RELOC: RECURSIVE LOCAL CAUSAL MODELING FOR EXPLAINING AND PREDICTING LLM-BASED MULTI-AGENT BEHAVIOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM)-based multi-agent systems (MAS) are increasingly deployed to solve complex tasks; thus learning how a local intervention on one agent causally affects the overall outcome is necessary. As the MAS scales, however, the space of joint interventions grows combinatorially, making a global causal model prohibitively expensive and preventing measured effects from extending to unseen intervention queries. We introduce Recursive Local Causal Modeling (*ReLoC*), which progressively builds an approximate global causal model by learning and composing local causal models of individual agents, each built from a few interventions. Specifically, *ReLoC* learns how each agent responds to interventions on task-defined concepts. Starting from the outcome agent, *ReLoC* recursively learns how upstream agents produce these concepts, following the communication graph back to the task inputs. *ReLoC* eventually composes these local models into the global model. Across three benchmarks, three topologies, and four LLMs, *ReLoC* accurately predicts how each agent responds to concept interventions, how these changes propagate through the system, and how the final answer changes under unseen interventions, consistently outperforming an LLM that directly reasons over the full MAS under the same intervention budget.

1 INTRODUCTION

Large language model (LLM) based multi-agent systems (MAS) are increasingly deployed to solve complex tasks by distributing work across many specialized agents (Li et al., 2023; Chen et al., 2024; Hong et al., 2024; Wu et al., 2024). Diagnosing failures and understanding MAS behavior require exploring how agents and their interactions affect outcomes (Lin et al., 2026). Causal analysis studies this behavior by intervening on agent messages and tracing changes in downstream responses and outcomes (Giloni et al., 2025; Shen et al., 2025; Ma et al., 2026; Shah, 2026). Exhaustively estimating intervention effects is costly, however, because the number of combinations grows combinatorially with editable concepts (Sussex et al., 2021; Tigas et al., 2022; 2023).

Causal analyses quantify agent influence and propagation (Triantafyllou et al., 2024; Giloni et al., 2025; Shen et al., 2025; Triantafyllou et al., 2025), while failure attribution localizes problematic agents and steps (Zhang et al., 2025; Chen et al., 2026; Zhang et al., 2026; Kong et al., 2026). These summaries do not specify agent responses to new input combinations: if removing either of two evidence sources leaves an answer unchanged, these measurements alone do not determine the answer when both sources are removed (Arif et al., 2026). Causal proxies and simulator surrogates learn reusable intervention responses (Wu et al., 2023; Dyer et al., 2024), while causal abstraction formalizes and learns consistency across models (Zennaro et al., 2023; Massidda et al., 2023; Felekis et al., 2026; Lorenz & Tull, 2026). Generalization to unseen intervention combinations further depends on shared structure and experimental coverage (Bravo-Hermsdorff et al., 2023; Agarwal et al., 2023; Dou et al., 2026). For an LLM-based MAS, upstream changes create new downstream inputs, so local responses must compose to capture concept propagation and outcomes. Acquisition must therefore account for how local responses compose globally. Active design targets graph features, mechanisms, or causal queries (Tigas et al., 2022; Toth et al., 2022; Tigas et al., 2023; Gao & Sejdinovic, 2026). The challenge is to acquire, within a limited agent-call budget, the local response information needed to model this propagation under unseen joint interventions.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We address this challenge with Recursive Local Causal Modeling (*ReLoC*), which, from local to global, progressively builds an approximate global causal model of an LLM-based MAS with limited interventions on each agent. This divide-and-conquer strategy intervenes on each agent locally instead of all agents jointly, and reuses the rest. *ReLoC* therefore builds a global causal model efficiently and predicts final answers for unseen interventions quickly, without further agent executions.

Concretely, *ReLoC* represents agent messages using task-defined concepts and learns local causal models of agent responses to interventions on their input concepts. These interventions are realized through edits to the corresponding text in task inputs or upstream messages. Starting from the outcome agent, it learns a local causal model by executing that agent in isolation and using prediction errors to guide subsequent interventions, including joint concept interventions when individual interventions are insufficient. It then follows the communication graph upstream, learning local causal models of how parent agents produce these concepts, until reaching the task inputs. For each intervention query, *ReLoC* composes these local causal models along the affected graph paths, in execution order, to predict how concept changes propagate across agents and affect the final answer.

We evaluate *ReLoC* on three benchmarks, three topologies, and four LLMs, comparing predicted agent responses, concept propagation, and final answers with actual system executions. *ReLoC* achieves 62.8–86.5% local response fidelity for the inputs received by each agent and 53.0–76.8% propagation fidelity when composing the models without further system executions. We further test whether this modeling extends to unseen interventions, measuring final-answer fidelity as the fraction of predictions matching the executed system's final answer. Across 23,400 held-out queries, *ReLoC* achieves 85.6–93.7% final-answer fidelity under an intervention budget of 32 agent calls per case, exceeding an LLM prompted with the whole system by 8.0–45.7 percentage points. On the two frontier LLMs, where whole-system training is feasible, it also outperforms a model learned from whole-system executions by 5.5–5.7 percentage points.

2 Related Work

Causal analysis and diagnosis of multi-agent behavior. Communication structure alone does not reveal which agents determine an outcome or where a failure originates. Intervention-based analyses address the first question by measuring influence and propagation, including how effects pass through subsequent agent actions and environment states (Giloni et al., 2025; Shen et al., 2025; Triantafyllou et al., 2025). Failure taxonomies and agent-step attribution address the second by relating unsuccessful executions to specific coordination problems and decisions (Cemri et al., 2025; Zhang et al., 2025; Kong et al., 2026; Sun et al., 2026). Complete execution contexts, counterfactual replay, and tool-verified diagnostic memory strengthen this attribution by examining the evidence available to agents and testing suspected errors (Chen et al., 2026; Li et al., 2026; Zhang et al., 2026; Yu et al., 2026; Shah, 2026; Ma et al., 2026). Together, these approaches show why system behavior must be examined through intermediate interactions (Lin et al., 2026; Arif et al., 2026), but do not specify responses to new combinations of inputs. ReLoC models those conditional responses so that local evidence can support predictions of how other interventions propagate through the system.

Causal models for intervention prediction. Reusable intervention models need both behavioral fidelity and a basis for extrapolating beyond measured edits. Causal proxies and simulator surrogates learn to reproduce intervention responses (Wu et al., 2023; Dyer et al., 2024), while causal abstraction formalizes and learns the correspondence that makes cross-model interventions comparable (Beckers et al., 2020; Massidda et al., 2023; Zennaro et al., 2023; Felekis et al., 2026; Lorenz & Tull, 2026). Generalization further requires assumptions connecting experimental conditions: shared intervention factors or low-rank outcomes with sparse interactions provide such structure (Bravo-Hermsdorff et al., 2023; Agarwal et al., 2023; Dou et al., 2026). For a composed model, however, fidelity at individual mechanisms does not establish fidelity along an entire path. The local parent-set evaluation of Counterfactual Chains illustrates this distinction, leaving multi-hop propagation unverified (Nussbaum-Hoffer et al., 2026). ReLoC uses the communication graph to connect agent-local models and tests their composition separately.

Intervention acquisition for causal modeling. Intervention design depends on what the learned model must answer. Resolving an unknown graph or selected graph features differs from identifying effects on a target response, which may not require full graph recovery (Wang et al., 2020; Sussex et al., 2021). Mechanism learning and query-directed design extend acquisition to response uncer-

tainty and quantities computed from the causal model (Tigas et al., 2022; Toth et al., 2022; Tigas et al., 2023; Gao & Sejdinovic, 2026); causal Bayesian optimization instead values experiments for finding interventions with better outcomes (Aglietti et al., 2020). This distinction matters for composition: reducing prediction error at one agent helps only if it improves downstream predictions. Given the communication graph, ReLoC spends its budget on responses along the outcome agent's upstream dependencies. Coverage gaps and observed errors guide local interventions, while held-out propagation tests whether this information suffices across the composed path.

3 PRELIMINARIES

Multi-agent systems. We consider an LLM-based multi-agent system that solves a task by executing a set of agents $\mathcal{A}$ on a set of task inputs $\mathcal{S}$. Each agent $a \in \mathcal{A}$ reads the outputs of its parents $\mathrm{pa}(a) \subseteq \mathcal{S} \cup \mathcal{A}$ and produces a message. The communication recorded in an execution therefore forms a directed acyclic communication graph $G = (V, E)$ with $V = \mathcal{S} \cup \mathcal{A}$ and $E = \{(v, a) : v \in \mathrm{pa}(a)\}$. Agents execute along a topological order of G, and the outcome agent $o \in \mathcal{A}$ produces the final message, whose first concept is the final answer.

Concepts. Each node $v \in V$ takes a value $X_v \in \mathcal{X}_v$. The message of an agent a is described by K task-defined semantic concepts, $X_a = (X_{a,1}, \ldots, X_{a,K})$. Each concept takes one of several discrete values, with each value expressed through a corresponding textual realization in the agent's message. The first concept $X_{a,1}$ is the agent's answer to the task, while the others describe aspects of its assessment, such as whether the available evidence is sufficient. A task input X_s is a text, and we restrict it to a finite set $\mathcal{X}_s$ of versions, such as the original text, a counterfactual edit, or the text with its evidence withheld. Intervening on a task input therefore changes the semantic concepts expressed by that text. Every $\mathcal{X}_v$ is thus finite. We write x^0 for the values in the original execution, and x_W for the restriction of an assignment x to a node set W.

Local mechanisms. For each case, an agent's task, role, and other background information remain fixed. Its changing inputs are the values of its parents, $X_{\mathrm{pa}(a)}$. Agent a responds through a local mechanism $p_a(\cdot \mid \mathrm{do}(X_{\mathrm{pa}(a)} = x))$, which is a distribution because LLM decoding is not always deterministic. Executing a alone with its parent inputs set to x and the same background information samples from this mechanism. We call this a local intervention, or probe, which costs one agent call.

Intervention queries. A query $q = (\mathrm{do}(X_U = u), R)$ sets the nodes $U \subseteq V$ to the values u and restores the edges $R \subseteq E$: for each $(v, a) \in R$, agent a receives the original value x^0_v, while all other edges carry the intervened values. The remaining agents are then executed in topological order, and X^q_o denotes the outcome agent's resulting message. Queries include editing one or several task inputs ($U \subseteq \mathcal{S}$), overriding intermediate messages ($U \subseteq \mathcal{A}$), and editing a task input while restoring one downstream edge to its original message. Because agents interact only through their messages, a query replaces only the mechanisms of U and the values on R, and leaves every other local mechanism p_a unchanged. This modularity is what allows local models to be composed.

Problem. Given the original execution x^0 and a budget of B local interventions, the goal is to learn a global causal model that describes local agent responses and concept propagation. For a held-out query $q \in \mathcal{Q}$, composing the local models predicts the intermediate messages and final answer without executing the system. Responses to held-out queries are used only for evaluation. We test the model's description of intermediate behavior and its final-answer predictions separately.

4 METHOD

4.1 LOCAL CAUSAL MODELS

For each agent a, ***ReLoC*** keeps the records D_a of its local interventions, each a pair (x, y) of an input assignment x to $\mathrm{pa}(a)$ and the observed message y. D_a is initialized with the original execution $(x^0_{\mathrm{pa}(a)}, x^0_a)$. The local causal model $\hat{p}_a(y \mid x)$ estimates $p_a(y \mid \mathrm{do}(X_{\mathrm{pa}(a)} = x))$ in one of three ways, and each prediction records which way was used. If x has been measured, $\hat{p}_a(y \mid x)$ is the empirical frequency of y among the records with input x, and the prediction has *measured support*.

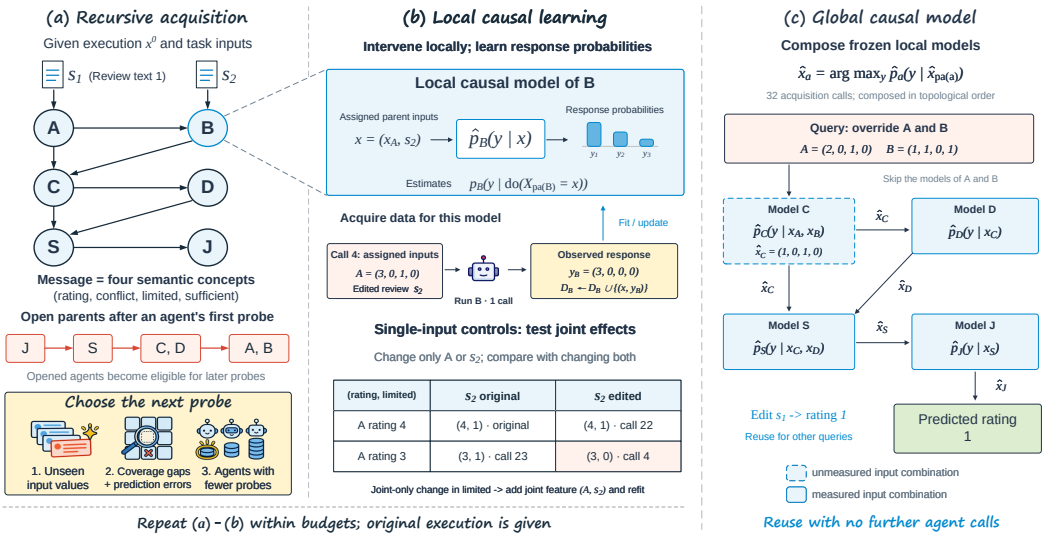


Figure 1: *ReLoC* on an example case. (a) From the outcome agent, recursively select upstream agents for local interventions. (b) Learn local causal models from intervention records. Matched controls change A's message or s_2 alone to test their joint effect. (c) Compose the local causal models along the communication graph, passing the most probable message at each step without agent calls.

