What Should Agents Say? Action-state Communication for Efficient Multi-Agent Systems

Anonymous ACL submission

Abstract

Multi-agent systems (MAS) built on large language models are typically organized around roles, pipelines, and turn schedules, while the content that agents pass to one another is often left as unconstrained natural language. However, this free-form communication can rapidly inflate token usage, consume the shared context window, and ultimately affect both system performance and inference cost. We analyze five common inter-agent communication strategies across two MAS topologies, finding that no fixed strategy is universally optimal. Instead, effective inter-agent messages consistently preserve action-centered information needed by downstream agents. Building on this, we propose the **PACT** (**P**rotocolized **A**ction-state **C**ommunication and **T**ransmission), which treats inter-agent communication as a public state-update problem and projects each raw agent output into a compact action-state record before it enters shared history. Across different MAS topologies, PACT consistently improves the performance–cost trade-off, achieving comparable or stronger task performance with substantially fewer tokens. The gains extend to production coding harnesses: PACT lifts OpenHands' resolve rate at -10% tokens-per-resolved, and is resolve-neutral on SWE-agent while halving input tokens.

1 Introduction

Large language models (LLMs) are increasingly deployed as multi-agent systems (MAS) (Hong et al., 2024; Cemri et al., 2026). In these systems, several agents collaborate to solve complex tasks that a single model call handles poorly (Wu et al., 2024; Chen et al., 2024; Li et al., 2023). Such collaboration can take many forms, including debate (Du et al., 2024; Li et al., 2024), sequential pipelines (Zhang et al., 2024; Zhao et al., 2025; Zou et al., 2025), retrieval-based relays (Chen et al., 2025), and tool-using agents (Yao et al., 2023). The

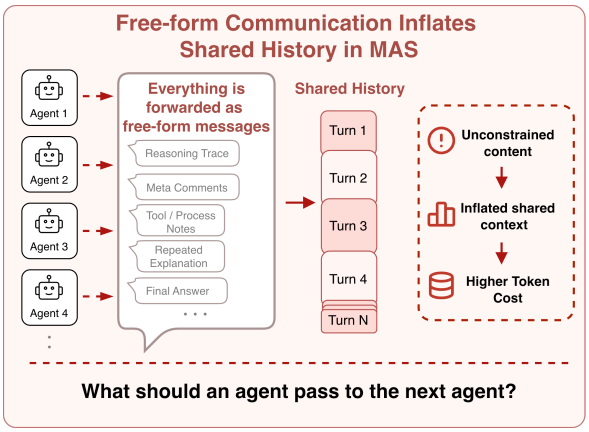

Figure 1: Free-form inter-agent messages accumulate in the shared history, forcing downstream agents to process unconstrained context and increasing token cost.

paradigm is already widely deployed in agentic coding assistants (Anthropic, 2026a; OpenAI, 2026a) and open agent platforms (Wang et al., 2025; Yang et al., 2024), decomposing a task across many cooperating model calls and outperform single-shot prompting. This effectiveness carries a cost: a MAS issues many model calls whose outputs feed one another, so it consumes far more tokens, often many times more, than a single model solving the same task (Bai et al., 2026; Yu et al., 2026).

This issue becomes especially severe when modern reasoning models are used in multi-turn settings (Anthropic, 2026b; OpenAI, 2026b; Team et al., 2026): once an agent forwards a long internal reasoning trace, every downstream agent must repeatedly process it as part of the shared dialogue history (Ramesh and Li, 2025). As shown in Figure 1, verbose or noisy messages are repeatedly reprocessed by downstream agents, causing token costs to compound across turns and potentially exhausting the context window before the task is completed, which can ultimately degrade performance (Li et al., 2024; Chen et al., 2025). However, existing MAS research has largely focused on roles, turn-taking schedules, and tool al-

location (Wu et al., 2024; Chen et al., 2024; Wang et al., 2024a; Qian et al., 2024), while leaving the inter-agent message itself as mostly unconstrained free-form natural language. We argue that the inter-agent message is a central design lever: **the content of messages between agents directly shapes both MAS performance and token cost**.

Specifically, we ask: *what should an agent send to another agent in a MAS*? We conduct a diagnostic analysis of inter-agent communication across two different MAS settings: a split-evidence interaction and a sequential pipeline. The analysis compares five common strategies: forwarding the agent's full free-form output, using native concise generation, keeping only the final conclusion, producing a short summary of the turn, and passing only the role artifact (§3). The analysis shows that no single strategy is uniformly best across topologies: forwarding the full output is expensive and redundant, generic shortening can remove task-relevant information, conclusion-only messages are too lossy when the receiver lacks surrounding state, and artifact-only messages capture useful task content but still leave the receiver to infer the sender's intended action and grounding evidence. We therefore view inter-agent communication as a public state-update problem: the message should not summarize the sender's private reasoning, but should update the shared state with the minimal action-relevant information needed by later agents.

Motivated by this observation, we introduce **PACT** (**P**rotocolized **A**ction-state **C**ommunication and **T**ransmission), a minimal and harness-agnostic communication protocol that projects each non-terminal agent's raw output into a compact public action-state message before it is appended to the shared history. Rather than exposing the sender's full generation transcript, PACT retains only the receiver-facing information needed for continuation: the action taken or required next, the task-relevant state, and the resulting artifact to be used downstream. Across model scales and MAS settings, PACT consistently reduces token usage while preserving and often improving task performance. We further port PACT to two real-world agentic coding harnesses, OpenHands (Wang et al., 2025) and SWE-agent (Yang et al., 2024). The same communication rule yields substantial reductions in token usage with minimal performance degradation, indicating PACT is also practical as an inference-time communication protocol for existing agentic applications.

