Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.36917

RoundValue: A Complete-Trajectory Audit of Repair, Harm, and Correct-Candidate Non-Uptake in Multi-agent Reasoning

Zhiqi Hu

Lyle School of Engineering, Southern Methodist University, Dallas, USA
zhiqi@smu.edu

Abstract. Additional rounds of multi-agent deliberation may correct, preserve, or worsen an earlier answer, but endpoint accuracy alone cannot reveal these trajectory-level events. RoundValue provides a trajectory audit of the full five-round, seven-call-per-round debate process using 350 stored trajectories from seven 50-task model–benchmark conditions. For locally served profiles, outputs are capped at 768 tokens. The framework measures round-level accuracy, adjacent correction and degradation transitions, six task-level trajectory archetypes, and the relationship between blind Stage-1 correct-candidate emergence and Writer ever-repair. Across conditions, R1-to-R5 accuracy changes range from +6 to −6 percentage points. Repair occurs in six conditions, while answer degradation affecting initially correct responses also appears in six. On Human Annotated Reasoning Problems (HARP), Mistral-7B shows at least one incorrect intermediate checkpoint in 8 of 10 R1-correct trajectories, with five recovering by R5. Correct candidates emerge without Writer repair in six of seven conditions, while two repairs occur without detected structured candidate emergence. These results show that additional rounds can both repair and degrade answers, and that candidate emergence and Writer repair are distinct empirical events.

Keywords: RoundValue, multi-agent deliberation, trajectory audit, answer degradation

1. Introduction

Multi-agent reasoning systems repeatedly generate candidates, critique them, and aggregate the discussion into a final answer. Their behavior is commonly compressed into a single endpoint score or a comparison between first-round and final-round accuracy. That view is incomplete: the same net change can arise from stable answers, several repaired errors, or repairs and degradations that cancel one another. It also misses cases in which a structured candidate contains the reference option, but the later Writer checkpoint remains wrong. Understanding a stored multi-round run therefore requires tracking both Writer answer states and candidate-option emergence.

For every task, the stored record contains all five Writer checkpoints and the structured outputs of the debate nodes. This supports three complementary views of the same trajectory. Adjacent-round transitions identify corrections and degradations hidden by aggregate accuracy. Task-level archetypes distinguish persistent correctness, persistent wrongness, repair, temporary repair, harm,

DOI: 10.54254/2755-2721/2026.36917

and recovery. Blind Stage-1 candidate analysis records whether the reference option appears before any later Writer repair.

RoundValue operationalizes complete-trajectory auditing and is applied to seven model–benchmark conditions containing 350 trajectories in total. All available complete trajectories in these selected conditions contribute to the analysis. Conditions are reported separately because model, benchmark, serving provider, architecture, and quantization are not controlled treatments. The comparison describes recurring trajectory events while preserving differences among deployment settings.

Three research questions organize the study:

RQ1: How do Writer correctness and adjacent correction/degradation events evolve across five rounds?

RQ2: Which task-level trajectory archetypes occur, and how do their frequencies vary across conditions?

RQ3: Among R1-wrong trajectories, how does blind Stage-1 correct-candidate emergence relate to Writer ever-repair?

The results support an observational conclusion. Repair, degradation, recovery, and correct-candidate non-uptake all occur in the stored trajectories, but their frequencies are strongly condition-dependent. Because each task has only one sampled trajectory, these frequencies describe realized events rather than intrinsic task-level probabilities and do not identify the mechanism behind any transition.

2. Related work

Multi-agent debate allocates additional test-time computation to alternative hypotheses, criticism, and revision [1]. Self-refinement similarly uses iterative feedback to revise model outputs [2]. Carefully matched comparisons show that no debate strategy dominates across all evaluated settings [3]. These findings motivate examining intermediate trajectories rather than assuming that more interaction monotonically improves an answer.

Selective collaboration routes tasks among multi-agent configurations [4]. Related work gates debate using an initial signal [5]. Early-stopping methods likewise seek to terminate reasoning when additional computation is unlikely to help [6]. These lines of work motivate measurements at intermediate checkpoints. The present study supplies descriptive measurements of answer-state transitions and candidate-to-Writer outcomes within fixed-horizon trajectories; it does not infer which trajectory should have continued prospectively.