Otherwise, if D_a contains at least two distinct inputs, *ReLoC* extrapolates with a factorized count estimator over input features $\phi(x) = (\phi_1(x), \phi_2(x), \dots)$. Each concept of an input message and each input text is one feature. With Laplace smoothing, the estimator is

$$\hat{p}_a(y \mid x) \;\propto\; (N_y + 1) \prod_j \frac{N_{y,j}\big(\phi_j(x)\big) + 1}{N_y + m_j}, \tag{1}$$

where y ranges over the messages observed in D_a, N_y is the number of records with response y, $N_{y,j}(z)$ is the number of these records whose j-th feature equals z, and m_j is the number of distinct values of feature j in D_a and x. Such a prediction is a *statistical extrapolation*; the worked examples in Appendix D show how the counts determine predictions for unmeasured inputs. If D_a contains fewer than two distinct inputs, a single response cannot distinguish a constant mechanism from an unmeasured effect, and the prediction is *unsupported*.

Joint features. Equation (1) treats features independently given the response, so it misses responses that depend jointly several inputs, such as two edits that change the response only together. *ReLoC* therefore adds a joint feature only when matched controls reveal such a dependency, as illustrated by B's limited-applicability concept in Figure 1b. Suppose that a record (x, y) is not predicted correctly by $\hat{p}_a$ and x differs from x^0 on a set S of at least two inputs. For each $T \subseteq S$, let $x^{(T)}$ take the values of x on T and of x^0 elsewhere. *ReLoC* measures these controls, which are drawn from the catalog and charged to the budget. If each control yields a single response $y^{(T)}$ and the contrast $\sum_{T \subseteq S} (-1)^{|S|-|T|}\, \mathbf{1}\big[y_k^{(T)} = z\big]$ is nonzero for some concept k and value z, the joint effect of S is not explained by its individual effects. *ReLoC* then adds x_S to ϕ and refits Equation (1) on D_a.

4.2 Recursive Acquisition

Starting from the outcome agent as shown in Figure 1a, *ReLoC* maintains a set of opened agents $\mathcal{O}$, initialized to $\{o\}$. Candidate local interventions for agent a come from a catalog $\mathcal{C}_a$ of input assignments, which is fixed before learning and constructed without reading any response. $\mathcal{C}_a$ covers every value of each input and bounded joint assignments of several inputs, together with the matched controls of each joint assignment. At each step, *ReLoC* selects an opened agent and one of its candidates, executes the agent alone on this input, and adds the record to D_a. After the first local intervention of a, its agent parents $\mathrm{pa}(a) \cap \mathcal{A}$ join $\mathcal{O}$. The recursion therefore follows G upward from o and ends at the task inputs, which need no model. Acquisition stops when B local interventions

have been spent or the catalogs of opened agents are exhausted; the original execution is given and not charged. Details can be found in Appendix B.

Allocation. Before adding a record, *ReLoC* compares the observed answer with the prediction of $\hat{p}_a$. Let e_a count mispredicted answers among the n_a records of a that received a prediction, and let $\epsilon_a = (e_a + 1)/(n_a + 2)$ be the smoothed local error rate. *ReLoC* ranks opened agents in three tiers. First, agents whose best candidate introduces an input value unseen in D_a, namely a new answer of an input message or a new version of an input text, are preferred, with fewer local interventions first. Second, an agent whose best candidate covers new input conditions, namely pairs of input values and, for agents with over two inputs, the full joint assignment, receives the score $(1 + \epsilon_a)\, c_a^{\text{new}}/c_a^{\text{rem}}$, where c_a^{new} is the number of conditions the candidate newly covers and c_a^{rem} is the number of conditions still uncovered in $\mathcal{C}_a$. Third, the remaining agents are ordered by their number of local interventions. The score directs interventions to agents whose local causal models have coverage gaps and make errors, and the smoothing gives newly opened agents a defined priority. Within-agent candidate ranking and matched-control scheduling are given in Appendix B.

4.3 Composition

Given a query $q = (\mathrm{do}(X_U = u), R)$, *ReLoC* first determines the required agents $\mathcal{A}^q$: an agent $a \notin U$ belongs to $\mathcal{A}^q$ if G contains a directed path from a to o whose nodes avoid U and whose edges avoid R. Overrides and restored edges thus cut the corresponding upstream dependencies, and agents outside $\mathcal{A}^q$ are never evaluated (e.g., A and B in Figure 1c). For an edge (v, a), the input that a receives from v is

$$\hat{x}^q_{v\to a} = \begin{cases} x^0_v & \text{if } (v,a) \in R, \\ u_v & \text{if } v \in U, \\ \hat{x}^q_v & \text{otherwise,} \end{cases} \qquad \hat{x}^q_a = \arg\max_y \hat{p}_a\big(y \mid (\hat{x}^q_{v\to a})_{v\in \mathrm{pa}(a)}\big), \tag{2}$$

where $\hat{x}^q_s = x^0_s$ for task inputs $s \notin U$. *ReLoC* evaluates Equation (2) for the agents in $\mathcal{A}^q$ in the topological order of G and returns their predicted messages, including $\hat{x}^q_o$, without executing the system. The prediction has measured support if every composed step does, is unsupported if some step is unsupported, and is a statistical extrapolation otherwise. Support refers to the inputs encountered during composition; local response fidelity is evaluated separately on realized parent states. The query loop in Algorithm 1 reuses these local models, so one acquisition answers all queries in $\mathcal{Q}$.

5 Experiments

Multi-agent systems. We build systems on three benchmarks. CEBaB (Abraham et al., 2022) asks for a five-level restaurant rating from two review passages, with human-written counterfactual edits. FEVER (Thorne et al., 2018) and SciFact (Wadden et al., 2020) ask whether evidence supports or refutes a claim, or is insufficient; we split up to 16 sentences around the evidence into two passages, and the edit withholds the evidence. Each system has two task inputs and one of three topologies with 6, 8, or 10 agents (sequential, parallel, hierarchical). Every agent reports an answer and three Boolean concepts: conflicting evidence, limited applicability, and sufficient evidence. We run all systems at temperature zero with two frontier LLMs, Hy3 (Tencent Hy Team, 2026) and Doubao-Seed-2.0-pro (Doubao) (ByteDance Seed, 2026), and two small open-weight LLMs, Qwen3-4B (Yang et al., 2025) and Llama-3.2-3B-Instruct (Llama-3.2-3B) (Grattafiori et al., 2024), giving 36 settings. Details can be found in Appendix C.

Queries. Each case has six held-out queries: editing the first, the second, or both task inputs; overriding one or two intermediate messages; and editing the first input while restoring one downstream edge. The ground truth is the answer of the executed system. The 24 CEBaB, 151 FEVER, and 150 SciFact test cases yield 23,400 queries.

Baselines. *Probe-order* baselines share *ReLoC*'s local models and composition and change only acquisition: **Full-local** probes agents along the execution order, **One-shot** along the reverse order with a fixed plan, and **Generic** by response entropy. *Outcome-only* baselines predict the final answer without modeling agents: **Global LLM** prompts the system's own LLM with all roles, wiring, the

Table 1: Local response fidelity, propagation fidelity, trajectory exact match, and final-answer fidelity (%) for *ReLoC*. Mechanism scores are averaged within cases, then across cases; final-answer fidelity uses all 5,850 queries per LLM. The last column gives the share and count of incorrect intermediate trajectories among queries with a correctly predicted final answer.

LLM	Local response	Propagation	Trajectory exact match	Final answer	Correct final answer, wrong trajectory
Hy3	73.8	64.4	36.0	88.4	59.7 (3,086/5,172)
Doubao	62.8	53.0	19.6	85.6	77.5 (3,879/5,006)
Qwen3-4B	84.0	73.5	47.4	93.7	49.9 (2,732/5,480)
Llama-3.2-3B	86.5	76.8	50.8	85.6	41.1 (2,060/5,010)

original transcript, and calibration executions; **Global learned** fits a count estimator to the calibration answers; and **XGBoost** fits gradient-boosted trees to the same labels and features. **Invariant** (*no change*) returns the original answer. Details can be found in Appendix C.1.

Protocol and metrics. Each method receives 32 agent calls per case, with checkpoints at 8 and 16; a local intervention costs one call and a calibration execution one per executed agent. All models are fitted before evaluation and remain fixed; XGBoost hyperparameters are selected on development queries only. **Final-answer fidelity** is the fraction of queries whose predicted answer matches the executed one, also reported for queries with *changed* and *unchanged* final answers. For intermediate messages, **local response fidelity** scores local models on realized parent inputs, **propagation fidelity** scores messages composed from the intervention, and **trajectory exact match** requires every scored message to match; all require agreement on all four fields. Confidence intervals use 2,000 paired bootstrap repetitions over case clusters sharing source families or documents.

5.1 Modeling Local Responses and Propagation

***ReLoC*'s causal model captures how the MAS responds.** We first test whether the learned model reproduces the system itself, with results in Table 1. Local models reproduce realized messages exactly, on all four fields, 62.8–86.5% of the time, and composing them from the intervention alone reproduces 53.0–76.8% of intermediate messages. Trajectory exact match, which requires every scored message to be exact, reaches 19.6–50.8%. Final answers are more robust than full trajectories: 41.1–77.5% of correct answers pass through at least one imperfect message. Full-local and One-shot show the same pattern, so it is shared by local composition rather than specific to recursive acquisition. Outcome-only baselines predict no intermediate messages and cannot be scored here.

Concepts and restored edges. Boolean concept fields help most with small LLMs: removing them and refitting on the same records lowers final-answer fidelity by 3.4 points with Qwen3-4B and 4.5 with Llama-3.2-3B, while with frontier LLMs the answer alone is nearly sufficient (the full model differs by -1.3 and $+0.8$ points). For edge restoration, *ReLoC* answers both the edit and the edit with a restored edge correctly in 81–93% of pairs; since the executed answer differs between the two in only 37–100 pairs per LLM, this mainly tests consistency of the restoration operation.

5.2 Predicting Unseen Interventions

***ReLoC* outperforms outcome-only prediction.** We next test whether the same causal model extends to the 23,400 held-out queries. With the frontier LLMs, *ReLoC* reaches 88.4% (Hy3) and 85.6% (Doubao) fidelity, as shown in Table 2. It exceeds Global LLM by 8.0 and 12.5 points, Global learned by 5.7 and 5.5, XGBoost by 3.3 and 3.2, and Invariant by 5.0 and 4.1, all with 95% intervals above zero, and ranks first or second in 14 of the 18 frontier settings. With the small LLMs, *ReLoC* attains the highest overall fidelity with Qwen3-4B (93.7%) and exceeds Global LLM by 17.2 and 45.7 points; with Llama-3.2-3B it is within 1.6 points of the best baseline.

The advantage grows as the LLM weakens. Global LLM must simulate all agents in one pass. It drops from 86.9% on Seq-6 to 71.3% on Hier-10 with Hy3 and reaches only 39.9% with Llama-3.2-3B. *ReLoC* only asks the LLM to execute one agent on one input, which even the 3B model does

Table 2: Final-answer fidelity (%) on the 23,400 held-out test queries under a budget of 32 agent calls per case. Within each LLM and setting, the best method is in **green bold** and the second best is underlined; *All* is the fidelity over the 5,850 queries of an LLM. Paired intervals are in Tables 6 and 7.

