EMERGENT COMMUNICATION TOPOLOGIES FROM LOCAL REASONING REFINEMENTS IN MULTI-AGENT SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective collaboration in Large Language Model (LLM)-based multi-agent systems (MAS) requires communication that adapts to how agents can improve one another's evolving reasoning. However, existing approaches based on task or query information or overall agent assessments do not directly identify which reasoning states can provide useful corrections to a particular peer. We introduce **refinement-specific communication** and instantiate it in **LORIT**, a training-free framework that organizes communication around directional, recipient-specific opportunities for reasoning improvement. Agents assess how their current reasoning states can correct or complete one another, and these local relations induce a global communication structure that accommodates mutual refinement and evolves with reasoning. This couples reasoning updates with topology self-adaptation without imposing a global agent ranking or optimizing a separate graph-level objective. Experiments across diverse reasoning and coding tasks demonstrate broad improvements over single-agent and multi-agent baselines with both weak and strong LLM backbones. With stronger refinement assessment, LORIT enables groups of smaller models to achieve collective performance comparable to or exceeding that of larger single-agent models. These results highlight the potential of refinement-specific communication to turn complementary information in imperfect individual solutions into stronger collective reasoning.

1 INTRODUCTION

Large language model (LLM) (Qwen et al. (2025); Singh et al. (2025)) based multi-agent systems (MAS) provide a natural paradigm for distributed reasoning, where multiple agents explore different solutions and improve their decisions through interaction (Wu et al. (2023); Chan et al. (2024); Liu et al. (2025)). But communication is not inherently beneficial: incorrect arguments may propagate across agents, premature consensus may suppress alternative solutions, and dense interaction may repeatedly transmit redundant information (Shen et al. (2025); Zhang et al. (2025b)). Effective multi-agent reasoning therefore depends not only on what individual agents reason, but also on which reasoning states should influence which others.

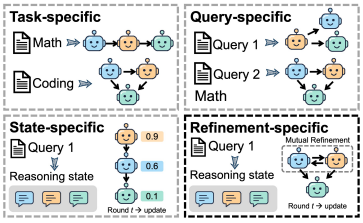


Figure 1: Comparison of communication topology paradigms.

Communication topology has consequently become an important component of LLM-based MAS. Early systems commonly relied on predefined interaction patterns, such as sequential workflows and all-to-all debate, reusing the same communication structure across different queries (Qian et al., 2024; Du et al., 2023). However, a fixed structure may be poorly matched to heterogeneous problems, while excessive communication can introduce redundancy and amplify errors (Zhang et al., 2025b). Research has therefore moved toward adaptive topology design, progressing from **task-specific** optimization to **query-specific** selection and generation. Along this progression, existing approaches have explored predefined interaction structures (Qian et al., 2025), learned or optimized communication graphs (Zhuge et al., 2024; Liu et al., 2023; Zhang et al., 2025a), adaptive pruning (Zhang et al., 2025b; Wang et al., 2025), and task-conditioned topology generation (Sun et al., 2026; Li et al., 2026). Yet LLM stochasticity makes collaboration needs depend on agents' evolving reasoning

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.

states, not merely on the task or query. This limitation motivated a further shift to **state-specific** organization, where communication adapts to agents' actual responses without external training, such as through dynamic organization based on agents' contribution scores (Tastan et al., 2026).

Despite this progress, a common limitation remains: adapting communication to a task, query, or current response does not by itself determine who can improve whose reasoning. Topologies determined from task or query information alone may not track changing opportunities for refinement during reasoning states. While state-specific organization can address this mismatch, when influence is guided by agreement with the collective response, it may still overlook recipient-specific contributions. A highly rated agent may have little to offer a particular peer, while a lower-rated agent may provide exactly the correction that peer needs. These limitations are particularly relevant to difficult problems, where most initial answers can be incomplete or incorrect and agreement can reflect shared errors, as well as open-ended long-horizon tasks such as scientific discovery, where alternative hypotheses may warrant continued exploration. Even a response with an incorrect final answer may contain information that helps another agent revise its reasoning. Effective collaboration therefore calls for communication that adapts to these evolving opportunities for refinement. Such opportunities can also be reciprocal: two agents may correct different parts of each other's reasoning without either being better overall. We therefore ask a more fundamental question:

How can local reasoning refinements support collaboration and induce a global communication topology that accommodates complementary perspectives and cyclic refinement relations?

Our key insight is to organize communication around recipient-specific opportunities for reasoning refinement, which we term **refinement-specific communication** (Figure 1). We instantiate this perspective in **LoRIT** (***Lo**cal **R**efinement-**I**nduced **T**opology*), a training-free MAS framework that couples reasoning refinement with topology evolution. In contrast to using predefined connections or overall agent rankings, LoRIT constructs communication topology from opportunities for agents to refine one another at each round, then uses the resulting structure to guide reasoning updates. This enables state-dependent collaboration without searching over candidate graphs using a separate graph-level scoring objective, addressing both the need to adjust topology to evolving reasoning states and the limitations of a single agent ranking in capturing mutual refinement and cyclic dependencies. To adapt to different task backgrounds and difficulty levels, LoRIT enables topology construction and updates without tunable graph parameters or edge-level reinforcement learning.

We make the following contributions:

1. We introduce **refinement-specific communication**, which grounds influence in directional, recipient-specific reasoning refinements, and mathematically prove that communication topology can emerge directly from agents' reasoning states through these relations without predefined structures, external training, or agent rankings.
2. We propose **LoRIT**, a training-free MAS framework that transforms local refinement relations into a minimal, refinement-preserving communication topology and dynamically re-induces it as agent reasoning evolves.
3. We evaluate LoRIT across multiple LLM backbones and diverse reasoning and coding benchmarks, demonstrating broad improvements over representative baselines, and analyze how local refinement relations give rise to global collaboration in multi-agent systems, enabling them to arrive at correct answers.

2 LoRIT

We show the overview of LoRIT in Figure 2. The key idea is to convert pairwise refinements into a component topology through strongly connected components (SCCs) condensation and transitive reduction, expand it for synchronous agent updates, and re-induce it as reasoning evolves. Final readout aggregates states from source components of the topology induced after the last update.

2.1 Local Reasoning Refinement

Given a query Q, we consider N LLM agents $\mathcal{A} = \{A_1, \ldots, A_N\}$, each of which initially solves Q independently. At collaboration round t, agent A_i maintains a reasoning state $S_i^{(t)} = (a_i^{(t)}, r_i^{(t)})$, consisting of its answer and reasoning trace, and $\mathcal{S}^{(t)} = \{S_1^{(t)}, \ldots, S_N^{(t)}\}$ denotes the collection of

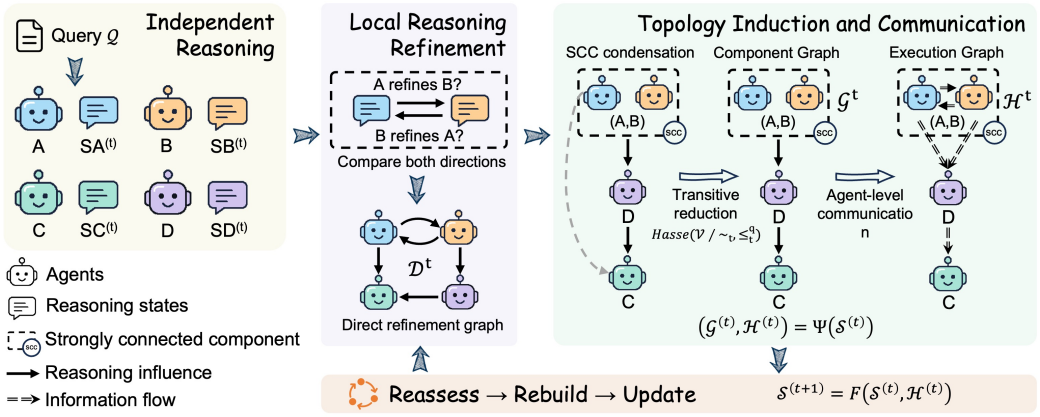


Figure 2: Overview of LoRIT. Given a query Q, agents independently generate initial reasoning states. The refinement graph $\mathcal{D}^{(t)}$ induces a component topology $\mathcal{G}^{(t)}$ through SCC condensation and transitive reduction. Its agent-level expansion $\mathcal{H}^{(t)}$ guides synchronous reasoning updates, followed by topology re-induction and evolution.

current states. Independent initialization preserves diverse reasoning trajectories before communication begins, exposing different hypotheses, derivations, and potential failure modes.

Directional refinement. We say that $S_i^{(t)}$ refines $S_j^{(t)}$, written $S_i^{(t)} \succ_t S_j^{(t)}$, if it provides information that can concretely improve $S_j^{(t)}$, e.g., by correcting an error, supplying a missing step, identifying an overlooked constraint, or resolving an inconsistency. Agreement on the final answer alone does not constitute refinement. For each pair of states, LoRIT independently evaluates the two directional judgments using the backbone LLM with a structured comparison prompt.

$$R_t(i,j) = \mathbb{1}\left[S_i^{(t)} \succ_t S_j^{(t)}\right], \qquad R_t(j,i) = \mathbb{1}\left[S_j^{(t)} \succ_t S_i^{(t)}\right] \tag{1}$$

Both directions may hold when two states provide complementary refinements, while neither needs to hold when no useful refinement exists. We set $R_t(i,i) = 0$. These judgments induce the direct refinement graph $\mathcal{D}^{(t)} = (\mathcal{V}, \mathcal{E}_{\text{ref}}^{(t)})$, where $\mathcal{V} = \{1, \ldots, N\}$ and

$$(i,j) \in \mathcal{E}_{\text{ref}}^{(t)} \iff R_t(i,j) = 1. \tag{2}$$

Each edge records a positive directional refinement judgment. Refinement measures pairwise usefulness rather than an intrinsic importance score of an agent: a reasoning state may refine one state while providing no useful information to another. The local judgments are not required to be globally transitive or acyclic; LoRIT derives global structure from their induced reachability. We evaluate these judgments against human annotations and examine their robustness and downstream utility through perturbation tests and paired reasoning updates (Appendix D.1).

2.2 Reasoning-Induced Communication Topology

The direct refinement graph $\mathcal{D}^{(t)}$ may contain cycles and transitively redundant edges, which can lead to unnecessary token consumption and amplify error signal. Rather than resolving these through additional learned scores or graph heuristics, LoRIT derives the communication structure from the order induced by refinement reachability.

From refinement reachability to a partial order. For $i, j \in \mathcal{V}$, define

$$i \preceq_t j \iff i = j \text{ or } i \rightsquigarrow j \text{ in } \mathcal{D}^{(t)} \tag{3}$$

where $i \rightsquigarrow j$ denotes directed reachability. This relation forms a preorder. Mutual reachability, $i \sim_t j$ iff $i \preceq_t j$ and $j \preceq_t i$, partitions $\mathcal{D}^{(t)}$ into SCCs. Let $q_t(i) = [i]_t$ denote the corresponding quotient map. The induced relation

$$[i]_t \preceq_t^q [j]_t \iff i \preceq_t j \tag{4}$$

forms a partial order over the SCCs.

