When Attribution Is Not Repair: Responsibility and Intervention in LLM Multi-Agent Systems

Anonymous authors
Paper under double-blind review

Abstract

Debugging tools for LLM multi-agent systems often assume that finding where an error entered a trace tells us where to fix it. We test this directly. A failed trace exhibits *source–repair mismatch* (SRM) when a logged action would repair the final answer but no such action lies at an annotated error source. On 116 repairable failures, two independent blinded Source annotations give SRM rates of 0.543 and 0.733, and 0.388 of traces stay mismatched even when every event marked by either annotator counts as a source. As a consequence, a policy restricted to source events can recover at most 61.2% of candidate-oracle logged repairs (71/116 repairable traces), corresponding to 31.7% absolute Repair@1 on the 224-failure cohort, however accurate its localization. We release a benchmark with six-role event responsibility labels, a 1,600-item human-adjudicated test set, and checker-scored single-intervention repair. We also introduce RAISER, which ranks logged answers with responsibility evidence and passes that evidence to a frozen verifier over a three-item shortlist. On 196 failures held out from development, aligned graph and role channels improve ranking NDCG over a matched local ranker (Holm p=0.007), permuting them within each trace erases the gain, and the full system raises Repair@1 from 65.3% to 70.9% of failures (p=0.027). On development and validation, the same system recovers 86.8% and 85.6% of candidate-oracle repairs, +1.5 and +4.4 oracle points over matched verifier controls. A preregistered decomposition on 1,200 further failures locates the remaining headroom in recognizing an already-shortlisted correct answer. Code: https://anonymous.4open.science/r/Trace-R. Data: https://anonymous-hf.com/a/psa97qn014rw/.

1 Introduction

Multi-agent LLM systems now split tasks across planners, specialists, tool users, critics, and verifiers (Chen et al., 2024; Wu et al., 2023; Huang et al., 2025). When such a system returns a wrong answer, a developer wants to know two things: which agent introduced the error, and what change would have produced the right answer. Most of the recent debugging literature treats these as one question. Attribution benchmarks score a method on whether it recovers the faulty agent, decisive step, or issue span (Zhang et al., 2025c; Deshpande et al., 2025; Chen et al., 2026; In et al., 2026), and intervention-based debuggers typically act at the location an attribution step proposes (Ma et al., 2025; Wang et al., 2026; Bonagiri et al., 2026).

The two questions have different answers more often than one might expect. Figure 1 shows the typical pattern. An early agent proposes a wrong hypothesis, a second agent builds on it, a reviewer fails to perform an assigned check, and the final decision is wrong. Yet a downstream agent has already recomputed the correct answer, and the final decision simply ignored it. Replacing the final decision with that downstream answer fixes the task; editing the source event, the natural target of attribution, may not. An attribution method can be perfectly accurate and still point to the wrong place to intervene.

This paper makes the gap measurable and studies how to close it. Given a failed trace, a policy sees the event log, the agents' contracts, and every answer an agent explicitly concluded. It must

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.



Figure 1: Source–repair mismatch. The answers 0.5, 3, and 85 are taken from an AgentVerse/GSM8K failure in our cohort (Table 12); the intermediate roles are drawn schematically. The event that introduced the error ($\mathcal{S}_\tau$) and the event whose logged answer repairs the task ($\mathcal{R}_\tau$) are disjoint: a source-first repair returns 0.5, majority voting returns 3, and only the downstream critic's 85 passes the checker.

choose *one* logged answer to replace the failed final decision. A deterministic task checker, hidden from the policy, scores the result. Under this protocol, source localization and repair are separate, well-defined targets, and we can ask how often they coincide, what it costs to conflate them, and what a repair-oriented method should use instead.

In summary, our contributions are three-fold.

- Firstly, we formalize the attribution–repair gap as *source–repair mismatch* and show that it persists across two independent blinded Source annotations and even under their union (Section 5.2). A coverage bound (Proposition 1) turns this into a concrete cost: any policy that repairs only at annotated sources recovers at most 61.2% of candidate-oracle logged repairs (31.7% absolute Repair@1 on the 224-failure cohort).
- Secondly, we build a joint responsibility and repair benchmark that labels events with six operational roles, includes a human-adjudicated test set for the duty-based roles, and scores repair under a fixed single-intervention budget (Section 5.1).
- Thirdly, we propose RAISER, a repair selector that ranks logged answers with responsibility evidence and passes the same evidence to a frozen verifier. Held-out tests show that candidate-aligned graph and role evidence improves ranking and, when the same evidence is exposed to the frozen verifier, raises final Repair@1 from 65.3% to 70.9% (p=0.027); development and validation results measure the same final system against broader controls (Sections 5.3 and 5.4).

2 RELATED WORK

Our work sits between two lines of research that have so far been evaluated against different targets. We first review failure attribution, which asks where an error originated, and then work that moves from diagnosis to intervention, which asks how a failed execution can be recovered. We close with responsibility and verification, the two ingredients RAISER uses to connect them.

Failure attribution. Who&When and TRAIL localize faulty agents, decisive steps, and issue spans (Zhang et al., 2025c; Deshpande et al., 2025); TraceElephant and MP-Bench broaden trace coverage and admit multiple valid attributions (Chen et al., 2026; In et al., 2026). Aegis contributes injected-fault trajectories, CORRECT transfers diagnostic knowledge across systems, and MAST organizes failure modes (Kong et al., 2025; Yu et al., 2025; Cemri et al., 2025). Localization methods range from specialist reasoning and temporal modeling to spectrum and graph analysis (Zhu et al., 2026a; Zhang et al., 2025a; Banerjee et al., 2025; Zhu et al., 2026c; Liu et al., 2026; Li et al., 2026; Ge et al., 2025; Zhang et al., 2025b). All of these are evaluated against an origin label. Our Source role and the SRM analysis (Section 5.2) adopt the same target, but we pair it with a second, executable one, the logged answer that repairs the task, and show that the two diverge on a large share of failures; attribution methods also reappear as front ends of our repair ranker (Table 9).

From diagnosis to intervention. DoVer, CHIEF, AgentTrace, CausalFlow, and zero-replay debugging link localization to interventions or recovery estimates (Ma et al., 2025; Wang et al., 2026;

Wang, 2026; Bonagiri et al., 2026; Kang et al., 2026). VerifyMAS verifies failure hypotheses, AgentLocate aggregates independent verification views, AgentDebugX conducts structured investigation and recovery, and AgentRx audits trajectory constraints (Qiao et al., 2026; Xia et al., 2026; Zhu et al., 2026b; Barke et al., 2026). Closest to our setting, Luan et al. (2026) ask whether rerun-based repair beats plain resampling, and AgentForesight audits trajectory prefixes for early intervention (Zhang et al., 2026); Adaptive Influence Graphs and AgentScope sharpen localization itself (Bakish et al., 2026; Bi et al., 2026). These systems share our goal of turning diagnosis into recovery but act in different spaces, from reruns to trajectory regeneration. We isolate the smallest executable intervention, replacing the final decision with one logged answer, which makes selection errors directly measurable; VerifyMAS and DeepDebug are adapted to this budget as baselines (Section 5.3), and native regeneration is reported separately (Section E).

Responsibility and verification. Structural-model accounts separate causal responsibility from blame (Chockler and Halpern, 2004); we separate an operational duty breach from intervention success. Process supervision and self-correction assess or revise intermediate reasoning (Lightman et al., 2024; Gou et al., 2024; Shinn et al., 2023; Madaan et al., 2023), and self-verification, Chain-of-Verification, universal self-consistency, and prover–verifier procedures supply candidate-checking primitives (Weng et al., 2022; Dhuliawala et al., 2023; Chen et al., 2023; Jiang et al., 2025; Sedoc et al., 2026). We borrow the distinction between responsibility and blame to define duty-based roles such as Missed Verifier, and we use an off-the-shelf verifier as the last stage of RAISER. What is new is how the two meet: responsibility evidence is computed per candidate and passed to the verifier, and Section 5.4 measures when this evidence changes the repair decision.