			CEBaB			FEVER			SciFact			
LLM	Type	Method	Seq-6	Par-8	Hier-10	Seq-6	Par-8	Hier-10	Seq-6	Par-8	Hier-10	All
Hy3	*No change*	Invariant	37.5	50.0	52.1	82.9	89.6	79.2	86.9	92.7	86.8	83.4
	Outcome only	Global learned	40.3	54.2	**58.3**	81.7	86.0	81.0	85.6	91.8	85.3	82.7
		XGBoost	37.5	54.2	52.1	88.0	88.4	79.2	92.1	94.1	86.9	85.1
		Global LLM	59.7	43.1	41.7	87.2	83.0	69.9	90.9	89.7	77.4	80.4
	Probe order only	Generic	16.7	15.3	11.1	74.7	65.2	56.5	80.3	67.3	53.8	62.5
		One-shot	37.5	54.2	52.1	92.2	**91.3**	79.8	93.7	**95.3**	88.6	87.0
		Full-local	70.8	57.6	39.6	**92.6**	90.4	**84.4**	**94.2**	95.0	89.9	**88.5**
	Ours	*ReLoC*	**79.9**	**61.1**	47.2	91.5	90.4	82.0	93.8	94.6	**90.6**	88.4
Doubao	*No change*	Invariant	31.2	50.0	47.9	84.4	82.0	79.4	90.2	87.0	84.1	81.5
	Outcome only	Global learned	37.5	52.8	47.2	81.6	77.7	80.0	88.6	83.9	85.0	80.1
		XGBoost	33.3	47.2	45.8	87.5	83.3	79.5	91.3	86.1	85.8	82.4
		Global LLM	47.2	54.2	43.8	80.4	75.7	68.4	81.3	77.9	66.7	73.1
	Probe order only	Generic	11.1	12.5	10.4	64.8	61.1	55.4	73.1	64.2	55.3	58.6
		One-shot	31.9	59.7	**50.0**	83.6	82.5	79.0	91.9	87.7	85.4	82.2
		Full-local	54.2	52.8	33.3	87.7	86.2	**83.7**	92.1	**89.9**	86.6	84.7
	Ours	*ReLoC*	**61.1**	**76.4**	38.2	**88.0**	**86.5**	83.6	**92.8**	88.8	**86.8**	**85.6**
Qwen3-4B	*No change*	Invariant	46.5	57.6	54.9	97.4	**98.3**	98.0	91.8	97.0	96.7	93.3
	Outcome only	Global learned	41.7	53.5	41.7	96.8	98.2	**98.5**	**92.9**	97.3	**96.9**	93.0
		XGBoost	50.0	57.6	50.7	97.7	97.6	98.0	92.3	96.8	96.7	93.3
		Global LLM	51.4	59.7	**56.2**	85.2	79.5	81.6	71.8	73.8	76.9	76.5
	Probe order only	Generic	32.6	18.1	16.0	78.6	93.0	84.7	72.2	87.4	82.6	78.6
		One-shot	**56.9**	39.6	54.9	96.6	97.9	96.9	88.7	**97.4**	96.1	92.3
		Full-local	55.6	60.4	41.0	**98.1**	**98.3**	97.9	92.7	96.8	95.7	93.3
	Ours	*ReLoC*	49.3	**71.5**	47.2	97.5	**98.3**	98.0	92.1	**97.4**	96.7	**93.7**
Llama-3.2-3B	*No change*	Invariant	**77.8**	57.6	59.0	80.8	87.3	**90.7**	86.4	82.6	88.0	84.4
	Outcome only	Global learned	**77.8**	**87.5**	**61.1**	85.3	**90.5**	90.4	89.1	85.4	88.1	**87.2**
		XGBoost	**77.8**	57.6	60.4	80.5	88.4	90.5	86.6	83.1	**88.3**	84.7
		Global LLM	20.8	38.9	34.7	39.3	21.7	46.9	47.3	41.0	47.3	39.9
	Probe order only	Generic	47.9	33.3	22.9	72.3	77.0	75.1	72.3	63.9	61.9	67.8
		One-shot	**77.8**	68.8	60.4	85.2	90.1	89.0	88.7	**86.1**	83.6	85.8
		Full-local	**77.8**	83.3	59.0	87.6	88.7	89.1	90.8	84.2	87.0	86.8
	Ours	*ReLoC*	76.4	61.1	58.3	**88.1**	88.5	89.8	**91.0**	84.1	82.0	85.6

reliably: its measured-support predictions are all correct. *ReLoC*'s margin over Global LLM therefore widens from 8.0 points (Hy3) to 12.5 (Doubao), 17.2 (Qwen3-4B), and 45.7 (Llama-3.2-3B).

***ReLoC* predicts change without losing stability.** Most queries leave the answer unchanged, so the changed queries show whether a method captures propagated effects, as shown in Figure 2. *ReLoC* predicts 45–47% of changed final answers with the frontier LLMs and Llama-3.2-3B, against 29–46% for Global learned, 12–34% for One-shot, and 10–28% for XGBoost in Table 7, while keeping 93–98% fidelity on unchanged queries. Global LLM reaches a similar changed fidelity only by over-predicting change: its fidelity on unchanged queries falls to 41.7–87.0%. Pooled over all LLMs, *ReLoC* has the highest changed fidelity at 44.8%, as shown in Table 3.

Most of the gain comes from avoiding false changes. On Qwen3-4B and Llama-3.2-3B, unchanged queries account for 96.8% and 94.7% of *ReLoC*'s net gain over Global LLM. Thus, the larger margins with small LLMs mainly reflect fewer false changes. For Qwen3-4B, Invariant already

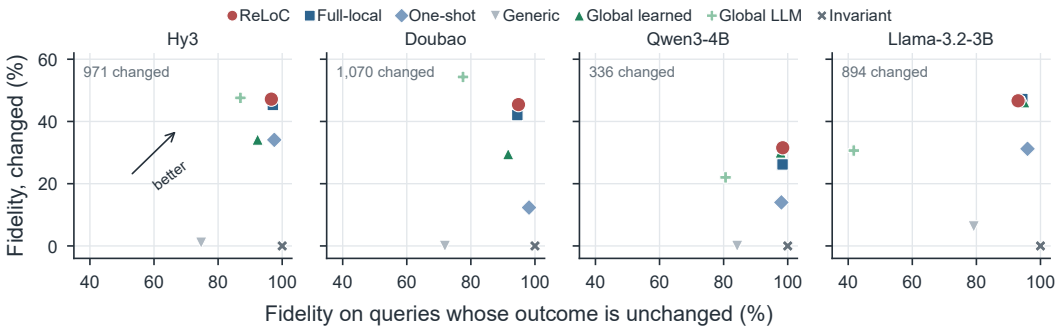


Figure 2: Final-answer fidelity on changed versus unchanged queries under a budget of 32 calls. Invariant sits at $(100, 0)$. *ReLoC* predicts more changed final answers than Global learned and One-shot while keeping 93–98% fidelity on unchanged queries; Global LLM reaches similar changed fidelity only with frontier LLMs, and only by predicting change too often.

reaches 93.3%, close to *ReLoC*'s 93.7%, while *ReLoC* predicts only 31.5% of changed answers correctly. High overall fidelity can therefore coexist with limited fidelity on changed queries.

5.3 Acquisition and Budget

Recursive acquisition pays off where local conditions are scarce. The probe-order baselines isolate acquisition. *ReLoC* exceeds Generic by 15–27 points for every LLM and One-shot by 1.4–3.4 points for the frontier LLMs and Qwen3-4B. On FEVER and SciFact, whose answers take three values, 32 calls cover the relevant local conditions under any order, so *ReLoC* and Full-local reach similar fidelity. On FEVER and SciFact Par-8, the two methods even allocate calls identically, as shown in Table 8. On CEBaB, whose ratings take five values, targeting errors matters: *ReLoC* exceeds Full-local by 6.7 points with Hy3 and 11.8 with Doubao, including 23.6 on Par-8 with Doubao. With Qwen3-4B, it remains 3.7 points higher across the three topologies, though 6.2 lower on Seq-6.

***ReLoC* scales quickly with budget.** From 8 to 16 calls, *ReLoC* improves by 14.5–15.3 points with every LLM. At 32 calls it is the best local method with Doubao and Qwen3-4B, 0.1 points behind Full-local with Hy3, and 1.2 points behind with Llama-3.2-3B. See Appendix E for details.

Early gains come from supporting more queries. Across the four LLMs, 3,292 of the 3,505 net additional correct predictions from 8 to 16 calls (93.9%) come from queries that were unsupported at 8 calls. The early gain therefore mainly comes from acquiring enough local observations to complete composition. Later calls can then refine these predictions.

5.4 Error Analysis

Composition is near-exact where measured. When every composed step has measured support, *ReLoC*'s answer is 98.0% correct, and local models reproduce 84.4–100% of realized messages on measured joint inputs. Extrapolated predictions are 73.4% of queries and 85.3% correct overall (Table 3). On CEBaB their fidelity is 48.2% (Figure 3a). Joint queries cost *ReLoC* only 1.1–4.1 points relative to single ones, against 7.0 points for Global learned on joint edits, because *ReLoC* models how edits combine inside the system.

Errors start at unmeasured inputs. The first agent whose predicted answer is wrong had received a measured input in only 13–34% of *ReLoC*'s errors, against 62–69% of the steps of correct predictions. In 79–89% of missed changes, this agent is predicted to repeat its original answer, suggesting that extrapolation often fails to carry a change through an unmeasured input. Measuring the relevant input conditions is therefore a direct lever for further gains.

Limiting cases. On Hier-10, with ten agents, a four-input integrator, and five-level ratings, only 14–16% of *ReLoC*'s predictions have measured support with the frontier LLMs, and Invariant scores

Table 3: Final-answer fidelity (%) pooled over the four LLMs, by query type and by whether the final answer changes (number of queries in parentheses), and fidelity of *ReLoC* grouped by the support status of its predictions (share of queries in parentheses). Unchanged and changed counts exclude 88 queries with missing ground truth (0, 15, 55, and 18 for the four LLMs); those queries remain in the fidelity denominator as incorrect. Support shares use all 23,400 queries.

	ReLoC	Full-local	Global learned	Global LLM
Edit first input	89.9	**90.0**	89.1	70.7
Edit second input	**90.3**	90.0	88.8	72.3
Edit both inputs	**88.8**	88.3	82.1	72.9
Override one message	86.6	**87.1**	**87.1**	62.4
Override two messages	**82.5**	**82.5**	77.3	58.9
Edit and restore edge	91.8	**92.0**	90.1	67.6
Final answer unchanged (20,041)	95.8	**96.1**	94.3	71.8
Final answer changed (3,271)	**44.8**	42.8	35.5	42.5
Measured support (26.6%)	98.0	–	–	–
Statistical extrapolation (73.4%)	85.3	–	–	–

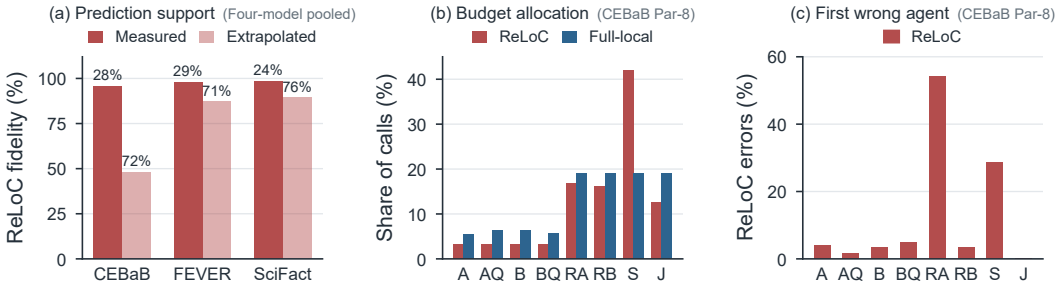


Figure 3: Anatomy of *ReLoC*'s errors, pooled over the four LLMs. *(a)* Final-answer fidelity of predictions whose every step has measured support versus statistical extrapolation; labels give the share of queries. *(b)* Share of agent calls per agent on CEBaB Par-8. *ReLoC* concentrates calls on the integrator S. *(c)* First agent in execution order whose predicted answer is wrong, among *ReLoC*'s errors on CEBaB Par-8: most errors start at the under-probed reconciler RA.

higher (47.9% and 52.1% versus 38.2% and 47.2%). On Par-8 with Llama-3.2-3B, error-guided allocation spends 41.9% of calls on the integrator S (19.0% for Full-local). Pooled over the four LLMs, Figure 3b shows the same concentration, with S at 42%. The reconciler RA is the first wrong agent in 54% of Llama's errors on this setting, and Llama copies an overridden rating down the chain in 20 of 23 cases, which *ReLoC* misses. The same allocation yields a 23.6-point gain over Full-local with Doubao, which suggests securing minimal coverage of every agent before refining by error.

6 Conclusion

We presented ReLoC for building a global causal model of how interventions change agent decisions, how concept changes propagate through a MAS, and how they affect its final outcome. ReLoC learns local causal models from a few interventions, recurses from the outcome agent to the task inputs, and composes these models along the system's communication graph to obtain the global model. Experiments on three benchmarks, three topologies, and four LLMs show that these models capture agent responses and concept propagation, with 62.8–86.5% local response fidelity and 53.0–76.8% propagation fidelity. Across 23,400 unseen intervention queries, the resulting global models achieve 85.6–93.7% final-answer fidelity, exceeding an LLM prompted with the whole system by 8.0–45.7 percentage points under the same intervention budget. These results support learning and composing local causal models as an approach to understanding system behavior from individual agent responses through to the effects of unseen interventions. Future work can strengthen this account by targeting acquisition at changed intermediate states and sharing information across agents, and extend it to free-form messages and communication structures that change during execution.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and readability improvements and to provide limited feedback during experimental planning. All AI-assisted outputs were critically reviewed and validated by the authors. The authors take full responsibility for the methodology, experiments, results, and final content of this work.