3. Methodology

3.1. Fixed checkpointed debate protocol

The protocol has five rounds, R1–R5, and seven logical model calls per round. In Stage 1, a Planner, Analyst, and Critic independently receive the public task. Their structured outputs form a deterministic packet. In Stage 2, another Planner, Analyst, and Critic receive that packet together with the preceding Writer checkpoint when one exists. A Writer then produces the canonical option and a concise reasoning summary.

Figure 1 shows the fixed two-stage topology used in every round. The public task is first sent independently to the Stage-1 Planner, Analyst, and Critic, and their structured outputs are combined into a deterministic Stage-1 packet. At R2–R5, these Stage-1 roles remain blind to the previous

DOI: 10.54254/2755-2721/2026.36917

Writer checkpoint, so their structured candidate_answer fields provide a diagnostic of whether the reference option can re-enter the discussion independently of direct checkpoint copying.

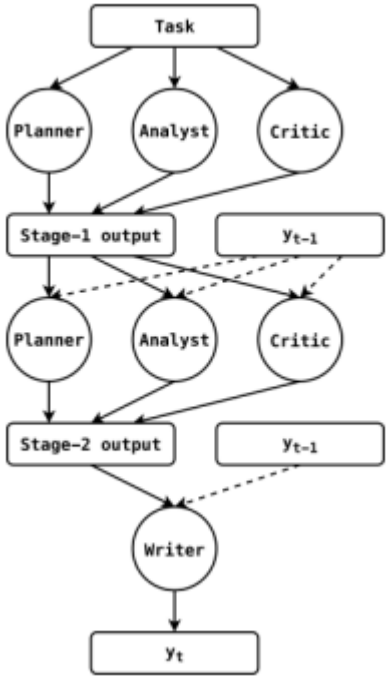

Figure 1. Fixed two-stage debate topology (photo credit: original)

Figure 1 also shows how Stage 2 receives the Stage-1 packet together with the preceding Writer checkpoint when one exists. A second Planner, Analyst, and Critic process this combined context; their outputs form the Stage-2 packet, and the Writer produces the canonical checkpoint. One selected model supplies all roles within a run, and the scorer normalizes the Writer's answer to one option and uses binary exact-option correctness; reasoning text cannot rescue an incorrect option. This fixed topology is treated as part of the deployment protocol because the communication structure can affect both beneficial information diffusion and error propagation in multi-agent systems [7].

3.2. Complete-trajectory evidence

Seven complete 50-task conditions are audited. Five serving profiles are compared on the same HARP-50 subset: DeepSeek-V4-Flash, Generative Pre-trained Transformer (GPT-5-nano), GPT-4o-mini, locally served Qwen2.5-7B-Instruct Q5_K_M, and locally served Mistral-7B-Instruct-v0.3 Q8_0. DeepSeek-V4-Flash is also audited on Massive Multitask Language Understanding (MMLU-Pro-50) and Logical Question Answering (LogiQA-50). The shared DeepSeek × HARP run belongs to both descriptive slices, yielding seven unique conditions and 350 unique task trajectories.

MMLU-Pro provides a more difficult extension of multi-task language understanding evaluation [8]. HARP supplies the human-annotated mathematical reasoning subset [9]. LogiQA 2.0 supplies the natural-language logical reasoning subset [10]. The three subsets are analyzed as complementary diagnostic settings rather than interchangeable samples from one population.

The local runs preserve the benchmark subset, task identities, role prompts, topology, five-round horizon, temperature 0.3, scoring, and audit definitions. They use a 768-token output safety cap and

DOI: 10.54254/2755-2721/2026.36917

an observed 4096-token Ollama runtime context on one RTX 5070 Ti laptop. Architecture, quantization, serving stack, and context behavior differ, so the local profiles are deployment replications rather than controlled model ablations.