SCC membership reflects mutual reachability, not semantic equivalence or direct refinement between every pair of member states. Nontrivial SCCs capture cyclic refinement dependencies. Quotienting removes cycles at the component level while retaining the individual reasoning states for subsequent communication.

Minimal component topology. LoRIT retains only the cover relations of this quotient partial order, yielding

$$\boxed{\mathcal{G}^{(t)} = \mathrm{Hasse}(\mathcal{V}/\!\sim_t, \preceq_t^q) = \mathrm{TR}\left(\mathrm{Condensation}(\mathcal{D}^{(t)})\right)} \tag{5}$$

where TR denotes transitive reduction (Aho et al., 1972). Thus, $\mathcal{G}^{(t)}$ contains only the immediate dependencies of the quotient refinement order while preserving its reachability. For the fixed quotient node set, it is the unique minimum-edge graph satisfying

$$\mathrm{Reach}(\mathcal{G}^{(t)}) = \mathrm{Reach}\left(\mathrm{Condensation}(\mathcal{D}^{(t)})\right) \tag{6}$$

This limits direct inter-component communication by removing links whose reachability is already provided by other paths. It theoretically avoids additional edge-selection scores or sparsity hyperparameters, since the refinement relations uniquely determine the component topology. The minimality is structural: it concerns component-level refinement reachability rather than lossless reasoning propagation or minimum token cost. Statements and proofs are provided in Appendix A.

2.3 Communication and Topology Co-Evolution

The component topology $\mathcal{G}^{(t)}$ captures the global refinement order, but reasoning updates occur at the level of individual agents. We thus expand it into an agent-level execution graph $\mathcal{H}^{(t)} = (\mathcal{V}, \mathcal{E}_{\mathcal{H}}^{(t)})$.

Agent-level communication. Within each SCC, LoRIT retains the original refinement edges. Across two components connected by a Hasse edge, all states in the upstream component are exposed to all agents in the downstream component:

$$\begin{aligned}\mathcal{E}_{\mathcal{H}}^{(t)} = \{(i,j) \in \mathcal{E}_{\mathrm{ref}}^{(t)} : q_t(i) = q_t(j)\} \\ \cup \{(i,j) : (q_t(i), q_t(j)) \in \mathcal{E}(\mathcal{G}^{(t)})\}\end{aligned} \tag{7}$$

Hence, $\mathcal{D}^{(t)}$ represents directly judged local refinements, $\mathcal{G}^{(t)}$ represents the induced component-level order, and $\mathcal{H}^{(t)}$ specifies the communication actually executed by the agents. The inter-component all-to-all exposure gives every downstream agent access to the upstream component without introducing representative selection. Despite this expansion, the execution graph preserves the structural reachability of the original refinement graph:

$$\boxed{\mathrm{Reach}(\mathcal{H}^{(t)}) = \mathrm{Reach}(\mathcal{D}^{(t)})} \tag{8}$$

The proof is provided in Appendix A. With this, all agents can update synchronously, each using its own state and incoming neighbors' states from the same round-t snapshot:

$$S_j^{(t+1)} = F\left(S_j^{(t)}, \{S_i^{(t)} : i \in \mathcal{N}^-_{\mathcal{H}^{(t)}}(j)\}\right) \tag{9}$$

where F is instantiated by the backbone LLM, conditioned on Q, to critically incorporate useful incoming information. Since $\mathcal{G}^{(t)}$ explicitly communicates only immediate component dependencies, indirect influence may propagate across successive collaboration rounds rather than through every transitive dependency within a single round.

Reasoning–topology co-evolution. Reasoning revision changes the states from which the topology was induced. LoRIT therefore re-evaluates the refinement relations and reconstructs the communication structure after each round:

$$\boxed{(\mathcal{G}^{(t)}, \mathcal{H}^{(t)}) = \Psi(\mathcal{S}^{(t)}), \qquad \mathcal{S}^{(t+1)} = F(\mathcal{S}^{(t)}, \mathcal{H}^{(t)}).} \tag{10}$$

Reasoning determines communication, while communication shapes subsequent reasoning. The process runs for at most T rounds and may terminate early when both the reasoning states and induced structure stabilize, as indicated by an unchanged execution-graph edge set between consecutive rounds.

2.4 Final Response Aggregation

After the final collaboration round T, LoRIT derives the system-level response from the remaining refinement-maximal reasoning states. Specifically, let

$$\mathcal{M}^{(T)} = \left\{ C \in \mathcal{V}/\sim_T : \deg^-_{\mathcal{G}^{(T)}}(C) = 0 \right\} \tag{11}$$

denote the components with no incoming refinement relation in the final component topology. Under our edge convention, these components represent reasoning states for which no other component provides a further refinement. LoRIT collects the states belonging to these components and applies a final readout using the same backbone LLM:

$$\hat{a} = \rho\left(\left\{ S_i^{(T)} : q_T(i) \in \mathcal{M}^{(T)} \right\} \right), \tag{12}$$

where ρ synthesizes a single answer by resolving differences among the remaining reasoning states based on their reasoning content rather than their support counts. If only one state remains, its answer is returned directly. This refinement-based readout completes the reasoning process without introducing an additional ranking score or learned aggregation module. Prompts and implementation details are provided in Appendix B.

3 Experiments

3.1 Experimental Setup

Benchmarks. We evaluate LoRIT on benchmarks spanning **math**, **science**, **general knowledge**, and **coding**. ■ **Math reasoning benchmarks** include GSM8K (Cobbe et al., 2021), GSM-Hard (Gao et al., 2023), AQUA-RAT (Ling et al., 2017), and AIME-2023, AIME-2024, and AIME-2025 (Zhang & Math-AI, 2025). ■ **Scientific reasoning benchmark** includes GPQA-Diamond (Rein et al., 2023), ■ **General knowledge benchmarks** include MMLU (Hendrycks et al., 2021) and MMLU-Pro (Wang et al., 2024), and ■ **Coding benchmarks** include HumanEval (Chen et al., 2021) and the coding scenario of MultiAgentBench (Zhu et al., 2025) (MAB-Coding). Detailed evaluation metrics are provided in Appendix C.

Backbone LLMs and baselines. We conduct our experiments using 9 LLM backbones: `Qwen3-{0.6B, 1.7B, 4B, 8B, 14B, 30B-A3B}` (Yang et al., 2025), `GLM-4-9B` (Glm et al., 2024), `Mistral3-8B` (Liu et al., 2026) and `GPT-5 mini` (Singh et al., 2025). Meanwhile, `GPT-5` (Singh et al., 2025), `GPT-4.1, GPT-4.1-mini` (Achiam et al., 2023), and `DeepSeek-V3` (Liu et al., 2024) are used for benchmarking only. The default maximum generation length is 8192 tokens, with a sampling temperature of 0.5. Our single-agent baselines include **Single Agent**, **Chain-of-Thought (CoT)** prompting (Wei et al., 2022) and **Self-Consistency** (Wang et al., 2022). Multi-agent baselines include **Full-Mesh**, **DyLAN** (Liu et al., 2023), **MacNet** (Qian et al., 2025), **G-Designer** (Zhang et al., 2024), **TopoDIM** (Sun et al., 2026), and **SelfOrg** (Tastan et al., 2026). Implementation details for LoRIT and all baselines are provided in Appendix C.

3.2 Main Baseline Results

Table 1 presents the baseline results across three backbone scales. ❶ With a relatively weak backbone LLM, Qwen3-1.7B, coordination is particularly challenging because agents must identify useful corrections among imperfect reasoning states despite their limited individual capabilities. In this setting, LoRIT achieves an average score of 57.62, outperforming the strongest baseline by 4.85 points and ranking first on eight of the nine benchmarks. ❷ With Qwen3-8B, LoRIT achieves the best performance on all nine benchmarks, reaching an average score of 71.04 with a 5.60-point advantage over the strongest baseline. The gains are notable on challenging reasoning tasks: LoRIT improves over the best baseline by 3.94, 5.55, and 5.05 percentage points on GSM-Hard, AIME, and GPQA, respectively. The benefits also extend to knowledge and coding, with gains of 8.14 percentage points on MMLU-Pro and 1.87 percentage points in HumanEval pass@1. ❸ LoRIT remains effective with the large backbone LLM Qwen3-30B-A3B, achieving the highest score on all nine benchmarks and an average score of 74.74, exceeding the strongest baseline by 3.63 points. On AIME, it reaches 36.67%, a gain of 7.78 percentage points over the best baseline. It also achieves the highest MAB-Coding score of 45.00, improving over the strongest baseline by 1.75 points. These results show that refinement-specific communication benefits both limited-capability agents and stronger backbones across diverse reasoning and coding tasks.

Table 1: Main results on reasoning and coding benchmarks with different scales of backbone LLMs (`Qwen3-1.7B, 8B and 30B-A3B`). Best and second-best available results within each backbone block are highlighted with red and blue backgrounds, respectively. We report accuracy on reasoning benchmarks, pass@1 on HumanEval, and benchmark scores on MAB-Coding. Avg. denotes the average accuracy across all benchmarks. Higher values indicate better performance.