REFERENCES

Eldar D Abraham, Karel D'Oosterlinck, Amir Feder, Yair Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. CEBaB: Estimating the causal effects of real-world concepts on NLP model behavior. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1278.

Abhineet Agarwal, Anish Agarwal, and Suhas Vijaykumar. Synthetic combinations: A causal inference framework for combinatorial interventions. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3d17b7f7d52c83ab6e97e2dc0bda2e71-Abstract.html.

Virginia Aglietti, Xiaoyu Lu, Andrei Paleyes, and Javier González. Causal bayesian optimization. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108, pp. 3155–3164. PMLR, 2020. URL https://proceedings.mlr.press/v108/aglietti20a.html.

Nokimul Hasan Arif, Qian Lou, and Mengxin Zheng. Conjunctive prompt attacks in multi-agent LLM systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34175–34191, 2026. doi: 10.18653/v1/2026.acl-long.1577. URL https://aclanthology.org/2026.acl-long.1577/.

Sander Beckers, Frederick Eberhardt, and Joseph Y. Halpern. Approximate causal abstractions. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115, pp. 606–615, 2020. URL https://proceedings.mlr.press/v115/beckers20a.html.

Gecia Bravo-Hermsdorff, David Watson, Jialin Yu, Jakob Zeitler, and Ricardo Silva. Intervention generalization: A view from factor graph models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/88139fdcc82fc597090620d77b023282-Abstract-Conference.html.

ByteDance Seed. Seed2.0 model card: Towards intelligence frontier for real-world complexity. Model card, 2026. URL https://arxiv.org/abs/2607.00248.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei A. Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/b1041e52d3be19f0a9bc491657488e4a-Abstract-Datasets_and_Benchmarks_Track.html.

Mengzhuo Chen, Junjie Wang, Fangwen Mu, Yawen Wang, Zhe Liu, Huanxiang Feng, and Qing Wang. Seeing the whole elephant: A benchmark for failure attribution in LLM-based multi-agent systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19888–19905, 2026. doi: 10.18653/v1/2026.acl-long.912. URL https://aclanthology.org/2026.acl-long.912/.

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. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/578e65cdee35d00c708d4c64bce32971-Abstract-Conference.html.

Jitong Dou, Lingrui Luo, Bing Zhu, Hengliang Luo, Mingjun Zhong, and Yurong Cheng. DSCF: Dual-source counterfactual fusion for high-dimensional combinatorial interventions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20879–20886, 2026. doi: 10.1609/aaai.v40i25.39228. URL https://ojs.aaai.org/index.php/AAAI/article/view/39228.

Joel Dyer, Nicholas Bishop, Yorgos Felekis, Fabio Massimo Zennaro, Anisoara Calinescu, Theodoros Damoulas, and Michael Wooldridge. Interventionally consistent surrogates for complex simulation models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/26b8e3dc3a21fcd660d80c63b767f324-Abstract-Conference.html.

Yorgos Felekis, Theodoros Damoulas, and Paris Giampouras. Distributionally robust causal abstractions. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.04842.

Erdun Gao and Dino Sejdinovic. ActiveCQ: Active estimation of causal quantities. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/85ff89c606e8b79743d3db2222ed7dcb-Abstract-Conference.html.

Amit Giloni, Chiara Picardi, Roy Betser, Shamik Bose, Aishvariya Priya Rathina Sabapathy, and Roman Vainshtein. CAIR: Counterfactual-based agent influence ranker for agentic AI workflows. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 18940–18966, 2025. doi: 10.18653/v1/2025.emnlp-main.958. URL https://aclanthology.org/2025.emnlp-main.958/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6507b115562bb0a305f1958ccc87355a-Abstract-Conference.html.

Fanqi Kong, Ruijie Zhang, Huaxiao Yin, Guibin Zhang, Xiaofei Zhang, Ziang Chen, Zhaowei Zhang, Xiaoyuan Zhang, Song-Chun Zhu, and Xue Feng. Aegis: Automated error generation and attribution for multi-agent systems. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10006419.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html.

Jiazheng Li, Emine Yilmaz, Bei Chen, and Thu Le. Towards self-improving error diagnosis in multi-agent systems. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 2063–2077, 2026. doi: 10.18653/v1/2026.findings-acl.98. URL https://aclanthology.org/2026.findings-acl.98/.

Bohan Lin, Kuo Yang, Zelin Tan, Yingchuan Lai, Chen Zhang, Guibin Zhang, Xinlei Yu, Miao Yu, Xu Wang, Yudong Zhang, and Yang Wang. AgentAsk: Multi-agent systems need to ask. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026. doi: 10.18653/v1/2026.acl-long.1294. URL https://aclanthology.org/2026.acl-long.1294/.

Robin Lorenz and Sean Tull. Causal and compositional abstraction, 2026. URL https://arxiv.org/abs/2602.16612. Preprint.

Ming Ma, Jue Zhang, Fangkai Yang, Yu Kang, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. DoVer: Intervention-driven auto debugging for LLM multi-agent systems. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007537.

Riccardo Massidda, Atticus Geiger, Thomas Icard, and Davide Bacciu. Causal abstraction with soft interventions. In *Proceedings of the Second Conference on Causal Learning and Reasoning*, volume 213, pp. 68–87, 2023. URL https://proceedings.mlr.press/v213/massidda23a.html.

Nirit Nussbaum-Hoffer, Nitay Calderon, Liat Ein-Dor, and Roi Reichart. LLM explainability with counterfactual chains and causal graphs, 2026. URL https://arxiv.org/abs/2606.05972. Preprint.

Jaineet Shah. Causal agent replay: Counterfactual attribution for LLM-agent failures, 2026. URL https://arxiv.org/abs/2606.08275. Preprint.

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*, 2025. doi: 10.18653/v1/2025.emnlp-main.623. URL https://aclanthology.org/2025.emnlp-main.623/.

Kai Sun, Wenqiang Li, Bo Dong, Yuxin Lin, Jingyao Zhang, and Bin Shi. Scope delineation before localization: A two-stage framework for enhancing failure attribution in multi-agent systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33108–33116, 2026. doi: 10.1609/aaai.v40i39.40594. URL https://ojs.aaai.org/index.php/AAAI/article/view/40594.

Scott Sussex, Caroline Uhler, and Andreas Krause. Near-optimal multi-perturbation experimental design for causal structure learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 777–788, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/06d5ae105ea1bea4d800bc96491876e9-Abstract.html.

Tencent Hy Team. Hy3. Official model page, 2026. URL https://hy.tencent.ai/research/hy3. Accessed September 24, 2026.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 809–819, 2018. doi: 10.18653/v1/N18-1074. URL https://aclanthology.org/N18-1074/.

Panagiotis Tigas, Yashas Annadani, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. Interventions, where and how? experimental design for causal models at scale. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1752.

Panagiotis Tigas, Yashas Annadani, Desi R. Ivanova, Andrew Jesson, Yarin Gal, Adam Foster, and Stefan Bauer. Differentiable multi-target causal bayesian experimental design. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 34263–34279, 2023. URL https://proceedings.mlr.press/v202/tigas23a.html.

Christian Toth, Lars Lorch, Christian Knoll, Andreas Krause, Franz Pernkopf, Robert Peharz, and Julius von Kügelgen. Active bayesian causal inference. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16261–16275, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/675e371eeeea99551ce47797ed6ed33e-Abstract-Conference.html.

Stelios Triantafyllou, Aleksa Sukovic, Debmalya Mandal, and Goran Radanovic. Agent-specific effects: A causal effect propagation analysis in multi-agent MDPs. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48578–48607, 2024. URL https://proceedings.mlr.press/v235/triantafyllou24a.html.

Stelios Triantafyllou, Aleksa Sukovic, Yasaman Zolfimoselo, and Goran Radanovic. Counterfactual effect decomposition in multi-agent sequential decision making. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 60072–60098, 2025. URL https://proceedings.mlr.press/v267/triantafyllou25a.html.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7534–7550, 2020. doi: 10.18653/v1/2020.emnlp-main.609. URL https://aclanthology.org/2020.emnlp-main.609/.

Tian-Zuo Wang, Xi-Zhu Wu, Sheng-Jun Huang, and Zhi-Hua Zhou. Cost-effectively identifying causal effects when only response variable is observable. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 10060–10069. PMLR, 2020. URL https://proceedings.mlr.press/v119/wang20w.html.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=BAakY1hNKS.

Zhengxuan Wu, Karel D'Oosterlinck, Atticus Geiger, Amir Zur, and Christopher Potts. Causal proxy models for concept-based model explanations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 37313–37334, 2023. URL https://proceedings.mlr.press/v202/wu23b.html.

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. URL https://arxiv.org/abs/2505.09388.

Yifan Yu, Moyan Li, Shaoyuan Xu, Jinmiao Fu, Xinhai Hou, Fan Lai, and Bryan Wang. CORRECT: COndensed eRror RECognition via knowledge transfer in multi-agent systems. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.24088.

Fabio Massimo Zennaro, Máté Drávucz, Geanina Apachitei, W. Dhammika Widanage, and Theodoros Damoulas. Jointly learning consistent causal abstractions over multiple interventional distributions. In *Proceedings of the Second Conference on Causal Learning and Reasoning*, volume 213, pp. 88–121, 2023. URL https://proceedings.mlr.press/v213/zennaro23a.html.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who is inducing failure in the LLM agentic systems? In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10007726.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? On automated failure attribution of LLM multi-agent systems. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 76583–76599, 2025. URL https://proceedings.mlr.press/v267/zhang25cq.html.

A ALGORITHM

Algorithm 1 summarizes *ReLoC*. Lines 2–11 perform the recursive acquisition of Section 4.2, and lines 12–14 answer each query by the composition of Section 4.3.

Algorithm 1: *ReLoC*

Input: graph G, original execution x^0, catalogs $\{\mathcal{C}_a\}_{a\in\mathcal{A}}$, budget B, queries $\mathcal{Q}$
Output: predictions $\{\hat{x}_o^q\}_{q\in\mathcal{Q}}$ with support status

1 $D_a \leftarrow \{(x^0_{\mathrm{pa}(a)}, x^0_a)\}$ for all $a \in \mathcal{A}$; $\mathcal{O} \leftarrow \{o\}$
2 **while** *fewer than B local interventions are spent and some $a \in \mathcal{O}$ has candidates in $\mathcal{C}_a$* **do**
3 select an agent $a \in \mathcal{O}$ and a candidate $x \in \mathcal{C}_a$ by the allocation tiers
4 $\hat{y} \leftarrow \arg\max_y \hat{p}_a(y \mid x)$ unless unsupported; execute a alone on x and observe y
5 **if** *$\hat{y}$ exists* update e_a, n_a with $\hat{y}_1 \neq y_1$
6 $D_a \leftarrow D_a \cup \{(x, y)\}$
7 **if** *$\hat{y} \neq y$ and x differs from x^0 on at least two inputs*
8 schedule the matched controls of x; add the joint feature once they show a nonzero contrast
9 **end**
10 $\mathcal{O} \leftarrow \mathcal{O} \cup (\mathrm{pa}(a) \cap \mathcal{A})$
11 **end**
12 **foreach** $q \in \mathcal{Q}$ **do**
13 determine $\mathcal{A}^q$ and evaluate Equation (2) along the topological order of G
14 **end**

B CATALOG CONSTRUCTION AND CANDIDATE RANKING

Value domains. For a task input, the catalog uses its versions in $\mathcal{X}_s$: the edited text, the text with its evidence withheld, and the original text. When the edit itself withholds the evidence, the first two versions coincide. For an input message, the catalog uses every combination of an answer with the three Boolean concepts, ordered so that consecutive values of the same answer vary the Boolean concepts.

Catalog. For each agent a, $\mathcal{C}_a$ contains three kinds of input assignments. Single-input assignments set one input to each value of its domain. Joint assignments set a pair of inputs, and for agents with more than two inputs also the full set of inputs, to aligned or shifted positions of their domains. For a pair that contains a task input, the catalog instead enumerates all versions of the task input against one value of the other input. Every joint assignment is preceded by all of its proper subsets, which serve as its matched controls. Duplicate assignments and assignments equal to $x^0_{\mathrm{pa}(a)}$ are removed. The construction uses only the domains and never reads a response.

Candidate ranking. Within an agent, *ReLoC* ranks candidates lexicographically by the following keys, and ties keep the catalog order.

1. Whether the candidate is primary, that is, every task input takes its original or edited version.
2. The number of input values not yet seen in D_a, where a message is represented by its answer and a text by its version.
3. The number of new input conditions, defined as in Section 4.2.
4. When the second or third key is positive, fewer changes of Boolean concepts relative to x^0.
5. The number of Boolean values not yet seen in D_a for the corresponding input.
6. The Hamming distance to the nearest measured input.