Figure 2 summarizes the evidence flow. The seven conditions are organized into two descriptive slices: five model profiles on HARP-50 and one model profile across three benchmarks. Every complete five-round trajectory contributes to the condition-level audit, and the shared DeepSeek × HARP condition is counted only once in the total of 350 trajectories. The analysis recomputes Writer correctness from raw stored checkpoints and reference options and re-reads the Stage-1 node outputs from the original trajectory files. Using the same task trajectory as the unit for accuracy, transitions, archetypes, and candidate emergence keeps the measurements aligned and makes every aggregate count traceable to stored model outputs.

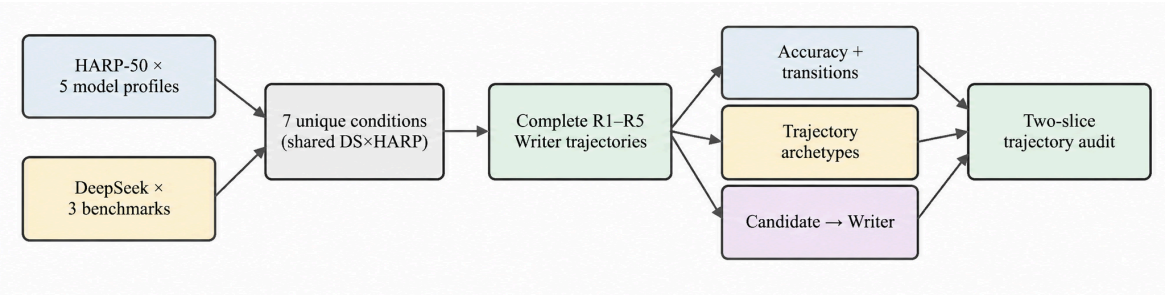

Figure 2. RoundValue complete-trajectory audit pipeline (photo credit: original)

3.3. Audit definitions

Let $Q_{i,t} \in \{0,1\}$ denote Writer correctness for task i after round t. For adjacent rounds, a correction transition occurs when $Q_{i,t} = 0$ and $Q_{i,t+1} = 1$; a degradation transition occurs when $Q_{i,t} = 1$ and $Q_{i,t+1} = 0$. The two unchanged states are recorded separately.

For an R1-wrong trajectory, every repair is:

$$R_i = 1[Q_{i,1} = 0 \wedge \max_{2 \le t \le 5} Q_{i,t} = 1] \tag{1}$$

For an R1-correct trajectory, every harm is:

$$H_i = 1[Q_{i,1} = 1 \wedge \min_{2 \le t \le 5} Q_{i,t} = 0] \tag{2}$$

Each task is assigned to exactly one of six archetypes: correct throughout; harmed but correct again at R5; harmed and wrong at R5; repaired and correct at R5; repaired but wrong again at R5; or wrong throughout. "Final correct" describes the R5 state and does not assert that the trajectory remained correct continuously after its first repair.

For R1-wrong tasks only, blind correct-candidate emergence is present when the reference option appears in the structured candidate_answer field of any Stage-1 Planner, Analyst, or Critic at R2–R5. This gold-derived annotation is computed only after the run and is not an input available to the agents. Crossing emergence with ever repair yields four exhaustive outcomes: neither emergence nor

DOI: 10.54254/2755-2721/2026.36917

repair; emergence without repair; repair without detected emergence; and emergence with repair. This cross-tabulation records co-occurrence over R2–R5 and does not impose temporal ordering between the first detected candidate emergence and the first Writer repair.

3.4. Statistical reporting

The task trajectory is the observational unit. Counts and proportions are reported separately for every condition, without cross-condition pooling. Conditional repair and harm rates are accompanied by 95% Wilson score intervals [11]. The intervals describe finite samples of tasks under one sampled trajectory per task; they do not account for trajectory-to-trajectory variation under repeated stochastic runs.

The reported quantities use different, explicit denominators. Round accuracy always uses all 50 tasks in a condition. Correction and degradation counts are specific to each adjacent-round boundary, so one task can contribute transitions at several boundaries. Ever-repair and ever-harm are trajectory-level indicators and count each eligible task at most once; their denominators are the R1-wrong and R1-correct subsets, respectively. Candidate–Writer outcomes are restricted to R1-wrong trajectories because the question is whether a later correct candidate coincides with any subsequent repair. Keeping these denominators separate prevents a high repair proportion in a small error pool from being read as a large condition-wide gain.