3 PROBLEM FORMULATION

We model a failed multi-agent execution as a trace $\tau = (\mathcal{A}, \mathcal{E}, \mathcal{C}, \mathcal{O}, y)$, which consists of agents $\mathcal{A}$, a chronologically ordered sequence of events $\mathcal{E} = \{e_1, \ldots, e_T\}$, role contracts $\mathcal{C}$ that specify what each agent may and must do, cross-agent obligations $\mathcal{O}$, and a task reference y that no policy observes. Each event e_t records its agent, content, stage, tool state, and explicit references to earlier events. Obligations make checking duties explicit: an obligation $o = (e_s, a_v, \phi, t_a, t_d, A) \in \mathcal{O}$ requires agent a_v to check condition ϕ on upstream event e_s between activation time t_a and deadline t_d, provided access condition A holds.

Given this structure, we describe how each event relates to the failure through six operational responsibility roles $\mathcal{K} = \{\text{SRC}, \text{PROP}, \text{MV}, \text{CV}, \text{SYM}, \text{RL}\}$ (Table 1), and attach to every event e_t a multi-label vector whose entry $\ell_{t,k} \in \{0, 1\}$ indicates role $k \in \mathcal{K}$. The MV role is defined normatively through obligations: for an obligation o, let D_o, A_o, and B_o indicate that the duty was assigned, that the agent had access or opportunity, and that the duty was breached; MV holds when $D_o A_o B_o = 1$. None of the roles refers to whether an intervention succeeds, which lets us test the relationship between responsibility and repair instead of assuming it.

Table 1: The six operational responsibility roles in $\mathcal{K}$ and the symbols used in the paper.

Symbol	Role	An event with this role ...
SRC	Source	introduces the harmful state.
PROP	Propagator	inherits or amplifies an upstream harmful state.
MV	Missed Verifier	held an activated checking duty with access or opportunity and did not validly discharge it ($D_o A_o B_o$=1).
CV	Contract Violation	breaches a role, authority, tool, safety, or workflow contract in $\mathcal{C}$.
SYM	Symptom	exposes the failure without originating it.
RL	Repair Leverage	is predicted to carry repair value.

To measure repair, we consider the smallest intervention a debugger can execute on a completed trace. Let $a_t = \text{Parse}(e_t)$ denote the answer that event e_t explicitly concludes, with $a_t = \perp$ if it concludes none, and let $\mathcal{Q}(\tau)$ be the set of logged event–answer pairs with $a_t \neq \perp$. A policy selects one pair $\hat{q} \in \mathcal{Q}(\tau)$, and the operator $I(\tau, \hat{q})$ replaces the failed final decision with its answer. A deterministic checker $h(\hat{y}, y) \in \{0, 1\}$, which returns 1 when a final answer $\hat{y}$ is correct, scores the result as $r_\tau = h(I(\tau, \hat{q}), y)$, and Repair@1 averages r_τ over all failed traces, counting traces without

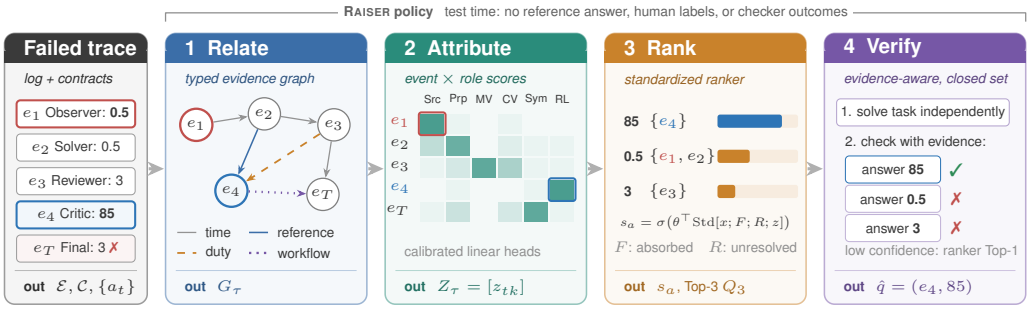

Figure 2: RAISER on the trace of Figure 1 (red: Source event e_1; blue: repair event e_4). (1) Deterministic parsers build a typed evidence graph. (2) Calibrated sparse heads score six roles per event. (3) Events are grouped by the answer they conclude; the group $\{e_1, e_2\}$ contains the Source and absorbs little responsibility, while $\{e_4\}$ ranks first. (4) A frozen verifier solves the task and checks the Top-3 groups, falling back to the ranker when unsure. The selected answer 85 replaces the failed final answer 3, and only then does the task checker score the result. Stages 1–3 make no generative calls; each stage can be swapped independently (Section 5.4).

a candidate as failures. The policy never observes y, human labels, Source annotations, or checker outcomes.

With roles and repairs defined on the same events, the two targets can be compared directly. For each trace, the Source set and the successful-repair set are

$$\mathcal{S}_\tau = \{e_t : \ell_{t,\mathrm{SRC}} = 1\}, \qquad \mathcal{R}_\tau = \{e_t : a_t \neq \bot \wedge h(a_t, y) = 1\}, \tag{1}$$

where the Source labels come from an independent annotation under a fixed rubric, never from the model being evaluated. We call the answer of any event in $\mathcal{R}_\tau$ a *correction*: an answer that some agent already concluded in the failed trace and that passes the checker. A trace is repairable when $\mathcal{R}_\tau \neq \varnothing$, and it exhibits source–repair mismatch when none of its corrections lies at a source:

$$\mathrm{SRM}(\tau) = \mathbf{1}[\mathcal{R}_\tau \neq \varnothing \wedge \mathcal{S}_\tau \cap \mathcal{R}_\tau = \varnothing]\,. \tag{2}$$

Over a cohort, $\hat{p}_{\mathrm{SRM}}$ is the fraction of repairable traces with $\mathrm{SRM}(\tau) = 1$. This rate directly bounds any policy that uses attribution as a candidate filter.

Proposition 1 (Source-restricted repair ceiling)**.** *Fix the Source rubric, answer parser, checker, and intervention operator. Let $\pi_{\mathcal{S}}$ be any policy whose choice on every trace lies in $\mathcal{S}_\tau$. On a failed-trace cohort $\mathcal{T}$ with repairable subset $\mathcal{T}_{\mathrm{rep}}$,*

$$\mathrm{Repair@1}(\pi_{\mathcal{S}}) \;\leq\; \frac{|\mathcal{T}_{\mathrm{rep}}|}{|\mathcal{T}|}\,\big(1 - \hat{p}_{\mathrm{SRM}}\big). \tag{3}$$

The bound is attained by a selector that knows which source events carry a correct answer.

The proof is a set-coverage argument (Section A.1). It holds for perfect localization and constrains only methods that use attribution as a candidate filter; a method that also considers non-source events escapes it. The point is practical: the dominant way attribution feeds repair today, intervening at the localized event, has a measurable ceiling well below what the logged trace already contains.

4 RAISER: RESPONSIBILITY-INFORMED REPAIR SELECTION

RAISER has four stages (Figure 2): build a typed evidence graph, estimate event roles, rank candidate answers by how they absorb responsibility, and verify a shortlist. Its output is one logged event–answer pair.

The first stage turns the flat event log into a structure on which responsibility can be reasoned about. Errors rarely stay where they enter: they travel along explicit references, are inherited by agents that build on earlier outputs, and escape when an agent with a checking duty does not act. Whether a later answer can repair the failure depends on the same paths. We therefore represent each trace as a typed graph whose edges record how information and duties connect events.

Trace Responsibility Evidence Graph. The TREG $G_\tau = (V, E, \psi)$ has one node per event and edges $E = E_{\text{temporal}} \cup E_{\text{data}} \cup E_{\text{ref}} \cup E_{\text{duty}} \cup E_{\text{workflow}}$, where duty and workflow edges cover MustCheck, Authority, Contract, Delegation, Handoff, and Trust relations. Deterministic parsers recover chronology, explicit references, agent transitions, and contract or checking cues. All experiments use this automatically extracted skeleton; adjudicated edges serve only as extraction targets (Section F).