The counts c_a^{new} and c_a^{rem} in the allocation score are computed over primary candidates.

Matched controls. After a mispredicted record whose input differs from x^0 on at least two inputs, the record is kept pending. When the agent is selected again and its best candidate introduces no new input answer, *ReLoC* replaces this candidate with a missing matched control of the pending record. The joint feature is tested once all controls are measured. If some control yields more than one response, the record stays pending, because a variable control cannot separate a joint effect from decoding noise.

C Experimental Details

Topologies. Table 4 lists the agents of the three topologies. s_1 and s_2 denote the two task inputs, and J is the outcome agent. The parallel and hierarchical systems assess each task input by a factual agent and a qualification agent and reconcile the two assessments before integration; the hierarchical system adds a conflict agent and a dependence audit.

Table 4: Agents, inputs, and responsibilities of the three topologies.

Topology	Agent	Inputs	Responsibility
Seq-6	A	s_1	assess the factual bearing of the first input
	B	A, s_2	assess the second input and revise A when warranted
	C	A, B	check consistency between the two assessments
	D	C	check whether qualifications limit the conclusion
	S	C, D	integrate consistency and applicability
	J	S	decide the task
Par-8	A, B	s_1; s_2	assess factual evidence in each input
	AQ, BQ	s_1; s_2	assess scope and qualifications in each input
	RA	A, AQ	reconcile the assessments of the first input
	RB	B, BQ	reconcile the assessments of the second input
	S	RA, RB	integrate both branches
	J	S	decide the task
Hier-10	all agents of Par-8, where S reads RA, RB, C, and D		
	C	RA, RB	assess cross-input conflict
	D	RA, RB, AQ, BQ	audit shared evidence and dependence

Queries. The first three queries replace one or both task inputs with their edited versions. The fourth overrides the message of A with the second label in the allowed label set, marked as limited and not sufficient, and the fifth additionally overrides the message of B with the first label, marked as conflicting. The sixth edits s_1 and restores the edge from A to its first receiver, which is B in Seq-6 and RA otherwise.

Data. Table 5 gives the partition sizes. Partitions follow the original source families, and derived edits stay with their family. In the test partition, the FEVER cases come from 102 families and the SciFact cases from 86 families. Thirteen FEVER source documents and one SciFact source document appear in more than one partition, so the confidence intervals resample connected clusters of families and source documents. Reference labels and annotations are removed before acquisition.

Table 5: Number of cases per partition.

Benchmark	Discovery	Development	Test
CEBaB (Abraham et al., 2022)	42	24	24
FEVER (Thorne et al., 2018)	100	30	151
SciFact (Wadden et al., 2020)	100	30	150

Models and prompts. Hy3 (Tencent Hy Team, 2026), Doubao-Seed-2.0-pro (ByteDance Seed, 2026) (version `doubao-seed-2-0-pro-260215`), and the open-weight checkpoints `Qwen/Qwen3-4B` (Yang et al., 2025) and `LLM-Research/Llama-3.2-3B-Instruct` (Grattafiori et al., 2024) run at temperature zero with thinking disabled and at most 512 output

tokens. Every agent receives the task, the allowed labels, its responsibility, its inputs, identifiers of the sources from which those inputs derive, and the following instruction. Source text is included only for direct source parents; downstream agents receive parent messages and source identifiers. The task, allowed labels, role, and source identifiers remain fixed for each case.

```
Perform your stated role on the real evidence provided. The task is a
    restaurant rating or evidence-conditioned claim verification. Do not
    infer a label from the fact that material was edited or withheld.
    Missing evidence is not evidence of falsity. Account for
    contradictory evidence and conditions that limit applicability.
    Intermediate reports are fallible assessments, not authoritative
    answers. Use your role to assess and synthesize them. Return only one
     JSON object with exactly these keys: {"label": "one allowed label",
    "conflict": false, "limited": false, "sufficient": true}. The
    booleans must reflect evidence conflict, applicability limitations,
    and evidence sufficiency respectively. Choose the label by reasoning
    about the supplied task and evidence; never copy a previous label
    automatically. The label must be a JSON string exactly equal to one
    item in allowed_labels. Even when evidence is insufficient, return
    your best estimate among those labels and set sufficient to false. N/
    A, UNKNOWN, null and extra keys are not valid responses. Do not
    substitute a label based on formatting alone.
```

Global LLM receives the following instruction together with the task, the responsibilities and wiring of all agents, the original evidence and transcript, the query, and its calibration examples.

```
Predict the final response of the specified multi-agent system after the
    proposed intervention. You have the task, all role responsibilities
    and wiring, the original evidence and complete original transcript,
    and any independent calibration examples. Predict the system's
    behavior, which may differ from the correct answer to the task. An
    override replaces that output; other roles then execute in dependency
     order. A restored edge sends its original message to the named
    receiver while other edges retain the intervened value. Use all
    available information and reason through the consequences. Return one
     JSON object with exactly these keys: {"label": "one allowed label",
    "conflict": false, "limited": false, "sufficient": true}. The label
    must be a JSON string exactly equal to one item in allowed_labels.
    Even when uncertain, predict an allowed system label and set
    sufficient to false. N/A, UNKNOWN, null and extra keys are not valid
    responses.
```

C.1 BASELINES

Probe-order baselines. All three use the same local response models and composition as *ReLoC*. **Full-local** probes agents in turn along the execution order and selects local inputs by coverage. **One-shot** follows the reverse order with a fixed probe plan, without adapting it to observed responses. **Generic** selects agents by the entropy of their recorded responses, with a bonus for agents probed less often, then selects local inputs by coverage. These comparisons isolate the effect of recursive, error-guided acquisition.

Outcome-only baselines. **Global LLM** uses the same LLM as the system to predict its final answer in one pass. It receives the task, all agent roles and wiring, the original evidence and transcript, the query, and calibration executions under interventions distinct from the held-out queries. Its prompt is given above. **Global learned** fits a count estimator to the original final answer and system-level calibration answers, using the intervened state and intervention indicators as features. **XGBoost** uses the same calibration labels and features with a separate tree-based classifier for each case, as detailed below. This comparison tests whether a more flexible predictor improves on the count estimator under sparse calibration.

Invariant. This baseline always returns the original final answer. Its fidelity is the fraction of queries that leave the answer unchanged, providing a reference for the outcome-only predictors.

XGBoost final-answer predictor. For each case, we fit a separate classifier to the original final answer and the nonempty final answers from the same calibration executions used by Global learned. No new system executions are acquired. The features match Global learned: the intervention is applied to the original state, and the resulting fields are encoded. Indicators of which fields were intervened on, and of whether an edge was restored, enter as separate features. Each text feature is converted to an integer using only the values present in that case's training rows; a value unseen in training receives -1. When the training rows contain only one answer, that answer is predicted directly and no trees are fit. If the calibration records are missing or no training row is usable, the prediction is unsupported and counts as incorrect. The executed responses on the test queries are attached only after the prediction has been made.

We fix the learning rate at 0.3 and select the tree depth from $\{1, 2\}$ and the number of trees from $\{10, 20\}$ on the 576 available queries from the development run stopped early: CEBaB Par-8 and Seq-6 with Hy3 and Doubao. Both depths score 247/576 with 10 trees and 246/576 with 20 trees. The tie is resolved in favor of the smaller model, depth 1 with 10 trees, and this configuration is applied unchanged to all 23,400 test queries. The classifier uses a multiclass probability objective and random seed 20260922. Paired 95% intervals use 2,000 bootstrap repetitions with the same seed, resampling the 186 source-connected family clusters jointly across topologies for each LLM. All 5,850 queries per LLM remain in the overall denominator, including missing ground truth and unsupported predictions. The four-LLM comparison is descriptive.

D WORKED EXAMPLES

We trace three cases from the sealed test runs, selected after evaluation to illustrate acquisition, local modeling, composition, edge restoration, and failure. All use a budget of 32 local agent calls. The examples describe the systems' observed responses, which need not be the correct task answers.

D.1 CASE 1: FROM A RESTAURANT REVIEW TO LOCAL MEASUREMENTS

CEBaB 000891 | Hy3 | Seq-6

Task. Predict the overall restaurant rating from 1 to 5.
Original s_1. They could forget about the corn, but as usual the rest of the food was **excellent**.
Edited s_1'. They could forget about the corn, but as usual the rest of the food was **inedible**.
Original s_2. The Sunday special of lobster, potatoes and corn was a good deal.

What the six agents are. Each agent is a separate role-conditioned call to the same Hy3 backend, with temperature 0, thinking disabled, and a 512-token output limit. A, B, C, D, S, and J are the identifiers in the experiment. The names below describe their functions; the last column gives their exact role instructions. The task and allowed labels $\{1, 2, 3, 4, 5\}$ accompany every call.

ID	Role name	Receives	Role instruction
A	First-evidence assessor	s_1	Assess the factual bearing of the first evidence on the task.
B	Second-evidence assessor	A, s_2	Independently assess the second evidence and revise the first assessment when warranted.
C	Consistency checker	A, B	Check consistency between the two assessments; preserve disagreement.
D	Applicability checker	C	Check whether qualifications and uncertainty limit the conclusion.
S	Integrator	C, D	Integrate the consistency and applicability assessments without assuming agreement.
J	Final judge	S	Decide the original task using the integrated evidence and its limitations.

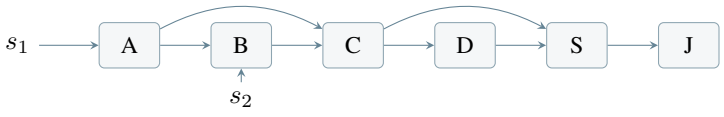
A reads the food review. B reads the other review and A's report. C, D, S, and J see only their listed parent reports and source identifiers, not the review text or the complete conversation. Each call receives a fresh request with the fixed task, role, allowed labels, parent inputs, and source lineage. No conversational history is carried between local probes.

What an agent returns. Every response is a JSON object with four fields. `label` is the proposed task answer; `conflict` indicates contradictory evidence; `limited` indicates a qualification limiting applicability; and `sufficient` indicates sufficient evidence. We write these fields in that order as a tuple, with $0/1$ for false/true. For example,

```
{"label":"4", "conflict":false, "limited":true, "sufficient":false}
```

means rating 4, no reported conflict, limited applicability, and insufficient evidence. The protocol still requires a rating when evidence is insufficient.

The original execution. This is the case underlying Figure 1. All six agents return $(4, 0, 1, 0)$; the final rating is 4. Every dependency is shown below.



D.2 Case 1 Continued: What Is Executed During Acquisition

The agent and its local causal model. The agent is the Hy3 call specified above. Its local causal model is a separate response estimator maintained by *ReLoC*: it maps that agent's parent inputs to a distribution over its observed four-field responses. For example, $\hat{p}_B(y \mid A, s_2)$ predicts the second-evidence assessor's response. Learning updates a table of interventions and fitted counts. A different table and estimator are maintained for each agent and each case.

Opening the graph from the outcome. Call 1 probes J and opens S. Call 2 probes S and opens C and D. Call 3 probes C and opens A and B. Opening an agent makes it eligible for a later probe; it does not execute that agent. The first six calls are J, S, C, B, D, A. Each call executes only the selected agent on a catalog input, and its response becomes one local training record.

Call	1	2	3	4	5	6	7	8
Agent	J	S	C	B	D	A	S	C
Call	9	10	11	12	13	14	15	16
Agent	J	B	D	S	C	J	B	D
Call	17	18	19	20	21	22	23	24
Agent	S	C	J	B	D	B	B	B
Call	25	26	27	28	29	30	31	32
Agent	B	B	S	S	S	S	S	S

The final allocation is A: 1, B: 9, C: 4, D: 4, S: 10, J: 4. The original execution initializes all six models and is not charged to these 32 calls.

A complete local request. At call 4, B receives the following user payload, paired with the common system instruction in Appendix C. That instruction requires independent assessment of the supplied evidence, forbids inferring a label merely because evidence was edited or withheld, and requires the four-field JSON response. The payload is shown with whitespace added for readability.

```
{
  "task": "Predict the overall restaurant rating from 1 to 5.",
  "allowed_labels": ["1", "2", "3", "4", "5"],
  "role": "Independently assess the second evidence and revise the first assessment when
      warranted.",
  "inputs": {
    "A": {"label": "3", "conflict": false,
         "limited": true, "sufficient": false},
    "source_b": "The Sunday special of lobster, potatoes and corn was a good deal, even with
        all the noise."
  },
  "source_lineage": {"A": ["food"],
                "source_b": ["service_or_ambiance"]}
}
```

Observed response and stored record

B returns `{"label":"3", "conflict":false, "limited":false, "sufficient":false}`. The stored pair is

$$\big(\{A = (3, 0, 1, 0),\ s_2 = s_2'\},\quad y_B = (3, 0, 0, 0)\big).$$

Here s_2' is the full review text in the payload. Only B executes; its supplied A message is assigned by the intervention. No downstream agent runs, and the local cost is one call.