Backbone LLM: Qwen3-1.7B (Small)										
Method	GSM8K	GSM-Hard	AQUA-RAT	AIME	GPQA	MMLU	MMLU-Pro	HumanEval	MAB-Coding	Avg.
Single-Agent Reasoning										
Single Agent	80.29	52.97	75.98	8.89	26.77	60.90	44.39	62.73	16.13	47.67
CoT (Wei et al., 2022)	78.24	53.07	70.08	11.11	30.30	62.45	45.22	62.73	17.38	47.84
Self-Consistency (Wang et al., 2022)	81.85	56.08	78.52	14.44	35.35	64.85	49.79	66.46	15.50	51.43
Multi-Agent Reasoning										
Full-Mesh	82.49	58.23	79.13	8.89	33.33	63.98	50.37	60.87	18.13	50.60
DyLAN (Liu et al., 2023)	77.36	55.84	75.25	7.78	34.85	61.58	50.54	67.08	17.88	49.80
MacNet (Qian et al., 2025)	75.00	52.50	74.58	7.78	37.37	64.51	51.12	68.18	18.63	49.96
G-Designer (Zhang et al., 2024)	78.89	59.65	76.48	5.56	30.81	68.25	53.62	67.08	18.63	51.00
TopoDIM (Sun et al., 2026)	76.54	57.86	76.14	11.11	35.86	56.77	49.04	64.60	17.75	49.52
SelfOrg (Tastan et al., 2026)	80.81	56.90	74.18	12.22	39.90	68.69	55.61	67.70	18.88	52.77
Ours										
LoRIT	83.11	64.31	79.05	18.89	43.94	72.76	62.51	72.05	22.00	57.62
Backbone LLM: Qwen3-8B (Medium)										
Method	GSM8K	GSM-Hard	AQUA-RAT	AIME	GPQA	MMLU	MMLU-Pro	HumanEval	MAB-Coding	Avg.
Single-Agent Reasoning										
Single Agent	90.83	63.91	81.10	13.33	43.43	77.15	64.67	86.96	33.88	61.70
CoT (Wei et al., 2022)	86.58	61.64	83.46	23.33	47.47	78.65	64.84	85.09	33.00	62.67
Self-Consistency (Wang et al., 2022)	91.58	66.48	84.28	21.11	48.99	78.84	68.50	88.82	35.38	64.89
Multi-Agent Reasoning										
Full-Mesh	86.13	61.64	85.83	23.33	47.47	76.42	66.50	86.96	35.00	63.25
DyLAN (Liu et al., 2023)	81.57	68.36	83.86	23.33	48.48	76.89	69.16	87.58	38.13	64.15
MacNet (Qian et al., 2025)	90.50	66.07	78.74	23.33	43.94	78.47	60.60	81.99	37.13	62.31
G-Designer (Zhang et al., 2024)	92.15	65.47	81.10	21.11	45.96	80.88	65.00	86.34	38.88	64.10
TopoDIM (Sun et al., 2026)	83.00	59.59	79.53	20.00	39.39	72.90	61.43	80.47	36.88	59.24
SelfOrg (Tastan et al., 2026)	82.50	66.81	83.07	25.56	50.51	82.59	68.58	90.68	38.63	65.44
Ours										
LoRIT	94.47	72.30	89.76	31.11	55.56	85.16	77.30	92.55	41.13	71.04
Backbone LLM: Qwen3-30B-A3B (Large)										
Method	GSM8K	GSM-Hard	AQUA-RAT	AIME	GPQA	MMLU	MMLU-Pro	HumanEval	MAB-Coding	Avg.
Single-Agent Reasoning										
Single Agent	95.24	74.40	87.80	23.33	55.56	82.07	71.32	90.06	41.50	69.03
CoT (Wei et al., 2022)	93.98	72.09	84.65	25.56	54.04	83.31	69.83	86.34	40.63	67.83
Self-Consistency (Wang et al., 2022)	95.04	74.15	86.74	23.33	56.57	81.88	71.24	91.30	41.25	69.06
Multi-Agent Reasoning										
Full-Mesh	94.58	71.46	86.61	25.56	58.08	83.17	66.58	88.82	42.50	68.60
DyLAN (Liu et al., 2023)	93.25	74.57	87.54	26.67	51.01	81.43	75.23	92.55	42.13	69.38
MacNet (Qian et al., 2025)	92.36	74.02	85.89	25.56	53.03	82.98	72.49	91.30	42.63	68.92
G-Designer (Zhang et al., 2024)	94.85	75.25	83.50	22.22	54.55	84.77	71.90	90.68	43.25	69.00
TopoDIM (Sun et al., 2026)	93.50	68.31	81.69	23.33	51.52	79.18	70.99	89.44	40.88	66.54
SelfOrg (Tastan et al., 2026)	95.44	74.68	88.58	28.89	57.58	85.58	73.48	92.55	43.25	71.11
Ours										
LoRIT	97.80	79.74	92.76	36.67	60.10	87.73	79.72	93.17	45.00	74.74

3.3 Scaling across LLM Backbones

Consistent Gains across Scales. We investigate how refinement-based collaboration scales with individual reasoning capability, particularly when single agents struggle to produce complete and correct solutions. To this end, we evaluate LoRIT across six Qwen3 backbones on challenging mathematical and scientific reasoning benchmarks, GSM-Hard and GPQA, alongside two coding benchmarks, HumanEval and MAB-Coding. Table 2 demonstrates consistent improvements over single-agent reasoning across all evaluated settings. The gains are particularly substantial for smaller backbones: LoRIT improves GSM-Hard accuracy by **16.20 percentage points** with Qwen3-0.6B and GPQA accuracy by **17.17 percentage points** with Qwen3-1.7B. As backbone capability increases, absolute gains on these reasoning benchmarks generally diminish but remain positive. These results highlight the value of refinement-based collaboration in the limited-capability regime, where individual solutions are frequently imperfect, while demonstrating continued benefits as individual reasoning becomes more reliable.

Relative Error Reduction. We report relative error reduction (RER) to quantify the proportional decrease in error rate over single-agent reasoning. For method m, we compute $\mathrm{RER}_m = \frac{a_m - a_{\text{single}}}{1 - a_{\text{single}}}$,

Table 2: Performance of LoRIT across Qwen3 backbones. Gain denotes the absolute improvement over Single Agent: percentage points for GSM-Hard, GPQA, and HumanEval, and score points for MAB-Coding.

	GSM-Hard			GPQA			HumanEval			MAB-Coding		
Backbone	Single Agent	LoRIT	Gain	Single Agent	LoRIT	Gain	Single Agent	LoRIT	Gain	Single Agent	LoRIT	Gain
0.6B	21.03	**37.23**	+16.20	10.61	**26.26**	+15.65	26.21	**36.65**	+10.43	5.12	**10.25**	+5.13
1.7B	52.97	**64.31**	+11.34	26.77	**43.94**	+17.17	62.73	**72.05**	+9.32	16.13	**22.00**	+5.87
4B	56.97	**67.02**	+10.05	34.85	**49.49**	+14.64	79.50	**86.34**	+6.84	27.88	**34.88**	+7.00
8B	63.91	**72.30**	+8.39	43.43	**55.56**	+12.13	86.96	**92.55**	+5.59	33.88	**41.13**	+7.25
14B	67.82	**75.13**	+7.31	48.48	**57.07**	+8.59	89.44	**93.17**	+3.73	35.38	**42.75**	+7.37
30B-A3B	74.40	**79.74**	+5.34	55.56	**60.10**	+4.54	90.06	**93.17**	+3.09	41.50	**45.00**	+3.50

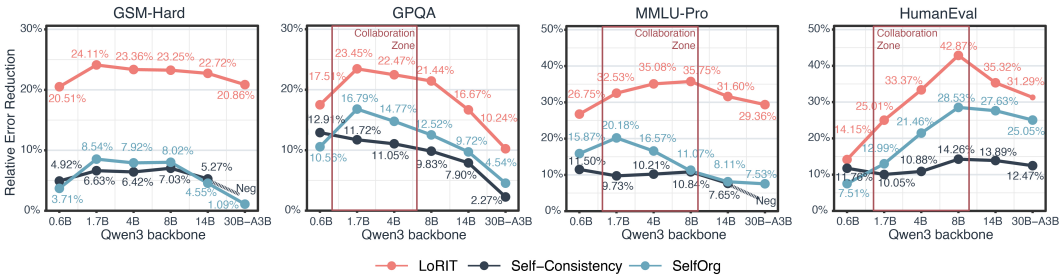


Figure 3: Relative error reduction (RER) of LoRIT, Self-Consistency, and SelfOrg over single-agent reasoning across Qwen3 backbones on four benchmarks. *Neg* denotes negative RER, indicating an increase in error rate relative to the single-agent baseline. Vertical scales differ across panels.

where a denotes accuracy or pass@1 expressed as a proportion. Self-Consistency provides a baseline for independent sampling and answer aggregation without multi-agent collaboration. Figure 3 shows that LoRIT achieves higher RER than both baselines across all evaluated settings. With Qwen3-8B, LoRIT reaches 35.75% RER on MMLU-Pro, compared with 10.84% for Self-Consistency and 11.07% for SelfOrg; on HumanEval, the corresponding values are 42.87%, 14.26%, and 28.53%. Benefits persist with stronger backbones: LoRIT retains 20.86% and 29.36% RER on GSM-Hard and MMLU-Pro with Qwen3-30B-A3B, respectively, while Self-Consistency yields negative RER on both benchmarks. Compared with the relatively smooth RER trends of the communication-free baseline, MAS baselines exhibit task-dependent peaks that mark empirical ranges of particularly strong collaboration gains. We refer to these ranges as **Collaboration Zone**. For LoRIT, these occur around Qwen3-1.7B to 8B across all four datasets with different tasks.

Table 3: Graph statistics across Qwen3 backbones. $\bar{\rho}_D$: refinement edge density in $\mathcal{D}^{(t)}$; $\bar{e}_G$: mean component-edge count in $\mathcal{G}^{(t)}$; $\bar{k}$: mean SCC count.

	GSM-Hard			GPQA		
Backbone	$\bar{\rho}_D$	$\bar{e}_G$	$\bar{k}$	$\bar{\rho}_D$	$\bar{e}_G$	$\bar{k}$
0.6B	0.11	5.62	7.58	0.17	5.19	7.31
1.7B	0.50	1.54	2.15	0.64	1.67	2.56
4B	0.54	1.54	2.29	0.59	1.39	2.38
8B	0.66	1.77	2.22	0.67	1.60	2.45
14B	0.50	1.67	2.11	0.54	1.57	2.31
30B-A3B	0.42	1.85	2.13	0.50	1.44	2.29

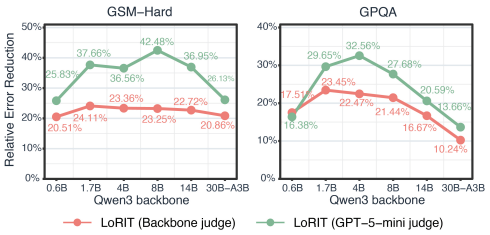
To further examine the structural patterns associated with these Collaboration Zones, we report the edge density of $\mathcal{D}^{(t)}$, the average number of edges in $\mathcal{G}^{(t)}$, and the average number of SCCs in Table 3. On GSM-Hard and GPQA, Qwen3-0.6B produces sparse refinement graphs with densities of 0.11 and 0.17, respectively, and predominantly singleton components, averaging 7.58 and 7.31 SCCs. This limited identified refinement connectivity accompanies RER values below the task-specific peaks. Nevertheless, the high error rates of its single-agent baselines allow even moderate proportional error reductions to translate into large absolute accuracy gains. Across the broader intermediate-scale range from 1.7B to 8B, refinement density increases to 0.50–0.67, while the average number of SCCs falls to approximately two to three. The resulting component topologies contain only 1.39–1.77 edges on average, organizing the denser local relations into mutually reachable groups with few inter-component links. This structural regime overlaps with the Collaboration Zones and is consistent with sufficient reasoning capability enabling agents to provide informative complementary content, recognize recipient-specific corrections, and incorporate them into revised solutions. LoRIT organizes these opportunities through refinement-induced topologies, achieving higher RER than both independent aggregation in Self-Consistency and contribution-guided organization in SelfOrg. These results support grounding communication in concrete opportunities to improve a peer's current reasoning.

3.4 Collaboration with Stronger Judges

Inspired by the Collaboration Zones and their associated graph statistics, we ask whether better refinement assessment can strengthen collaboration among different scales of LLM agents. We hypothesize that, given sufficient reasoning capability, more accurate identification of recipient-specific refinements helps agents exploit complementary information in imperfect reasoning states, enabling collective intelligence to approach or match that of larger models. We test this hypothesis by replacing the backbone-based refinement judge in LoRIT with GPT-5-mini while keeping the reasoning agents unchanged.