4. Results

4.1. Aggregate accuracy conceals task turnover

All result figures use the same two-panel separation. Panel (a) compares five model/serving profiles on the frozen HARP-50 tasks; panel (b) compares three benchmarks with DeepSeek-V4-Flash fixed. DeepSeek-V4-Flash × HARP-50 is the single shared condition and is marked with a star in both panels. It is counted once among the seven unique conditions.

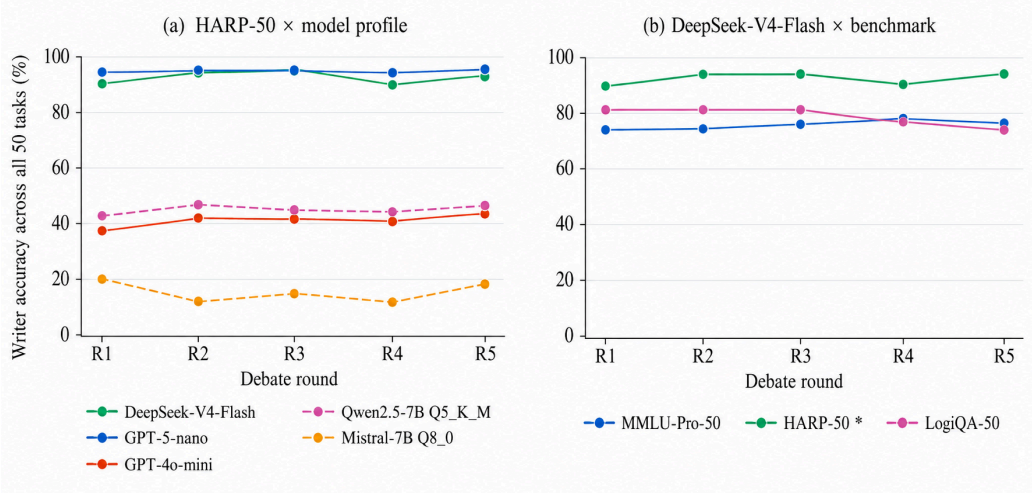

Figure 3. Five-round Writer accuracy by evidence slice (photo credit: original)

Figure 3 uses all 50 tasks per condition. R1-to-R5 accuracy changes are +2 percentage points for DeepSeek × MMLU-Pro, +4 for DeepSeek × HARP, 0 for GPT-5-nano × HARP, +6 for GPT-4o-mini × HARP, +4 for Qwen × HARP, −2 for Mistral × HARP, and −6 for DeepSeek × LogiQA.

DOI: 10.54254/2755-2721/2026.36917

Intermediate rounds are not monotonic: Mistral falls from 20% at R1 to 12% at R2, while DeepSeek × LogiQA remains at 82% through R3 before declining.