Role estimation. The Role Model uses role-specific sparse linear heads. Let $x_t^{(S)}$ denote the structured features of event e_t (position, agent, stage, tool state, relation counts) and $x_t^{(T)}$ its word 1–2-gram TF–IDF vector. For role k, structured and text heads produce $s_{tk}^{(S)}$ and $s_{tk}^{(T)}$, which are blended with weight α_k. For Source and RL, a representation choice $\rho_k \in \{\text{split}, \text{event}, \text{context}\}$ may instead feed a joint logistic head with weights β_k on $[x_t^{(S)}; x_t^{(\rho_k)}]$, where $x_t^{(\rho_k)}$ is the event or context representation. The calibrated score is

$$z_{tk} = c_k(u_{tk}\,[\lambda_k + (1-\lambda_k)\,g_{tk}])\,, \qquad u_{tk} = \begin{cases} \alpha_k s_{tk}^{(S)} + (1-\alpha_k) s_{tk}^{(T)}, & \text{split}, \\ \sigma\Big(\beta_k^\top [x_t^{(S)}; x_t^{(\rho_k)}]\Big)\,, & \text{joint}, \end{cases} \tag{4}$$

where c_k is Platt calibration (Guo et al., 2017), and g_{tk} is an evidence gate with floor λ_k that lets MV condition on verifier or obligation evidence (g=1 for other roles). The Propagator head reads the event together with its graph predecessors. Representation, blend, gate, calibration, and thresholds are chosen on a development fold disjoint from both training and test traces.

Responsibility ranking. Every distinct parseable answer a defines a candidate sink Q_a, the set of events supporting it. Let m_{ik} be event i's responsibility mass for role channel k, derived from its calibrated score z_{ik}, $c_j \in [0,1]$ the sink capacity of supporting event j, and $d(i,j)$ the directed path cost over typed edges. At scale s, the fraction of responsibility that candidate a absorbs, and the fraction left unresolved, are

$$F_{aks} = \frac{\sum_i m_{ik} \max_{j \in Q_a} c_j\, e^{-d(i,j)/s}}{\sum_i m_{ik}}, \qquad R_{aks} = 1 - F_{aks}, \tag{5}$$

with $s \in \{0.75, 1.5, 3.0\}$ and zero flow for zero-mass channels; F_a and R_a stack these values over channels and scales. Intuitively, a good repair candidate sits downstream of the error, absorbs the responsibility that flowed into the failure, and leaves little of it unaccounted for. Together with local candidate features x_a (independent support, temporal closure, source contamination, recovery margin, downstream contradiction) and local role summaries z_a (the role scores of a's supporting events), these form the input of a sparse, class-balanced logistic ranker

$$s_a = \sigma\big(\theta^\top \mathrm{Std}\big([x_a;\, F_a;\, R_a;\, z_a]\big) + b\big)\,, \tag{6}$$

where σ is the logistic function, θ and b are learned parameters, and $\mathrm{Std}(\cdot)$ standardizes every block with statistics from 1,145 historical training traces. A head of the same family on x_a alone is the matched local ranker. An earlier sign-constrained variant, $s_a = w_L^\top x_a + w_F^\top F_a - w_R^\top R_a$ with $w_F, w_R \geq 0$, standardized only x_a; we keep it as a diagnostic in Section 5.4.

Evidence-aware verification. The top three candidates (all of them if fewer exist) are passed, anonymized and in hash order, to a frozen verifier (Qwen3.8-27B). Alongside each candidate the verifier receives the ranker's rank prior and six responsibility summaries (Source, Propagator, missed-verifier, contract-violation, repair-leverage, and boundary support), presented as fallible evidence. It solves the task independently and chooses one logged candidate; outputs with confidence below 0.8 fall back to the ranker's Top-1. The policy can only return a logged answer, and traces without a parseable candidate count as failures. The full procedure is Algorithm 1.

5 Experiments

5.1 Benchmark and Setup

Responsibility data. We normalize traces from Aegis, CORRECT-Error, MAST-MAD, TRAIL, TraceElephant, and Who&When (Kong et al., 2025; Yu et al., 2025; Cemri et al., 2025; Deshpande

et al., 2025; Chen et al., 2026; Zhang et al., 2025c). A model-consensus benchmark (484 traces, 5,516 events) is labeled by three open models with a 2-of-3 merge; 2,637 silver traces are used only for training. For external evaluation we collected a **human MV/CV Gold Test** of 1,600 items from 648 traces, disjoint from all training data. Two annotators label every item, and a blind expert resolves all 585 disagreements or uncertain cases. For SRM, two further blinded annotators label all 1,267 events of the 116 repairable traces for Source, without seeing prior labels, repair outcomes, or selector identities (Section B).

Repair cohorts. We use four cohorts. The *development panel* has 224 failures from AgentVerse, DyLAN, LLM-Debate, and MacNet on GSM8K, MATH, MMLU, and SciBench; it informed method design. A *validation cohort* of 346 Aegis failures was frozen before its first evaluation; its raw IDs overlap the development split, but no history or full record is shared. A *component test* of 1,200 failures was sampled after every component of the original pipeline was frozen; it requires at least three distinct candidate answers and excludes any query or history overlap with earlier data. The 196 eligible failures left after all earlier exclusions form a *held-out cohort*, scored once after the ranking heads were frozen. All final-system numbers use one current answer parser, which leaves 22 development failures without a candidate. Because eligibility and parser versions differ, absolute rates are not comparable across cohorts.

Baselines and statistics. We compare against earliest answer, answer majority, a BGE-small embedding selector (Xiao et al., 2023), direct and contract-aware prompting of Qwen, Devstral Small 2 (Mistral AI, 2025), and DeepSeek-V4-Pro (DeepSeek-AI, 2026), and adaptations of VerifyMAS (Qiao et al., 2026) and DeepDebug (Zhu et al., 2026b) that keep their inference decomposition but return one logged candidate. Two matched controls use the same verifier model without responsibility information: *query-only, all candidates* sees every candidate, and *query-only, local Top-3* sees the local ranker's shortlist. These two controls and RAISER are run with the current parser; the broader baselines reuse their frozen decisions on the same failed executions. All policies get one intervention and no reference.

Metrics. We use three metrics.

1) *SRM rate.* $\hat{p}_{\mathrm{SRM}}$ (Equation (2)) is the fraction of repairable traces in which no correction lies at an annotated source, with trace-bootstrap 95% intervals.
2) *Percentage of the candidate oracle.* Only failures whose log contains a correction can be repaired by a logged-answer policy, so we report Repair@1 as a percentage of the candidate oracle, which selects a correction whenever one exists (136 of 224 development and 229 of 346 validation failures under the current parser; on the component and held-out tests, the oracle over each three-item shortlist). Shortlist coverage uses the same unit; intervals are Wilson 95% intervals over repairable traces, and differences (Δ) are in oracle points.
3) *Ranking quality.* For a ranker alone, Rank@1 is the share of failures whose top-ranked candidate is a correction, candidate AP is average precision over all candidates, and NDCG is averaged over traces with corrections as relevant items.

Binary outcomes are compared with exact McNemar tests over discordant traces and ranking metrics with paired sign-flip randomization (20,000 draws); comparison families are Holm-corrected.

5.2 RQ1: Does the event where an error originates coincide with the event that can repair it?

Attribution-first debugging assumes that the event where an error entered the trace is also where it should be fixed. If this held, every repairable failure would have a correction at an annotated source, and SRM would be near zero. RQ1 therefore asks how often a repairable failure has no correction at any source, whether this rate survives a change of annotator and of Source definition, and how much repair a policy gives up by intervening only at sources.

Under the original broad-rubric labels, 82 of 116 repairable failures are mismatched ($\hat{p}_{\mathrm{SRM}}$=0.707, CI $[0.621, 0.784]$). The two blinded replications give 0.543 (63/116) and 0.733 (85/116). The annotators agree on 68.9% of events (κ=0.330), so annotation uncertainty affects the estimated *magnitude* of SRM. It does not determine the qualitative conclusion: taking the union of both blinded annotations

Table 2: Single-intervention repair on the 224-failure development cohort, as % of the candidate oracle (136 repairable failures under the current parser). Matched rows and RAISER are run with the current parser; the other rows reuse frozen decisions on the same failed executions. Δ is RAISER minus the row in oracle points; Holm p is over the full comparison family.