Our contributions are summarized as follows:

- Through a systematic analysis of communication strategies in inter-agent message in MAS, we show that no fixed policy is universally optimal, indicating that what agents pass to one another is a central but underexplored design dimension.
- We introduce **PACT**, a training-free and harness-agnostic communication protocol for MAS, defining the boundary between private computation and public communication.
- Experiments on two MAS settings show that PACT substantially improves the performance–cost frontier, reducing token usage by 38.7% on average across baselines and model scales while preserving or improving task performance. On real-world coding harnesses, PACT reduces tokens-per-resolved by 47% on SWE-agent, demonstrating its practical significance.

2 Related Work

Multi-agent systems. LLM-based multi-agent systems (MAS) coordinate several model instances to solve tasks beyond a single call. Research has largely explored *who* talks and *when*: role specialisation and persona self-collaboration (Wang et al., 2024c; Li et al., 2023), peer debate and critique (Du et al., 2024; Liang et al., 2023), general multi-agent conversation frameworks (Wu et al., 2024; Chen et al., 2024), and parallel output aggregation (Wang et al., 2024a). These designs differ in roles or orchestration, but uniformly leave the content of each inter-agent message to free-form natural language. We examine through systematic experiments and show that the content one agent pass to the next matters in both performance and cost.

Token consumption in MAS. The token cost of multi-agent coordination arises from repeated context reuse: one agent's output becomes part of another agent's input, and later agents must reprocess earlier messages. This effect is especially pronounced when agents forward deliberative content, such as chain-of-thought reasoning (Wei et al., 2022), because the same reasoning traces are repeatedly read by downstream agents (Zeng et al., 2025). Prior work studies the *shape* of inter-agent messages and shows that disciplined communication affects accuracy and cost (Zou et al., 2025; Chen et al., 2025; Yu et al., 2026). While studies pay attention to who should the agent talk to under

various topologies (Zhang et al., 2025; Shen et al., 2025), what content should be communicated is largely ignored. Instead, PACT defines a communication invariant over shared history: only action, grounded state, and reusable result are made public, while private deliberation remains outside the inter-agent channel.

Agentic harnesses for real-world tasks. Beyond research scaffolds, agentic harnesses are increasingly built to solve real-world problems, most prominently software engineering (Hong et al., 2024; Wang et al., 2024b; Qian et al., 2024): SWE-bench (Jimenez et al., 2024) evaluates agents on real GitHub issues, and OpenHands (Wang et al., 2025) and SWE-agent (Yang et al., 2024) are widely used loops that interleave reasoning, tool calls, and environment feedback over long trajectories. Because such harnesses accumulate long tool-augmented histories, the per-turn message and shared-history budget can easily scale up. We show that passing the action-state related information only is enough to reach comparable results while significantly reducing the token usage.

3 Diagnostic Analysis of Inter-Agent Communication

Following the question raised in §1, we examine how an agent's output should be handled in the inter-agent message, isolating this single choice while holding all other MAS design factors fixed. We analyze five communication strategies that cover today's common choices, in two structurally different MAS settings, at various model scales.

3.1 Two MAS Settings

Setting A — Split-Evidence Interaction. We use a symmetric two-agent setting where the evidence needed to solve the task is split across agents. Each agent receives only partial information, so neither agent can reliably answer the question alone. The agents must exchange task-relevant evidence over multiple turns and combine their partial views to produce the final answer.

Setting B — Sequential pipeline. In this setting, the task is processed sequentially among the agents, each agent will take the output of previous agents as input and solve the task building on earlier intermediate artifacts. We adopt a widely studied role-specialised pipeline (Zou et al., 2025; Yu et al., 2026) of four agents run in fixed order—**Planner**, **Critic**, **Refiner**, and **Solver**. The Planner produces a plan, the Critic reviews it and emits a critique, the Refiner returns a revised plan, and the Solver produces the benchmark answer.

Models and settings. All experiments run on Qwen3-8B, Qwen3-14B, and Qwen3-32B (Yang et al., 2025). For the split-evidence interaction setting, we evaluate on HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020). Each answer depends on combining multiple supporting paragraphs and requires evidence exchange. Each question is supportted by ten paragraphs, including two gold supporting paragraphs and eight distractors, and evenly split them between two agents. For the sequential pipeline setting, we evaluate on three categories of benchmarks: (i) mathematical reasoning, including AIME2024 (Maxwell-Jia, 2024) and AIME2025 (math-ai, 2025); (ii) scientific reasoning, using GPQA-Diamond (Rein et al., 2024); and (iii) commonsense question answering, using OpenBookQA (Mihaylov et al., 2018). Other experiment details are deferred to Appendix A.

3.2 Inter-Agent Communication Strategies

We analyze five strategies that cover the common ways a MAS can pass an agent's reasoning and output on the inter-agent channel. **Full Content** appends the agent's freely generated output, including its internal reasoning trace, to the shared history. **Concise Generation** lets the model operate in its native non-thinking mode, which usually produces shorter responses. **Conclusion Only** passes the conclusion / final answer portion to the next agent. **Brief Summary** explicitly asks the agent to pass a short free-form summary to the next agent. **Artifact Only** reduces the message to the role artifact alone (e.g., the plan or critique), without the surrounding action description or supporting state.