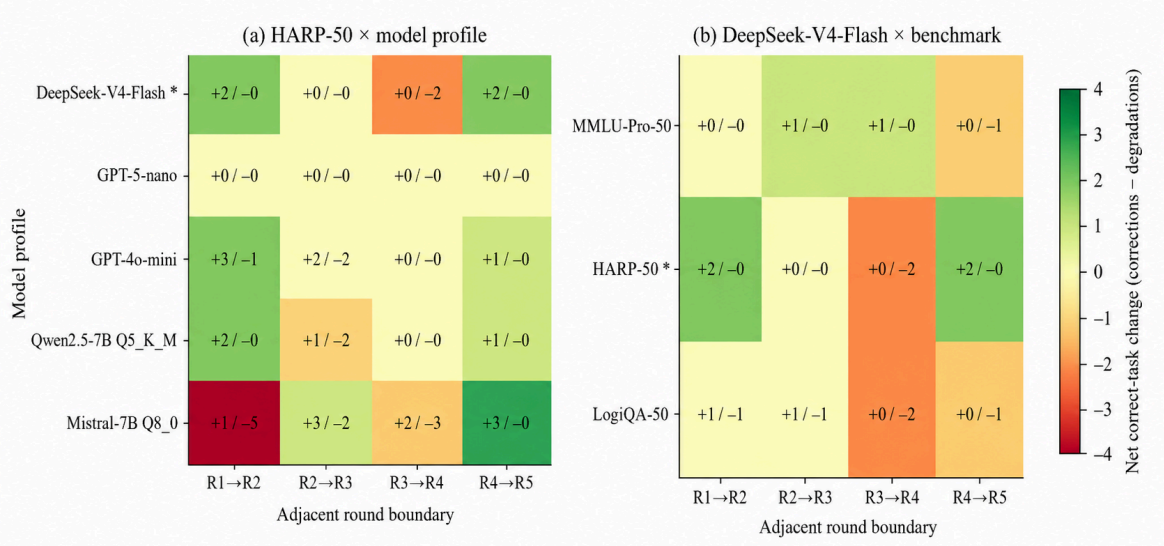

Figure 4. Adjacent-round correction and degradation counts (photo credit: original)

Figure 4 exposes the task turnover behind these curves. Each cell reports the number of adjacent corrections and degradations. DeepSeek × HARP gains two correct tasks at R1→R2, loses two at R3→R4, and gains two again at R4→R5. Its aggregate curve therefore hides both beneficial and harmful transitions. Mistral shows the strongest instability: R1→R2 contains one correction but five degradations, while R4→R5 contains three corrections and no degradations.

These observations rule out a monotonic interpretation of additional rounds in the sampled trajectories. They do not determine whether the changes are caused by debate, sampling variability, model-specific aggregation, or another component.

4.2. Persistent outcomes dominate, but repair and harm coexist

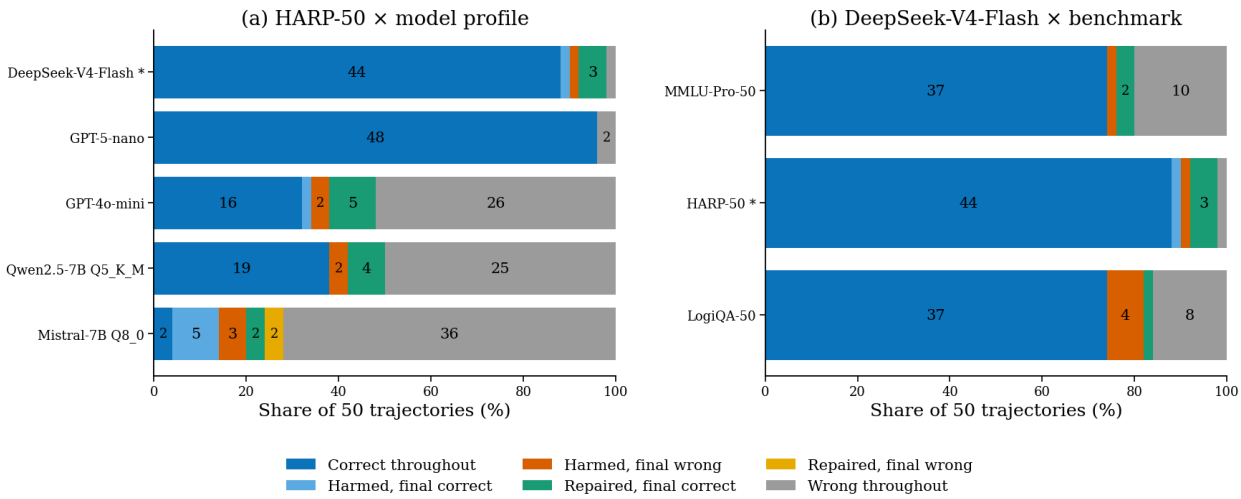

Figure 5. Task-level trajectory archetypes by evidence slice (photo credit: original)

DOI: 10.54254/2755-2721/2026.36917

Figure 5 partitions all 50 tasks in each condition into the six mutually exclusive archetypes. GPT-5-nano is maximally stable in this sample: 48 trajectories are correct throughout, and two are wrong throughout. At the other extreme, Mistral has only two trajectories correct throughout, 36 wrong throughout, five harmed then correct again by R5, three harmed and still wrong at R5, two repaired and correct at R5, and two repaired but wrong again at R5.