Group	Policy	% oracle [95% CI]	Δ	Holm p
Heuristic	Earliest answer	27.9 [21.1, 36.0]	+58.8	<0.001
	Answer majority	64.7 [56.4, 72.2]	+22.1	0.051
Encoder	BGE-small selector	64.0 [55.6, 71.6]	+22.8	0.035
LLM selector	Direct Qwen	68.4 [60.2, 75.6]	+18.4	0.152
	Contract-aware Qwen	68.4 [60.2, 75.6]	+18.4	0.152
	Contract-aware Devstral	67.6 [59.4, 74.9]	+19.1	0.152
	Direct DeepSeek-V4-Pro	75.0 [67.1, 81.5]	+11.8	0.517
	Contract-aware DeepSeek-V4-Pro	69.1 [60.9, 76.3]	+17.6	0.158
Debugging	VerifyMAS-style verification	75.0 [67.1, 81.5]	+11.8	0.517
	DeepDebug-style investigation	72.1 [64.0, 78.9]	+14.7	0.326
Matched	Query-only Qwen3.8, all candidates	85.3 [78.4, 90.3]	+1.5	1.000
	Query-only Qwen3.8, local Top-3	85.3 [78.4, 90.3]	+1.5	1.000
Ours	**RAISER**	**86.8** [80.0, 91.5]	–	–

mechanically expands the observed Source set and can only reduce mismatch, yet 38.8% of repairable traces remain source-disjoint (Figure 3). Thus our strongest event-level stress test is deliberately conservative with respect to the two observed Source label sets.

The union of the two annotations is the strongest test available to us. Counting an event as a source whenever *either* annotator marks it can only shrink mismatch, yet 45 of 116 traces (0.388, CI $[0.302, 0.474]$) remain disjoint. Six of these have no Source event in the union at all; excluding them without relabeling still leaves 39 of 110 (0.355). Mismatch is not an artifact of empty annotations.

This mismatch has a direct cost for repair. By Proposition 1, even an oracle that intervenes only at union-source events can repair at most 71 of the 116 repairable traces, so a source-restricted policy is capped at 71/116 = 61.2% of candidate-oracle logged repairs. On the full 224-failure cohort, the same bound is 71/224 = 31.7% absolute Repair@1. On the 45 union-mismatch traces, every repair lies downstream of the annotated sources, and the earliest-answer heuristic succeeds on only 28.9% of them.

5.3 RQ2: CAN A REPAIR-ORIENTED SELECTOR RECOVER THE CORRECTIONS ALREADY PRESENT IN FAILED TRACES?

A correction is a logged answer that some agent explicitly concluded in the failed trace and that passes the task checker (Section 3). RQ1 shows that corrections often lie away from the annotated sources, so a useful repair policy must find them without seeing the reference answer. RQ2 asks how much of the candidate oracle RAISER recovers, how it compares with prompting, debugging, and matched verifier baselines, and whether its ranking also improves attribution methods.

We begin with the development cohort. In Table 2, RAISER repairs 118 of the 136 repairable failures (86.8% of the oracle), the highest of all fourteen policies. It leads the prompting and debugging baselines by 11.8 to 19.1 oracle points, including DeepSeek-V4-Pro and the VerifyMAS-style verifier, and heuristic and encoder selectors by 22 to 59 points. Both matched controls, which use the same verifier model without responsibility information, reach 85.3%. The shortlist contains 93.4% of the oracle repairs.

The same pattern holds on the validation cohort. On its 346 failures (Table 10), RAISER recovers 85.6% of the oracle, against 81.2% for both matched controls: 4.4 oracle points, with 19 wins and 9 losses against the all-candidate control (p=0.087) and 16 wins and 6 losses against the local Top-3 control (p=0.053). The responsibility ranker with a query-only verifier sits in between at 83.4%, so the ranker and the evidence-aware verifier each add part of the gain. The shortlist contains 96.9% of the oracle repairs.

The gain does not come from showing the verifier more candidates: it sees 2.09 candidates per failure on average against 2.84 for the all-candidate control, and stages 1–3 add no generative calls.

Finally, repair ranking complements attribution rather than replacing it. Existing attribution methods can be turned into repair methods by acting at their top-ranked event, which recovers 60–66% of the oracle on the development cohort under the earlier parser. Re-ranking their Top-3 answers with RAISER's original ranking stage raises each to 80–83%, a gain of 13.8 to 22.4 oracle points ($p \leq 0.001$) with no new model calls (Table 9). What remains is coverage: attribution shortlists contain only 90.5–91.4% of the available repairs, which is Proposition 1 in practice: attribution helps order repairs but costs coverage when used as a filter.

5.4 RQ3: WHAT DOES RESPONSIBILITY CONTRIBUTE, AND WHERE IS THE HEADROOM?

The previous results establish RAISER as a strong repair selector but not which components drive the gain. We answer this with a preregistered pipeline decomposition, a responsibility-channel ablation on a fresh cohort, and an end-to-end test connecting the ablated rankers to the verifier. Before sampling the 1,200-failure component cohort, we froze the answer parser, local proposer, responsibility-based residual proposer, Role Model, TREG extractor, role-free ranker (M2), boundary-flow ranker, shortlist size $K{=}3$, verifier prompts, threshold, and fallback. The primary endpoint was shortlist coverage; each subsequent row changes exactly one component (Table 13).

Pipeline decomposition. Three findings follow.

Coverage is already high. A local proposer places a correct answer in the shortlist on 72.3% of failures, and a responsibility-derived residual candidate leaves this unchanged (5 wins, 8 losses; only 134 of 1,200 shortlists differ).

The verifier carries the ranking. The verifier sees candidates in hash order, so the ranker matters only on fallback (0.1 oracle points). As a standalone ranker, the original boundary-flow head is 2.3 points below the role-free ranker (15 wins, 35 losses), which we traced to its implementation: only local features were standardized, and the responsibility blocks were heavily regularized.

Recognition is where the headroom is. Row (a) recovers 80.6% of the repairs its shortlist contains; the rest is lost in choosing among three candidates.

Responsibility-channel ablation. We standardized every feature block on the 1,145 historical training traces, fit four additive logistic rankers of the same family (local only, + automatic TREG, + learned role channels, + both), froze them, and scored all 196 traces of the ranking confirmation set once. All rows share candidates, training data, and labels (Table 3). Two controls act on the frozen full model: one rotates the responsibility block among candidates within each trace, keeping its values but breaking their alignment; the other replaces it by its training mean.

The full model improves NDCG by 0.0115 (95% CI $[0.0041, 0.0206]$, Holm $p{=}0.007$) and candidate AP from 0.814 to 0.835. Rank@1 rises from 70.9% to 73.0% (four positive flips, none negative; exact $p{=}0.125$). The controls identify the source of the gain. Permuting the responsibility block within each trace lowers NDCG to 0.713, below the local-only ranker, so the model relies on which candidate the evidence belongs to rather than on extra dimensions. The gain appears only when graph and role evidence are combined, as in Section 4, where role mass is transported along graph edges.

From ranking to repair. We connected both frozen rankers to the frozen verifier on the same 196 failures (Table 3B); their shortlists differ on 37 traces. With query-only verification, repair is 66.3% versus 65.3% (3 wins, 1 loss), showing that the verifier discards most ordering information. When the verifier also receives the ranker's per-candidate evidence (Section 4), repair rises to 70.9%, 5.6 points above the local system (16 wins, 5 losses, CI $[1.0, 10.2]$, $p{=}0.027$) and 4.6 points above the same shortlist with a query-only verifier ($p{=}0.064$). Coverage moves only from 82.1% to 82.7%, so the gain is in recognition, the bottleneck identified above: the evidence-aware verifier overrides a correct ranker Top-1 on 20 traces instead of 29.