3.3 From Action-Centered Messages to Structured Handoffs

Figure 2 shows that no common communication strategy is uniformly satisfactory across the two MAS settings. The useful signal in the inter-agent communication is usually action-centered, but it must also expose the state that makes the handoff reliable. These findings motivate a protocolized handoff that aligns the content of inter-agent messages and makes them compact and stable.

Passing the full content is redundant. Full Content forwards the entire free-form output, including

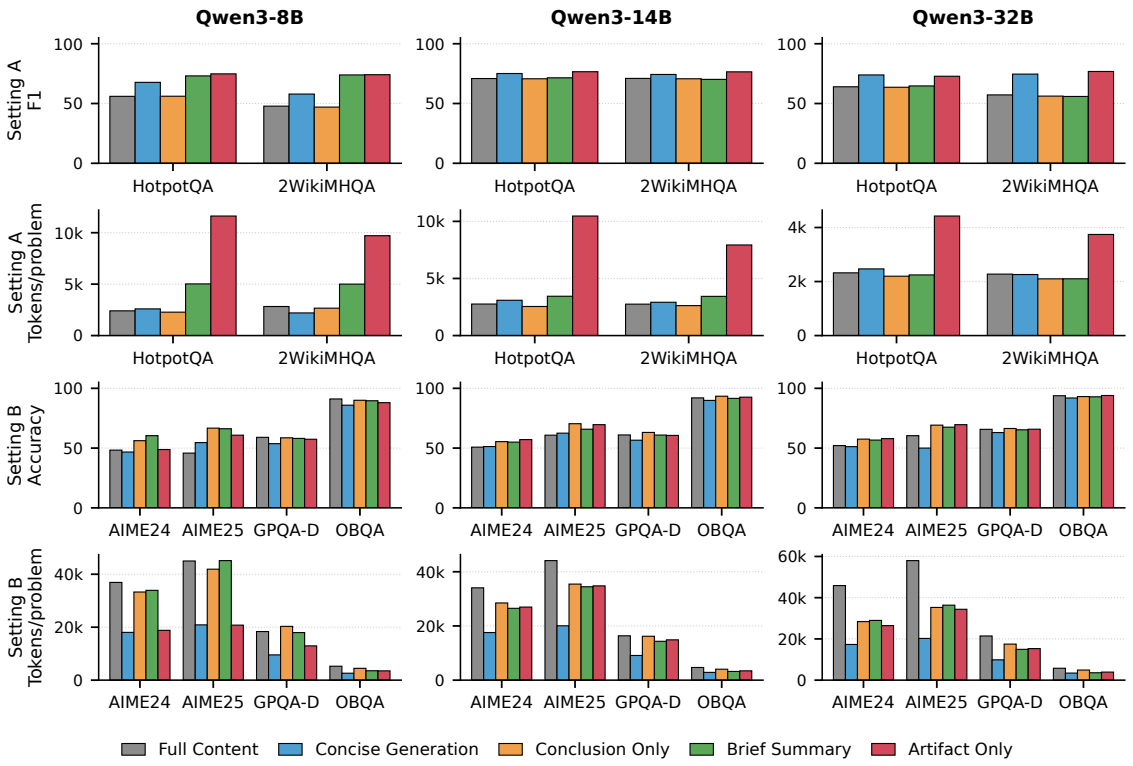

Figure 2: Five inter-agent communication strategies in two MAS settings at three model scales. Top two rows: Setting A interaction (F1 and tokens per problem); bottom two rows: Setting B pipeline (Accuracy and tokens).

the explanations, which is consistently expensive and rarely gives the best performance. In the sequential pipeline it has the largest token cost at every model scale, while being clearly behind the best-performing alternatives. In the interaction setting it is also not the strongest choice. These results show that forwarding the full transcript is an inefficient default: much of what is passed is not the information the receiver needs to act on.

Being short is not enough. Concise Generation reduces token usage by suppressing long reasoning traces, and it performs well in the interaction setting, where the receiver often only needs a surfaced fact. However, the same strategy is weak in the role-specialized pipeline, where intermediate agents must produce useful plans, critiques, and refinements before the final solver acts. Brief Summary gives a different kind of short message, but its behavior is unstable: it is strong for some 8B interaction and pipeline runs, yet falls behind simpler strategies at larger scales. These results suggest that generic conciseness is not a reliable communication rule: a message can be short while still omitting the information needed by the next agent.

Conclusion-only messages are topology-dependent. Keeping the visible final output while removing the intermediate traces works relatively well in the sequential pipeline, where the fixed role order already tells each agent how to interpret the previous output. In the interaction setting, however, Conclusion Only is much weaker, because the partner needs not only a local conclusion but also the evidence observed by the sender and the missing information still needed for the answer. Thus, removing thinking processes is helpful, but keeping only the conclusion is not a general solution: it depends on whether the surrounding MAS topology already supplies the missing action and state information.