Table 1 reports the conditional opportunity and risk denominators. DeepSeek × HARP repairs 3 of 4 R1 errors, but this high point estimate rests on a very small denominator. GPT-4o-mini and Qwen expose larger R1-error pools but repair only 5 of 31 and 4 of 29, respectively. Mistral harms 8 of its 10 initially correct trajectories. DeepSeek × LogiQA harms 4 of 41 R1-correct trajectories and ends six percentage points below R1 despite one R1-error repair.

Table 1. Conditional repair and harm counts by condition

Condition	R1→R5 accuracy	R1 wrong	Ever repaired	R1 correct	Ever harmed
DeepSeek × MMLU-Pro	76%→78%	12	2 (16.7%)	38	1 (2.6%)
DeepSeek × LogiQA	82%→76%	9	1 (11.1%)	41	4 (9.8%)
DeepSeek × HARP	92%→96%	4	3 (75.0%)	46	2 (4.3%)
GPT-5-nano × HARP	96%→96%	2	0 (0.0%)	48	0 (0.0%)
GPT-4o-mini × HARP	38%→44%	31	5 (16.1%)	19	3 (15.8%)
Qwen2.5-7B × HARP	42%→46%	29	4 (13.8%)	21	2 (9.5%)
Mistral-7B × HARP	20%→18%	40	4 (10.0%)	10	8 (80.0%)

4.3. Correct options can appear without later Writer repair

Figure 6 crosses blind Stage-1 correct-candidate emergence with Writer ever-repair among R1-wrong trajectories. Correct candidates appear without repair in six of seven conditions: 4 DeepSeek × MMLU-Pro tasks, 2 GPT-5-nano tasks, 8 GPT-4o-mini tasks, 7 Qwen tasks, 12 Mistral tasks, and 3 DeepSeek × LogiQA tasks. Under this field-level diagnostic, the presence of the reference option is therefore not sufficient for Writer repair during R2-R5 in the observed trajectories.

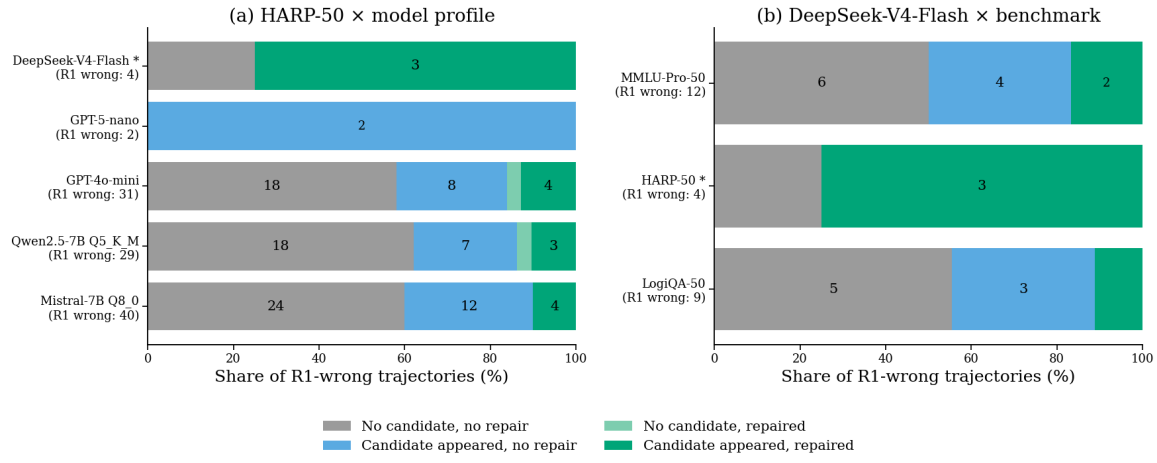

Figure 6. Correct-candidate emergence and Writer ever-repair (photo credit: original)

Two repairs—one GPT-4o-mini task and one Qwen task—occur without a detected correct structured Stage-1 candidate. Candidate emergence is therefore not necessary under this operational

DOI: 10.54254/2755-2721/2026.36917

definition. Conversely, emergence without repair does not identify where later non-uptake arose: the correct option may have been weakly or incorrectly supported, and the relevant loss could occur during comparison, criticism, aggregation, or Writer generation. The measured result is a candidate–Writer outcome gap, not a diagnosis of its cause.