Across cohorts, the final system is above matched controls by +5.6 points on held-out failures ($p{=}0.027$), +4.4 oracle points on validation ($p{=}0.053$), and +1.5 on development.

Table 3: Responsibility ablation on 196 failures held out from all development. **(A)** All rows rank the same 825 candidates with the same model family and training traces; ΔNDCG is relative to local only for the additive rows and to the full model for the controls, with paired sign-flip p Holm-corrected over five contrasts. **(B)** Each ranker's Top-3 is passed to the frozen Qwen3.8-27B verifier. Coverage is the share of failures whose shortlist contains a correct answer; Repair@1 is given as a share of failures and, in parentheses, as % of the shortlist oracle. Δ and exact McNemar p are against the first row of the panel.

A. Ranking

	Ranker features	Cand. AP	NDCG	Rank@1	ΔNDCG (p)
Additive	Local only	0.814	0.787	70.9	–
	+ automatic TREG	0.819	0.786	70.9	−0.001
	+ learned role channels	0.815	0.793	72.4	+0.006
	+ TREG + role channels (full)	**0.835**	**0.798**	**73.0**	+0.012 (0.007)
Controls	Full, responsibility permuted	0.614	0.713	53.6	−0.085 (<0.001)
	Full, responsibility neutralized	0.741	0.770	67.3	−0.028 (<0.001)

B. End-to-end repair

	Ranker + verifier input	Coverage	Rank@1	Repair@1	Δ (p)
	Local + query only	82.1	70.9	65.3 (79.5)	–
	Responsibility + query only	82.7	73.0	66.3 (80.2)	+1.0 (0.625)
	Responsibility + responsibility evidence	82.7	73.0	**70.9** (**85.8**)	+5.6 (0.027)

5.5 RQ4: Is the responsibility measurement itself reliable?

The role labels are useful beyond repair only if they can be predicted and checked. The Role Model reaches Macro-F1 0.382 on the model-consensus benchmark against 0.324 for the best baseline (Table 7); graph-predecessor context raises Propagator F1 from 0.148 to 0.247 (p=0.022). On the independent human Gold Test it has the best Macro-F1 (0.378; Table 8) and higher Macro-AP than structured LR (0.395 versus 0.347, p=0.047).

Beyond fixed-prefix replacement, the selection interface also runs with real framework checkpoints in LangGraph and AutoGen, with identical outcomes across the two runtimes (Section E).

6 Limitations

Source labels use blinded LLM annotation rather than human experts (κ=0.33); the union stress test supports the qualitative claim, while human-adjudicated Sources would better estimate its magnitude. The bound is specific to source-restricted logged-answer repair. Earlier cohorts informed development; the final end-to-end test uses 196 held-out failures, and a new cohort would further test that gain. Public benchmark tasks may appear in pretraining data.

7 Conclusion

Error origin and repair location often differ: under the conservative Source union, source-restricted intervention covers at most 61.2% of candidate-oracle logged repairs (31.7% absolute Repair@1). On held-out failures, candidate-aligned responsibility evidence raises final Repair@1 from 65.3% to 70.9% when shared with the verifier. The final system also recovers 86–87% of candidate-oracle repairs on development and validation. Attribution and repair should therefore be measured separately.

Ethics Statement

The responsibility roles in this work describe software agents, not people, and should not be used for employment, legal, or disciplinary decisions. Human annotators labeled software trace records and contributed no personal or behavioral data. Dataset provenance, model involvement, and disagreement resolution are documented in Section B.

Reproducibility Statement

Code is available at https://anonymous.4open.science/r/Trace-R and data at https://anonymous-hf.com/a/psa97qn014rw/. We release normalized annotations, both Source workbooks, human Gold labels with provenance, frozen predictions and thresholds, all selector decisions, cohort manifests with hashes, runtime branch caches, and builders and tests. Cached decisions reproduce every reported metric without new model calls. Model identifiers, decoding settings, and seeds are listed in Section G.

AI Use Statement

Generative models were used for language editing and formatting, as disclosed components of the model-consensus labeling and Source annotation, and as selector and rerun baselines. They did not produce or alter the human Gold labels, expert adjudications, reference answers, or reported outcomes. The authors reviewed all AI-assisted text and code and take responsibility for the content.

References

Yarden Bakish, Amir Dudai, Roy Ganz, Oren Nuriel, Elad Ben Avraham, Mor Shpigel Nacson, and Ron Litman. Adaptive influence graphs for failure attribution in multi-agent systems. *arXiv preprint arXiv:2608.24361*, 2026.

Adi Banerjee, Anirudh Nair, and Tarik Borogovac. Where did it all go wrong? A hierarchical look into multi-agent error attribution. In *NeurIPS 2025 Workshop on Evaluating the Evolving LLM Lifecycle*, 2025. arXiv:2510.04886.

Shraddha Barke, Arnav Goyal, Alind Khare, Avaljot Singh, Suman Nath, and Chetan Bansal. AgentRx: Diagnosing AI agent failures from execution trajectories. *arXiv preprint arXiv:2602.02475*, 2026.

Jiayi Bi, Yanjie Gao, Yuanmin Xie, Liqun Li, Tianyin Xu, Fan Yang, and Mao Yang. Diagnosing with insights: Structured analysis of agent failures via behavioral abstractions. *arXiv preprint arXiv:2609.02371*, 2026.

Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. CausalFlow: Causal attribution and counterfactual repair for LLM agent failures. *arXiv preprint arXiv:2605.25338*, 2026.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? *arXiv preprint arXiv:2503.13657*, 2025.

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*, pages 19888–19905, 2026.

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 *International Conference on Learning Representations*, 2024.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*, 2023.

Hana Chockler and Joseph Y. Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

DeepSeek-AI. DeepSeek-V4-Pro API model documentation. https://api-docs.deepseek.com/quick_start/pricing/, 2026.

Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. TRAIL: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*, 2025.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*, 2023.

Yu Ge, Linna Xie, Zhong Li, Yu Pei, and Tian Zhang. Who is introducing the failure? Automatically attributing failures of multi-agent systems via spectrum analysis. *arXiv preprint arXiv:2509.13782*, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330, 2017.

Jen-Tse Huang, Jiaxu Zhou, Tailin Jin, Xuhui Zhou, Zixi Chen, Wenxuan Wang, Youliang Yuan, Michael Lyu, and Maarten Sap. On the resilience of LLM-based multi-agent collaboration with faulty agents. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 26202–26226, 2025.

Yeonjun In, Mehrab Tanjim, Jayakumar Subramanian, Sungchul Kim, Uttaran Bhattacharya, Wonjoong Kim, Sangwu Park, Somdeb Sarkhel, and Chanyoung Park. Rethinking failure attribution in multi-agent systems: A multi-perspective benchmark and evaluation. *arXiv preprint arXiv:2603.25001*, 2026.

Yuhua Jiang, Yuwen Xiong, Yufeng Yuan, Chao Xin, Wenyuan Xu, Yu Yue, Qianchuan Zhao, and Lin Yan. PAG: Multi-turn reinforced LLM self-correction with policy as generative verifier. *arXiv preprint arXiv:2506.10406*, 2025.

Dong Ho Kang, Hyeonjeong Cha, and Daein Weon. Knowledge-based zero-replay debugging of multi-agent LLM traces. *arXiv preprint arXiv:2606.14805*, 2026.

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. *arXiv preprint arXiv:2509.14295*, 2025.

Jiazheng Li, Emine Yilmaz, Bei Chen, and Dieu-Thu Le. Towards self-improving error diagnosis in multi-agent systems. *arXiv preprint arXiv:2604.17658*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Yang Liu, Hongjiang Feng, Junsong Pu, and Zhuangbin Chen. MASPrism: Lightweight failure attribution for multi-agent systems using prefill-stage signals. *arXiv preprint arXiv:2605.07509*, 2026.

Zhongwen Luan, Xiaoyu Zhang, Ming Hu, Yue Yang, Jiongchi Yu, and Xiaohong Chen. Repair or resample? rethinking failure debugging in LLM multi-agent systems. *arXiv preprint arXiv:2608.25920*, 2026.