Artifact-only messages identify the right content but not the right protocol. Artifact Only provides the clearest signal about what should be communicated. It often achieves the best in the interaction setting and is competitive in the sequential pipeline, showing that the receiver benefits most from the sender's action-related work product rather than from the full generation trace. However, Artifact Only is not yet a communication protocol. In the interaction setting it can achieve high accuracy only with very large messages. As shown in Figure 3, Artifact Only runs more turns than other strategies, indicating that suppressing all pream-

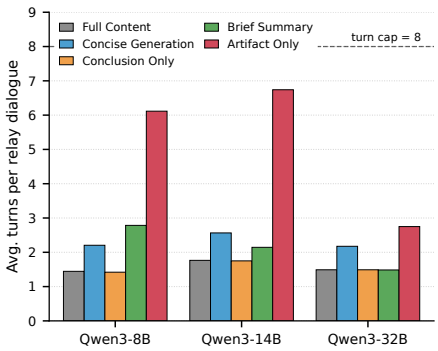

Figure 3: Average agent turns per interaction dialogue: *Artifact Only* runs more turns than other strategies, driving its $\sim 3\times$ token cost.

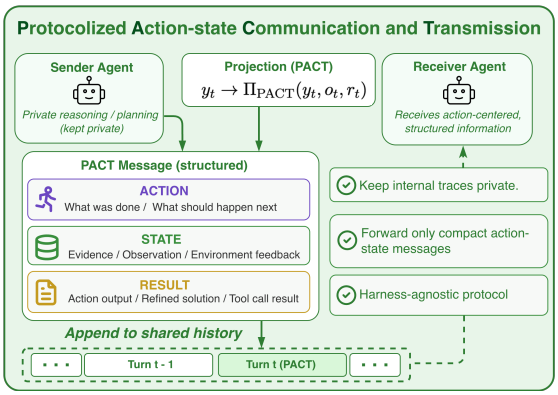

Figure 4: Overview of PACT

ble also suppresses the closure cues agents use to stop. In the pipeline, it is efficient and competitive at larger scales, but weaker for the smaller model, suggesting that an artifact alone may omit useful state or handoff information when the sender is less capable. The artifact is therefore the right starting point, but it must be made structured and compact.

4 PACT: Protocolized Action-state Communication and Transmission

The analysis result in §3 suggests that the central challenge in inter-agent communication is not merely message length, but the type of information that is preserved for downstream agents. Rather than exposing the sender's full generation transcript, an inter-agent message should communicate the sender's action-state: the action taken, the task-relevant state or evidence supporting it, and the result that the next agent needs in order to continue. Building on this observation, we propose **Protocolized Action-state Communication and Transmission** (**PACT**), shown in Figure 4, a communication protocol that restricts shared inter-agent history to compact action-state messages while excluding intermediate process-level content.

4.1 General setup

Consider a multi-agent system with a shared history H_t at turn t. An agent with role or action r_t receives its local observation o_t and the shared history H_t, then produces a raw output y_t. In a standard MAS, this raw output is directly appended to the shared history:

$$H_{t+1}^{\text{std}} = H_t^{\text{std}} \oplus y_t, \tag{1}$$

where $\oplus$ denotes appending a message. This update rule is simple, but it makes every part of y_t public, including reasoning traces, thought explanations, repeated statements, and the final artifact. These elements are all retained and re-read by later agents.

The idea of PACT is simple: before an agent output becomes shared history, we project it into a compact public message space. Formally, PACT replaces the raw message y_t and updates the history as

$$H_{t+1}^{\text{PACT}} = H_t^{\text{PACT}} \oplus \Pi_{\text{PACT}}(y_t, o_t, r_t). \tag{2}$$

Here Π_{PACT} is a sender-side projection that determines what information is allowed to enter the inter-agent channel.

4.2 Action-state message space

PACT represents each public inter-agent message as an action-state record with three receiver-facing fields: ACTION, STATE, and RESULT. ACTION states what the sender has done, or what the next agent should do; this makes the handoff explicit rather than leaving the receiver to infer the sender's intent. STATE records the evidence, observation, environment feedback, or tool result that grounds the message; this allows the receiver to consume the message as a grounded public state rather than an unsupported conclusion. RESULT contains the action output itself, such as a retrieved fact, a refined solution, or a tool-call result. Together, these fields define the information that is allowed to enter the shared inter-agent channel.

This message space also specifies the boundary of PACT. Each agent may still reason, self-check, or explore alternatives as needed to produce its output. The constraint applies only to what becomes part of the public inter-agent message space. Under PACT, a message retains only the information that is relevant to the action-state record: what action was taken, what state or evidence grounds it, and

what result should be passed downstream. Intermediate process content such as hidden reasoning traces or redundant restatements is excluded before the message is appended to the shared history. In this way, PACT separates private computation from public communication , preventing intermediate deliberation from compounding across later context windows.

4.3 Protocol properties

Since PACT is intended to be a protocol over the inter-agent channel, rather than a template tied to a particular MAS scaffold, it has the following properties. First, PACT does not constrain the agent's private computation. The agent can still perform internal reasoning and planning in its original manner; PACT only controls which part of the generated output is retained as public communication. This differs from methods that suppress reasoning or alter the agent's internal solving process and preserves the capability of the agent. Second, PACT is orthogonal to standard MAS design choices. It does not introduce new agents, change the turn schedule, train a compressor, or assume a particular role structure. The same action-state interface can be serialized differently across settings: a split-evidence interaction may write the action, state, and result explicitly, while a role-specialized pipeline may encode part of the action through the role schedule and retain a more compact role artifact. This protocol works for controlled MAS scaffolds and complex tool-using harnesses like agentic coding systems.

