INTERVENED PARTIAL REPLAY FOR MULTI-TURN AGENT TRAJECTORY REPAIR

Anonymous authors
Paper under double-blind review

ABSTRACT

The growing presence of AI agents in real-world applications is contrasted by their very limited reliability in task execution, resulting in agent failures that are intrinsically context-dependent and laborious to fix on a case-by-case basis. Failures in multi-turn interactions often occur at long horizons that are hard to foresee at task initiation. To understand and mitigate these failures, we consider measuring agent reliability in multi-turn interactions as life testing and construct local insertion fixes through agent-agent interactions. Inspired by repair mechanisms in conversations and programs, we propose an offline replay method to produce quality instruction interventions using best-of-N sampling, which induce behavioral changes in the agent trajectory leading to repair. Our replay-based approach delegates trajectory intervention to a separate agent, thereby eliminating the cognitive capacity needed for self-initiated repair. Using a single localized feedback around failure within the agent trajectory, we reduce the inference cost and demonstrate significant performance improvement on existing long-horizon agent tasks without any model capability enhancement, as quantified by the success rate and turn-indexed survival metrics. The results offer insights to designing verbalized controls to shape multiagent interactions that underlie many existing AI agent operations.

1 INTRODUCTION

AI agents based on large language models (LLMs) are an essential endpoint for delivering the values of generative AI to specialized software applications. One of the major challenges of these systems is the high failure rate (Cemri et al., 2025), leading to compromised reliability. Moreover, the patterns of agent failures depend on the domain and structure of the workflow (Song et al., 2025; Ye et al., 2025; Gu et al., 2025), making their resolution labor-intensive. Long-horizon agent actions that proceed over multiple turns of interactions are important for real-world implementations, such as in customer support (Yao et al., 2025), medical diagnostics (Tu et al., 2025), and generative coding (Jain et al., 2025). Ongoing efforts primarily focus on designing taxonomy-based failure diagnostics and prevention methods (Cemri et al., 2025; Gu et al., 2025). In multi-turn interactions, performance degradation is attributed to the limitations of the inherent memory mechanisms of AI systems (Maharana et al., 2024; Laban et al., 2026), error accumulation and compounding (Lin et al., 2025; Cheng et al., 2026), resulting in long-horizon degradation (Kwan et al., 2024; Graf et al., 2026). For typical problems, updating models is not always the most convenient solution because of unanticipated side effects associated with an upgrade (Chen et al., 2024) and the cost of adjusting other components in an existing pipeline (Belcak et al., 2025). Existing research on AI adaptation focuses on model weight finetuning and test-time scaling (Xiao & Snoek, 2024; Jiang et al., 2025) to accommodate changes in the environment, human needs, and requirements. These techniques still face challenge in multi-turn interactions because of ballooning cost, latency, and unspecific targeting

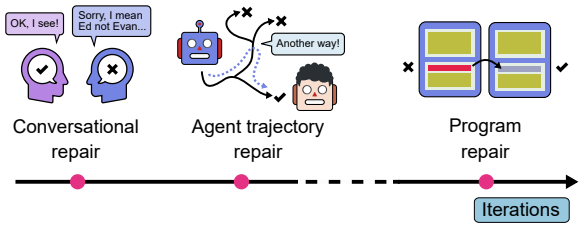


Figure 1: The repair process for agent trajectories integrates aspects from conversational repair (turn-taking) and program repair (iterative fault localization). The number of interactions increases from left to right.

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.

that affects downstream applications. Specifically for agent failure mitigation, two promising classes of approaches are emerging: On one hand, enhancing metacognition can counteract the deficiency in an agent's self-assessment of its past history and uncertainty (Steyvers & Peters, 2025; Johnson et al., 2024; Liu & Van Der Schaar, 2025); on the other hand, advancing cooperative infrastructure can improve system resilience in out-of-distribution scenarios in long interaction horizons and open-ended environment (Chacon-Chamorro et al., 2025; Xian et al., 2025b). These approaches converge on designing structured guidance to facilitate agents to recover from errors and thereby avert failures.

Driven by these limitations, we consider agent failure mitigation through replay-based repair. Across symbolic systems, repair processes manifest distinctively from fully conversational (Albert & de Ruiter, 2018) to fully programmatic (Monperrus, 2018). Interactions between AI agents straddle both sides (Fig. 1), so the design of repair mechanisms may employ the essential elements of both worlds. AI agents are engineered systems so their repair should focus on minimal adjustments to recover desirable behavior.[1] The sequential nature of multi-turn interactions (Zhang et al., 2025a) indicates that repairing an agent trajectory *in situ* only needs to re-direct a subset of missteps, rather than all steps, to avert failures (called interruption in (Orseau & Armstrong, 2016)) while still maintaining the ability to reproduce prior results in a rollback. Attaining these goals leaves room for exploring other avenues for solutions.

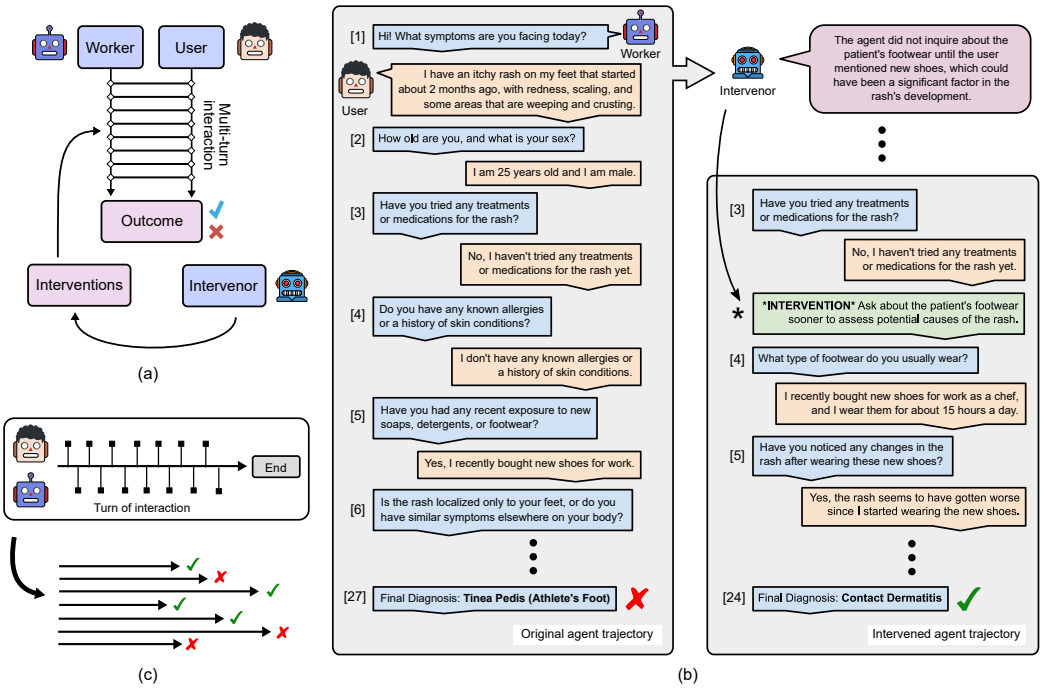


Figure 2: (a) Agent-agent interaction diagram including the agents representing the worker, user, and intervenor. (b) An example to illustrate the effect of an intervention that leads to performance repair in the agent trajectory. On the right trajectory, the footwear question was initiated earlier and in the context of the rash. (c) Illustration of life test of multi-turn systems with binary outcomes.