D.3 Case 1 Continued: Learning a Joint Response at B

The complete learned record table. B, the second-evidence assessor, has two parents: A's four-field message and the literal review text s_2. Let s_2' be the version with "even with all the noise" added. Its final local model stores exactly the following ten observations, including the initial execution. The highlighted rows and the original observation form the matched controls.

Call	A's message	Review	B's response	Assignment
Original	$(4, 0, 1, 0)$	s_2	$(4, 0, 1, 0)$	original inputs
4	$(3, 0, 1, 0)$	s_2'	$(3, 0, 0, 0)$	change both
10	$(2, 0, 1, 0)$	s_2	$(2, 0, 1, 0)$	change A
15	$(1, 0, 1, 0)$	s_2	$(1, 0, 0, 0)$	change A
20	$(5, 0, 1, 0)$	s_2	$(5, 0, 1, 0)$	change A
22	$(4, 0, 1, 0)$	s_2'	$(4, 0, 1, 0)$	change review
23	$(3, 0, 1, 0)$	s_2	$(3, 0, 1, 0)$	change A
24	$(2, 0, 1, 0)$	s_2'	$(2, 0, 1, 0)$	change both
25	$(1, 0, 1, 0)$	s_2'	$(3, 0, 1, 0)$	change both
26	$(5, 0, 1, 0)$	s_2'	$(5, 0, 1, 0)$	change both

How the model gains a joint feature. At call 4, B has only its original record and the new input is unsupported. Calls 22 and 23 complete the missing controls. Across the original, call-23, call-22, and call-4 records, `limited` is respectively true, true, true, and false. Thus, for $h(y) = \mathbf{1}[y_{\text{limited}} = \text{false}]$,

$$\underbrace{h(y^{A,s_2})}_{1} - \underbrace{h(y^{A})}_{0} - \underbrace{h(y^{s_2})}_{0} + \underbrace{h(y^{\varnothing})}_{0} = 1.$$

Each assignment has one response. At call 23, the log records the nonzero contrast in `limited` and adds (A, s_2) as a joint feature. This describes B's measured response to the supplied inputs.

What the fitted model contains. Its six categorical features are

$$\phi_B(x) = \big(A.\text{label}, A.\text{conflict}, A.\text{limited}, A.\text{sufficient}, s_2, (A, s_2)\big).$$

The text and the joint assignment are treated as categorical values. The output classes are the six distinct *complete messages* in the response column, not just five ratings. For instance, $(3, 0, 0, 0)$ occurs once and $(3, 0, 1, 0)$ occurs twice. For the former class, the counts for $A.\text{label} = 3$, $s_2 = s_2'$, and the call-4 joint input are each one. These response and feature counts, with Laplace smoothing $\alpha = 1$, specify the estimator in Equation (1).

Two ways to use the same local model

Measured input. For $(A = (3, 0, 1, 0), s_2')$, the recorded response is $(3, 0, 0, 0)$. Its empirical probability is $1/1$, so this prediction has measured support.

Unmeasured input. An assignment absent from this table is scored against all six observed response classes using the counts above. The largest normalized score supplies the full predicted message; its status is statistical extrapolation.

All ten records contribute to the final model. The contrast establishes a feature to include; this case alone does not measure its contribution to held-out prediction fidelity.

D.4 Case 1 Continued: The Six Fitted Models and a Numerical Prediction

One model per agent. After acquisition, the six models have the following sizes. A "response" is a distinct four-field JSON message. The record count includes the original observation; the feature count includes any validated joint feature.

Model	Parent inputs	Records	Features	Responses	Joint feature
$\hat{p}_A$	review s_1	2	1	2	–
$\hat{p}_B$	A, review s_2	10	6	6	(A, s_2)
$\hat{p}_C$	A, B	5	8	5	–
$\hat{p}_D$	C	5	4	5	–
$\hat{p}_S$	C, D	11	8	5	–
$\hat{p}_J$	S	5	4	5	–

A: a two-record model of the food assessor. The complete table for A is $s_1 \mapsto (4, 0, 1, 0)$ from the original execution and $s_1' \mapsto (1, 0, 1, 0)$ from call 6. Thus the food-edit query retrieves $(1, 0, 1, 0)$ with empirical probability 1. The text is matched as a complete categorical value; this lookup makes no new Hy3 call.

C: a model of the consistency checker. C has eight input features: the four fields of A and the four fields of B. Its entire training table is below. Write $r_k = (k, 0, 1, 0)$. Every row appears once, so each output class has count $N_{r_k} = 1$. C has no validated joint feature.

Acquired at	A's message	B's message	C's response
Original	r_4	r_4	r_4
Call 3	r_3	r_3	r_3
Call 8	r_2	r_2	r_2
Call 13	r_1	r_1	r_1
Call 18	r_5	r_5	r_5

Computing C's prediction for the food edit. A and B supply $(1, 0, 1, 0)$ and $(1, 0, 0, 0)$. This pair is unmeasured because B's `limited` field differs from the call-13 input. In Equation (1), each rating feature has $m_j = 5$. For output r_1, its two matching rating counts give factors $2/6$ and $2/6$; every other output gets $1/6$ and $1/6$. The Boolean-feature factors and class priors are identical across the five outputs, including the smoothed factor for the unseen `limited=false`. Hence

$$w(r_1) : w(r_2) : w(r_3) : w(r_4) : w(r_5) = 4 : 1 : 1 : 1 : 1.$$

Normalization gives the following model probabilities, reconstructed from the sealed records without agent calls.

Candidate response	r_1	r_2	r_3	r_4	r_5
Food edit	0.500	0.125	0.125	0.125	0.125
Joint message override	$2/7$	$2/7$	$1/7$	$1/7$	$1/7$

The food edit selects r_1. For the joint override $A = (2, 0, 1, 0)$, $B = (1, 1, 0, 1)$, ratings 1 and 2 tie; the implementation's deterministic ordering of serialized responses selects r_1. All five stored response classes have Boolean fields $(0, 1, 0)$, so C's estimator cannot output the actual $(1, 1, 0, 1)$ response in that query. These are estimator probabilities, not an empirical distribution from repeated Hy3 executions.

J: the final judge's measured mapping. J stores $S = r_k \mapsto J = r_k$ for each $k \in \{1, 2, 3, 4, 5\}$, measured in the original execution and calls 1, 9, 14, and 19. When composition supplies $S = r_1$, J's model returns r_1 by exact lookup. This observed mapping applies to those five complete inputs; it does not establish that the judge always copies S.

D.5 Case 1 Continued: Reusing the Models Across Queries

After 32 calls, all local models are frozen. For an input edit, *ReLoC* propagates predicted messages through the graph. For an override, it inserts the supplied message and skips that agent's model. Below, M means that the composed parent assignment was measured; E denotes extrapolation. "Actual" is the separately executed system response, used for evaluation.

Query 1: replace "excellent" with "inedible" in s_1

Only s_1 changes. A predicts the new rating, B combines A's message with the unchanged s_2, and C, D, S, and J propagate the result. No agent is executed while making these predictions.

Agent	Inputs	Predicted message	Actual message	Support
A	s_1'	$(1,0,1,0)$	$(1,0,1,0)$	M
B	$\hat{x}_A, s_2$	$(1,0,0,0)$	$(1,0,0,0)$	M
C	$\hat{x}_A, \hat{x}_B$	$(1,0,1,0)$	$(1,0,1,0)$	E
D	$\hat{x}_C$	$(1,0,1,0)$	$(1,0,1,0)$	M
S	$\hat{x}_C, \hat{x}_D$	$(1,0,1,0)$	$(1,0,1,0)$	M
J	$\hat{x}_S$	$(1,0,1,0)$	$(1,0,1,0)$	M

All six predicted messages match this execution. C nevertheless receives an unmeasured pair, so the query as a whole is an extrapolation.

Query 2: override A and B with two supplied messages

Set $A = (2,0,1,0)$ and $B = (1,1,0,1)$. Neither A nor B needs a model evaluation; composition starts at C and proceeds through D, S, and J.

Agent	Inputs	Predicted message	Actual message	Support
C	overridden A, B	$(1,0,1,0)$	$(1,1,0,1)$	E
D	$\hat{x}_C$	$(1,0,1,0)$	$(1,1,0,1)$	M
S	$\hat{x}_C, \hat{x}_D$	$(1,0,1,0)$	$(1,1,0,1)$	M
J	$\hat{x}_S$	$(1,0,1,0)$	$(1,1,0,1)$	M

The final rating is correct, but all three Boolean concepts are wrong from C onward. Downstream M marks support for the *predicted* inputs, which already differ from the actual inputs. It does not certify the correctness of the propagated state.

All six queries from the same acquisition. The table retains both failures. Querying the frozen models costs zero additional agent calls; the reference executions used to score these predictions have a separate evaluation cost.

Intervention	Predicted rating	Actual rating	Support
Edit s_1	1	1	E
Edit s_2	4	4	M
Edit both inputs	1	3	E
Override A	2	2	M
Override A and B	1	1	E
Edit s_1; restore A→B	1	4	E

D.6 Case 2: The Claim, Evidence, and Eight-Agent System

FEVER 68733 | Hy3 | Par-8

Claim. Steve Buscemi spent his entire career in the pharmaceutical industry.

Task labels. SUPPORT, REFUTE, NOT_ENOUGH.

Interventions. Replace s_1 with `[Evidence withheld]`; then compare this edit with the same edit plus restoration of A→RA.

The evidence supplied to the system. The two inputs are successive parts of the supplied Steve Buscemi source context. The following are literal excerpts; ellipses omit other film and television credits, not edits made for the intervention. Both full strings are preserved with the case records.

Original source text (excerpts)

s_1: "Buscemi has starred and supported in successful Hollywood and indie films , including Parting Glances , New York Stories , Mystery Train , Reservoir Dogs , Desperado , Con Air , Armageddon , The Grey Zone , Ghost World , Big Fish , and The Sopranos ." ...

s_2: "From 2010 to 2014 , he portrayed Enoch " Nucky " Thompson in the critically acclaimed series Boardwalk Empire , which earned him two Screen Actors Guild Awards , a Golden Globe , and two nominations for an Emmy Award . He made his directorial debut in 1996 , with Trees Lounge , in which he also starred ." ...

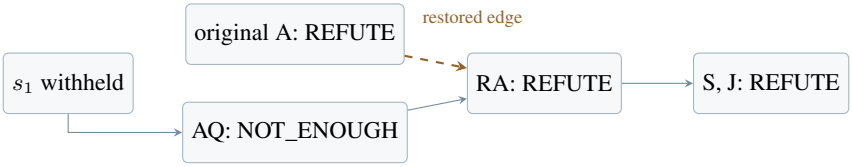
The agents and their information. All eight use Hy3 with the same decoding settings and four-field JSON protocol as Case 1. Every call receives the claim and allowed labels. Only A, AQ, B, and BQ see source text; the remaining agents see parent reports and source identifiers. The names describe the roles; the instruction column reproduces the actual role text.

ID	Role name	Receives	Role instruction
A	First-source assessor	s_1	Assess factual evidence in the first source.
AQ	First-source qualifier	s_1	Assess scope, qualifications and evidential limitations in the first source.
B	Second-source assessor	s_2	Assess factual evidence in the second source.
BQ	Second-source qualifier	s_2	Assess scope, qualifications and evidential limitations in the second source.
RA	First-branch reconciler	A, AQ	Reconcile the factual and scope assessments of the first source.
RB	Second-branch reconciler	B, BQ	Reconcile the factual and scope assessments of the second source.
S	Branch integrator	RA, RB	Integrate both branches, preserving unresolved conflicts and limitations.
J	Final judge	S	Decide the original task from the integrated assessment; exercise independent judgment.

Original state and fitted models. All eight agents originally return $(\mathrm{REFUTE}, 0, 0, 1)$. Acquisition spends one call each on A, AQ, B, and BQ and seven each on RA, RB, S, and J: 32 calls in total. The four source-agent models each store two records and one text feature. RA and RB each store eight records and eight report-field features; S stores eight records and nine features, including the validated joint feature $(\mathrm{RA}, \mathrm{RB})$; J stores eight records and four features. The next page expands RA's fitted model, which determines how the restored edge affects the first branch.

D.7 Case 2 Continued: The Reconciler Model and Edge Restoration

Two receivers of the edited evidence. In Par-8, A and AQ both read s_1, and RA reconciles their reports. Withholding s_1 makes both agents output NOT_ENOUGH. Restoring A→RA supplies A's *original* REFUTE message to RA, while AQ still supplies its changed NOT_ENOUGH message. It does not restore s_1 or the AQ branch.



The diagram shows the affected branch and the observed labels under restoration. The unchanged B/BQ→RB branch also feeds S.

Observed and predicted propagation. Write R for REFUTE and N for NOT_ENOUGH. Each entry below is *predicted / actual*; the first two rows show the labels delivered to RA, including the edge-specific replacement.