5 Experiments

5.1 Experiment setup

We evaluate on the same two MAS settings introduced in §3: the two-agent split-evidence interaction (Setting A) and the sequential pipeline (Setting B), over the same six benchmarks. All experiments use Qwen3-8B, Qwen3-14B, and Qwen3-32B (Yang et al., 2025). We compare PACT against three representative MAS communication paradigms. **Chain of Agents (CoA)** (Zhang et al., 2024) chains worker agents in sequence, with each agent reading only the most recent peer message rather than the full inter-agent history. **Text-based MAS (TextMAS)** (Zou et al., 2025) uses role-specialised agents that collaborate through unconstrained natural-language messages and retain the full message history in the shared channel. **Multi-Agent Debate** (Du et al., 2024) runs multiple agents that propose initial answers and then debate over several rounds, converging on a final answer by majority vote. We report token-overlap F1 for the interaction setting and exact-match accuracy for the pipeline, alongside the average total tokens per problem. Implementation details for each baseline are given in Appendix A.

Table 1: **PACT vs. baselines on the two-agent split-evidence interaction (Setting A).** Tok = avg total tokens per problem. Best F1 and lowest tokens per benchmark within each model in **bold**.

	HotpotQA		2WikiMHQA	
Method	F1	Tok	F1	Tok
Qwen3-8B				
CoA	68.6	7,281	65.6	6,744
TextMAS	59.5	9,384	58.0	7,753
Multi-Agent Debate	69.6	13,520	68.2	10,669
PACT (ours)	**69.9**	**6,704**	**68.7**	**6,100**
Qwen3-14B				
CoA	54.8	6,782	55.2	6,452
TextMAS	45.8	8,652	41.8	7,160
Multi-Agent Debate	48.4	12,133	36.7	9,561
PACT (ours)	**56.8**	**5,996**	**59.2**	**5,130**
Qwen3-32B				
CoA	54.2	5,871	56.3	5,339
TextMAS	50.1	8,857	46.4	7,300
Multi-Agent Debate	60.3	12,258	58.7	9,467
PACT (ours)	**61.5**	**4,821**	**61.3**	**4,039**

5.2 Main Results

Tables 1 and 2 show that PACT shifts the performance–cost frontier across both evaluated MAS settings, demonstrating the effectiveness of passing the action-state information.

Grounded handoffs outperform full-history communication. In the split-evidence interaction setting, a successful message must do more than state a local conclusion: it must surface the relevant evidence and clarify what information is still needed from the other agent. This requirement exposes the weakness of unconstrained communication. For example, TextMAS retains the full conversation history, but its free-form generations introduce redundant reasoning and force later turns to re-read unnecessary content. As shown in Table 1, it consumes more tokens than PACT but still performs worse on both benchmarks, indicating that preserving more text does not necessarily provide a more useful handoff. In contrast, as shown in the example in Figure 5, PACT fits the interaction structure by using an explicit action-state format,

Table 2: **PACT vs. baselines on the four-agent sequential pipeline (Setting B).** Tok = avg total tokens per problem. PACT row shaded; best accuracy and lowest tokens per benchmark within each model in **bold**.

	AIME24		AIME25		GPQA-D		OBQA		**Mean**	
Method	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok	Acc	Tok
Qwen3-8B										
CoA	55.0	38,072	63.7	47,279	59.5	22,563	90.4	4,851	67.2	28,191
TextMAS	48.3	36,911	45.8	45,004	59.0	20,357	91.1	5,263	61.1	26,884
Multi-Agent Debate	56.4	140,863	**70.8**	169,785	57.7	85,351	92.0	32,784	69.2	107,196
PACT (ours)	**57.5**	**31,414**	67.5	**41,071**	**60.4**	**19,960**	**92.5**	**4,304**	**69.5**	**24,187**
Qwen3-14B										
CoA	54.9	28,368	64.5	35,807	61.6	18,789	93.9	4,533	68.7	21,874
TextMAS	50.8	34,066	60.8	44,111	61.0	16,366	92.0	4,675	66.2	24,805
Multi-Agent Debate	55.2	138,081	**74.3**	156,895	61.8	81,878	93.2	30,704	71.1	101,890
PACT (ours)	**55.4**	**26,994**	71.6	**32,548**	**62.5**	**15,935**	**95.2**	**3,559**	**71.2**	**19,759**
Qwen3-32B										
CoA	57.3	29,593	71.2	35,818	65.7	18,535	93.1	3,811	71.8	21,939
TextMAS	52.1	45,867	60.4	57,984	65.7	21,428	93.8	5,770	68.0	32,762
Multi-Agent Debate	58.4	134,679	**75.8**	157,744	66.4	77,888	93.7	28,394	73.6	99,676
PACT (ours)	**59.2**	**25,574**	72.7	**30,970**	**68.5**	**16,271**	**94.4**	**3,517**	**73.7**	**19,083**

turning each turn into a grounded handoff rather than a free-form explanation or an unsupported answer fragment. This allows PACT to achieve the best performance while using fewer tokens.

Compact artifacts outperform extended deliberation. In the sequential pipeline setting, the communication requirement is different: each downstream agent needs the artifact-related information required to update its own planning. This exposes a different limitation of unrestricted multi-agent interaction. Although Multi-Agent Debate can improve reasoning through repeated answer-level deliberation, its communication is not organized around the intermediate artifacts needed, causing substantial token overhead. As shown in Table 2, PACT achieves comparable or better performance on most benchmarks while using only 19%–23% of the tokens required by Multi-Agent Debate. This suggests that the pipeline benefits less from extended deliberation than from passing compact action-related artifacts that downstream agents can directly use.