In the present work, we quantitatively study the benefits of local interventions in long-horizon agent actions (>10–20 turns of interactions), we consider failure monitoring in an agent trajectory as life testing (Epstein, 1954), which anchors the reliability assessment and enhancement challenge within repeated evaluations (Ascher & Feingold, 1984; Lawless, 2002) (Sec. 3.1). Inspired by insertion repair (Wilkinson & Weatherall, 2011; Albert & de Ruiter, 2018) in communications and third-party processes in organizations (French & Bell, 1999), we introduce an intervention initiator (i.e., *intervenor*) external to the agentic system to induce trajectory repair. The design of our method (Sec. 3.2), intervened partial replay (IPR), also incorporates elements of turn-taking and fault localization in repair processes (Fig. 1), which leaves a small inference footprint (Sec. 3.3) in the main agents

[1] Repairable systems are established in engineering practices that concern system lifetime and behavior under internal and external degradations that lead to failures (Ascher & Feingold, 1984).

handling the task completion since the thinking processes from self-initiated reflections are sourced externally (Bouzenia et al., 2025). Our main contributions include: (1) the formulation of turn-indexed life test for LLM agents to study their reliability in multi-turn interactions; (2) development of a best-of-N (BoN) sampling algorithm for agent trajectory repair that identifies potential interventions favorable to enhancing the reliability of multi-turn interactions; (3) event-based analysis of agent failures for quantifying the effects of instruction interventions.

2 Related works

Structured multiagent interactions Agent-agent communication (Chen et al., 2026) is a key paradigm in facilitating multiagent collaboration (Feng et al., 2025) for LLM agents. Structured interactions among agents are generally engineered according to a protocol that stipulates role-playing and turn-taking. Examples for LLM agent interaction protocols include the sequential order in multiagent debate (Du et al., 2024) or execution and aggregation patterns in decentralized agent workflows. In either case, the structured multiagent interactions benefit from role differentiation among the agents in a team or organizational setting. Agent communications can be improved through protocol pruning (Li et al., 2024b; Zhang et al., 2025b), which reduces unnecessary inputs and interference from specific participants in the interactions.

Agent failures and reliability Production-level AI agents face significant challenges with high failure rates coupled with system complexity and scale (Pan et al., 2025; Kim et al., 2025). The challenges in making AI agents reliable take on different layers of meaning (Rabanser et al., 2026). For repeated agent actions such as tool use, reliability can be realized through robustness certification against environmental perturbations (Yeon et al., 2025). For multiagent interactions, reliability can be enhanced through tooling infrastructure or behavioral science such as using a separate LLM agent to provide feedback (Kirstein et al., 2025; Chakraborty et al., 2025) in the course of interactions.

Inference-time improvement Inference-time scaling based on refinement allows models to improve its generation through meta-generators that iterates over preliminarily generated responses (Didolkar et al., 2024; Ji-An et al., 2025), inspired by metacognitive monitoring in human problem-solving (Flavell, 1976). This class of approaches can be carried out by the model (or agent) alone within the entire context window or through external memory infrastructure that stores the past history of intermediate results. Inference-time scaling has been shown to improve adversarial robustness (Zaremba et al., 2025) and general capabilities (Kamoi et al., 2024). The main class of approaches uses self-correction but suffers from self-bias (Xu et al., 2024). The other class of methods employs an external critic to evaluate intermediate results to provide signals for refining the responses (Estornell et al., 2025; Dong et al., 2025; Tang et al., 2025). These approaches are generally limited to tasks requiring single-turn interactions. Their extensions to long-horizon, multi-turn tasks remain underexplored.

3 Intervened replay for multi-turn reliability

3.1 Life test for multi-turn reliability

Turn-indexed life test Life tests (LTs) assess the distribution of persistent system operation before failure occurs (Epstein, 1954; Lawless, 2002). We model failure occurrences in multi-turn interactions as discrete events to assess system reliability (Thompson, 1981). We define a *turn-indexed LT* using the turn of interaction to parametrize time since it is a natural choice for multi-turn conversations (Sacks et al., 1974). In this setting, the test yields data $\mathcal{D} = \{(T_i, S_i)_{i=1}^{|\mathcal{D}|}\}$ with the total number of turns T_i and a success/failure ($S_i = 1$ or 0) record for each tested example.

Reliability quantification We reinterpret the corresponding metric for reliability modeling (Ascher & Feingold, 1984) as the mean turn to failure (MTTF), which is an outcome-based population average over all test cases associated with a task. For turn-indexed life test, we can empirically reconstruct time-bounded MTTF through the reliability function (aka. survival function), $R(t)$, by

$$\text{MTTF}(T_-, T_+) = \sum_{T_- < T \leq T_+} R(T), \qquad T_+ > T_{\max} = \max\{T_i\}_{i=1}^{|\mathcal{D}|} \tag{1}$$

The sum over turn indices runs from the start of the interaction T_- (e.g., $T_- \geqslant 0$) to a designated endpoint T_+ that exceeds all task completion requirements ($T_{\max}$). $R(T)$ represents the marginal probability of no failure (i.e., survival) until the Tth turn of interaction. In a life test, $R(T)$ is generally estimated from empirical data (Epstein, 1954; Lawless, 2002), which we also adopt in this work. As shown in Fig. 2c, in multi-turn interactions between two agents, failures don't necessarily lead to immediate termination of subsequent actions because their influence is assessed at the end of the interaction.

Def. 1 Reliable multi-turn system *Given the upper turn limit for the interactions T_+ according to design constraints (e.g., context limit, timeout, cost), and the intended task i requires T_i turns (or steps) to complete ($T_i < T_+$). The agentic system is considered reliable for all tasks, $\{i = 1, ..., |\mathcal{D}|\}$, if the MTTF of interactions satisfies MTTF $> T_{\max}$ (Eq. 1). Two multi-turn systems (i.e., built with different models or hyperparameters) can compare reliability through their respective MTTF metrics.*

The MTTF in Def. 1 may be estimated through an agent testing protocol, which has a (practical) design limit T_+ defined by utility or cost. For example, $T_+ = 10$ is used in Deshpande et al. (2025) for multiple task domains; $T_+ = 20$ is used in Ferber et al. (2026) for evaluating medical agents.

3.2 Intervened partial replay

Failure assumptions and intervention design In this work, we use *instruction interventions* to refer to context-specific instructions used to alter agent trajectory to alter worker performance and prevent policy violations (Algorithm 1). For a single agent task carried out step-by-step, it is assumed that (**A1**) the task failure (assessed at the outcome in Fig. 2) of an agent has a discrete temporal onset in its trajectory, referred to as failure points. (**A2**) There can be multiple independent failure points in a single trajectory that result in a task failure. (**A3**) Failure points emerged at T_ε affect subsequent steps ($T > T_\varepsilon$) in the multi-turn interaction due to the negative consequences (i.e., accumulation and compounding) of error conditioning (Cheng et al., 2026). Therefore, the instruction intervention provides a prior and timely reminder to the agent to prevent it from occurring.