	GSM-Hard			GPQA		
Scale	Backbone	GPT-5-mini	Gain	Backbone	GPT-5-mini	Gain
0.6B	37.23	**41.43**	+4.20	26.26	**25.25**	-1.01
1.7B	64.31	**70.68**	+6.37	43.94	**48.48**	+4.54
4B	67.02	**72.70**	+5.68	49.49	**56.06**	+6.57
8B	72.30	**79.24**	+6.94	55.56	**59.09**	+3.53
14B	75.13	**79.71**	+4.58	57.07	**59.09**	+2.02
30B-A3B	79.74	**81.09**	+1.35	60.10	**61.62**	+1.52

(a) Accuracy (%)



(b) Relative error reduction

Figure 4: Effect of refinement judge strength on LoRIT. (a) Accuracy using the reasoning backbone or GPT-5-mini as the refinement judge. Gain denotes the absolute improvement of the GPT-5-mini judge over the backbone judge, in percentage points. (b) Relative error reduction on GSM-Hard and GPQA, computed against the same single-agent baseline at each backbone scale.

Figure 4 shows accuracy gains of 6.94 percentage points on GSM-Hard with Qwen3-8B and 6.57 points on GPQA with Qwen3-4B. A human audit of shared Qwen3-8B states further shows that GPT-5-mini improves refinement-detection F1 by 6.80 and 10.97 points, respectively (Appendix D.1). These results support better refinement assessment as a means of strengthening collaboration while keeping the reasoning agents unchanged.

Table 4: Graph statistics of LoRIT with GPT-5-mini as the refinement judge. Subscripts indicate increases (red) or decreases (blue) relative to using the reasoning backbone as the judge in Table 3.

	GSM-Hard			GPQA		
Backbone	$\bar{\rho}_D$	$\bar{e}_G$	k	$\bar{\rho}_D$	$\bar{e}_G$	k
0.6B	$0.20_{(\uparrow 0.09)}$	$3.36_{(\downarrow 2.26)}$	$4.47_{(\downarrow 3.11)}$	$0.30_{(\uparrow 0.13)}$	$3.55_{(\downarrow 1.64)}$	$4.19_{(\downarrow 3.12)}$
1.7B	$0.26_{(\downarrow 0.24)}$	$3.28_{(\uparrow 1.74)}$	$4.19_{(\uparrow 2.04)}$	$0.29_{(\downarrow 0.35)}$	$3.85_{(\uparrow 2.18)}$	$4.52_{(\uparrow 1.96)}$
4B	$0.22_{(\downarrow 0.32)}$	$3.04_{(\uparrow 1.50)}$	$4.30_{(\uparrow 2.01)}$	$0.23_{(\downarrow 0.36)}$	$3.04_{(\uparrow 1.65)}$	$4.29_{(\uparrow 1.91)}$
8B	$0.24_{(\downarrow 0.42)}$	$3.15_{(\uparrow 1.38)}$	$4.42_{(\uparrow 2.20)}$	$0.25_{(\downarrow 0.42)}$	$2.98_{(\uparrow 1.38)}$	$4.50_{(\uparrow 2.05)}$
14B	$0.20_{(\downarrow 0.30)}$	$3.09_{(\uparrow 1.42)}$	$4.35_{(\uparrow 2.24)}$	$0.18_{(\downarrow 0.36)}$	$2.77_{(\uparrow 1.20)}$	$4.43_{(\uparrow 2.12)}$
30B-A3B	$0.22_{(\downarrow 0.20)}$	$3.21_{(\uparrow 1.36)}$	$4.45_{(\uparrow 2.32)}$	$0.16_{(\downarrow 0.34)}$	$2.55_{(\uparrow 1.11)}$	$4.57_{(\uparrow 2.28)}$

Table 4 reports the corresponding graph statistics. For backbones from 1.7B to 30B-A3B, refinement density decreases, while the numbers of SCCs and inter-component edges increase. For example, on GSM-Hard with Qwen3-8B, refinement density decreases from 0.66 to 0.24 and the average SCC count increases from 2.22 to 4.42, alongside an accuracy improvement from 72.30% to 79.24%. **Higher accuracy with fewer judged refinement edges suggests that effective collaboration depends on selectively identifying and organizing useful refinement relations.** The resulting topology separates agents into more mutually reachable groups and connects these groups through additional component-level links. These findings support our central premise: imperfect reasoning states can contain exploitable complementary information, and refinement assessment plays a key role in organizing this information into effective collaboration.

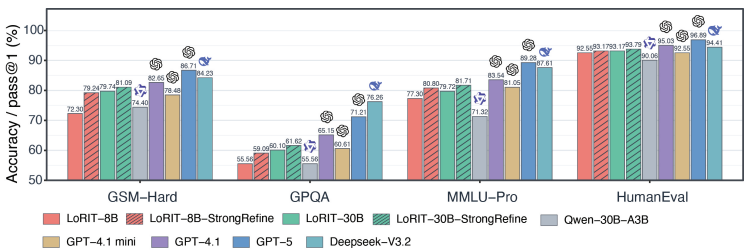


Figure 5: Comparison of LoRIT with single-agent model baselines across four benchmarks. Solid and hatched LoRIT bars use the reasoning backbone and GPT-5-mini as the refinement judge respectively. The 30B setting refers to Qwen3-30B-A3B.

Comparison with Frontier Models. We further compare LoRIT with larger and proprietary single-agent models in Figure 5. Using Qwen3-8B reasoning agents and GPT-5-mini as the refinement judge, LoRIT-8B-StrongRefine outperforms single-agent Qwen3-30B-A3B on all four benchmarks, including **gains of 4.84 percentage points** on GSM-Hard and **9.48 points** on MMLU-Pro, and reaches the same performance level as GPT-4.1 mini. These results indicate that better identification of recipient-specific refinements helps smaller reasoning agents collaborate more efficiently. With improved collaboration, LoRIT can potentially boost smaller models to the performance of stronger individual models.

4 Further Analysis on LoRIT

Appendix D evaluates heterogeneous agent teams, tool and memory augmentation, and accuracy–cost trade-offs across backbone scales. Appendix E further examines agent count, transitive reduction, cross-SCC broadcast, and dynamic topology re-induction, clarifying their contributions to collective performance and inference cost.

5 Related Work

Collaborative Reasoning with LLM Agents. LLM-based multi-agent systems combine specialized roles and iterative interaction for complex tasks. AutoGen (Wu et al., 2023) and ChatDev (Qian et al., 2024) support structured agent collaboration, while multi-agent debate (Du et al., 2023) and ChatEval (Chan et al., 2024) use dialogue for reasoning and evaluation. DMAD (Liu et al., 2025) and CONSENSAGENT (Pitre et al., 2025) address homogeneous reasoning and sycophantic agreement, respectively. Self-Consistency (Wang et al., 2022) aggregates independently sampled reasoning paths without inter-agent communication. LoRIT determines communication through recipient-specific refinement relations among agents' evolving reasoning states.

Communication Topology Design. Communication topology shapes the performance and efficiency of multi-agent systems. MacNet (Qian et al., 2025) studies collaboration across network scales and DAG topologies with local actor–critic refinement. GPTSwarm (Zhuge et al., 2024) optimizes node prompts and inter-agent connections, while G-Designer (Zhang et al., 2024) learns query-conditioned topologies through a variational graph autoencoder. AgentPrune (Zhang et al., 2025b) prunes spatial and temporal connections, whereas AgentDropout (Wang et al., 2025) removes low-contribution agents and refines the remaining communication structure. TopoDIM (Sun et al., 2026) generates topologies supporting conditioned, feedback, and debate interactions. These methods organize communication through structural design, learned generation, or utility-based pruning. LoRIT derives its component topology directly from directional refinements among current reasoning states through reachability-preserving operations, without a separate graph-level scoring objective or learned edge-selection policy.

State-Dependent Multi-Agent Organization. Adaptive coordination uses information generated during execution to organize agents. DyLAN (Liu et al., 2023) selects active agents based on intermediate responses and uses importance estimates for team selection. AgentNet (Yang et al., 2026) combines task routing with experience-driven adaptation of agent capabilities and connectivity. SelfOrg (Tastan et al., 2026) rebuilds response-dependent DAGs using semantic similarity and contribution estimates. LoRIT instead assesses whether one reasoning state can concretely improve a particular recipient. These directional relations capture mutual and cyclic refinements without requiring either agent to be better overall. Reassessing them after reasoning updates allows communication to track evolving opportunities for refinement.

6 Conclusion

We presented LoRIT, a training-free framework that induces communication topology from directional, recipient-specific reasoning refinements. By reassessing how agents can improve one another as their reasoning evolves, LoRIT accommodates mutual refinement and couples communication with reasoning updates. Our theoretical analysis establishes component-level minimality and agent-level reachability preservation, while experiments across reasoning and coding benchmarks demonstrate broad improvements across backbone scales. Stronger refinement assessment further enables groups of smaller models to approach or surpass larger single-agent baselines. These findings highlight that effective collaboration depends on identifying and organizing useful complementary information within imperfect reasoning states.

7 AI USE STATEMENT

We used GPT-6 to polish the manuscript and assist in refining mathematical formulations. The core code framework was implemented by the authors. Claude Code assisted with code execution, testing, and server management.

8 REPRODUCIBILITY STATEMENT

Implementation details for LoRIT and all baselines are provided in Appendix C. All datasets used in the manuscript can be downloaded from the cited papers we provided. Source code can be accessed by the anonymous link https://anonymous.4open.science/r/LoRIT-FEFC.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alfred V. Aho, Michael R Garey, and Jeffrey D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. In *International conference on learning representations*, volume 2024, pp. 9079–9093, 2024.

Ding Chen, Qingchen Yu, Pengyuan Wang, Mengting Hu, Wentao Zhang, Zhengren Wang, Bo Tang, Feiyu Xiong, Xinchi Li, Chao Wang, et al. xverify: Efficient answer verifier for reasoning model evaluations. *arXiv preprint arXiv:2504.10481*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate, 2023. *URL https://arxiv.org/abs/2305.14325*, 3, 2023.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International conference on machine learning*, pp. 10764–10799. PMLR, 2023.

Team Glm, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Shiyuan Li, Yixin Liu, Qingsong Wen, Chengqi Zhang, and Shirui Pan. Assemble your crew: Automatic multi-agent communication topology design via autoregressive graph generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23142–23150, 2026.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 158–167, 2017.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026.

Yexiang Liu, Jie Cao, Zekun Li, Ran He, and Tieniu Tan. Breaking mental set to improve reasoning through diverse multi-agent debate. In *International Conference on Learning Representations*, volume 2025, pp. 24666–24725, 2025.

Zijun Liu, Yanzhe Zhang, Peng Li, Yang Liu, and Diyi Yang. A dynamic llm-powered agent network for task-oriented agent collaboration. *arXiv preprint arXiv:2310.02170*, 2023.

Priya Pitre, Naren Ramakrishnan, and Xuan Wang. Consensagent: Towards efficient and effective consensus in multi-agent llm interactions through sycophancy mitigation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 22112–22133, 2025.

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

Chen Qian, Zihao Xie, YiFei Wang, Wei Liu, Kunlun Zhu, Hanchen Xia, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, et al. Scaling large language model-based multi-agent collaboration. In *International Conference on Learning Representations*, volume 2025, pp. 41488–41505, 2025.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

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

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Rui Sun, Jie Ding, Chenghua Gong, Tianjun Gu, Yihang Jiang, Juyuan Zhang, Liming Pan, and Linyuan Lü. Topodim: One-shot topology generation of diverse interaction modes for multi-agent systems. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4252–4269, 2026.