Stronger models make better use of compact communication. The scale trend further supports the role of sparse public communication. Under PACT, stronger models require less explicit inter-agent communication while achieving better task performance. In the sequential pipeline, PACT's average token usage decreases by 21.1% from Qwen3-8B to Qwen3-32B, while increasing the mean accuracy by 4.2 points. A similar reduction appears in the interaction setting, where PACT's token usage decreases as the model scale increases on both benchmarks. This suggests that stronger agents can better exploit compact evidence and refined artifacts, without requiring the sender's full reasoning process to be repeatedly exposed in the shared history. However, under baseline settings, larger models continue to spend tokens producing and consuming redundant public content. Thus, pure model scaling cannot fix communication inefficiency: lacking a protocol, powerful models still waste tokens on redundant public messages, while PACT enables more private computation and keeps agent communication concise.

5.3 Ablation Study

Table 3 shows how each field in PACT contributes to effective communication. Removing ACTION reduces F1 from 69.9 to 64.9, showing that the receiver needs an explicit signal about what information is being provided or requested. Removing STATE also hurts performance, which indicates that the action result is less useful when it is not grounded in the sender's observed evidence. The largest degradation appears when both action and state are removed and only RESULT is forwarded: F1 falls to 64.3 and token usage increases by 12.9%. This suggests that an unsupported result creates ambiguity for the receiver, leading to both worse task performance and less efficient interaction. Overall, the ablation confirms that PACT's advantage comes from the complete action-state handoff: the action clarifies how the message should be used, the state grounds the message in evidence, and the result

Table 3: PACT field-ablation on HotpotQA with Qwen3-8B. A = Action, S = State, R = Result.

	F1	Tok	ΔF1	ΔTok
PACT (A+S+R)	69.9	6,704	—	—
w/o A (S+R)	64.9	6,826	−5.0	+1.8%
w/o S (A+R)	65.2	6,741	−4.7	+0.6%
w/o A, S (R only)	64.3	7,571	−5.6	+12.9%

carries the factual contribution.

6 PACT on Agentic Coding Harnesses

The settings so far use predefined MAS scaffolds. We now test whether the communication-content rule transfers to external production agentic coding harnesses it was not designed for, such as **OpenHands** (CodeActAgent) (Wang et al., 2025) and **SWE-agent** (Yang et al., 2024), on **SWE-bench Verified** (Jimenez et al., 2024), which solves real-world GitHub issues in real environments.

6.1 Porting PACT as a proxy hook

PACT is implemented as a proxy hook that updates the public messages passed between agent turns in-flight. The hook has two components. First, each turn must emit, before its tool call, a structured `<summary>` block containing *Action Required*, *Observed State*, and *Planned Effect*—the §4 schema rephrased with forward-looking coding semantics. Second, before each turn the proxy rewrites prior assistant messages to keep only the `<summary>` block and the tool calls, removing the intermediate process and any free-form prose; tool results are left intact, so the inter-turn channel reduces to the information of action-state summary, tool calling and tool results.

6.2 Results on coding harnesses

As shown in Table 4, PACT shifts the efficiency frontier in both coding harnesses. On **OpenHands**, PACT improves both effectiveness and efficiency: the SWE-bench Verified resolve rate increases from 19.40% to 23.00% (+18 resolved instances), while tokens-per-resolved decreases by 10.3% and average completion tokens per call decrease by 5.3%. On **SWE-agent**, PACT produces an even larger efficiency gain, reducing input tokens from 314.6 M to 156.0 M (−50.4%), which is the dominant cost in this long-context loop with only a small resolve-rate change, while tokens-per-resolved drops by approximately 47%. Thus, the efficiency gain is consistent across both production coding harnesses.

Table 4: **PACT ported to production agentic coding harnesses** on SWE-bench Verified, Qwen3-14B.

OpenHands (CodeActAgent)

Metric	Baseline	**PACT**	Δ
Resolved	97/500	**115/500**	**+18**
Resolved %	19.40	**23.00**	**+3.60 pp**
Avg compl./call	870.6	**824.2**	**−5.3%**
Tokens / resolved	3.82 M	**3.43 M**	**−10.3%**

SWE-agent (default single-agent)

Metric	Baseline	**PACT**	Δ
Resolved	128/500	121/500	−7
Resolved %	25.6	24.2	−1.4 pp
Input tokens	314.6 M	**156.0 M**	**−50.4%**
LLM calls	21,398	**20,623**	**−3.6%**
Tokens / resolved	2.46 M	**1.30 M**	**−47%**

This indicates that PACT is not tied to the controlled MAS scaffolds used in earlier sections. The same action-state principle can be implemented as a lightweight proxy-level intervention and still improves the efficiency of external coding agents. It improves OpenHands, where verbose assistant turns are repeatedly carried through the interaction history, and remains approximately neutral on SWE-agent, where the main gain comes from reducing accumulated input context. In both cases, PACT reduces the cost per solved instance, suggesting that sparse public communication is a portable efficiency mechanism even when the underlying agent loop and tool interface differ.