Def 2. Intervention strategy *The task-dependent policy for an agent to interrupt and alter the trajectory of another's course of interaction towards a goal is an intervention strategy.*

Our focus here is on intervention strategies in dyadic multi-turn interactions carried out in sequence. For interaction between agents, we use the shorthand a and b to represent sequences of think-act-observe cycles[2]. Using the shorthand notation $[b_j, a_j]_{j \leq n} = (b_n, a_n), (b_{n-1}, a_{n-1}), ..., (b_1, a_1)$, we explicitly construct the likelihood function for dyadic interactions between autoregressive language models (Estornell & Liu, 2024). When the instruction intervention (v_k) is introduced at the kth turn, then the conditional probability of the multi-turn system at the Tth turn ($T \geqslant k$) is

$$P_\theta^{[1:T]}|v_k = \underbrace{\prod_{i=k}^{T} P_\theta((b_i, a_i)|[b_j, a_j]_{k \leq j \leq i-1}, v_k, [b_j, a_j]_{j \leq k-1})}_{\text{after intervention}} \cdot \underbrace{\prod_{i=1}^{k-1} P_\theta((b_i, a_i)|[b_j, a_j]_{j \leq i-1})}_{\text{before intervention}}. \quad (2)$$

Best-of-N trajectory repair For an intervention strategy implemented according to Eq. 2, the intervention (v_k) affects all the follow-up interactions between the two agents. Given this formulation, the best-of-N intervention within the proposal set $K = \{1, 2, ..., N\}$ amounts to the following

$$v_{\text{BoN}} = \underset{k \in K}{\text{argmax}}\, \mathbb{1}_{S(y, y^*)} \left\{ y \sim P_{\theta_{\text{worker}}}^{[1:T_h]} | v_k \right\}, \quad (3)$$

where S is a binary performance metric or evaluator that compares the outcome of the agent actions with the desired outcome (y^*). While it is possible to use an LLM judge to compute S, here we consider the situation in which y^* is known for simplifying the pipeline. In Eq. 3, the interaction horizon ends at T_h. For equally effective instruction interventions within the N candidates, the one that responds to an earlier failure point is taken (Zheng et al., 2025).

[2]This notational system is generic and not affected by the specific implementation of the LLM agent.

Algorithm 1 Best-of-N intervened partial replay (BoNIPR).

Inputs: N trials, upper turn limit T_+, transcript $C_{[1:T]}$ with T turns of interaction.
Outputs: Best intervention for the task. None if unsuccessful after all attempts.

BoNIPR(N, $T_{\max}$, $C_{[1:T]}$)
  $\mathcal{E} \leftarrow$ TraceError(N, $C_{[1:T]}$) ▷ Obtain error set $\mathcal{E} = \{(T_k, \varepsilon_k)\}_{k=1}^{N}$ from transcript
  $\{v_k\}_{k=1}^{N} \sim P_{\theta_{\text{intervenor}}}|\{\varepsilon_k\}_{k=1}^{N}$ ▷ Generate interventions based on error set information
  for $k \leftarrow 1$ to N **do** ▷ Iterate over candidate interventions
    $T \leftarrow T_k$, $v \leftarrow v_k$
    while $T < T_+$ **do**
      $b'_T, a'_T \sim P^{[1:T]}_{\theta_{\text{user}}}|v, P^{[1:T]}_{\theta_{\text{worker}}}|v$ ▷ Continue subsequent interactions
      $T \leftarrow T + 1$
    if $S(y, y^*) \leftarrow 1$ **then** ▷ Check if task completion is successful
      return T_k, v_k

3.3 Inference scaling analysis

In the following analysis, T_h indicates the total turn of interactions as in Eq. 3. In multiagent debate with m participating agents ($m \geqslant 2$ in a debate), the number of model calls for a single invocation of debate is $\sim mT_h$, so N invocations lead to $N_{\text{call}} = O(mNT_h)$ (Kim et al., 2025). A prior study of multi-turn BoN sampling used for few-turn ($T_h \leqslant 5$) code generation (Jain et al., 2025) requires repeatedly rolling out entire trajectories (i.e., full reroll) but doesn't consider the inherent reliability of the system. For yet longer interactions in complex, multi-step tasks, these approaches lead to large inference-scaling overhead.

For the BoNIPR approach (Algorithm 1), the intervenor agent is enacted only once for every complete multi-turn interaction between the worker and user agents. The intervenor agent can use tools to investigate the failure subject to a limited number of calls. Therefore, this step requires a constant number (c) of model calls. Assuming that $T_h >$ MTTF, for each IPR procedure, up to N additional interactions are needed for the system to recover from failure, we therefore need a total of

$$N_{\text{call}} = c + \text{MTTF} \cdot m + N \cdot m \cdot (T_h - \text{MTTF}) = O(Nm(T_h - \text{MTTF})) \sim O(mT_h). \quad (4)$$

The last step has $T_h - \text{MTTF} = \alpha T_h$, with $\alpha \in (0, 1)$. Consider MTTF as a design constant, then N in BoN sampling can be chosen such that $N\alpha \sim$ const, which leads to $N_{\text{call}} = O(mT_h)$.

4 Evaluating intervened partial replay

4.1 Agentic system implementation

LLMs and agents The worker agent is implemented as ReAct-type tool-use agents (Yao et al., 2023), while the user agent is passive (i.e., only performs the information exchange). This design is generic and compatible with many use cases of LLM agents. The implementation uses the agents are mini-versions of OpenAI GPT-4 and GPT-5 model families (GPT-4o-mini, GPT-4.1-mini, and GPT-5-mini) as the base model. Implementation of the BoN intervention follows Algorithm 1. The transcript analysis tool (Meng et al., 2025) uses another designated model (GPT-4o-mini) which we chose because of its fast inference. During execution, the base model of the user agent is set to have a temperature of one to simulate user variability (Yao et al., 2025), while the temperature for the worker agent is set to zero. Model inference is executed through API calls.

Multi-turn datasets We evaluated our algorithm using the τ-bench (Yao et al., 2025) and CRAFT-MD (Johri et al., 2025), where the multiagent interactions are both constructed between the corresponding worker and user agents[3]. The τ-bench contains simulated realistic customer service interactions in airline (50 examples) and retail (115 examples) settings. The expected outcome is that

[3]For τ-bench, the user agent simulates a customer seeking commercial services. For CRAFT-MD, the user agent simulates a patient seeking medical diagnosis.

the worker agent correctly modifies the corresponding user information in a custom-built database. $S(y, y^*)$ in Eq. 3 here is a hash string comparison. From CRAFT-MD, we chose the subsets on infectious diseases (214 examples) and neurology (137 examples) for cost-effective demonstration. They are transcribed from the US Medical Licensing Examination into a multi-turn interaction format (Johri et al., 2025). The length of interactions between the worker and user agent is generally at 10 turns or more for both datasets. $S(y, y^*)$ for this dataset is executed by an LLM judge (Li et al., 2024a) constructed with the same model as the worker.