Although the aggregation mechanisms differ, related evidence from heterogeneous-agent majority voting likewise shows that a correct minority answer can be suppressed during collective aggregation [12]. The comparison supports treating candidate availability and final uptake as separate measurements, while leaving the causal source of non-uptake unresolved in the present protocol.

4.4. Repair and harm occupy different conditional pools

Figure 7 places repair opportunity and harm risk on separate denominators while retaining the two evidence slices as columns. The top row asks, among tasks that are wrong at R1, how often any later checkpoint is correct. The bottom row asks, among tasks that are correct at R1, how often any later checkpoint becomes wrong. Vertical bars are Wilson 95% intervals.

A single endpoint change cannot summarize these profiles. DeepSeek × HARP has a high observed repair rate and low harm rate, but the repair denominator is four. GPT-4o-mini and Qwen exhibit both repair and nonzero harm. Mistral combines a low 4/40 repair rate with an 8/10 harm rate. The two measures answer different conditional questions and can move independently, while the wide intervals emphasize that some first-round error or correct pools are small.

Table 1 and Figure 7 also show why counts and conditional rates must be read together. Three repaired DeepSeek × HARP trajectories correspond to 75% because only four tasks are wrong at R1, whereas four repaired Mistral trajectories correspond to 10% because 40 tasks are wrong at R1. On the harm side, eight harmed Mistral trajectories correspond to 80% of the ten tasks that are correct at R1.

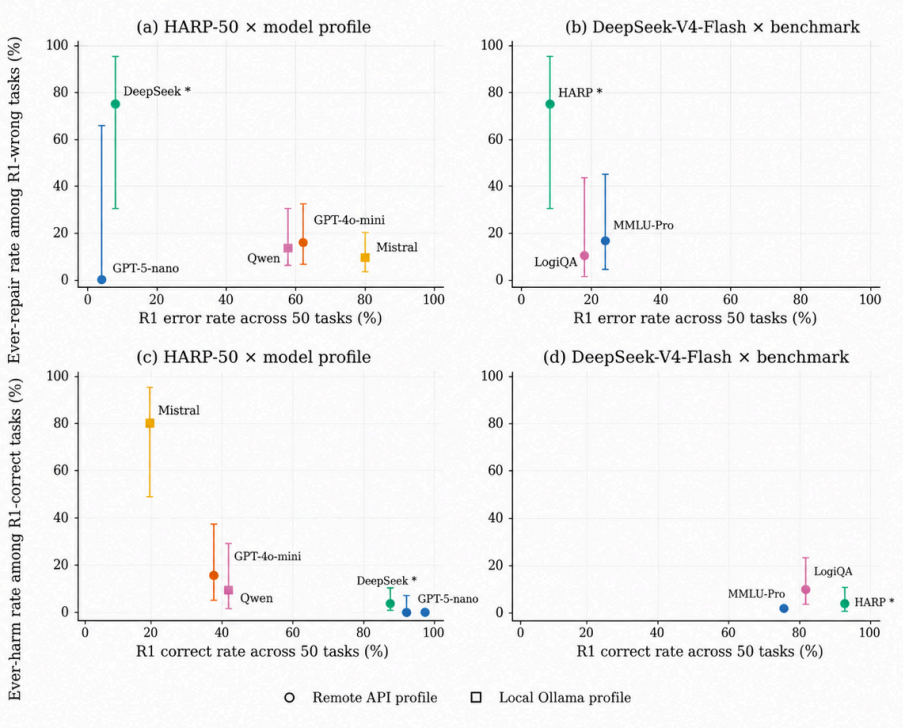

Figure 7. Conditional repair and harm rates by evidence slice (photo credit: original)

