs.*"

2WikiMultiHopQA hint: "*This is a multi-hop question. Find entities that appear in BOTH this evidence and your own paragraphs. They are the links in the reasoning chain.*"

A.10 Supplementary Ablation Details

This appendix reports supplementary content-modification controls and the evidence-partition scaling experiment.

Content-modification controls (*adaptor_clean_v1*, *compressed_bridge_safe_v1*). Two further single-class controls confirm that content rewriting underperforms framing. In *adaptor_clean_v1* (3 pairs, 450 runs per strategy), compressed messaging is -1.9 pp (ns), profile-adapted is -0.4 pp

(ns), while TF is $+11.3$ pp ($p<0.001$). In *compressed_bridge_safe_v1* (3 pairs, 449 runs per strategy), standard compression is $+2.1$ pp (ns), a bridge-safe variant is $+2.6$ pp (ns), and TF is $+13.8$ pp ($p<0.001$).

Evidence-partition scaling (16,410 runs). On the same MuSiQue 3-hop and 4-hop questions, we vary the number of evidence partitions N to control per-message information density. For an X-hop question with N partitions, the X supporting paragraphs are divided into N contiguous chunks assigned alternately to Agent A and Agent B; each agent's system prompt labels evidence by round and instructs it to share only the current round's chunk. Agent A initiates, and the pair exchanges messages for $\lceil N/2 \rceil$ full rounds (i.e., 1 round at $N{=}2$, 2 rounds at $N{=}3$ or $N{=}4$). At $N{=}2$ (single-pass baseline), the setup reduces to the standard midpoint split. The design covers $N \in \{2, 3\}$ for 3-hop and $N \in \{2, 3, 4\}$ for 4-hop, using 3 sender→GPT-5-Nano pairs (Claude Opus 4.6, GPT-5.2, Qwen3.5-35B-A3B).

Table 8: **Evidence-partition scaling on MuSiQue (16,410 runs).** Finer partitioning ($N{>}2$) distributes evidence across more messages, reducing per-message information density. Within each hop group, TF gain decreases as N increases, with the largest gain at $N{=}2$.

	3-hop		4-hop		
Metric	$N{=}2$	$N{=}3$	$N{=}2$	$N{=}3$	$N{=}4$
Comm. rounds	1	2	1	2	2
n (runs)	4,560	4,560	2,430	2,430	2,430
PT F1	.270	.350	.205	.340	.191
TF F1	.537	.416	.429	.416	.238
Δ F1 (pp)	+26.7	+6.6	+22.5	+7.6	+4.7
p	$< .001$	$< 10^{-7}$	$< .001$	$< 10^{-5}$	$< .001$

The nominal evidence share per chunk is $1/N$: 50% at $N{=}2$, 33% at $N{=}3$, and 25% at $N{=}4$. These are averages across chunks; integer paragraph counts make individual chunks unequal when N does not divide the hop count (e.g., a 1/2-paragraph split for 3-hop questions at $N{=}2$). When a single message is information-rich, the receiver faces a high-stakes integration challenge. TF's provenance framing and integration scaffold directly address this, yielding $+22$–27 pp. As N increases and information spreads across multiple smaller messages, each becomes easier to process without explicit framing, reducing TF's marginal benefit to $+5$–8 pp.

A.11 Representative Case Studies

Table 9 presents four cases where TF rescues a passthrough failure (F1 rises to 1.0) and one failure-to-rescue case, illustrating the mechanism and its boundary conditions.

The rescue cases illustrate both forms of receiver blindness (RB; Section 4.1): explicit ("waiting for partner's evidence") and implicit (confident single-source answers). In the failure-to-rescue case, both PT and TF score 0, so TF does not resolve the failure. Separately, 3 of 36 configurations (8.3%) have negative point estimates; two are non-significant and one is significantly negative.

A.12 Collab-Overcooked: Pair-Level Results

Table 10 reports pair-level success rates on all 30 recipes and on the L5–L6 patty subset, supporting the sender- and receiver-dependent claims in Section 5.6.

A.13 Collab-Overcooked: Coordination Rescue Case

Table 11 presents a representative rescue case from the Collab-Overcooked domain, where TF resolves a coordination breakdown on a complex patty recipe (L5). Unlike QA, the receiver-side failure here is procedural: the receiver repeatedly requests task guidance from the sender but does not act on the sender's replies.