Message or agent	Withhold s_1	Withhold s_1 + restore A→RA
A→RA	N / N	R / R
AQ→RA	N / N	N / N
RA	N / N	R / R
RB (unchanged branch)	R / R	R / R
S	N / N	R / R
J	N / N	R / R

Which models are evaluated? Without restoration, composition uses all eight agents. With restoration, A has no remaining path to J and is omitted; its original message is already available. AQ is still required because its changed message reaches RA. *ReLoC* therefore evaluates AQ, B, BQ, RA, RB, S, and J, using the same frozen models for both queries.

Inside $\hat{p}_{\mathrm{RA}}$. RA's model has eight measured inputs and two observed response classes, $r = (\mathrm{R}, 0, 0, 1)$ with count 5 and $n = (\mathrm{N}, 0, 0, 1)$ with count 3. The grid gives all eight records, one per populated cell. Here P denotes SUPPORT; input Boolean fields are $(0, 0, 1)$ except in the starred cell, where AQ has $(0, 0, 0)$. Outputs r, n denote full messages. No joint feature is present.

A's label \ AQ's label	R	P	N
R	r	r	r
P	r	r	unmeasured
N	n	n^*	n

Withholding s_1 supplies $(\mathrm{N}, 0, 0, 0)$ from both parents. The count estimator gives $\hat{p}(n) = 0.9489$ and $\hat{p}(r) = 0.0511$. With restoration it receives A's original r and AQ's $(\mathrm{N}, 0, 0, 0)$, giving $\hat{p}(r) = 0.5466$ and $\hat{p}(n) = 0.4534$. Both inputs require extrapolation because the complete messages differ from the grid's measured inputs. These calculated distributions explain why the frozen RA model switches its predicted label.

Under restoration, all seven evaluated agents' full messages match the execution. Without restoration, RA already mispredicts `sufficient` as true; S and J also mispredict Boolean concepts, despite correct final labels. The contrast concerns the system's behavior under these two interventions.

D.8 Case 3: An Unmeasured Reconciler Input Blocks a Change

CEBaB 000850 | Llama-3.2-3B | Par-8

Original s_1. Overly robust sauces kill the taste of what was ordered.

Original s_2. Very noisy, difficult to converse.

Task. Predict the overall restaurant rating from 1 to 5.

Original execution. Every agent returns $(1, 0, 0, 1)$.

Query. Override A with $(2, 0, 1, 0)$; leave both texts unchanged.

This case uses the eight roles, instructions, and wiring listed in Case 2, with the restaurant-rating task and labels $\{1, 2, 3, 4, 5\}$. All roles use `Llama-3.2-3B-Instruct`, temperature 0, and a 512-token output limit. A/AQ assess the sauces review, B/BQ assess the noise review, RA/RB reconcile each pair, S integrates the branches, and J returns the rating. The models are fitted afresh for this case and backend.

The budget and the missing input. *ReLoC* spends half of its 32 calls on S. RA receives four probes, all with matching ratings from A and AQ. Together with the original observation, they cover the five diagonal rating pairs below, always with Boolean fields $(0, 0, 1)$. They never measure the query's pair: a limited, insufficient rating-2 message from A and the original rating-1 message from AQ.

Calls	A	AQ	B	BQ	RA	RB	S	J
ReLoC	1	1	1	1	4	4	16	4
Full-local	2	2	2	1	6	6	6	6

RA record	A's message	AQ's message	RA's response	Source
Original	$(1, 0, 0, 1)$	$(1, 0, 0, 1)$	$(1, 0, 0, 1)$	observation
Probe	$(2, 0, 0, 1)$	$(2, 0, 0, 1)$	$(2, 0, 0, 1)$	call 3
Probe	$(5, 0, 0, 1)$	$(5, 0, 0, 1)$	$(5, 0, 0, 1)$	call 10
Probe	$(4, 0, 0, 1)$	$(4, 0, 0, 1)$	$(4, 0, 0, 1)$	call 14
Probe	$(3, 0, 0, 1)$	$(3, 0, 0, 1)$	$(3, 0, 0, 1)$	call 18
Query	$(2, 0, 1, 0)$	$(1, 0, 0, 1)$	$(2, 0, 0, 1)$	evaluation only

The fitted distribution and first error. RA has eight report-field features, five output classes with one record each, and no joint feature. For the query, its normalized weights for ratings $1, 2, 3, 4, 5$ are $(2/7, 2/7, 1/7, 1/7, 1/7)$, all with Boolean fields $(0, 0, 1)$. Matching A's rating favors class 2; matching AQ's rating favors class 1. The unseen Boolean values contribute equal factors to all classes. The first two classes tie, and the fixed ordering selects $(1, 0, 0, 1)$; actual RA returns $(2, 0, 0, 1)$. S and J then reproduce the predicted rating 1 on measured composed inputs.

Agent	Role in the query	Predicted message	Actual message	Support
A	supplied override	$(2, 0, 1, 0)$	$(2, 0, 1, 0)$	–
AQ	unchanged evidence	$(1, 0, 0, 1)$	$(1, 0, 0, 1)$	M
RA	first wrong agent	$(1, 0, 0, 1)$	$(2, 0, 0, 1)$	E
RB	unchanged branch	$(1, 0, 0, 1)$	$(1, 0, 0, 1)$	M
S	integrates RA and RB	$(1, 0, 0, 1)$	$(2, 0, 0, 1)$	M
J	final rating	$(1, 0, 0, 1)$	$(2, 0, 0, 1)$	M

A concrete comparison. Full-local and Global learned predict the actual rating 2. Full-local's two additional RA records measure rating pairs $(1, 2)$ and $(2, 1)$ with Boolean fields $(0, 0, 1)$; RA follows A in both. These observations distinguish the parents in a disagreement, although the query's Boolean values remain unmeasured. *ReLoC*'s five diagonal records leave the conflicting parent ratings tied.

E Additional Results

Per-setting paired intervals. Table 6 gives the difference in fidelity between *ReLoC* and Invariant, Global learned, Global LLM, and the probe-order baselines in all 36 settings, with paired 95% bootstrap intervals over clusters of cases that share source families or documents. Differences whose interval lies above zero are in bold. The largest gains of *ReLoC* over the local baselines occur on CEBaB. Losses with intervals below zero larger than 3 points occur on the hierarchical CEBaB systems of both frontier LLMs and on CEBaB Par-8 with Llama-3.2-3B, as discussed in Section 5.4, and also on CEBaB Seq-6 with Qwen3-4B and on SciFact Hier-10 with Llama-3.2-3B. Three further losses against Full-local lie between 1.1 and 2.4 points.

XGBoost comparison. Table 7 reports XGBoost fidelity separately on queries with changed and unchanged final answers, together with paired differences from Global learned and *ReLoC*; overall and per-setting fidelity are in Table 2. Missing ground truth affects 0, 15, 55, and 18 queries for Hy3, Doubao, Qwen3-4B, and Llama-3.2-3B; unsupported XGBoost predictions affect 0, 12, 36, and 18 queries, respectively. These categories can overlap and are retained as incorrect in overall fidelity.

Few observed final answers. XGBoost improves over Global learned by 2.5 points with Hy3 and 2.3 with Doubao, but remains 3.3 and 3.2 points below *ReLoC*, respectively, as shown in Table 2, with paired intervals in Table 7. With Qwen3-4B its difference from Global learned is small and its interval includes zero; with Llama-3.2-3B it loses 2.5 points. Its final-answer fidelity on changed queries is only 10.4–28.3%. Thus, replacing the count estimator does not close the frontier-model gap. With typically only 4–7 observed final answers per case, including the original answer, these results support sparse system-level supervision as a limitation beyond estimator simplicity; they do not isolate the number of observations as the sole cause.

Allocation of the budget. Table 8 lists the share of agent calls that *ReLoC* and Full-local spend on each agent, pooled over the four LLMs. On Par-8 and Hier-10, the task-input agents A, AQ, B, and BQ receive one call each from *ReLoC*, which is enough because their only inputs are the finite versions of a task input. On Seq-6 this holds for A; B also reads A's message. *ReLoC* shifts calls toward multi-input agents where it observes errors: the integrator S on CEBaB Par-8, the second-input agent B on CEBaB Seq-6, and the reconcilers and conflict agent on FEVER and SciFact Hier-10. Full-local does not shift calls toward the agents where *ReLoC* observes errors. On FEVER and SciFact Par-8, the two methods divide calls among agents in the same proportions, consistent with their nearly equal fidelity reported in Section 5.3.

Fidelity across budgets. Figure 4 reports fidelity at the checkpoints of 8, 16, and 32 agent calls per case. *ReLoC* gains 14.5–15.3 points from 8 to 16 calls with every LLM. At 32 calls it is the best local method with Doubao and Qwen3-4B, 0.1 points behind Full-local with Hy3, and 1.2 points behind with Llama-3.2-3B. At 8 calls, *ReLoC* reaches 68.5–77.1% against 77.5–88.6% for Full-local: the outcome-first recursion spends its first calls on the outcome agent and the integrator, so upstream agents still hold only their original record, and 17% of *ReLoC*'s predictions are unsupported, against 6–10% for Full-local. Once every agent has been opened, at roughly two calls per agent, error-guided refinement takes over. Among the outcome-only baselines, Global learned stays near Invariant at every budget, because a few whole-system executions per case cannot cover six distinct queries. Global LLM improves with more calibration executions for the frontier LLMs and Qwen3-4B (from 60.8% to 80.4% with Hy3) but stays below *ReLoC* at every budget and with every LLM.

Table 6: Final-answer fidelity of *ReLoC* minus Invariant, Global learned, Global LLM, and the probe-order baselines (percentage points) under a budget of 32 calls, with paired 95% intervals in brackets. Bold differences have intervals above zero. *All* pools the nine settings of an LLM.

Benchmark	Topology	Invariant	Global learned	Global LLM	Generic	One-shot	Full-local
Hy3							
CEBaB	Seq-6	**+42.4** [+34.7, +49.3]	**+39.6** [+30.6, +48.6]	**+20.1** [+10.4, +28.5]	**+63.2** [+55.6, +70.8]	**+42.4** [+34.0, +50.0]	**+9.0** [+2.8, +16.0]
CEBaB	Par-8	+11.1 [-1.4, +23.6]	+6.9 [-5.6, +19.4]	**+18.1** [+8.3, +27.8]	**+45.8** [+37.5, +54.2]	+6.9 [-5.6, +19.4]	+3.5 [-5.6, +12.5]
CEBaB	Hier-10	-4.9 [-15.3, +4.9]	-11.1 [-20.1, -2.8]	+5.6 [-2.1, +13.9]	**+36.1** [+27.8, +43.8]	-4.9 [-15.3, +4.9]	**+7.6** [+2.8, +13.2]
FEVER	Seq-6	**+8.6** [+4.7, +13.2]	**+9.8** [+6.5, +13.4]	**+4.3** [+1.5, +7.5]	**+16.8** [+11.2, +23.2]	-0.7 [-3.0, +1.4]	-1.1 [-2.1, -0.1]
FEVER	Par-8	+0.8 [-1.8, +3.4]	**+4.4** [+1.6, +7.7]	**+7.4** [+3.6, +11.3]	**+25.2** [+19.3, +31.4]	-0.9 [-2.9, +1.2]	+0.0 [+0.0, +0.0]
FEVER	Hier-10	**+2.8** [+0.4, +5.4]	+1.0 [-3.3, +5.3]	**+12.1** [+7.4, +17.1]	**+25.5** [+18.4, +32.9]	+2.2 [-1.5, +5.8]	-2.4 [-4.4, -0.5]
SciFact	Seq-6	**+6.9** [+3.2, +11.0]	**+8.2** [+5.1, +11.9]	**+2.9** [+0.3, +5.4]	**+13.4** [+8.6, +18.3]	+0.1 [-2.6, +2.7]	-0.4 [-1.5, +0.7]
SciFact	Par-8	**+1.9** [+0.2, +4.0]	**+2.8** [+0.5, +5.1]	**+4.9** [+2.4, +7.5]	**+27.2** [+21.8, +32.6]	-0.8 [-2.3, +0.7]	-0.4 [-1.5, +0.5]
SciFact	Hier-10	**+3.8** [+1.8, +6.1]	**+5.2** [+2.5, +8.0]	**+13.1** [+9.7, +16.7]	**+36.8** [+29.6, +43.5]	**+2.0** [+0.1, +4.2]	+0.7 [-0.6, +2.1]
All		**+5.0** [+3.6, +6.5]	**+5.7** [+4.5, +7.1]	**+8.0** [+6.5, +9.6]	**+25.9** [+22.5, +29.5]	**+1.4** [+0.3, +2.6]	-0.1 [-0.7, +0.5]
Doubao							
CEBaB	Seq-6	**+29.9** [+20.8, +38.2]	**+23.6** [+13.2, +34.0]	**+13.9** [+2.1, +24.3]	**+50.0** [+41.0, +59.0]	**+29.2** [+19.4, +38.9]	+6.9 [+0.0, +13.9]
CEBaB	Par-8	**+26.4** [+13.2, +38.9]	**+23.6** [+10.4, +36.1]	**+22.2** [+11.1, +32.6]	**+63.9** [+56.2, +70.8]	**+16.7** [+5.6, +26.4]	**+23.6** [+14.6, +32.6]
CEBaB	Hier-10	-9.7 [-18.8, -0.7]	-9.0 [-18.8, -0.7]	-5.6 [-13.9, +1.4]	**+27.8** [+18.1, +38.2]	-11.8 [-22.2, -2.1]	+4.9 [+0.0, +10.4]
FEVER	Seq-6	+3.5 [-0.2, +7.8]	**+6.4** [+3.5, +9.7]	**+7.6** [+4.1, +11.3]	**+23.2** [+17.7, +28.9]	**+4.4** [+1.1, +8.3]	+0.2 [-1.1, +1.6]
FEVER	Par-8	**+4.5** [+0.2, +8.6]	**+8.8** [+4.9, +13.2]	**+10.8** [+7.3, +14.3]	**+25.4** [+19.5, +32.4]	+4.1 [-0.1, +7.9]	+0.3 [-0.5, +1.4]
FEVER	Hier-10	**+4.2** [+0.7, +8.0]	**+3.5** [+0.1, +7.1]	**+15.1** [+10.8, +19.4]	**+28.1** [+22.1, +34.7]	**+4.5** [+1.1, +8.2]	-0.1 [-1.7, +1.5]
SciFact	Seq-6	+2.6 [-0.2, +5.9]	**+4.2** [+1.8, +6.7]	**+11.4** [+7.9, +15.0]	**+19.7** [+14.2, +25.6]	+0.9 [-1.4, +3.3]	+0.7 [-0.3, +1.7]
SciFact	Par-8	+1.8 [-0.4, +4.1]	**+4.9** [+2.1, +8.2]	**+10.9** [+7.0, +14.9]	**+24.6** [+18.8, +30.5]	+1.1 [-1.1, +3.5]	-1.1 [-2.3, -0.1]
SciFact	Hier-10	**+2.7** [+0.6, +5.1]	+1.8 [-1.0, +4.7]	**+20.1** [+15.2, +25.1]	**+31.4** [+25.5, +37.8]	+1.3 [-0.9, +3.8]	+0.2 [-1.3, +1.8]
All		**+4.1** [+2.7, +5.8]	**+5.5** [+4.1, +7.1]	**+12.5** [+10.5, +14.7]	**+27.0** [+23.9, +30.4]	**+3.4** [+2.0, +4.9]	**+0.9** [+0.3, +1.6]