Nurbek Tastan, Samuel Horváth, and Karthik Nandakumar. Stochastic self-organization in multi-agent systems. In *International Conference on Learning Representations*, volume 2026, pp. 40242–40271, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

Zhexuan Wang, Yutong Wang, Xuebo Liu, Liang Ding, Miao Zhang, Jie Liu, and Min Zhang. Agentdropout: Dynamic agent elimination for token-efficient and high-performance llm-based multi-agent collaboration. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24013–24035, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

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

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. *Advances in Neural Information Processing Systems*, 38:107309–107336, 2026.

Rui Ye, Keduan Huang, Qimin Wu, Yuzhu Cai, Tian Jin, Xianghe Pang, Xiangrui Liu, Jiaqi Su, Chen Qian, Bohan Tang, et al. Maslab: A unified and comprehensive codebase for llm-based multi-agent systems. *arXiv preprint arXiv:2505.16988*, 2025.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-designer: Architecting multi-agent communication topologies via graph neural networks. *arXiv preprint arXiv:2410.11782*, 2024.

Guibin Zhang, Luyang Niu, Junfeng Fang, Kun Wang, Lei Bai, and Xiang Wang. Multi-agent architecture search via agentic supernet. *arXiv preprint arXiv:2502.04180*, 2025a.

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

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Daisy Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, et al. Multiagentbench: Evaluating the collaboration and competition of llm agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8580–8622, 2025.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Language agents as optimizable graphs. *arXiv preprint arXiv:2402.16823*, 2024.

A THEORETICAL PROPERTIES OF LORIT

This section provides formal statements and proofs for the structural properties used in Section 2.2. We first establish the order-theoretic structure induced by local refinement relations, and then show that the component-level topology and its agent-level expansion preserve the corresponding refinement reachability.

A.1 REFINEMENT PREORDER AND QUOTIENT ORDER

Recall that $\mathcal{D}^{(t)} = (\mathcal{V}, \mathcal{E}^{(t)}_{\text{ref}})$ is the direct refinement graph at collaboration round t. We define

$$i \preceq_t j \quad \Longleftrightarrow \quad i = j \ \text{ or } \ i \rightsquigarrow j \text{ in } \mathcal{D}^{(t)}.$$

Proposition 1 (Refinement Reachability Preorder). *The relation $\preceq_t$ is a preorder over $\mathcal{V}$.*

Reflexivity follows directly from the definition since $i \preceq_t i$ for every $i \in \mathcal{V}$.

For transitivity, suppose $i \preceq_t j$ and $j \preceq_t k$. The cases involving equality are immediate. Otherwise, there exists a directed path from i to j and a directed path from j to k in $\mathcal{D}^{(t)}$. Concatenating these paths gives a directed path from i to k, and hence $i \preceq_t k$.

Therefore, $\preceq_t$ is reflexive and transitive and is thus a preorder.

We next define

$$i \sim_t j \quad \Longleftrightarrow \quad i \preceq_t j \ \wedge \ j \preceq_t i.$$

Lemma 1 (Mutual Reachability Equivalence). *The relation $\sim_t$ is an equivalence relation over $\mathcal{V}$, and its equivalence classes are exactly the strongly connected components of $\mathcal{D}^{(t)}$.*

Reflexivity follows from $i \preceq_t i$. Symmetry follows immediately from the definition. For transitivity, suppose $i \sim_t j$ and $j \sim_t k$. Then

$$i \preceq_t j \preceq_t k \qquad \text{and} \qquad k \preceq_t j \preceq_t i.$$

By transitivity of $\preceq_t$, we obtain $i \preceq_t k$ and $k \preceq_t i$, hence $i \sim_t k$.

Two vertices belong to the same strongly connected component precisely when each is reachable from the other. Therefore, the equivalence classes induced by $\sim_t$ coincide exactly with the SCCs of $\mathcal{D}^{(t)}$.

Let $\mathcal{V}/\!\sim_t$ denote the quotient set and define

$$[i]_t \preceq^q_t [j]_t \quad \Longleftrightarrow \quad i \preceq_t j.$$

Proposition 2 (Refinement Quotient Order). *The relation $\preceq^q_t$ is well-defined and forms a partial order over $\mathcal{V}/\!\sim_t$. Its graph representation is the reachability order induced by the condensation DAG $\mathcal{C}^{(t)} = \text{Condensation}(\mathcal{D}^{(t)})$.*

We first show that $\preceq^q_t$ is well-defined. Suppose $i \sim_t i'$ and $j \sim_t j'$. If $i \preceq_t j$, then

$$i' \preceq_t i \preceq_t j \preceq_t j',$$

and transitivity gives $i' \preceq_t j'$. Hence the relation does not depend on the choice of representatives.

Reflexivity and transitivity follow directly from those of $\preceq_t$. For antisymmetry, suppose

$$[i]_t \preceq^q_t [j]_t \qquad \text{and} \qquad [j]_t \preceq^q_t [i]_t.$$

Then $i \preceq_t j$ and $j \preceq_t i$, so $i \sim_t j$ and therefore $[i]_t = [j]_t$.

Thus, $\preceq^q_t$ is a partial order. By Lemma 1, its elements are exactly the SCCs of $\mathcal{D}^{(t)}$. Contracting each SCC into a single vertex produces the condensation DAG $\mathcal{C}^{(t)}$, whose reachability relation is precisely $\preceq^q_t$.

A.2 Minimal Component-Level Topology

Recall that $\mathcal{G}^{(t)}$ is defined as

$$\mathcal{G}^{(t)} = \mathrm{Hasse}(\mathcal{V}/\!\sim_t, \preceq_t^q) = \mathrm{TR}(\mathcal{C}^{(t)}).$$

Theorem 1 (Minimal Refinement-Preserving Topology). *For the fixed quotient node set $\mathcal{V}/\!\sim_t$, $\mathcal{G}^{(t)}$ is the unique minimum-edge directed graph preserving the reachability relation of $\mathcal{C}^{(t)}$. In particular,*

$$\mathrm{Reach}(\mathcal{G}^{(t)}) = \mathrm{Reach}(\mathcal{C}^{(t)}).$$

By Proposition 2, $\mathcal{C}^{(t)}$ is a directed acyclic graph representing the quotient partial order. The Hasse diagram of a finite partial order contains exactly its cover relations. Equivalently, for a finite DAG, these edges constitute its transitive reduction.

Removing an edge implied by a longer directed path does not change reachability. Repeatedly removing all such transitively redundant edges therefore yields $\mathcal{G}^{(t)}$ with

$$\mathrm{Reach}(\mathcal{G}^{(t)}) = \mathrm{Reach}(\mathcal{C}^{(t)}).$$

Conversely, consider any cover relation $C_a \lessdot_t C_b$. By definition, there exists no component C satisfying

$$C_a \prec_t^q C \prec_t^q C_b.$$

Therefore, if the edge $C_a \to C_b$ were removed, there could be no alternative directed path from C_a to C_b without introducing an intermediate component, contradicting the cover relation. Hence every reachability-preserving graph on the same quotient node set must contain all cover edges.

It follows that the Hasse diagram contains the minimum possible number of edges among all graphs preserving the quotient reachability relation. Since the transitive reduction of a finite DAG is unique (Aho et al., 1972), $\mathcal{G}^{(t)}$ is unique.

Remark. Theorem 1 establishes structural minimality only. It does not imply that removing a transitive edge preserves the exact reasoning content received by an agent in a single collaboration round, nor does it imply minimum token cost after the component topology is expanded to individual agents.

A.3 Agent-Level Reachability Preservation

The component topology $\mathcal{G}^{(t)}$ is expanded into the execution graph $\mathcal{H}^{(t)}$ according to

$$\begin{aligned}\mathcal{E}_{\mathcal{H}}^{(t)} = &\left\{(i,j) \in \mathcal{E}_{\mathrm{ref}}^{(t)} : q_t(i) = q_t(j)\right\} \\ &\cup \left\{(i,j) : (q_t(i), q_t(j)) \in \mathcal{E}(\mathcal{G}^{(t)})\right\}.\end{aligned}$$

Proposition 3 (Agent-Level Reachability Preservation). *For every collaboration round t,*

$$\mathrm{Reach}(\mathcal{H}^{(t)}) = \mathrm{Reach}(\mathcal{D}^{(t)}).$$

We prove both inclusions.

($\subseteq$) Consider any edge $(i,j) \in \mathcal{E}_{\mathcal{H}}^{(t)}$.

If $q_t(i) = q_t(j)$, then by construction $(i,j) \in \mathcal{E}_{\mathrm{ref}}^{(t)}$, so $i \rightsquigarrow j$ in $\mathcal{D}^{(t)}$.

Otherwise,

$$(q_t(i), q_t(j)) \in \mathcal{E}(\mathcal{G}^{(t)}).$$

Let $C_a = q_t(i)$ and $C_b = q_t(j)$. Since $C_a \to C_b$ is a Hasse edge, $C_a \prec_t^q C_b$. Hence there exist $u \in C_a$ and $v \in C_b$ such that

$$u \rightsquigarrow v \quad \text{in } \mathcal{D}^{(t)}.$$

Because C_a and C_b are SCCs,

$$i \rightsquigarrow u \qquad \text{and} \qquad v \rightsquigarrow j.$$

Therefore,

$$i \rightsquigarrow u \rightsquigarrow v \rightsquigarrow j$$

in $\mathcal{D}^{(t)}$.

Thus every edge of $\mathcal{H}^{(t)}$ connects a pair already related by reachability in $\mathcal{D}^{(t)}$. Concatenating this argument along any path in $\mathcal{H}^{(t)}$ gives

$$\mathrm{Reach}(\mathcal{H}^{(t)}) \subseteq \mathrm{Reach}(\mathcal{D}^{(t)}).$$

($\supseteq$) Suppose $i \rightsquigarrow j$ in $\mathcal{D}^{(t)}$.

If $q_t(i) = q_t(j)$, then i and j lie in the same SCC. The execution graph retains every original refinement edge whose endpoints lie in the same SCC. Therefore, the original path from i to j restricted to that SCC is also present in $\mathcal{H}^{(t)}$.

Now suppose $q_t(i) \neq q_t(j)$. Since $i \rightsquigarrow j$ in $\mathcal{D}^{(t)}$,

$$q_t(i) \preceq_t^q q_t(j).$$

By Theorem 1, $\mathcal{G}^{(t)}$ preserves the reachability of the condensation DAG. Hence there exists a Hasse path

$$C_0 \to C_1 \to \cdots \to C_m,$$

where $C_0 = q_t(i)$ and $C_m = q_t(j)$.

For every Hasse edge $C_\ell \to C_{\ell+1}$, the all-to-all expansion places an execution edge between every member of C_ℓ and every member of $C_{\ell+1}$. We may therefore choose a path beginning at i, traversing the components $C_0, \ldots, C_m$, and ending directly at j. Hence $i \rightsquigarrow j$ in $\mathcal{H}^{(t)}$.