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. *arXiv preprint arXiv:2512.06749*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023.

Mistral AI. Devstral Small 2. https://huggingface.co/mistralai/Devstral-Small-2-24B-Instruct-2512, 2025.

Hezhe Qiao, Hanghang Tong, Ee-Peng Lim, Bing Liu, and Guansong Pang. VerifyMAS: Error-focused analysis of multi-agent systems via hypothesis verification. *arXiv preprint arXiv:2605.17467*, 2026.

João Sedoc, Baotong Zhang, and Dean Foster. Trust but verify: Prover-verifier deliberation for selective LLM prediction. *arXiv preprint arXiv:2605.25133*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Yawen Wang, Wenjie Wu, Junjie Wang, and Qing Wang. From flat logs to causal graphs: Hierarchical failure attribution for LLM-based multi-agent systems. *arXiv preprint arXiv:2602.23701*, 2026.

Zhaohui Geoffrey Wang. AgentTrace: Causal graph tracing for root cause analysis in deployed multi-agent systems. *arXiv preprint arXiv:2603.14688*, 2026.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561*, 2022.

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. *arXiv preprint arXiv:2308.08155*, 2023.

Yufei Xia, Anjun Gao, Yueyang Quan, Zhuqing Liu, and Minghong Fang. AgentLocate: Multi-agent error localization via independent multi-perspective verification. *arXiv preprint arXiv:2607.07989*, 2026.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-Pack: Packaged resources to advance general Chinese embedding. *arXiv preprint arXiv:2309.07597*, 2023.

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. *arXiv preprint arXiv:2509.24088*, 2025.

Boxuan Zhang, Jianing Zhu, Zeru Shi, Dongfang Liu, and Ruixiang Tang. AgentForesight: Online auditing for early failure prediction in multi-agent systems. *arXiv preprint arXiv:2605.08715*, 2026.

Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who is inducing failure in the LLM agentic systems? *arXiv preprint arXiv:2509.03312*, 2025a.

Heng Zhang, Yuling Shi, Xiaodong Gu, Haochen You, Zijian Zhang, Lubin Gan, Yilei Yuan, and Jin Huang. GraphTracer: Graph-guided failure tracing in LLM agents for robust multi-turn deep search. *arXiv preprint arXiv:2510.10581*, 2025b.

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. *arXiv preprint arXiv:2505.00212*, 2025c.

Chenyang Zhu, Spencer Hong, Jingyu Wu, Kushal Chawla, Yuhui Tang, Youbing Yin, Nathan Wolfe, Erin Babinsky, and Daben Liu. RAFFLES: Reasoning-based attribution of faults for LLM systems. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics*, pages 7659–7688, 2026a.

Kunlun Zhu, Xuyan Ye, Zhiguang Han, Yuchen Zhao, Bingxuan Li, Weijia Zhang, Muxin Tian, Xiangru Tang, Pan Lu, James Zou, Jiaxuan You, and Heng Ji. AgentDebugX: An open-source toolkit for failure observability, attribution, and recovery in LLM agents. *arXiv preprint arXiv:2607.18754*, 2026b.

Taiyu Zhu, Yifan Wu, Weilin Jin, Ying Li, and Gang Huang. StepFinder: A temporal semantic framework for failure attribution in multi-agent systems. *arXiv preprint arXiv:2606.03467*, 2026c.

A METHOD DETAILS

A.1 PROOF OF PROPOSITION 1

For any trace, a source-restricted policy chooses an event in $\mathcal{S}_\tau$, and its intervention succeeds only if that event is in $\mathcal{R}_\tau$. Hence $r_\tau(\pi_\mathcal{S}) \le \mathbf{1}[\mathcal{S}_\tau \cap \mathcal{R}_\tau \neq \varnothing]$. Non-repairable traces contribute zero. Summing over $\mathcal{T}$,

$$\text{Repair@1}(\pi_\mathcal{S}) \le \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}_{\text{rep}}} \mathbf{1}[\mathcal{S}_\tau \cap \mathcal{R}_\tau \neq \varnothing] = \frac{|\mathcal{T}_{\text{rep}}|}{|\mathcal{T}|}(1 - \hat{p}_{\text{SRM}}),$$

using $\hat{p}_{\text{SRM}} = |\mathcal{T}_{\text{rep}}|^{-1} \sum_{\tau \in \mathcal{T}_{\text{rep}}} \text{SRM}(\tau)$. A selector that picks a correct source event whenever one exists attains equality. Under the A/B union, 116 traces are repairable and 45 disjoint, so at most 71 overlaps remain and the ceiling on the 224-trace cohort is $71/224 = 0.317$.

Monotonicity of the union. Since $\mathcal{S}_\tau^A, \mathcal{S}_\tau^B \subseteq \mathcal{S}_\tau^A \cup \mathcal{S}_\tau^B$, a trace disjoint under the union is disjoint under each annotator, so $\hat{p}_{\text{SRM}}(A \cup B) \le \min\{\hat{p}_{\text{SRM}}(A), \hat{p}_{\text{SRM}}(B)\}$. The union is therefore a conservative estimate relative to the two observed label sets, though not a population lower bound under expert labels.

A.2 SELECTION ALGORITHM

Algorithm 1 RAISER closed logged-candidate repair

Require: failed trace τ; TREG G_τ; role scores Z_τ; ranker s_θ; verifier V; threshold $\gamma = 0.8$
Ensure: selected logged intervention $\hat{q}$

$\mathcal{Q}(\tau) \leftarrow$ GROUPLOGGEDANSWERS(τ)
if $\mathcal{Q}(\tau) = \varnothing$ **then return** $\perp$
end if
for $q \in \mathcal{Q}(\tau)$ **do**
  $(x_q, F_q, R_q, z_q) \leftarrow$ EVIDENCE(q, G_τ, Z_τ); $s_q \leftarrow s_\theta\big(\text{Std}[x_q; F_q; R_q; z_q]\big)$
end for
$Q_3 \leftarrow$ TOPK$(\mathcal{Q}(\tau), s, 3)$
$(a_v, J, q_v, u) \leftarrow$ VERIFY$(V, \tau, Q_3, \{(s_q, z_q, F_q)\}_{q \in Q_3})$ ▷ evidence-aware
if VALID$(J, q_v) \wedge u \ge \gamma$ **then**
  $\hat{q} \leftarrow$ MAPEXACT(a_v, Q_3) if defined, else q_v
else
  $\hat{q} \leftarrow \arg\max_{q \in \mathcal{Q}(\tau)} s_q$
end if
return $\hat{q}$ ▷ $\hat{q}$ is always a logged answer; y is revealed only to the checker

A.3 TRAINING AND CROSS-FITTING

Stable trace hashes define five outer folds. For outer fold j, fold j is test, fold $(j{+}1) \bmod 5$ is development, and the remaining model-benchmark traces plus eligible silver data form the training pool. Each role objective is a class-balanced, tier-weighted logistic loss (weight 1 for benchmark events, 0.25 for silver) with ℓ_2 regularization. The development fold selects representation, regularization, MV supervision branch and gate, blend weights, calibration maps, and thresholds. The ranker uses three inner trace folds to select regularization before refitting on outer-training traces. Human Gold, Source annotations, future events, reference answers, and test outcomes are excluded from every fitted component.

Complexity. TREG construction is $O(T + |E|)$ and role scoring $O(KTd)$ for T events, $|E|$ relations, $K{=}6$ roles, and feature dimension d. Boundary features cost $O(KT|\mathcal{Q}|)$ after shortest-path preprocessing.

Table 4: Evidence tiers. Tiers are never pooled into one test metric.

Tier	Traces	Events/items	Positives	Use
Model-consensus benchmark	484	5,516	42 MV / 124 CV	Role test targets
Model silver	2,637	43,077	867 MV / 659 CV	Training only
Human MV/CV Gold Test	648	1,600	114 MV / 200 CV	External test
Blinded Event-Source audit	116	1,267	412 / 503 Source	SRM
Development repair cohort	224	2,412	–	Repair test

Table 5: Model-consensus label inventory by source dataset.