4.2 Trajectory Repair Procedure

The trajectory repair framework targets the worker agent for failure mitigation. It consists of an intervenor agent and external tools as a separate component (Fig. 2a) to the existing two-agent system (worker and user). The intervenor agent can access Transluce Docent tools (Meng et al., 2025) for analysis of the multi-turn conversation history (i.e., transcript) between the worker and user agents (see Fig. 3).

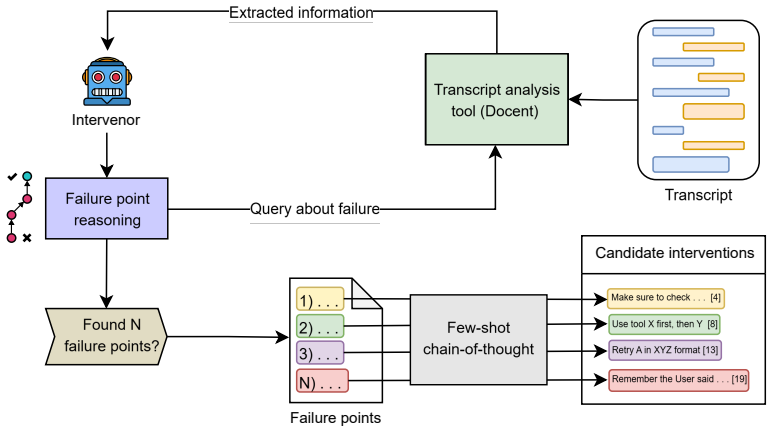


Figure 3: Schematic of trajectory analysis for identifying agent failure points and generating intervention proposals. The failure points are identified through interactions between the intervenor and the transcript analysis tools (Transluce Docent). The candidate interventions are constructed with the aim to fix the corresponding failure points.

(1) **Failure identification.** When reasoning about the failure points, the intervenor agent backtraces the error iteratively from the outcome of the trajectory with the help of the transcript analysis tools. This step produces the summary of failure (`failure-brief`), which contains a brief description of the failure, and the turn index of failure source (`failure-id`), which indicates the location of failure. This step functions as the `TraceError` function in Algorithm 1. The prompt patterns are explained in Appendix B.

(2) **Intervention generation.** For each identified failure point, the intervenor agent generates candidate instruction interventions in fixed formats such as "Do X instead of Y" or "Remember to do Z", etc. In our implementation, along with each failure summary and location in the previous step, the intervenor agent generates candidate intervention text (`intervention-text`) to correct the problem. In Algorithm 1, this amounts to obtaining $\{(T_k, v_k)\}_{k=1}^{N}$ through the intervenor $P_{\theta_{\text{intervenor}}}$ that iteratively uses the tool. The procedure terminates when a sufficient number of failure points are identified according to specification in the BoN sampling.

(3) **Repair induction.** Inserting the instruction intervention into the suggested location in the transcript initiates the repair. The intervention text is communicated as an extra system prompt to the worker agent to enforce its compliant action. The remainder of the user-worker agent interaction is resumed as indicated in Algorithm 1 and illustrated in Fig. 2b.

For the illustration of steps (1)-(2), an example of the failure point reasoning trajectory that produces the triplets (`failure-brief`, `failure-id`,`intervention-text`) is provided in Appendix C.1.

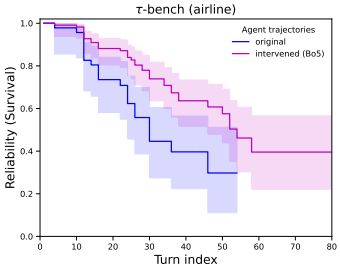
4.3 QUANTITATIVE METRICS

Reliability estimation We reconstruct the reliability function using Kaplan-Meier estimation (Akritas, 2004), a nonparametric approach computed with the cumulative product of the remaining fraction of samples (i.e., agents with unfinished task execution). The observation period starts from the first turn of interaction for all samples and the signal of the occurrence of failure is emitted at the end of the multi-turn interaction. We treat the agent trajectory data from turn-indexed LT as generally right-censored (Lagakos, 1979; Lawless, 2002) to account for the different lengths of multi-turn interactions among the samples. The estimated reliability at the Tth turn is $R(T) = \prod_{i \leq T} \left(1 - \frac{d_i}{n_i}\right)$, where d_i is the number of failed trajectories, and n_i the total number of unterminated trajectories, both of which are assessed at the ith turn. The agents that failed before the ith turn are taken out of the counts for n_i.

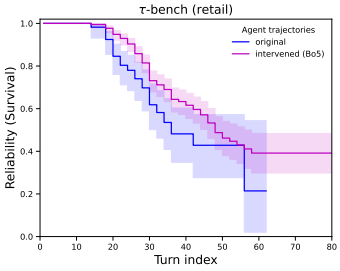
Performance metrics We assess the changes of an agent trajectory subject to instruction intervention along different dimensions. In our work, this is reflected directly in the relative performance gain (γ) from introducing the intervenor into the system, calculated using the success rate (SR) of agent trajectory before (SR_{orig}) and after (SR_{iv}) the introduction of the intervention, $\gamma = (SR_{\text{iv}} - SR_{\text{orig}})/SR_{\text{orig}}$. Another important metric is MTTF, which is calculated using the estimated reliability according to Eq. 1. In addition, the pass@N SR metric is also used for quantifying performance scaling.

Table 1: Success rate with different multiagent configurations involving single (sing., brown block) and the best-of-N intervention (Bo5 for $N = 5$, blue block) on τ-bench. Baselines are without an intervenor (None). The success rate (%) values are followed by their relative change (γ) in brackets in comparison with the baseline (no intervenor/interventions).

Worker & User	Intervenor	τ-bench airline	τ-bench retail
(Single trials) GPT-4o-mini	None	20.0	17.4
	None (Reflexion)	30.0 (50.0%↑)	31.3 (79.9%↑)
	GPT-4o-mini$^{(\text{sing.})}$	30.0 (50.0%↑)	29.6 (70.1%↑)
	GPT-5-mini$^{(\text{sing.})}$	**32.0** (60.0%↑)	**32.2** (85.1%↑)
(Best-of-5) GPT-4o-mini	None (Full RR)	38.0 (90.0%↑)	46.1 (165%↑)
	None (Partial RR)	44.0 (120%↑)	49.1 (182%↑)
	GPT-4o-mini$^{(\text{Bo5})}$	50.0 (150%↑)	46.1 (165%↑)
	GPT-5-mini$^{(\text{Bo5})}$	**52.0** (160%↑)	**50.8** (192%↑)
(Single trials) GPT-4.1-mini	None	42.0	35.7
	None (Reflexion)	42.0 (↔)	43.5 (21.8%↑)
	GPT-5-mini$^{(\text{sing.})}$	**56.0** (33.3%↑)	**50.4** (41.2%↑)
(Best-of-5) GPT-4.1-mini	None (Full RR)	62.0 (47.6%↑)	68.7 (92.4%↑)
	None (Partial RR)	58.0 (38.1%↑)	63.5 (77.9%↑)
	GPT-5-mini$^{(\text{Bo5})}$	**70.0** (66.7%↑)	**71.3** (99.7%↑)
(Single trials) GPT-5-mini	None	31.0	50.4
	None (Reflexion)	52.0 (67.7%↑)	**73.0** (44.8%↑)
	GPT-4o-mini$^{(\text{sing.})}$	**54.0** (74.2%↑)	64.3 (27.6%↑)
	GPT-5-mini$^{(\text{sing.})}$	44.0 (41.9%↑)	66.1 (31.1%↑)
(Best-of-5) GPT-5-mini	None (Full RR)	**72.0** (132%↑)	80.0 (58.7%↑)
	None (Partial RR)	64.0 (106%↑)	76.5 (51.8%↑)
	GPT-4o-mini$^{(\text{Bo5})}$	60.0 (93.5%↑)	**81.7** (62.1%↑)
	GPT-5-mini$^{(\text{Bo5})}$	54.0 (74.2%↑)	**81.7** (62.1%↑)