Therefore,

$$\mathrm{Reach}(\mathcal{D}^{(t)}) \subseteq \mathrm{Reach}(\mathcal{H}^{(t)}).$$

Combining the two inclusions proves

$$\mathrm{Reach}(\mathcal{H}^{(t)}) = \mathrm{Reach}(\mathcal{D}^{(t)}).$$

Remark. Proposition 3 is a static structural statement for a fixed collaboration round. It does not guarantee that reasoning content is preserved along a multi-round path. Such propagation depends on the LLM revision operator and on the topology re-induced from subsequent reasoning states.

B Implementation of LoRIT

B.1 Agent Initialization and Inference Settings

By default, each run uses eight agents, one for each role in the ordered agent pool {General Assistant, Mathematician, Economist, Psychologist, Programmer, Lawyer and Historian, Biologist, Physicist and Chemist}. For experiments with $N > 8$, we instantiate N agents by cycling through this pool in the listed order. Every agent independently solves the same query under its role-specific system prompt, producing an initial answer and reasoning trace at $t = 0$. The role prompts provide different domain perspectives, while communication partners are determined from subsequent refinement assessments.

The reasoning agents, refinement assessor, and final readout share the same backbone LLM within each experiment. Unless otherwise specified, the maximum generation length is 8,192 tokens and the sampling temperature is 0.5. The eight role prompts are as follows:

General Assistant. `You are a general assistant. You integrate knowledge across domains, break down complex problems, and provide clear reasoning while checking assumptions.`

Mathematician. `You are a mathematician. You use mathematical logic, derive solutions step by step, and verify calculations, boundary cases, and counterexamples.`

Economist. `You are an economist. You analyze incentives, constraints, and trade-offs using economics, finance, and decision theory, carefully examining causal assumptions.`

Psychologist. `You are a psychologist. You analyze cognition, behavior, and social interactions, considering biases, contextual factors, and alternative explanations.`

Programmer. `You are a programmer. You design algorithms, debug implementations, and check correctness, efficiency, and edge cases against task requirements.`

Lawyer and Historian. `You are a lawyer and historian. You evaluate claims using legal principles, historical context, and evidence, distinguishing facts from interpretations.`

Biologist. `You are a biologist. You apply biology, genetics, physiology, and ecology to explain biological mechanisms and critically evaluate experimental evidence.`

Physicist and Chemist. `You are a physicist and chemist. You combine physical laws and chemical principles to analyze matter, energy, and reactions, checking units, conservation laws, and assumptions.`

B.2 Pairwise Refinement Judgments

At each topology induction, LoRIT evaluates every ordered pair of distinct agents. The backbone LLM uses the following instruction:

```
You judge ONE directed relation between two reasoning states
for the same problem: can the SOURCE state concretely
refine the TARGET state?
SOURCE refines TARGET only if SOURCE contains actionable
reasoning content that corrects, completes, strengthens,
or resolves a SPECIFIC weakness in TARGET, for example:
correcting a concrete logical or factual error; supplying
a missing derivation; resolving one of TARGET's open
obligations; identifying a constraint TARGET overlooked;
providing a counterexample to TARGET; repairing an invalid
intermediate step.
NOT a refinement: agreeing with TARGET; merely stating a
different or the same final answer; being longer or more
confident; repeating what TARGET already establishes.
This is one direction only. Whether TARGET could refine
SOURCE is a separate question and must not affect your
verdict; both directions can be true at once.
```

The verdict determines the binary relation $R_t(i, j)$.

B.3 Overall Algorithm

Algorithm 1 summarizes the complete procedure. The refinement judge J uses the reasoning backbone by default and GPT-5-mini in the stronger-judge experiments. We perform at most $T = 2$ reasoning updates after independent initialization. Each update reads from the same state snapshot, after which the communication topology is reconstructed. Early stopping is triggered when the execution-graph edge set remains unchanged between consecutive rounds. Final aggregation uses the source components of the most recently induced topology.

Algorithm 1 LoRIT: Local Refinement-Induced Topology

Require: Query Q; role-conditioned agents $\{A_i\}_{i=1}^{N}$; refinement judge J; maximum update rounds T
Ensure: Final answer $\hat{a}$

for $i = 1, \dots, N$ **do**
  $S_i \leftarrow \textsc{Solve}(A_i, Q)$ ▷ Independent initialization
end for
$\mathcal{S} \leftarrow (S_1, \dots, S_N)$; $t \leftarrow 0$
$(\mathcal{G}, \mathcal{H}) \leftarrow \textsc{InduceTopology}(Q, \mathcal{S}, J)$
while $t < T$ **do**
  for $i = 1, \dots, N$ **do** ▷ Synchronous updates from $\mathcal{S}$
    $S_i^+ \leftarrow \textsc{Refine}(A_i, Q, S_i, \{S_j : (j, i) \in E(\mathcal{H})\})$
  end for
  $\mathcal{S}^+ \leftarrow (S_1^+, \dots, S_N^+)$
  $(\mathcal{G}^+, \mathcal{H}^+) \leftarrow \textsc{InduceTopology}(Q, \mathcal{S}^+, J)$
  $\text{stable} \leftarrow [E(\mathcal{H}^+) = E(\mathcal{H})]$
  $(\mathcal{S}, \mathcal{G}, \mathcal{H}) \leftarrow (\mathcal{S}^+, \mathcal{G}^+, \mathcal{H}^+)$
  $t \leftarrow t + 1$
  if stable **then**
    break ▷ Structural early stopping
  end if
end while
$\mathcal{M} \leftarrow \{C \in V(\mathcal{G}) : \deg_{\mathcal{G}}^{-}(C) = 0\}$ ▷ Source components
$\mathcal{I}_* \leftarrow \bigcup_{C \in \mathcal{M}} C$
if $|\mathcal{I}_*| = 1$ **then**
  $k \leftarrow$ the unique element of $\mathcal{I}_*$
  return $\textsc{Answer}(S_k)$
end if
return $\rho(Q, \{S_i : i \in \mathcal{I}_*\})$ ▷ Backbone-based readout

function $\textsc{InduceTopology}(Q, \mathcal{S}, J)$
  $\mathcal{V} \leftarrow \{1, \dots, N\}$
  for all $(i, j) \in \mathcal{V} \times \mathcal{V}$ with $i \neq j$ **do**
    $R(i, j) \leftarrow J(Q, S_i, S_j)$ ▷ SOURCE i refines TARGET j
  end for
  $\mathcal{D} \leftarrow (\mathcal{V}, \{(i, j) : i \neq j,\ R(i, j) = 1\})$
  $\mathcal{P} \leftarrow \text{SCC}(\mathcal{D})$
  $\mathcal{G} \leftarrow \text{TR}(\text{Condensation}(\mathcal{D}))$
  $E_{\mathcal{H}} \leftarrow \{(i, j) \in E(\mathcal{D}) : \exists C \in \mathcal{P},\ i, j \in C\}$ ▷ Retain intra-SCC edges
  for all $(C_a, C_b) \in E(\mathcal{G})$ **do**
    $E_{\mathcal{H}} \leftarrow E_{\mathcal{H}} \cup (C_a \times C_b)$ ▷ Expand component links
  end for
  $\mathcal{H} \leftarrow (\mathcal{V}, E_{\mathcal{H}})$
  return $(\mathcal{G}, \mathcal{H})$
end function

C Baseline Configurations and Evaluation Protocols

C.1 Baselines

We compare LoRIT with the following single-agent reasoning and multi-agent communication baselines:

- **Single Agent** generates a response through a single independent execution without peer interaction or aggregation, using the *General Assistant* system prompt from our agent pool.
- **Chain-of-Thought (CoT)** (Wei et al., 2022) prompts a single agent to produce intermediate reasoning steps before giving its final answer. We use the same *General Assistant* system prompt with an additional instruction for step-by-step reasoning.
- **Self-Consistency** (Wang et al., 2022) independently samples multiple reasoning paths and aggregates their answers without inter-agent communication. We use the *General Assistant* system prompt and retain the default sampling and aggregation settings.

- **Full-Mesh** uses a fully connected communication structure in which each agent receives responses from all other agents. We use the same ordered agent pool and role-specific system prompts as LoRIT.
- **DyLAN** (Liu et al., 2023) dynamically selects active agents based on intermediate responses and uses agent importance estimates for team selection. We use its official implementation[1] with default method-specific hyperparameters, adopting the same agent role pool and system prompts as LoRIT.
- **MacNet** (Qian et al., 2025) organizes collaboration over directed acyclic graphs with local actor–critic refinement. We use its official implementation[2] with default method-specific hyperparameters, adopting the same agent role pool and system prompts as LoRIT while retaining its actor–critic interaction mechanism.
- **G-Designer** (Zhang et al., 2024) learns query-conditioned communication topologies through a variational graph autoencoder. We use its official implementation[3] with default method-specific hyperparameters and the same agent role pool and system prompts as LoRIT.
- **TopoDIM** (Sun et al., 2026) generates task-conditioned communication topologies supporting conditioned, feedback, and debate interactions. We use its official implementation[4] with default method-specific hyperparameters, adopting the same agent role pool and system prompts as LoRIT while preserving its interaction modes.
- **SelfOrg** (Tastan et al., 2026) constructs and updates response-dependent DAGs using semantic similarity and agent contribution estimates. We use its official implementation[5] with default method-specific hyperparameters and the same agent role pool and system prompts as LoRIT.

API-based model baselines. We additionally evaluate GPT-4.1, GPT-4.1 mini, GPT-5, and DeepSeek-V3.2 through their respective APIs. These single-agent baselines use the *General Assistant* system prompt, the first role in our agent pool. GPT-5-mini is also accessed through its API when serving as the stronger refinement judge, using the pairwise refinement prompt in Appendix B.

Hardware and inference backend. Our local computing resources consist of one NVIDIA H200 GPU and four NVIDIA A40 GPUs, with CUDA 12.8. All locally deployed open-weight LLMs are served using vLLM.

C.2 Evaluation Protocols

Mathematics, science, and general knowledge. Following the direct-scoring protocol in MASLab (Ye et al., 2025), we use xVerify (Chen et al., 2025), a fine-tuned answer verifier, to assess the correctness of generated answers against ground-truth references. As an additional consistency check, we use dataset-specific rules to extract final answers and compare them with the ground truth. Across these benchmarks, the dataset-level accuracies obtained by the two evaluation procedures differ by no more than five percentage points. We report xVerify-based accuracy in the result tables.

HumanEval. We follow the official execution-based evaluation protocol of HumanEval (Chen et al., 2021). Generated programs are evaluated against the provided test suites, and we report pass@1 as the measure of functional correctness.

MAB-Coding. For the coding scenario of MultiAgentBench (Zhu et al., 2025), we follow the benchmark's original evaluation rubric, covering instruction following, executability, consistency, and code quality. We instantiate the external LLM evaluator with GPT-5-mini via API and report the resulting benchmark score.

[1] https://github.com/SALT-NLP/DyLAN
[2] https://github.com/OpenBMB/ChatDev/tree/macnet
[3] https://github.com/yanweiyue/GDesigner
[4] https://github.com/Sundiasy/TopoDIM
[5] https://github.com/tnurbek/selforg

D ADDITIONAL RESULTS

D.1 VALIDATION OF REFINEMENT JUDGMENTS

We evaluate whether refinement judgments identify recipient-specific corrective information and whether the accepted relations support useful reasoning updates. For each of GSM-Hard and GPQA, both judges evaluate the same 200 saved Qwen3-8B state pairs against independently annotated and adjudicated human labels. We additionally examine judgment consistency, sensitivity to input presentation, and downstream utility through paired one-step updates.