Table 6: Fidelity differences and paired 95% intervals, continued for Qwen3-4B and Llama-3.2-3B. The budget and notation are the same as above.

Benchmark	Topology	Invariant	Global learned	Global LLM	Generic	One-shot	Full-local
Qwen3-4B							
CEBaB	Seq-6	+2.8 [-6.9, +13.2]	+7.6 [-5.6, +20.1]	-2.1 [-13.2, +8.3]	**+16.7** [+6.2, +27.1]	-7.6 [-16.7, +1.4]	-6.2 [-10.4, -2.8]
CEBaB	Par-8	**+13.9** [+4.2, +24.3]	**+18.1** [+7.6, +29.2]	**+11.8** [+1.4, +23.6]	**+53.5** [+45.1, +61.1]	**+31.9** [+19.4, +43.8]	**+11.1** [+3.5, +19.4]
CEBaB	Hier-10	-7.6 [-16.7, +0.0]	+5.6 [-6.9, +17.4]	-9.0 [-20.1, +2.1]	**+31.2** [+22.9, +39.6]	-7.6 [-16.7, +0.0]	**+6.2** [+0.7, +12.5]
FEVER	Seq-6	+0.1 [-1.4, +2.0]	+0.7 [-0.6, +2.4]	**+12.3** [+7.3, +18.6]	**+18.9** [+13.2, +25.3]	+0.9 [-0.8, +2.8]	-0.7 [-1.6, +0.1]
FEVER	Par-8	+0.0 [-0.6, +0.6]	+0.1 [-1.2, +1.5]	**+18.9** [+13.9, +24.5]	**+5.3** [+2.7, +8.6]	+0.4 [-0.8, +1.9]	+0.0 [+0.0, +0.0]
FEVER	Hier-10	+0.0 [+0.0, +0.0]	-0.4 [-1.4, +0.2]	**+16.4** [+11.5, +22.0]	**+13.4** [+8.2, +19.0]	+1.1 [+0.0, +2.9]	+0.1 [+0.0, +0.4]
SciFact	Seq-6	+0.3 [-1.3, +2.3]	-0.8 [-2.6, +1.0]	**+20.3** [+13.7, +27.8]	**+19.9** [+14.2, +25.9]	**+3.4** [+0.1, +7.4]	-0.6 [-1.6, +0.1]
SciFact	Par-8	+0.4 [-0.2, +1.3]	+0.1 [-0.3, +0.7]	**+23.7** [+19.5, +28.0]	**+10.0** [+5.9, +14.6]	+0.0 [-0.5, +0.6]	+0.7 [+0.0, +1.9]
SciFact	Hier-10	+0.0 [+0.0, +0.0]	-0.2 [-1.1, +0.5]	**+19.8** [+14.8, +24.8]	**+14.1** [+9.6, +18.9]	+0.6 [+0.0, +1.6]	+1.0 [+0.0, +2.4]
All		+0.4 [-0.2, +0.9]	+0.7 [+0.0, +1.5]	**+17.2** [+14.2, +20.5]	**+15.1** [+12.8, +17.5]	**+1.4** [+0.6, +2.3]	+0.4 [-0.1, +0.8]
Llama-3.2-3B							
CEBaB	Seq-6	-1.4 [-4.2, +1.4]	-1.4 [-4.2, +1.4]	**+55.6** [+46.5, +63.2]	**+28.5** [+20.1, +37.5]	-1.4 [-4.9, +2.1]	-1.4 [-4.2, +1.4]
CEBaB	Par-8	+3.5 [+0.0, +7.6]	-26.4 [-31.2, -19.4]	**+22.2** [+13.9, +29.9]	**+27.8** [+20.1, +35.4]	-7.6 [-18.1, +3.5]	-22.2 [-29.2, -14.6]
CEBaB	Hier-10	-0.7 [-2.1, +0.0]	-2.8 [-6.9, +0.7]	**+23.6** [+11.8, +34.7]	**+35.4** [+25.0, +45.8]	-2.1 [-6.9, +1.4]	-0.7 [-2.1, +0.0]
FEVER	Seq-6	**+7.3** [+4.1, +10.9]	+2.8 [-0.1, +6.1]	**+48.8** [+40.3, +57.3]	**+15.8** [+12.0, +19.5]	**+2.9** [+0.3, +5.5]	+0.4 [-1.2, +2.2]
FEVER	Par-8	+1.2 [-2.1, +4.4]	-2.0 [-5.4, +1.2]	**+66.8** [+58.6, +74.3]	**+11.5** [+7.3, +16.1]	-1.5 [-5.0, +1.6]	-0.2 [-0.7, +0.0]
FEVER	Hier-10	-0.9 [-4.1, +2.2]	-0.6 [-4.7, +3.4]	**+42.9** [+33.4, +52.2]	**+14.8** [+10.3, +19.5]	+0.9 [-2.1, +3.9]	+0.8 [-1.6, +3.3]
SciFact	Seq-6	**+4.6** [+1.4, +7.8]	+1.9 [-1.0, +4.7]	**+43.7** [+35.8, +51.5]	**+18.7** [+13.3, +24.2]	+2.3 [-0.7, +5.4]	+0.2 [-0.9, +1.4]
SciFact	Par-8	+1.6 [-2.8, +6.1]	-1.3 [-5.7, +2.5]	**+43.1** [+34.4, +51.4]	**+20.2** [+14.4, +26.0]	-2.0 [-6.2, +2.5]	-0.1 [-0.8, +0.7]
SciFact	Hier-10	-6.0 [-9.2, -3.0]	-6.1 [-10.1, -2.1]	**+34.7** [+26.9, +42.6]	**+20.1** [+14.4, +25.9]	-1.6 [-4.5, +1.6]	-5.0 [-8.0, -2.3]
All		+1.2 [-0.1, +2.5]	-1.6 [-3.2, +0.1]	**+45.7** [+41.1, +50.5]	**+17.8** [+15.6, +20.4]	-0.1 [-1.5, +1.2]	-1.2 [-2.0, -0.4]

Table 7: XGBoost final-answer fidelity on changed and unchanged queries (%) and overall paired differences (percentage points, 95% family-cluster intervals) at the 32-call budget. Changed and unchanged scores exclude missing ground truth; overall differences retain all 5,850 queries per LLM.

LLM	Changed	Unchanged	XGBoost − Global learned	*ReLoC* − XGBoost
Hy3	28.3	96.5	+2.5 [+1.4, +3.6]	+3.3 [+2.1, +4.5]
Doubao	25.6	95.4	+2.3 [+1.1, +3.5]	+3.2 [+1.7, +4.9]
Qwen3-4B	10.4	99.3	+0.3 [-0.4, +1.0]	+0.4 [-0.1, +0.9]
Llama-3.2-3B	10.9	98.4	−2.5 [-3.6, -1.5]	+1.0 [-0.4, +2.3]

Table 8: Share (%) of the 32 agent calls per case spent on each agent, pooled over the four LLMs. Agents are listed in execution order. Entries are rounded and need not sum to 100.

Seq-6	*Method*	*A*	*B*	*C*	*D*	*S*	*J*
CEBaB	*ReLoC*	3	28	20	13	24	13
	Full-local	6	19	19	19	19	19
FEVER	*ReLoC*	3	18	21	17	22	18
	Full-local	3	22	19	19	19	19
SciFact	*ReLoC*	3	18	21	17	22	18
	Full-local	3	22	19	19	19	19

Par-8	*Method*	*A*	*AQ*	*B*	*BQ*	*RA*	*RB*	*S*	*J*
CEBaB	*ReLoC*	3	3	3	3	17	16	42	13
	Full-local	6	6	6	6	19	19	19	19
FEVER	*ReLoC*	3	3	3	3	22	22	22	22
	Full-local	3	3	3	3	22	22	22	22
SciFact	*ReLoC*	3	3	3	3	22	22	22	22
	Full-local	3	3	3	3	22	22	22	22

Hier-10	*Method*	*A*	*AQ*	*B*	*BQ*	*RA*	*RB*	*C*	*D*	*S*	*J*
CEBaB	*ReLoC*	3	3	3	3	18	14	15	14	14	13
	Full-local	6	6	6	6	13	13	13	13	13	13
FEVER	*ReLoC*	3	3	3	3	22	22	21	8	9	6
	Full-local	3	3	3	3	16	16	16	15	13	13
SciFact	*ReLoC*	3	3	3	3	22	22	21	8	9	6
	Full-local	3	3	3	3	16	16	16	16	13	13

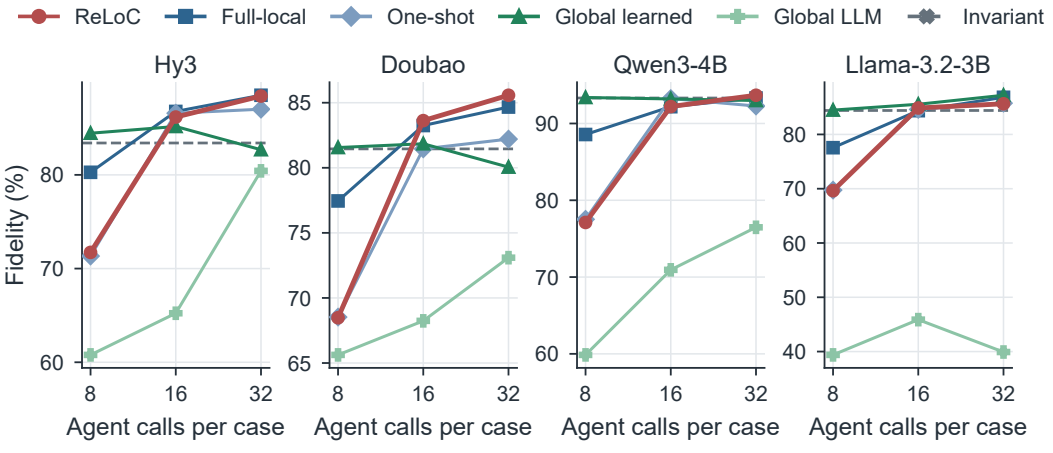


Figure 4: Final-answer fidelity at budgets of 8, 16, and 32 agent calls per case; the dashed line is Invariant. *ReLoC* gains 14.5–15.3 points from 8 to 16 calls, once the recursion reaches the task inputs, and then stays within 1.2 points of Full-local or above it.