(a)



(b)

Figure 4: Estimation of turn-indexed reliability in multi-turn agent interactions. Results for the τ-bench dataset are shown separately for (a) airline and (b) retail tasks. The intervened trajectories come from best-of-5 (Bo5) sampling, each with a different intervention. In both (a) and (b), the worker and user agents are both constructed with GPT-4.1-mini, and the intervenor with GPT-5-mini. The confidence band is calculated with the Greenwood formula (Borgan & Liestøl, 1990).

Table 2: Success rate with different multiagent configurations involving the best-of-N intervention (Bo3 for $N = 3$) on CRAFT-MD dataset. Baselines are reported when the intervenor is not present (None). The success rate (%) values are in black followed by their relative change (γ) in brackets in comparison with the baseline (no intervenor/interventions).

Worker & User	Intervenor	CRAFT-MD infectious diseases	CRAFT-MD neurology
GPT-4.1-mini	None	34.3	38.8
	GPT-5-mini$^{(\text{Bo3})}$	54.0 (57.4%↑)	50.0 (28.9%↑)
GPT-5-mini	None	40.9	50.5
	GPT-5-mini$^{(\text{Bo3})}$	56.2 (37.5%↑)	62.1 (23.1%↑)
GPT-5.4-mini	None	21.9	36.0
	GPT-5-mini$^{(\text{Bo3})}$	38.0 (73.3%↑)	49.0 (35.1%↑)

5 Results and Analysis

Quantifying agent failure mitigation On both τ-bench (Table 1) and CRAFT-MD (Table 2) datasets, inclusion of the behavior repair framework leads to considerable performance improvement across distinct categories of specialty tasks. To accommodate the larger number of samples in the biomedical tasks, we reduced the cost by using $N = 3$ in BoNIPR (Algorithm 1). The effectiveness of failure mitigation from including an intervention in the multi-turn interaction is assessed by the performance gain. We found that

(1) Having a custom instruction intervention can boost the performance at both the single trial and BoN level. At the single-trial level, we compared the IPR approach with self-reflection (Reflexion), in which the corrective behavior is self-initiated. At the BoN level ($N = 5$ in Table 1), we compared the BoNIPR approach with full and partial rerolls (RRs) of the agent trajectory using the same number of trials. Partial RRs use the same failure points identified by the intervenor but without the instruction intervention. The effect cannot be fully replicated by rerolling partial trajectories (which also requires first identifying the potential failure points).

(2) Selection of the LLM for constructing the intervenor agent can create a marginal benefit in performance. This indicates that changing the base model of the intervenor agent is a separate route for performance tuning without modifying the worker/user agents.

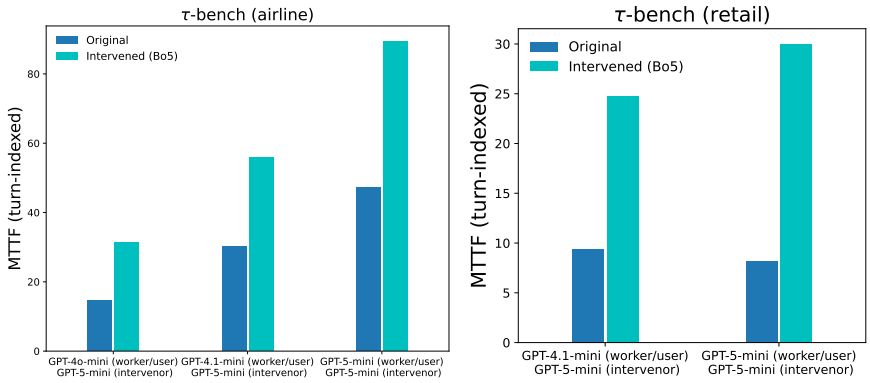


Figure 5: Reconstructed MTTFs for agentic systems constructed from different base LLMs.

Although our demonstrations used only the cost-efficient mini-version of OpenAI models (the GPT-4 or GPT-5 family), the performance metrics on τ-bench already rival or even surpass those executed with the full models[4], resulting in a cost reduction of about tenfold. Moreover, the reconstructed

[4]Official leaderboard at https://taubench.com/#leaderboard

MTTF for the dyadic interaction in Fig. 6 further illustrates this effect. The effective interventions show meaningful information (Appendix A). A full example of a successful intervention is provided in Appendix C.2.

Intervention strategy and scaling We compared the effectiveness of different intervention strategies by ranking the interventions before replaying the remaining transcript. Specifically, we compared the outcome of BoN samples based on the position of intervention being early (`early-first`) or late (`late-first`) within the transcript since error conditioning can undermine outcome at subsequent interactions (Cheng et al., 2026; Zheng et al., 2025). The natural order is the order of failure points iteratively identified from the failure point reasoning procedure (`TraceError` step in Algorithm 1). The results in Fig. 6 show that the scaling curve has a gradual plateauing behavior and the largest performance gain occurs in the first three identified failure points and constructed interventions, while the next few attempts show diminishing gain.

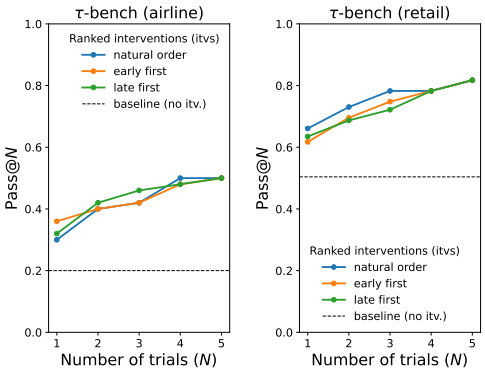


Figure 6: Scaling behavior of pass@N success rate across different position ranking (natural order, early-first, late-first) of the BoN interventions. Both examples use GPT-5-mini as the intervenor.

6 Discussions and Conclusion

Reliability is a core component in harnessing the value of AI systems (Pan et al., 2025). Existing safety-focused LLM agent designs have considered specialized agents to oversee emerging violations within the course of multiagent interactions (Xiang et al., 2025; Wang et al., 2025), but the actual mechanism behind their functionality and the benefit of specific role attribution are not usually investigated in depth or in isolation. The present quantitative study shows that the influenceability of AI agents can be harnessed for designing compact solutions for reliability and performance enhancement. Our work provides a proof-of-principle study for placing horizon limits in agent use for risk reduction (Xian et al., 2025a; Korbak et al., 2025). Besides improving the success rate, our approach also provides a framework for quantifying agentic system reliability and an empirical foundation for designing on-the-fly adaption (Jiang et al., 2025).