DOI: 10.54254/2755-2721/2026.36917

5. Conclusion

Across seven 50-task conditions, the sampled multi-round trajectories contain corrections, degradations, recovery, persistent error, and condition-specific instability. Aggregate round accuracy can hide offsetting task transitions, while structured correct candidates can appear without later Writer repair, showing that option emergence and subsequent Writer correctness are distinct recorded events under this diagnostic. These findings are descriptive rather than causal: each task has only one sampled trajectory, so ever-repair and ever-harm reflect realized five-round events rather than stable task-level probabilities. Conditional denominators are sometimes small—for example, four R1 errors for DeepSeek × HARP and ten R1-correct tasks for Mistral—and Wilson intervals do not account for benchmark-subset selection or model-sampling variation.

The seven conditions also do not form a factorial experiment; model family, provider, quantization, output limits, runtime context, and benchmark can differ, so cross-condition contrasts cannot isolate any single factor. Candidate emergence is measured only from the structured Stage-1 candidate_answer field at R2–R5 and does not capture ambiguous mentions or reasoning quality, while exact-option scoring reduces Writer checkpoints to binary states. Thus, RoundValue makes repair, harm, recovery, and candidate non-uptake visible without assigning them to a specific mechanism. Future work should repeat trajectories across seeds, expand independent task counts, and jointly evaluate option correctness, argument quality, confidence, and evidence provenance before attributing non-uptake to Writer selection.

References

[1] Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., & Mordatch, I. (2024). Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning (Vol. 235)*, 11733–11763. PMLR.

[2] Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., & Clark, P. (2023). Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (Vol. 36)*.

[3] Smit, A. P., Grinsztajn, N., Duckworth, P., Barrett, T. D., & Pretorius, A. (2024). Should we be going MAD? A look at multi-agent debate strategies for LLMs. In *Proceedings of the 41st International Conference on Machine Learning (Vol. 235)*, 45883–45905. PMLR.

[4] Yue, Y., Zhang, G., Liu, B., Wan, G., Wang, K., Cheng, D., & Qi, Y. (2025). MasRouter: Learning to route LLMs for multi-agent systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 15549–15572.

[5] Eo, S., Moon, H., Zi, E. H., Park, C., & Lim, H. (2025). Debate only when necessary: Adaptive multiagent collaboration for efficient LLM reasoning. arXiv preprint arXiv:2504.05047.

[6] Sun, R., Cheng, W., Li, D., Chen, H., & Wang, W. (2026). Stop when enough: Adaptive early-stopping for chain-of-thought reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 27250–27268.

[7] Shen, X., Liu, Y., Dai, Y., Wang, Y., Miao, R., Tan, Y., Pan, S., & Wang, X. (2025). Understanding the information propagation effects of communication topologies in LLM-based multi-agent systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing* (pp. 12347–12361).

[8] Wang, Y., Ma, X., Zhang, G., Ni, Y., Chandra, A., Guo, S., Ren, W., Arulraj, A., He, X., Jiang, Z., Li, T., Ku, M., Wang, K., Zhuang, A., Fan, R., Yue, X., & Chen, W. (2024). MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems (Vol. 37)*.

[9] Yue, A. S., Madaan, L., Moskovitz, T., Strouse, D. J., & Singh, A. K. (2024). HARP: A challenging human-annotated math reasoning benchmark. *arXiv preprint arXiv:2412.08819*.

[10] Liu, H., Liu, J., Cui, L., Teng, Z., Duan, N., Zhou, M., & Zhang, Y. (2023). LogiQA 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31*, 2947–2962.

DOI: 10.54254/2755-2721/2026.36917

[11] Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association, 22*(158), 209–212.
[12] He, C., Chen, Z., Yang, Z., Qiao, S., Ju, M., Liu, J., Wen, D., & Liu, G. (2026). Minority Sentinel: When to overturn majority voting in multi-agent LLM debates. *arXiv preprint arXiv:2606.29270*.