7 Conclusion

This paper studies what agents should pass to one another in MAS. Through diagnostic analyses across five common strategies on two MAS settings, we show that no common communication strategy is uniformly optimal, and the useful content in an inter-agent message is usually action-centered. Motivated by this, we propose **PACT** (**P**rotocolized **A**ction-state **C**ommunication and **T**ransmission), a communication protocol that restricts shared history to compact action-state messages while excluding intermediate content. Across controlled MAS settings and production coding harnesses, PACT improves the performance–cost trade-off by reducing redundant context while preserving or improving task performance. These results suggest that disciplining what enters the shared inter-agent channel is itself an effective mechanism for reducing redundant context growth in inter-agent communication, and we should not merely leave it as unconstrained natural-language outputs.

Limitations

PACT is designed for MAS settings where shared conversational history is a major source of token cost, its benefits on systems with short interactions or architectures that do not repeatedly expose prior agent outputs are not fully explored. Our experiments cover two controlled MAS topologies and two agentic coding harnesses, but they do not exhaust all forms of multi-agent collaboration, such as open-ended debate, tool-heavy planning, or dynamically routed agent networks.

References

Anthropic. 2026a. Claude Code. https://claude.com/product/claude-code.

Anthropic. 2026b. Introducing Claude Opus 4.7. https://www.anthropic.com/news/claude-opus-4-7.

Longju Bai, Zhemin Huang, Xingyao Wang, Jiao Sun, Rada Mihalcea, Erik Brynjolfsson, Alex Pentland, and Jiaxin Pei. 2026. How do ai agents spend your money? analyzing and predicting token consumption in agentic coding tasks. *arXiv preprint arXiv:2604.22750*.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, and 1 others. 2026. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2024. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. In *Proceedings of the 12th International Conference on Learning Representations*.

Weize Chen, Ziming You, Ran Li, Yitong Guan, Chen Qian, Chenyang Zhao, Cheng Yang, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2025. Optima: Optimizing effectiveness and efficiency for LLM-based multi-agent system. In *Proceedings of the 13th International Conference on Learning Representations*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, pages 11850–11881.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Steven Yau, Zijuan Lin, Liyang Zhou, and 1 others. 2024. Metagpt: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, volume 2024, pages 23247–23275.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can language models resolve real-world GitHub issues? In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*.

Guohao Li, Hasan Anil Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, pages 51991–52008.

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. 2024. Improving multi-agent debate with sparse communication topology. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7281–7294.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate.

math-ai. 2025. AIME 2025 dataset. https://huggingface.co/datasets/math-ai/aime25. Hugging Face dataset.

Maxwell-Jia. 2024. AIME 2024 dataset. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024. Hugging Face dataset.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

OpenAI. 2026a. Codex. https://openai.com/codex/.

OpenAI. 2026b. Introducing GPT-5.5. https://openai.com/index/introducing-gpt-5-5/.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, and 1 others. 2024. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 15174–15186.

Vignav Ramesh and Kenneth Li. 2025. Communicating activations between language model agents. *arXiv preprint arXiv:2501.14082*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof Q&A benchmark. In *Proceedings of the First Conference on Language Modeling (COLM)*.

Xu Shen, Yixin Liu, Yiwei Dai, Yili Wang, Rui Miao, Yue Tan, Shirui Pan, and Xin Wang. 2025. Understanding the information propagation effects of communication topologies in llm-based multi-agent systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12358–12372.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, and 1 others. 2026. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024a. Mixture-of-agents enhances large language model capabilities.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. 2024b. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, and 5 others. 2025. OpenHands: An open platform for AI software developers as generalist agents. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024c. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 257–279.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

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. 2024. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *Proceedings of the 12th International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the 11th International Conference on Learning Representations*.

Ye Yu, Heming Liu, Haibo Jin, Xiaopeng Yuan, Peng Kuang, and Haohan Wang. 2026. Learning to communicate: Toward end-to-end optimization of multi-agent language systems. *arXiv preprint arXiv:2604.21794*.

Yuting Zeng, Weizhe Huang, Lei Jiang, Tongxuan Liu, Xitai Jin, Chen Tianying Tiana, Jing Li, and Xiaohua Xu. 2025. S2-mad: Breaking the token barrier to enhance multi-agent debate efficiency. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 9393–9408.

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Yu, and Tianlong Chen. 2025. Cut the crap: An economical communication pipeline for llm-based multi-agent systems. In *International Conference on Learning Representations*, volume 2025, pages 75389–75428.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö Arık. 2024. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237.

Jiaxing Zhao, Hongbin Xie, Yuzhen Lei, Xuan Song, Zhuoran Shi, Lianxin Li, Shuangxue Liu, and Haoran Zhang. 2025. Connecting the dots: A chain-of-collaboration prompting framework for llm agents. *arXiv preprint arXiv:2505.10936*.

Jiaru Zou, Xiyuan Yang, Ruizhong Qiu, Gaotang Li, Katherine Tieu, Pan Lu, Ke Shen, Hanghang Tong, Yejin Choi, Jingrui He, and 1 others. 2025. Latent collaboration in multi-agent systems. *arXiv preprint arXiv:2511.20639*.

Appendix

A Experimental Details

This section collects the experimental settings for both the diagnostic analysis (§3) and the PACT experiments (§5, §6). Unless noted otherwise, all runs use vLLM with sampling temperature 0.6 and top-p 0.95.