Our work can inform future research along the following directions: (1) **Intervention design and planning** that can lead to effective scheduling of multiple interventions across the interaction horizons to guide agent actions without necessarily accessing the complete feedback signal (Chakraborty et al., 2025) at task completion. This can help mitigate failures that come from a gradual accumulation of errors that lead to performance degradation (Ascher & Feingold, 1984). (2) **Recursive improvement**, of which recursive failure mitigation is an integral part. Our approach can be used to generate data for continual test-time training of AI agents (Jiang et al., 2025) at scale to improve their cooperative behavior and bypass the costly requirements on stepwise verifiable rewards to handle more open-ended tasks in many real-world agent applications (Gunjal et al., 2026).

7 Limitations

Our demonstrations show that these external influences can be used for performance enhancements, but they rest on the assumptions that agent failures are discrete and independent, which are not always the case. Our approach is not immediately applicable to cases in which agent tools are maliciously poisoned with adversarial docstring or when the agent is negatively influenced by false information on the open web. Moreover, multi-turn interactions can take on different forms. If the connections between the actions executed in turns are not sufficiently high (i.e., the agent doing task 1, task 2, ... as unrelated tasks), then the effectiveness of our approach can also degrade.

AI USE STATEMENT

In this work, we used generative AI tools for formatting latex layout, fixing latex errors, search for relevant background references. It is also used partly for code completion and debugging.

REFERENCES

Michael G. Akritas. Nonparametric Survival Analysis. *Statistical Science*, 19(4):615–623, 2004. ISSN 0883-4237. URL https://www.jstor.org/stable/4144432.

Saul Albert and J. P. de Ruiter. Repair: The Interface Between Interaction and Cognition. *Topics in Cognitive Science*, 10(2):279–313, 2018. ISSN 1756-8765. doi: 10.1111/tops.12339. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/tops.12339. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/tops.12339.

Harold Ascher and Harry Feingold. *Repairable systems reliability: modeling, inference, misconceptions and their causes*. Marcel Dekker, 1984.

Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. Small Language Models are the Future of Agentic AI, June 2025. URL http://arxiv.org/abs/2506.02153. arXiv:2506.02153 [cs].

Ørnulf Borgan and Knut Liestøl. A Note on Confidence Intervals and Bands for the Survival Function Based on Transformations. *Scandinavian Journal of Statistics*, 17(1):35–41, 1990. ISSN 0303-6898. URL https://www.jstor.org/stable/4616153.

Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. RepairAgent: An Autonomous, LLM-Based Agent for Program Repair. In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering*, ICSE '25, pp. 2188–2200, Ottawa, Ontario, Canada, September 2025. IEEE Press. ISBN 979-8-3315-0569-1. doi: 10.1109/ICSE55347.2025.00157. URL https://dl.acm.org/doi/10.1109/ICSE55347.2025.00157.

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? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, October 2025. URL https://openreview.net/forum?id=fAjbYBmonr.

Manuela Chacon-Chamorro, Luis Felipe Giraldo, Nicanor Quijano, Vicente Vargas-Panesso, César González, Juan Sebastián Pinzón, Rubén Manrique, Manuel Ríos, Yesid Fonseca, Daniel Gómez-Barrera, and Mónica Perdomo-Pérez. Cooperative Resilience in Artificial Intelligence Multiagent Systems. *IEEE Transactions on Artificial Intelligence*, pp. 1–11, 2025. ISSN 2691-4581. doi: 10.1109/TAI.2025.3567371. URL https://ieeexplore.ieee.org/document/10988739.

Souradip Chakraborty, Mohammadreza Pourreza, Ruoxi Sun, Yiwen Song, Nino Scherrer, Furong Huang, Amrit Singh Bedi, Ahmad Beirami, Jindong Gu, Hamid Palangi, and Tomas Pfister. On the Role of Feedback in Test-Time Scaling of Agentic AI Workflows, July 2025. URL http://arxiv.org/abs/2504.01931. arXiv:2504.01931 [cs].

Jingdi Chen, Hanqing Yang, Zongjun Liu, and Carlee Joe-Wong. The Five Ws of Multi-Agent Communication: Who Talks to Whom, When, What, and Why - A Survey from MARL to Emergent Language and LLMs. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=LGsed0QQVq.

Lingjiao Chen, Matei Zaharia, and James Zou. How Is ChatGPT's Behavior Changing Over Time? *Harvard Data Science Review*, 6(2), March 2024. doi: 10.1162/99608f92.5317da47. URL https://hdsr.mitpress.mit.edu/pub/y95zitmz.

Yun Cheng, Xingyu Zhu, Haoyu Zhao, and Sanjeev Arora. Contextual Drag: How Errors in the Context Affect LLM Reasoning, March 2026. URL http://arxiv.org/abs/2602.04288. arXiv:2602.04288 [cs].

Kaustubh Deshpande, Ved Sirdeshmukh, Johannes Baptist Mols, Lifeng Jin, Ed-Yeremai Hernandez-Cardona, Dean Lee, Jeremy Kritz, Willow E. Primack, Summer Yue, and Chen Xing. MultiChallenge: A Realistic Multi-Turn Conversation Evaluation Benchmark Challenging to Frontier LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18632–18702, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.958. URL https://aclanthology.org/2025.findings-acl.958/.

Aniket Didolkar, Anirudh Goyal, Nan R. Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Sanjeev Arora. Metacognitive Capabilities of LLMs: An Exploration in Mathematical Problem Solving. *Advances in Neural Information Processing Systems*, 37:19783–19812, December 2024. doi: 10.52202/079017-0623. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2318d75a06437eaa257737a5cf3ab83c-Abstract-Conference.html.

Heng Dong, Kefei Duan, and Chongjie Zhang. Enhancing Decision-Making of Large Language Models via Actor-Critic. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 13984–14020. PMLR, October 2025. URL https://proceedings.mlr.press/v267/dong25c.html.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving Factuality and Reasoning in Language Models through Multiagent Debate. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 11733–11763. PMLR, July 2024. URL https://proceedings.mlr.press/v235/du24e.html. ISSN: 2640-3498.

Benjamin Epstein. Truncated Life Tests in the Exponential Case. *The Annals of Mathematical Statistics*, 25(3):555–564, September 1954. ISSN 0003-4851, 2168-8990. doi: 10.1214/aoms/1177728723. URL https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-25/issue-3/Truncated-Life-Tests-in-the-Exponential-Case/10.1214/aoms/1177728723.full.

Andrew Estornell and Yang Liu. Multi-LLM Debate: Framework, Principals, and Interventions. *Advances in Neural Information Processing Systems*, 37:28938–28964, December 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/32e07a110c6c6acf1afbf2bf82b614ad-Abstract-Conference.html.

Andrew Estornell, Jean-Francois Ton, Yuanshun Yao, and Yang Liu. ACC-Collab: An Actor-Critic Approach to Multi-Agent LLM Collaboration. In *The Thirteenth International Conference on Learning Representations*, October 2025. URL https://openreview.net/forum?id=nfKfAzkiez.