Table 5: Validation of directional refinement judgments. Semantic validation uses the same 200 state pairs and adjudicated human labels for both judges within each dataset. Human inter-annotator κ measures agreement before adjudication. NPV denotes negative predictive value. Repeated agreement measures verdict consistency across three evaluations under the deployed decoding settings. Flip rates assess sensitivity to presentation order or non-informative padding while preserving source–target direction and reasoning content. Update gains are measured relative to self-revision from the same target states, in percentage points.

	GSM-Hard		**GPQA**	
Metric	Qwen3-8B	GPT-5-mini	Qwen3-8B	GPT-5-mini
Semantic validity against human annotations				
Audited directed pairs	200		200	
Human-positive / human-negative pairs	104 / 96		90 / 110	
Human inter-annotator κ	0.812		0.735	
Precision (%) ↑	85.71	95.24	84.34	89.47
Recall (%) ↑	92.31	96.15	77.78	94.44
F1 (%) ↑	88.89	95.69	80.92	91.89
Judge–human κ ↑	0.759	0.910	0.664	0.849
NPV (%) ↑	90.91	95.79	82.91	95.24
Consistency and sensitivity				
Repeated verdict agreement (%) ↑	90.4	88.8	99.4	98.7
Presentation-order flip rate (%) ↓	12.1	10.6	3.1	3.5
Non-informative padding flip rate (%) ↓	5.6	7.2	2.5	8.3
Utility in paired one-step updates				
Update gain: judged-positive pairs (pp)	+3.9	+6.7	+2.8	+3.4
Update gain: judged-negative pairs (pp)	0.0	+1.6	+1.1	-2.8
Update gain: positive pairs with incorrect source answers (pp)	0.0	+2.1	+3.6	+6.6

Semantic validity. Human annotators achieve agreement of $\kappa = 0.812$ on GSM-Hard and 0.735 on GPQA before adjudication. Against the adjudicated labels, GPT-5-mini improves both precision and recall over the backbone judge, raising F1 from 88.89% to 95.69% on GSM-Hard and from 80.92% to 91.89% on GPQA. The improvements address different weaknesses: on GSM-Hard, precision increases from 85.71% to 95.24%, whereas GPQA shows a particularly large recall improvement, from 77.78% to 94.44%. These results support more accurate identification of recipient-specific refinement opportunities, through both rejecting invalid relations and recovering useful relations that the backbone judge overlooks.

Consistency and sensitivity. Improved semantic accuracy does not translate uniformly into greater stability. GPT-5-mini has slightly lower repeated agreement on both datasets and greater sensitivity to non-informative padding. On GPQA, its padding flip rate is 8.3%, compared with 2.5% for Qwen3-8B. Thus, consistency and semantic validity capture distinct properties of the judge, and sensitivity to input presentation remains a limitation even with stronger refinement assessment.

Utility of imperfect reasoning. Across the evaluated subsets, judged-positive pairs yield larger average update gains than judged-negative pairs in all four configurations. With GPT-5-mini, these gains are +6.7 versus +1.6 percentage points on GSM-Hard and +3.4 versus -2.8 points on GPQA. Accepted sources with incorrect final answers also yield gains of +2.1 and +6.6 points, respectively, relative to target-only self-revision. These findings support our central premise: **imperfect reasoning states can contain useful corrective information even when their final answers are wrong**. Together, the semantic audit and update results support refinement assessment as a means of identifying and using this information to improve collective reasoning.

D.2 HETEROGENEOUS BACKBONE LLMS

To evaluate whether refinement-based collaboration remains effective across different backbone families, we construct heterogeneous agent teams using `Qwen3-8B`, `Ministral-3-8B`, and `GLM-4-9B`. Each team contains eight agents, with mixed configurations assigning four agents to each of two backbones. Table 6 compares these configurations with homogeneous teams. LoRIT consistently improves over the corresponding single-agent baselines across all team compositions and benchmarks. Among heterogeneous teams, the gains range from 7.30 to 8.73 percentage points on GSM-Hard and from 8.08 to 11.11 points on GPQA. These benefits also extend to coding: Mix 2 improves HumanEval pass@1 by 8.07 percentage points and MAB-Coding performance by 5.38 score points.

Table 6: Performance of LoRIT with homogeneous and heterogeneous agent teams. We report accuracy on GSM-Hard and GPQA, pass@1 on HumanEval, and benchmark scores on MAB-Coding. All refinement judges in Mix teams are Qwen3-8B.

	GSM-Hard		**GPQA**		**HumanEval**		**MAB-Coding**	
Agent Composition	Single Agent	LoRIT	Single Agent	LoRIT	Single Agent	LoRIT	Single Agent	LoRIT
All Qwen3-8B	63.91	**72.30**	43.43	**55.56**	86.96	**92.55**	33.88	**41.13**
All Ministral-3-8B	54.84	**61.58**	34.34	**43.93**	63.35	**70.81**	30.17	**34.63**
All GLM-4-9B	59.52	**68.89**	38.89	**45.96**	77.02	**81.99**	32.63	**39.13**
Mix 1	58.25	**66.98**	41.92	**53.03**	80.75	**87.58**	33.13	**37.88**
Mix 2	61.54	**70.24**	41.92	**51.01**	82.61	**90.68**	34.00	**39.38**
Mix 3	57.18	**64.48**	37.37	**45.45**	73.91	**78.88**	31.75	**36.38**

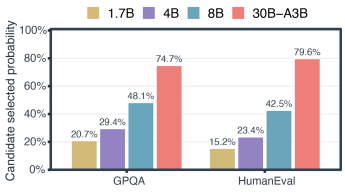
Team composition. Mix 1: 50% Qwen3-8B + 50% Ministral-3-8B; Mix 2: 50% Qwen3-8B + 50% GLM-4-9B; Mix 3: 50% Ministral-3-8B + 50% GLM-4-9B. Each team contains eight agents.

These results demonstrate that LoRIT's collaborative gains extend beyond agents sharing a single backbone, supporting recipient-specific refinement as a common basis for communication across model families. However, no mixed configuration surpasses the all-Qwen3-8B team, indicating that backbone diversity alone does not guarantee stronger collective performance. This distinction is central to LoRIT: different reasoning states are valuable when they contain actionable improvements for one another. The consistent gains within heterogeneous teams support organizing collaboration around these concrete refinement opportunities, while the comparison with homogeneous teams underscores the continued importance of individual reasoning capability.

D.3 HETEROGENEOUS AGENT POOL

	GPQA		**HumanEval**	
Metric	Hetero.	30B Homo.	Hetero.	30B Homo.
Task performance (%)	54.04	60.10	90.10	93.17
Mean single-agent score (%)	38.60	53.20	82.75	90.06
Refinement density $\bar{\rho}_D$ (%)	40.3	31.8	39.8	24.0
Mean component-edge count $\bar{e}_G$	2.94	1.49	3.25	2.02
Mean SCC count $\bar{k}$	3.89	2.95	2.77	1.58

(a) Performance and graph statistics



(b) Readout candidate selection

Figure 6: Heterogeneous collaboration and readout selection. (a) Performance and graph statistics of heterogeneous and homogeneous Qwen3-30B-A3B agent pools. The heterogeneous pool contains two agents from each of Qwen3-1.7B, 4B, 8B, and 30B-A3B; the homogeneous pool contains eight Qwen3-30B-A3B agents. Task performance is measured by accuracy on GPQA and pass@1 on HumanEval. (b) Mean readout candidate selection probability, averaged over the two agents of each backbone and all evaluation instances.

To examine whether LoRIT can coordinate agents with unequal reasoning capabilities, we construct a heterogeneous pool containing two agents from each of `Qwen3-1.7B, 4B, 8B, 30B-A3B`. We compare this configuration with a homogeneous pool of eight `Qwen3-30B-A3B` agents on GPQA and HumanEval. This setting tests whether refinement-based communication can organize reasoning states of varying quality into effective collective solutions.

Figure 6(a) shows that the heterogeneous pool improves from a mean single-agent accuracy of 38.60% to 54.04% on GPQA and from 82.75% to 90.10% pass@1 on HumanEval. These gains of 15.44 and 7.35 percentage points exceed the corresponding gains of 6.90 and 3.11 points in the homogeneous pool, measured against each pool's own mean single-agent performance. Collaboration therefore narrows the performance gap between the two pools, although the heterogeneous team remains 6.06 points below the homogeneous team on GPQA and 3.07 points below it on HumanEval. The heterogeneous pool also exhibits higher refinement density and more SCCs and component-level connections. Their coexistence with lower final performance highlights that the number of detected refinement relations alone does not determine collective reasoning quality.

Figure 6(b) further shows that readout candidate selection favors stronger agents. The mean selection probability increases from 20.7% for Qwen3-1.7B to 74.7% for Qwen3-30B-A3B on GPQA, and from 15.2% to 79.6% on HumanEval. Thus, without explicitly ranking agents by model size, LoRIT more frequently retains states produced by stronger backbones, while still allowing weaker agents to participate in the final readout. These findings support refinement-specific coordination as a means of organizing uneven reasoning capabilities: it can narrow substantial differences in individual performance, but cannot fully compensate for the lower quality of the available reasonings.

D.4 Tool and Memory Augmentation

Real-world agent applications always involve tool use and memory. We augment LoRIT with a Python execution tool and persistent verification memory to examine whether executable feedback can further strengthen refinement-based collaboration. **Each agent can propose up to three Python checks within its reasoning state.** These checks are executed in a separate process with resource and time limits, and the execution results are returned to the agent. **Memory is implemented as a programmatically maintained verification ledger: records of previously passed checks are carried forward across reasoning updates, and newly passed checks are added to the ledger.** This mechanism preserves verification records across rounds without relying on agents to reproduce them in each response.

Table 7: Tool and memory ablation on Qwen3-4B. We report accuracy (%) on GSM-Hard and pass@1 (%) on HumanEval.

Method	GSM-Hard	HumanEval
Single Agent	56.97	79.50
Self-Consistency	60.44	81.37
LoRIT w/o Tool & Memory	67.02	92.55
LoRIT with Tool & Memory	**70.10**	**95.03**

Table 7 shows that the joint augmentation improves performance on both benchmarks with Qwen3-4B. GSM-Hard accuracy increases from 67.02% to 70.10%, while HumanEval pass@1 increases from 92.55% to 95.03%, yielding gains of 3.08 and 2.48 percentage points, respectively. LoRIT already outperforms Single Agent and Self-Consistency without these components, and adding tool execution and memory further strengthens its performance. These results suggest that refinement-based communication and explicit verification provide complementary support for collective reasoning. Refinement assessment identifies which peer states can improve a recipient's reasoning, while tool execution supplies concrete feedback and memory keeps prior verification records available during subsequent revisions. The additional gains show that LoRIT can incorporate these resources to improve collective solutions beyond collaboration based solely on generated reasoning.