Dataset	Traces	Events	Src	Prop	MV	CV	Sym	RL
Aegis	72	899	182	66	35	104	28	76
CORRECT-Error	345	3,836	123	52	0	18	30	88
MAST-MAD	2	6	0	0	1	0	1	1
TRAIL	12	144	1	6	0	0	1	1
TraceElephant	25	291	5	15	0	0	4	2
Who&When	28	340	17	35	6	2	22	11

B Data and Annotation

Human Gold agreement. The pool contains 850 MV and 750 CV candidates. In the first round, observed agreement was 0.626 (κ=0.134) over 1,111 certain pairs. After the rubric was extended with explicit obligation, authority, opportunity, and discharge boundary cases, a positive-enriched supplement reached 0.894 agreement (κ=0.783) over 358 certain pairs. All 585 disagreements or uncertain items were resolved by a blind expert.

Event-Source audit. Annotators A and B labeled the same 1,267 blinded events under the same broad rubric, marking 412 and 503 Sources. Exact agreement is 0.689 (κ=0.330). No disagreement was overwritten; released rows retain both labels, confidences, and rationales. Neither workbook enters any fitted component.

Table 6: SRM under alternative Source definitions on the same 116 repairable traces.

Source definition	Level	Source events	Mismatch	SRM	95% CI
Original three-model broad	event	429	82	0.707	[0.621, 0.784]
Annotator A	event	412	63	0.543	[0.448, 0.629]
Annotator B	event	503	85	0.733	[0.647, 0.810]
A ∪ B (stress test)	event	654	45	0.388	[0.302, 0.474]
A ∩ B (positive agreement)	event	261	106	0.914	[0.862, 0.957]
Strict-first-source union	event	268	66	0.569	[0.474, 0.655]
Earliest event per faulty agent	agent	293	39	0.336	[0.250, 0.422]
All events by faulty agents	agent	736	22	0.190	[0.121, 0.267]

C Responsibility Results

Source-agent localization. Against 470 original benchmark faulty-agent labels, the Role Model reaches 0.691 agent top-1 accuracy, above random (0.474), last checkpoint (0.577), and structured LR (0.626), and tied with TF–IDF LR (0.694). We do not claim dominance on this source-only task.

Calibration and transfer. After development-fold Platt scaling, ECE is at most 0.016 for every role. Low ECE does not imply strong ranking for rare roles (MV AUPRC 0.174 with 42 positives). Strict leave-one-dataset-out transfer is weak; selecting per-role thresholds from 5, 10, or 20 labeled target traces improves the eligible-domain macro score by 0.007, 0.011, and 0.032.

Table 7: Event-role F1 on the model-consensus benchmark under trace-grouped five-fold cross-validation (484 traces, 5,516 events).

Method	Macro	Src	Prop	MV	CV	Sym	RL
Random	0.063	0.123	0.055	0.015	0.036	0.039	0.112
Last checkpoint	0.094	0.163	0.046	0.034	0.062	0.063	0.193
Local-event LR	0.228	0.395	0.139	0.162	0.208	0.193	0.274
Structured LR	0.242	0.416	0.131	0.161	0.276	0.125	0.345
TF–IDF LR	0.316	0.496	0.160	0.316	0.322	0.261	0.343
Structured MLP	0.226	0.457	0.123	0.104	0.211	0.163	0.301
Ungated linear	0.324	0.517	0.158	0.333	0.342	0.229	0.365
RAISER Role Model	**0.382**	**0.534**	**0.247**	**0.396**	**0.356**	**0.326**	**0.435**

Table 8: Human MV/CV Gold Test (1,600 items, 648 traces). F1 at threshold 0.5; AP is threshold-free. Role Model versus structured LR: Macro-AP $+0.049$ (CI $[+0.003, +0.089]$, p=0.047).

Method	MV F1	MV AP	CV F1	CV AP	Macro-F1	Macro-AP
Last verifier	0.260	0.226	0.274	0.358	0.267	0.292
Graph centrality	0.236	0.113	0.389	0.285	0.312	0.199
Structured LR	0.288	0.205	0.451	0.488	0.370	0.347
TF–IDF LR	0.254	**0.248**	0.366	**0.572**	0.310	**0.410**
RAISER Role Model	**0.293**	**0.248**	**0.463**	0.542	**0.378**	0.395
By annotation stratum (Role Model / structured LR Macro-AP)						
Certain agreement (n=1,015): 0.523 / 0.468				Expert-resolved (n=585): 0.404 / 0.318		

D REPAIR DETAILS

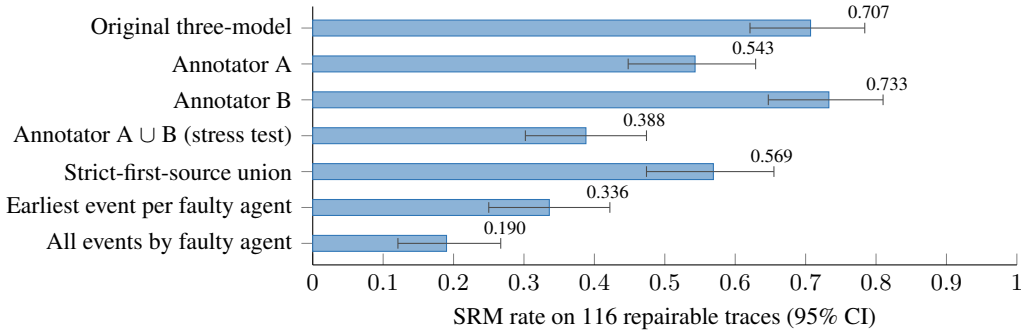

Figure 3: Source–repair mismatch under seven Source definitions, all on the same repair outcomes. The top four rows use the broad event-level rubric; their union can only lower SRM and serves as a stress test. The strict-first-source row changes the rubric, and the bottom two rows switch to agent-level definitions, a different estimand.

Earlier candidate-discovery evidence. On the development pool, removing the role-derived proposal from an earlier restricted-pool selector lowered Repair@1 from 0.446 to 0.397 (p=0.007), without an equal-budget replacement. On a 243-trace role-disjoint audit, rerun after correcting three candidate-parsing errors that were found after outcomes had been inspected, adding a responsibility residual candidate raised oracle coverage from 0.646 to 0.675 (Holm p=0.031); pruning back to the original budget retained $+0.012$ (p=0.25). The preregistered test in Table 13 was designed to confirm this effect and did not; we report both.

Failure cases. Majority succeeds and the earlier provenance selector fails on only 2 of 224 traces. Both contain conflicting downstream corrections: in an AgentVerse/MMLU trace, two evaluator events disagree and the later, incorrect one is ranked first; in a DyLAN/MMLU trace, verifier-

Table 9: Attribution-to-repair bridge on the 224 development failures, as % of the candidate oracle under the earlier parser (116 repairable failures), using RAISER's original ranking stage. Each attribution front end exposes its Top-3 answer groups; the frozen ranking stage then selects one. *Coverage* is the share of oracle repairs present in those three groups. p is the exact paired test against the front end's own Top-1.

Attribution front end	Coverage	Attribution Top-1	+ ranking stage	Δ	p
Last verifier	90.5	66.4	**80.2**	+13.8	0.001
Graph centrality	91.4	60.3	**82.8**	+22.4	<0.001
DoVer-style evidence	90.5	62.1	**81.9**	+19.8	<0.001

Table 10: Validation cohort of 346 Aegis failures, as % of the candidate oracle (229 repairable failures under the current parser). All rows use the current parser, Qwen3.8-27B, the 0.8 threshold, and the same fallback; three traces without a candidate count as failures. RAISER's Top-3 shortlist contains 96.9% of the oracle repairs. p is exact McNemar against RAISER.