Shangbin Feng, Wenxuan Ding, Alisa Liu, Zifeng Wang, Weijia Shi, Yike Wang, Zejiang Shen, Xiaochuang Han, Hunter Lang, Chen-Yu Lee, Tomas Pfister, Yejin Choi, and Yulia Tsvetkov. When One LLM Drools, Multi-LLM Collaboration Rules, February 2025. URL http://arxiv.org/abs/2502.04506. arXiv:2502.04506 [cs].

Dyke Ferber, Lars Hilgers, Christiane Höper, Benedict Kinny-Köster, Jan-Niklas Eckardt, Katharina Egger-Heidrich, Marius Bill, Martin M. K. Schneider, Jan Clusmann, Lejla Kadric, Marcel Oehme, Maximilian Mayrhofer-Schmid, Alexander Oeser, Georg Wölflein, Isabella C. Wiest, Jan Moritz Middeke, A. John Iafrate, Daniel Truhn, Dirk Jäger, and Jakob Nikolas Kather. Towards autonomous medical artificial intelligence agents. *Nature*, 655(8125):1282–1291, July 2026. ISSN 1476-4687. doi: 10.1038/s41586-026-10675-5. URL https://www.nature.com/articles/s41586-026-10675-5.

John H. Flavell. Metacognitive Aspects of Problem Solving. In *The Nature of Intelligence*. Routledge, 1976.

Wendell L. French and Cecil Bell. *Organization Development: Behavioral Science Interventions for Organization Improvement*. Prentice Hall, 6th edition, 1999. ISBN 978-0-13-242231-4.

Victoria Graf, Valentina Pyatkin, Nouha Dziri, Nathan Lambert, and Hannaneh Hajishirzi. TurnWise: The Gap between Single- and Multi-turn Language Model Capabilities, March 2026. URL http://arxiv.org/abs/2603.16759. arXiv:2603.16759 [cs].

Lei Gu, Yinghao Zhu, Haoran Sang, Zixiang Wang, Dehao Sui, Wen Tang, Ewen Harrison, Junyi Gao, Lequan Yu, and Liantao Ma. MedAgentAudit: Diagnosing and Quantifying Collaborative Failure Modes in Medical Multi-Agent Systems, October 2025. URL http://arxiv.org/abs/2510.10185. arXiv:2510.10185 [cs].

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean M. Hendryx. Rubrics as Rewards: Reinforcement Learning Beyond Verifiable Domains. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=c1bTcrDmt4.

Arnav Kumar Jain, Gonzalo Gonzalez-Pumariega, Wayne Chen, Alexander M. Rush, Wenting Zhao, and Sanjiban Choudhury. Multi-Turn Code Generation Through Single-Step Rewards. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 26700–26716. PMLR, October 2025. URL https://proceedings.mlr.press/v267/jain25a.html.

Li Ji-An, Hua-Dong Xiong, Robert Wilson, Marcelo G. Mattar, and Marcus K. Benna. Language Models Are Capable of Metacognitive Monitoring and Control of Their Internal Activations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, October 2025. URL https://openreview.net/forum?id=qTXlFwlggv.

Pengcheng Jiang, Jiacheng Lin, Zhiyi Shi, Zifeng Wang, Luxi He, Yichen Wu, Ming Zhong, Peiyang Song, Qizheng Zhang, Heng Wang, Xueqiang Xu, Hanwen Xu, Pengrui Han, Dylan Zhang, Jiashuo Sun, Chaoqi Yang, Kun Qian, Tian Wang, Changran Hu, Manling Li, Quanzheng Li, Hao Peng, Sheng Wang, Jingbo Shang, Chao Zhang, Jiaxuan You, Liyuan Liu, Pan Lu, Yu Zhang, Heng Ji, Yejin Choi, Dawn Song, Jimeng Sun, and Jiawei Han. Adaptation of Agentic AI, December 2025. URL http://arxiv.org/abs/2512.16301. arXiv:2512.16301 [cs].

Samuel G. B. Johnson, Amir-Hossein Karimi, Yoshua Bengio, Nick Chater, Tobias Gerstenberg, Kate Larson, Sydney Levine, Melanie Mitchell, Iyad Rahwan, Bernhard Schölkopf, and Igor Grossmann. Imagining and building wise machines: The centrality of AI metacognition, November 2024. URL http://arxiv.org/abs/2411.02478. arXiv:2411.02478 [cs].

Shreya Johri, Jaehwan Jeong, Benjamin A. Tran, Daniel I. Schlessinger, Shannon Wongvibulsin, Leandra A. Barnes, Hong-Yu Zhou, Zhuo Ran Cai, Eliezer M. Van Allen, David Kim, Roxana Daneshjou, and Pranav Rajpurkar. An evaluation framework for clinical use of large language models in patient interaction tasks. *Nature Medicine*, 31(1):77–86, January 2025. ISSN 1546-170X. doi: 10.1038/s41591-024-03328-5. URL https://www.nature.com/articles/s41591-024-03328-5.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When Can LLMs Actually Correct Their Own Mistakes? A Critical Survey of Self-Correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, November 2024. ISSN 2307-387X. doi: 10.1162/tacl_a_00713. URL https://doi.org/10.1162/tacl_a_00713.

Yubin Kim, Ken Gu, Chanwoo Park, Chunjong Park, Samuel Schmidgall, A. Ali Heydari, Yao Yan, Zhihan Zhang, Yuchen Zhuang, Mark Malhotra, Paul Pu Liang, Hae Won Park, Yuzhe Yang, Xuhai Xu, Yilun Du, Shwetak Patel, Tim Althoff, Daniel McDuff, and Xin Liu. Towards a Science of Scaling Agent Systems, December 2025. URL http://arxiv.org/abs/2512.08296. arXiv:2512.08296 [cs].

Frederic Kirstein, Terry Ruas, and Bela Gipp. What's Wrong? Refining Meeting Summaries with LLM Feedback. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 2100–2120, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.143/.

Tomek Korbak, Mikita Balesni, Buck Shlegeris, and Geoffrey Irving. How to evaluate control measures for LLM agents? A trajectory from today to superintelligence, April 2025. URL http://arxiv.org/abs/2504.05259. arXiv:2504.05259 [cs].

Wai-Chung Kwan, Xingshan Zeng, Yuxin Jiang, Yufei Wang, Liangyou Li, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. MT-Eval: A Multi-Turn Capabilities Evaluation Benchmark for Large Language Models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20153–20177, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1124. URL https://aclanthology.org/2024.emnlp-main.1124/.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs Get Lost In Multi-Turn Conversation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=VKGTGGcwl6.

S. W. Lagakos. General Right Censoring and Its Impact on the Analysis of Survival Data. *Biometrics*, 35(1):139–156, 1979. ISSN 0006-341X. doi: 10.2307/2529941. URL https://www.jstor.org/stable/2529941.

Jerald F. Lawless. *Statistical Models and Methods for Lifetime Data*. John Wiley & Sons, 2002. ISBN 978-1-118-03125-4.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods, December 2024a. URL http://arxiv.org/abs/2412.05579. arXiv:2412.05579 [cs].

Yunxuan Li, Yibing Du, Jiageng Zhang, Le Hou, Peter Grabowski, Yeqing Li, and Eugene Ie. Improving Multi-Agent Debate with Sparse Communication Topology. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7281–7294, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.427. URL https://aclanthology.org/2024.findings-emnlp.427/.