The TF preamble appended to the dialogue channel in this setting is:

Table 9: **Representative rescue and failure-to-rescue cases.** "PT answer" is the receiver's final output under passthrough; the last two columns compare PT and TF F1. Gold answers are ground-truth references. RB denotes receiver blindness, the audit category for receiver-side information neglect.

Task	Pair	Failure mode	PT answer	Gold	PT F1	TF F1
Rescue cases (TF corrects passthrough failure):						
wikimhop_94	GPT→Nano	Explicit RB	"waiting for partner's evidence"	Chengdu	0.0	1.0
wikimhop_40	Qwen→GPT	Implicit RB	Nerja, Spain	Sweden	0.0	1.0
hotpotqa_73	Claude→Qwen	Bridge entity missed	Hips Don't Lie	Rudebox	0.0	1.0
wikimhop_97	Qwen→Nano	Partial integration	"Gaps: None of my paragraphs specify..."	Illinois	0.06	1.0
Failure-to-rescue case:						
musique_11	Claude→Nano	Unresolved failure	"My partner likely holds evidence about..."	1932	0.0	0.0

Table 10: **Pair-level Collab-Overcooked results.** Binary task-completion rate (%) over 5 rollouts × 30 recipes (All) or 5 rollouts × 10 patty recipes (Patty). Aggregates computed from raw rollout outcomes across all pairs.

		All (30)		Patty (10)	
Sender	**Receiver**	**PT**	**TF**	**PT**	**TF**
GPT-5.2	DeepSeek	85.3	95.3	56.0	86.0
GPT-5.2	GPT-5.2	87.3	95.3	62.0	86.0
GPT-5.2	Qwen	66.7	82.7	12.0	48.0
DeepSeek	DeepSeek	85.3	95.3	60.0	86.0
DeepSeek	GPT-5.2	88.0	94.0	66.0	82.0
DeepSeek	Qwen	68.0	77.3	30.0	40.0
Qwen	DeepSeek	61.3	64.7	0.0	2.0
Qwen	GPT-5.2	65.3	66.0	2.0	2.0
Qwen	Qwen	50.0	55.3	0.0	0.0
Aggregate		73.0	80.7	32.0	48.0

```
[teammate-info] The following lines are the in-game dialogue
from your teammate {Chef}.  Treat them as task-relevant
teammate information.  Use your teammate's communicated
plans and observations to coordinate your next action and
avoid duplicating efforts.  They are NOT part of your output
template; do not quote or repeat them.  Continue to follow
the action format required by your role {Assistant}.
```

The trust component ("Treat them as task-relevant teammate information") directly addresses the failure mode by signaling that the preceding dialogue lines contain actionable instructions rather than background context to be acknowledged and re-requested.

A.14 Reproducibility

The main experiment uses temperature 0 and seed 42; each (pair, task family, strategy) cell contains 200 distinct tasks and is evaluated exactly once. The 43,200 outcome runs comprise 21,600 matched PT–TF task×pair contrasts. Robustness across seeds is verified separately (Section 5.2). The framework is implemented in Python 3.12 using AutoGen v0.4 (Wu et al., 2024). Models are accessed via API (GPT-5.2, Claude Opus 4.6) or self-hosted vLLM (Qwen3.5-35B-A3B served in BF16 on

Table 11: **Collab-Overcooked rescue case:** `mashed_cauliflower_and_lentil_patty`, GPT-5.2 (Chef) → DeepSeek-V4-Flash (Assistant). Under passthrough, the Assistant asks for the recipe 40+ times across 80 timesteps but never acts on the Chef's responses; under TF, the Assistant immediately executes the Chef's plan at T=0.

	Passthrough (SR = 0.0)	**Trust Framing (SR = 1.0)**
T=0	*Assistant:* "Chef, I don't have the recipe for mashed_cauliflower_and_lentil_patty. Could you please tell me the steps?"	*Chef:* "Please 1) pick up sweet_potato, put it on chopping_board0, and cut into slices; 2) then pick up bean and place on counter." *Assistant:* "I'll start by picking up sweet potato, placing it on the chopping board, and cutting it. Then I'll put the bean on the counter for you."
T=1–14	Assistant repeats the same request ("Chef, I still need the recipe…") each timestep; Chef's instructions are ignored.	Chef provides the full multi-step recipe; Assistant acknowledges and executes sequentially.
Result	0 orders completed (80-step timeout).	14 orders completed in 623 steps (recipe variants auto-generated).

two 64-GB MetaX XiYun C-series accelerator cards). Complete experimental logs, checkpoints, and analysis code will be released upon publication.