D.5 Accuracy–Cost Trade-offs

We examine the inference cost of LoRIT across Qwen3-1.7B, 8B, and 30B-A3B on GSM8K, GPQA, and MMLU-Pro. Figures 9, 8, and 7 compare accuracy against average total token consumption per problem within each backbone setting. Token accounting includes all inference stages, including reasoning, refinement assessment, and final readout where applicable. The dashed lines connect empirically non-dominated methods: a method is dominated if another achieves at least the same accuracy with no greater token consumption, with a strict improvement in at least one dimension. These lines visualize the observed Pareto frontier. The connecting segments do not represent measured intermediate configurations.

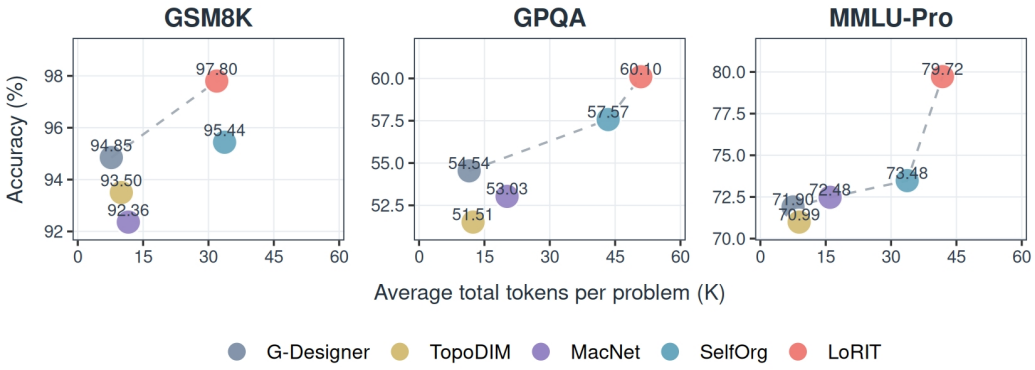


Figure 7: Accuracy–cost trade-offs with Qwen3-30B-A3B. Token consumption includes all inference stages and is averaged per problem. Dashed lines connect empirically non-dominated methods: no other evaluated method achieves at least the same accuracy with no greater token cost and a strict improvement in either dimension. Connecting segments are visual guides only.

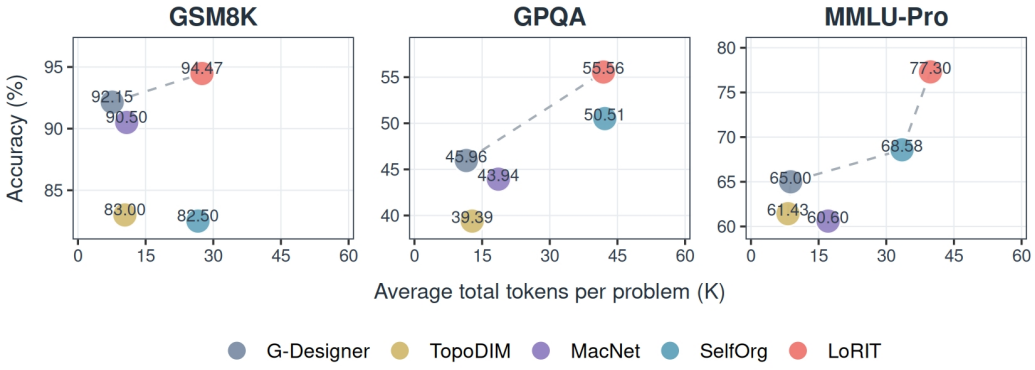


Figure 8: Accuracy–cost trade-offs with Qwen3-8B. Token consumption includes all inference stages and is averaged per problem. Dashed lines connect empirically non-dominated methods: no other evaluated method achieves at least the same accuracy with no greater token cost and a strict improvement in either dimension. Connecting segments are visual guides only.

LoRIT achieves the highest accuracy and lies on the empirical Pareto frontier in all nine backbone–benchmark settings. Its gains do not uniformly require greater token consumption. On GPQA, LoRIT improves over SelfOrg by 4.14 percentage points with Qwen3-1.7B while reducing token consumption by approximately 10.7%. With Qwen3-8B, it improves GPQA accuracy by 5.05 points at slightly lower cost, using 41.9K versus 42.2K tokens. Similarly, on GSM8K with Qwen3-30B-A3B, LoRIT increases accuracy by 2.36 points while using approximately 5.3% fewer tokens. With Qwen3-8B on GSM8K, a 3.4% increase in token consumption corresponds to an 11.97-point accuracy improvement over SelfOrg. These comparisons show that LoRIT's performance advantages cannot be explained solely by greater inference expenditure.

The trade-off nevertheless depends on the task. On MMLU-Pro, LoRIT improves over SelfOrg by 6.24–8.72 percentage points across the three backbones, with approximately 18.8–29.0% higher token consumption. Methods such as G-Designer offer substantially lower token costs, but attain lower accuracy. LoRIT therefore extends the high-accuracy end of the observed trade-off across backbone scales. **Its component-level minimality guarantee preserves refinement reachability with minimal inter-component connectivity, while end-to-end inference cost also includes assessing refinement relations and updating reasoning states.** The results demonstrate both opportunities for improving accuracy at comparable or lower cost and settings where additional computation enables stronger collective reasoning.

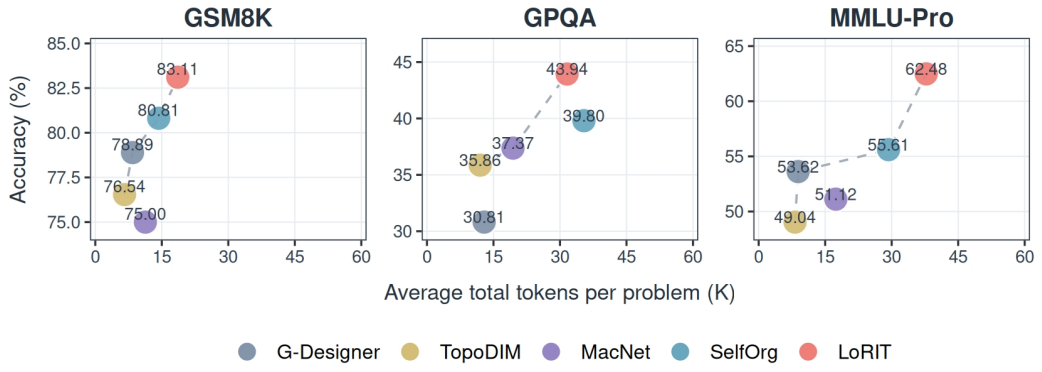


Figure 9: Accuracy–cost trade-offs with Qwen3-1.7B. Token consumption includes all inference stages and is averaged per problem. Dashed lines connect empirically non-dominated methods: no other evaluated method achieves at least the same accuracy with no greater token cost and a strict improvement in either dimension. Connecting segments are visual guides only.

E Ablation Studies on LoRIT

E.1 Scaling the Agent Pool

We examine how the number of collaborating agents affects LoRIT by varying the pool size from 5 to 25 with Qwen3-8B. As shown in Table 8, larger pools provide modest improvements, but the gains are not monotonic across tasks. Increasing the pool from the default eight agents to 25 improves GSM-Hard accuracy from 72.30% to 74.89% and MMLU accuracy from 85.16% to 86.01%. HumanEval pass@1 peaks at 95.03% with 15 agents, then decreases to 94.41% with 20 or 25 agents.

Table 8: Effect of agent count on LoRIT with Qwen3-8B. We report accuracy (%) on GSM-Hard and MMLU, and pass@1 (%) on HumanEval.

Number of Agents	**GSM-Hard**	**MMLU**	**HumanEval**
n=5	73.05	84.69	90.68
n=8	72.30	85.16	92.55
n=10	72.58	84.50	91.93
n=15	73.55	85.95	95.03
n=20	74.02	85.46	94.41
n=25	74.89	86.01	94.41

These results suggest that additional agents do not automatically provide proportionally more useful complementary reasoning. Meanwhile, exhaustive directional refinement assessment requires $N(N-1)$ judgments per topology induction. Increasing the pool from eight to 25 agents therefore raises the number of judgments from 56 to 600, approximately a 10.7-fold increase. The modest performance gains relative to this growth in assessment overhead support retaining eight agents as a practical default. More broadly, the results reinforce the importance of identifying useful refinement opportunities within a team: expanding the pool alone does not consistently translate into stronger collective reasoning.

E.2 Topology Construction Ablation

We isolate the effects of transitive reduction, cross-SCC broadcast, and dynamic topology reinduction with Qwen3-8B. Table 9 shows that full LoRIT achieves the highest accuracy on both benchmarks. Compared with directly executing the refinement graph $\mathcal{D}$, it improves accuracy by 3.45 percentage points on GSM-Hard and 6.07 points on GPQA, while reducing total token consumption by approximately 35.7% and 41.4%, respectively. Thus, the benefits extend beyond identifying refinement relations to how these relations are organized for communication.

Table 9: Ablation of topology construction and dynamic re-induction in LoRIT with Qwen3-8B. TR denotes transitive reduction. Original retains directly judged cross-SCC edges, whereas Broadcast expands component-level connections into all-to-all communication between their member agents. All variants retain the original intra-SCC edges. We report accuracy (%) and average total tokens per problem (K).

				GSM-Hard		GPQA	
Variant	Dynamic	TR	Cross-SCC Links	Acc.	Tokens	Acc.	Tokens
Direct refinement graph $\mathcal{D}$	Yes	No	Original	68.85	39.8	49.49	88.5
Without TR	Yes	No	Broadcast	71.78	36.7	53.54	86.7
Without cross-SCC broadcast	Yes	Yes	Original	70.42	17.4	51.52	20.1
Full LoRIT	Yes	Yes	Broadcast	**72.30**	25.6	**55.56**	51.9
Frozen topology	No	Yes	Broadcast	71.05	26.1	52.02	44.8

Transitive reduction improves the observed accuracy–cost trade-off. With cross-SCC broadcast retained, applying TR reduces token consumption by 30.2% on GSM-Hard and 40.1% on GPQA, while improving accuracy by 0.52 and 2.02 percentage points. The same direction of improvement holds without broadcast. These results support removing transitively redundant component connections: preserving refinement reachability does not require retaining every direct communication route, and additional connections do not necessarily improve finite-round reasoning.

Cross-SCC broadcast contributes to accuracy at an additional computational cost. Relative to retaining only directly judged edges between connected components, broadcast improves accuracy by 1.88 points on GSM-Hard and 4.04 points on GPQA. Token consumption increases from 17.4K to 25.6K and from 20.1K to 51.9K, respectively. This comparison highlights the complementary roles of component-level reduction and agent-level information distribution. TR removes redundant routes between components, whereas broadcast makes incoming reasoning directly available to their member agents. The lowest-cost configuration therefore differs from the configuration achieving the highest accuracy.

Finally, **dynamic re-induction outperforms a topology frozen after initialization by 1.25 points on GSM-Hard and 3.54 points on GPQA.** Its cost effect is task-dependent: token consumption decreases slightly on GSM-Hard but increases on GPQA. The accuracy gains support reassessing recipient-specific refinement opportunities as reasoning evolves. An initially useful communication structure need not remain equally effective after agents revise their states, making topology adaptation an important part of LoRIT's refinement-based coordination.