Models. Our evaluation includes models from the Qwen3 family (Yang et al., 2025) (8B, 14B and 32B in scale), allowing us to assess performance across both mid-scale and large-scale reasoning models.

Benchmarks. For the split-evidence interaction, we evaluate on HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020). For the sequential pipeline setting, we evaluate on three categories of benchmarks: (i) mathematical reasoning, including AIME2024 (Maxwell-Jia, 2024) and AIME2025 (math-ai, 2025); (ii) scientific reasoning, using GPQA-Diamond (Rein et al., 2024); and (iii) commonsense question answering, using OpenBookQA (Mihaylov et al., 2018). We report AIME24/25 and GPQA-Diamond as avg@8 where we average over different seeds to resolve sampling noise; single-seed benchmarks use seed 42.

Diagnostic analysis (§3). The diagnostic analyses evaluate five common inter-agent communication strategies in both MAS settings: *Full Content*, *Concise Generation*, *Conclusion Only*, *Brief Summary*, and *Artifact Only*. In the two-agent split-evidence interaction, we use a 5–5 context split, where each agent receives one gold supporting paragraph and four distractors. Agents communicate for at most 8 turns with early exit, and we set `max_new_tokens` to 8,192. The sequential pipeline uses the same five communication strategies under the fixed Planner–Critic–Refiner–Solver order, with benchmark-specific settings reported below.

PACT experiments. The four-agent Planner→Critic→Refiner→Solver pipeline (§3 Setting B, §5) and the two-agent split-evidence interaction (§5) reuse the corresponding analysis settings. The split-evidence interaction uses 4 alternating turns, `max_new_tokens`= 4,096 per turn, 5–5 split. Following Yu et al. (2026), the per-benchmark settings are given in Table 5.

Baselines (§5). The three MAS baselines are instantiated as follows. All baselines share the decoding settings, per-benchmark token budgets, and dataset splits described above; only the inter-agent communication protocol differs.

Chain of Agents (CoA). (Zhang et al., 2024) We use the same agent counts as the corresponding MAS settings (2 workers for the interaction setting, 4 for the pipeline). Each worker reads only the *single* most recent message from its predecessor—there is no rolling shared history, so the inter-agent channel carries only 1 round of context between consecutive workers.

Text-based MAS (TextMAS). (Zou et al., 2025) Role-specialised agents collaborating via free-form natural-language messages, with the full message history retained in the shared channel and no constraint on the message format. We instantiate it under the same MAS configurations: two symmetric agents with 4 alternating turns for Setting A, and the four-agent Planner→Critic→Refiner→Solver schedule for Setting B.

Multi-Agent Debate. (Du et al., 2024) 4 agents independently propose initial answers and then debate over 3 rounds, each round revising its answer conditioned on the other agents' previous-round answers; the final answer is the majority vote across agents after the last round. The same configuration is used in both settings.

Table 5: Per-benchmark evaluation settings for the sequential pipeline and the split-evidence interaction.

Benchmark	n	`max_new_tokens`
AIME 2024 / 2025	30 each	32,768
GPQA-Diamond	198	8,192
OpenBookQA	1,000	2,048
HotpotQA	7,405	4,096
2WikiMultiHopQA	12,576	4,096

Agentic coding harnesses (§6). OpenHands (CodeActAgent) and SWE-agent are evaluated on SWE-bench Verified with Qwen3-14B. PACT is applied as a LiteLLM proxy hook that rewrites prompts and parses responses in flight, with no SDK or docker-image changes.

B Split-Evidence Interaction Case Study: How PACT Changes the Message

Figure 5 contrasts the same interaction turn with and without PACT on a representative HotpotQA bridge question ("*Which magazine was started first,*

Without PACT (free-form, <think> kept) — ~480 tokens

<think> The question asks which magazine started first. My paragraph is about *Arthur's Magazine*; it says it was an American literary periodical published in Philadelphia, established in 1844. I have nothing on *First for Women*, so I should report my finding and ask the partner *(... ~420 tokens of deliberation ...)* </think>

Hi partner! I checked my documents. I have a paragraph on Arthur's Magazine, and it looks like it was established around 1844 in Philadelphia. I'm not fully certain, but could you see whether you have anything on First for Women and when it started, so we can decide which came first?

With PACT (three-field) — ~40 tokens

Action Required: the founding year of *First for Women*.
Environment State: "Arthur's Magazine was an American literary periodical first published in 1844."
Action Result: Arthur's Magazine was founded in 1844; need First for Women's date to compare.

Figure 5: Illustrative interaction turn with and without PACT (HotpotQA bridge question). PACT drops the <think> trace and hedged prose, keeping only the surfaced fact and the explicit request—an ~10× shorter message with the same actionable content. Long spans are elided as *(... tokens ...)*.

Arthur's Magazine or First for Women?"). Without PACT—essentially the TextMAS baseline—the agent forwards its full <think> deliberation and hedged prose, so the partner re-reads hundreds of tokens of reasoning about the agent's own paragraph. PACT keeps exactly the surfaced fact and the precise request: the ~10× shorter message carries the same actionable content, which is why downstream turns stay accurate while the shared history stops compounding.

C The Use of Large Language Models (LLMs)

Large Language Models (LLMs) were used solely for polishing the writing and improving the clarity of presentation. They were **NOT** involved in research ideation, methodology design, experiments, analysis, or any other substantive aspect of this work. All scientific contributions, results, and conclusions are the sole responsibility of the authors.