Policy	Repaired	% oracle [95% CI]	p
RAISER	196 / 229	**85.6** [80.5, 89.6]	–
Responsibility ranker, query-only verifier	191 / 229	83.4 [78.0, 87.7]	–
Query-only Qwen3.8, all candidates	186 / 229	81.2 [75.7, 85.8]	0.087
Query-only Qwen3.8, local Top-3	186 / 229	81.2 [75.7, 85.8]	0.053

like correction cues overweight an unsupported option. Both are ranking errors among available candidates, consistent with Section 5.4.

Component test protocol. The 1,200 traces were drawn by salted round-robin over framework and benchmark (AgentVerse 209, DyLAN 378, LLM-Debate 288, MacNet 325) from 1,396 eligible failures, before any candidate was compared with a reference. The verifier (qwen3.8:27b, temperature 0, seed 20260914) ran 2,534 deduplicated jobs; 2,523 parsed, and the remaining 11 followed the frozen fallback. Retries never used correctness labels. The protocol and cohort manifests are released with their hashes.

D.1 RESPONSIBILITY-CHANNEL RANKING ABLATION PROTOCOL

The 1,200-failure decomposition exposed two implementation issues in the original boundary-flow ranker: only local features were standardized, and the sign-constrained responsibility blocks were strongly regularized. We used that cohort only to develop the ablation family. Before selecting any further trace, we fit four logistic heads on the same 1,145 historical training traces (4,432 candidate groups): local only; local plus automatic TREG; local plus learned role and certificate channels; and their union. Every feature is standardized from the historical training records, and no confirmation outcome enters fitting or scaling. The serialized model hash begins `6605920e180c34a`.

The confirmation set contains all 196 eligible failures remaining after exact query and history exclusion of the historical data, Role-Model sources, and the 224-, 243-, 346-, and 1,200-failure cohorts, with 196 distinct queries and 825 candidates. Framework and benchmark counts were fixed in the cohort manifest (hash prefix `f58af0dea2fb06d1`) before features were computed, and candidate scores were written (hash prefix `3894a96bc0a4971f`) before checker labels were joined. NDCG inference uses 20,000 trace-paired bootstrap draws and 20,000 sign-flip draws, with Holm correction over five contrasts; the permutation and neutral controls give Holm p=0.00025 and 0.00040. The Rank@1 contrast has four wins, no losses, and exact p=0.125. The permutation control deterministically rotates all non-local responsibility columns among the candidate rows of each trace, preserving the trace's channel marginals while breaking alignment. The neutral control sets those columns to their historical training means. Neither control retrains the model or changes the candidate space.

End-to-end connection. Following the ranking analysis, a second staged protocol was frozen on the same 196 traces. Both revised rankers select exactly three candidates. Candidates are labeled

Table 11: Development-cohort Repair@1 by benchmark (absolute rates). Qwen is the current-parser query-only all-candidate control; majority and DeepSeek reuse frozen decisions.

Benchmark	n	Majority	DeepSeek-V4-Pro	Query-only Qwen3.8	RAISER
GSM8K	63	0.381	0.397	0.571	**0.587**
MATH	56	0.500	**0.589**	0.536	0.500
MMLU	39	0.308	0.385	0.590	**0.641**
SciBench	66	0.364	**0.439**	0.409	0.424

Table 12: Source–repair mismatch examples, selected by a fixed rule (earliest answer and majority fail, one case per environment). Event answers are checker-verified.

Environment / task	Source	Majority	Repair event	Why they differ
AgentVerse / GSM8K	E2: 0.5	E6: 3	E12 (Critic): 85	A downstream checker corrects the source.
DyLAN / MMLU	E1: B	E1: B	E6 (Mathematician): A	A downstream agent recomputes and corrects.
LLM-Debate / MATH	E3: 498	E3: 498	E4 (Assistant1): 284	An independent branch reaches the answer.

in a deterministic hash order rather than score order, and when the two shortlists are identical the query-only arms reuse the same verifier call. The evidence-aware arm keeps the responsibility shortlist and adds the ranker's frozen rank prior and the Source, Propagator, missed-verifier, contract-violation, repair-leverage, and boundary-support values, all described to the verifier as fallible evidence. Preparation reads outcome-blind event records and rejects any record with a nonzero success field; all prompts are serialized before the verifier is called, and checker labels are joined only at evaluation. The local and responsibility rankers change Top-3 membership on 37 of the 196 traces. Deduplication leaves 429 Qwen3.8-27B calls at temperature 0 and seed 20260923, all of which parse as valid. Hash prefixes: prepared jobs `7f684465bbd12ee8`, outputs `9ae7aa4d16f39c18`, per-trace records `618aef375f8b589d`. The evidence-aware system reaches Repair@1 0.709 against 0.653 for local/query-only (16 wins, 5 losses, exact p=0.0266, paired bootstrap CI $[0.010, 0.102]$) and against 0.663 for responsibility/query-only (14 wins, 5 losses, p=0.0636); responsibility/query-only versus local/query-only is 3 wins and 1 loss (p=0.625). The released per-trace file keeps verifier decisions, fallback status, and selected answer keys.

E STATEFUL AND NATIVE EXECUTION

The runs in this section use the selector interface that preceded the final evidence-aware system; they test whether the intervention ports to stateful execution.

Checkpointed LangGraph and AutoGen. We generated 150 current-model failures (93 tasks, five seeds) in a four-role solver–challenger–verifier–decider workflow and froze them before scoring any selector. For each trace, both policies receive the same candidate answers; the runtime restores the real checkpoint before the chosen event, re-executes that role under a common repair instruction, and regenerates all downstream events without access to the post-event transcript. Both policies repair 3 of 150 (0.020) in both runtimes; the frozen candidate oracle is 0.067, and cross-runtime outcome agreement is 1.000.

Native AgentVerse. Running the official AgentVerse code path on all 50 AgentVerse failures, with the selected event initializing the native advice state, RAISER repairs 19/50 (0.380) against 23/50 (0.460) for majority and 21/50 (0.420) with no hint (p=0.29). This operator conditions regeneration on advice rather than replacing the answer, and results are not comparable to fixed-prefix repair. We therefore treat native regeneration as a separate estimand and portability check, not as evidence for the paper's primary logged-answer claim: RAISER is designed for completed traces in which the intervention selects one already-logged conclusion, not as a universal trajectory-regeneration debugger.

Table 13: Preregistered component decomposition on 1,200 unseen failures, as % of the shortlist oracle (a correct answer is in the three-item shortlist on 72.3% of failures in row (a) and 72.1% in rows (b)–(d)). Each row changes one component relative to the row above. Paired exact tests: proposal, shortlist oracle -0.25 points (p=0.58); ranking, Rank@1 -2.3 points (p=0.007); verifier context, Repair@1 $+0.8$ points (p=0.51).

	Candidate set	Ranker	Verifier input	Rank@1	Repair@1
(a)	Local	Role-free M2	Query only	80.6	80.6
(b)	+ resp. residual	Role-free M2	Query only	80.9	80.3
(c)	as (b)	Boundary-flow	Query only	78.6	80.2
(d)	as (b)	Boundary-flow	+ resp. context	78.6	**81.0**

F Automatic Relation Extraction

Against adjudicated edges, the deterministic skeleton has near-perfect precision on every edge type, and recall near 1.0 for temporal, data-flow, reference, contract, handoff, delegation, and trust edges, but 0.206 for MustCheck and 0.594 for Authority. A learned pairwise extractor raises MustCheck AP from 0.253 to 0.288 (precision 0.153, recall 0.362 at an F2 operating point). All reported experiments use the deterministic skeleton so that no adjudicated information reaches test-time inputs.

G Implementation

The model-consensus layer uses qwen3.6:35b-a3b, gpt-oss:20b, and gemma4:31b at temperature 0 with a 32,768-token context. Selector baselines use temperature 0 and seed 2026; DeepSeek-V4-Pro runs with thinking disabled. The verifier is local qwen3.8:27b at temperature 0 with thinking disabled and deterministic answer anonymization (seed 20260822 on the development cohort, 20260829 on the untouched cohort). The embedding baseline is BGE-small-en-v1.5. Replay folds use seeds 2026–2030. Full prompts, token counts, and served model fingerprints are included in the release.