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, 2025. URL http://arxiv.org/abs/2510.07593. arXiv:2510.07593 [cs].

Tennison Liu and Mihaela Van Der Schaar. Position: Truly Self-Improving Agents Require Intrinsic Metacognitive Learning. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 81714–81727. PMLR, October 2025. URL https://proceedings.mlr.press/v267/liu25cw.html. ISSN: 2640-3498.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating Very Long-Term Conversational Memory of LLM Agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Kevin Meng, Vincent Huang, Jacob Steinhardt, and Sarah Schwettmann. Introducing Docent, March 2025. URL https://transluce.org/introducing-docent.

Martin Monperrus. Automatic Software Repair: A Bibliography. *ACM Comput. Surv.*, 51(1):17:1–17:24, January 2018. ISSN 0360-0300. doi: 10.1145/3105906. URL https://dl.acm.org/doi/10.1145/3105906.

Laurent Orseau and Stuart Armstrong. Safely interruptible agents. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, pp. 557–566, Arlington, Virginia, USA, June 2016. AUAI Press. ISBN 978-0-9966431-1-5.

Melissa Z. Pan, Negar Arabzadeh, Riccardo Cogo, Yuxuan Zhu, Alexander Xiong, Lakshya A. Agrawal, Huanzhi Mao, Emma Shen, Sid Pallerla, Liana Patel, Shu Liu, Tianneng Shi, Xiaoyuan Liu, Jared Quincy Davis, Emmanuele Lacavalla, Alessandro Basile, Shuyi Yang, Paul Castro, Daniel Kang, Joseph E. Gonzalez, Koushik Sen, Dawn Song, Ion Stoica, Matei Zaharia, and Marquita Ellis. Measuring Agents in Production, December 2025. URL http://arxiv.org/abs/2512.04123. arXiv:2512.04123 [cs].

Stephan Rabanser, Sayash Kapoor, Peter Kirgis, Kangheng Liu, Saiteja Utpala, and Arvind Narayanan. Towards a Science of AI Agent Reliability, February 2026. URL http://arxiv.org/abs/2602.16666. arXiv:2602.16666 [cs] version: 2.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. A Simplest Systematics for the Organization of Turn-Taking for Conversation. *Language*, 50(4):696–735, 1974. ISSN 0097-8507. doi: 10.2307/412243. URL https://www.jstor.org/stable/412243.

Kevin Song, Anand Jayarajan, Yaoyao Ding, Qidong Su, Zhanda Zhu, Sihang Liu, and Gennady Pekhimenko. Aegis: Taxonomy and Optimizations for Overcoming Agent-Environment Failures in LLM Agents, August 2025. URL http://arxiv.org/abs/2508.19504. arXiv:2508.19504 [cs].

Mark Steyvers and Megan A. K. Peters. Metacognition and Uncertainty Communication in Humans and Large Language Models. *Current Directions in Psychological Science*, pp. 09637214251391158, November 2025. ISSN 0963-7214. doi: 10.1177/09637214251391158. URL https://doi.org/10.1177/09637214251391158. Publisher: SAGE Publications Inc.

Yuzhi Tang, Tianxiao Li, Elizabeth Li, Chris J. Maddison, Honghua Dong, and Yangjun Ruan. LM Agents May Fail to Act on Their Own Risk Knowledge. In *Second Conference on Language Modeling*, August 2025. URL https://openreview.net/forum?id=OeYdS51k8F.

W. A. Thompson. On the Foundations of Reliability. *Technometrics*, 23(1):1–13, 1981. ISSN 0040-1706. doi: 10.2307/1267969. URL https://www.jstor.org/stable/1267969. Publisher: American Statistical Association, American Society for Quality.

Tao Tu, Mike Schaekermann, Anil Palepu, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Yong Cheng, Elahe Vedadi, Nenad Tomasev, Shekoofeh Azizi, Karan Singhal, Le Hou, Albert Webson, Kavita Kulkarni, S. Sara Mahdavi, Christopher Semturs, Juraj Gottweis, Joelle Barral, Katherine Chou, Greg S. Corrado, Yossi Matias, Alan Karthikesalingam, and Vivek Natarajan. Towards conversational diagnostic artificial intelligence. *Nature*, 642 (8067):442–450, June 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-08866-7. URL https://www.nature.com/articles/s41586-025-08866-7.

Charles L. Wang, Trisha Singhal, Ameya Kelkar, and Jason Tuo. MI9: An Integrated Runtime Governance Framework for Agentic AI, November 2025. URL http://arxiv.org/abs/2508.03858. arXiv:2508.03858 [cs].

Sue Wilkinson and Ann Weatherall. Insertion Repair. *Research on Language and Social Interaction*, 44(1):65–91, February 2011. ISSN 0835-1813, 1532-7973. doi: 10.1080/08351813.2011.544136. URL https://www.tandfonline.com/doi/full/10.1080/08351813.2011.544136.

R. Patrick Xian, Noah R. Baker, Tom David, Qiming Cui, A. Jay Holmgren, Stefan Bauer, Madhumita Sushil, and Reza Abbasi-Asl. Robustness tests for biomedical foundation models should tailor to specifications. *npj Digital Medicine*, 8(1):557, August 2025a. ISSN 2398-6352. doi: 10.1038/s41746-025-01926-2. URL https://www.nature.com/articles/s41746-025-01926-2.

R. Patrick Xian, Garry A. Gabison, Ahmed Alaa, Christoph Riedl, and Grigorios G. Chrysos. Reliable agent engineering should integrate machine-compatible organizational principles, December 2025b. URL http://arxiv.org/abs/2512.07665. arXiv:2512.07665 [cs].

Zhen Xiang, Linzhi Zheng, Yanjie Li, Junyuan Hong, Qinbin Li, Han Xie, Jiawei Zhang, Zidi Xiong, Chulin Xie, Carl Yang, Dawn Song, and Bo Li. GuardAgent: Safeguard LLM Agents

via Knowledge-Enabled Reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 68316–68342. PMLR, October 2025. URL https://proceedings.mlr.press/v267/xiang25a.html. ISSN: 2640-3498.

Zehao Xiao and Cees G. M. Snoek. Beyond Model Adaptation at Test Time: A Survey, November 2024. URL http://arxiv.org/abs/2411.03687. arXiv:2411.03687 [cs].

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. Pride and Prejudice: LLM Amplifies Self-Bias in Self-Refinement. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15474–15492, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.826. URL https://aclanthology.org/2024.acl-long.826/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Christine Ye, Sihan Yuan, Suchetha Cooray, Steven Dillmann, Ian L. V. Roque, Dalya Baron, Philipp Frank, Sergio Martin-Alvarez, Nolan Koblischke, Frank J. Qu, Diyi Yang, Risa Wechsler, and Ioana Ciuca. ReplicationBench: Can AI Agents Replicate Astrophysics Research Papers?, November 2025. URL http://arxiv.org/abs/2510.24591. arXiv:2510.24591 [cs].

Jehyeok Yeon, Isha Chaudhary, and Gagandeep Singh. Quantifying Distributional Robustness of Agentic Tool-Selection, October 2025. URL http://arxiv.org/abs/2510.03992. arXiv:2510.03992 [cs].

Wojciech Zaremba, Evgenia Nitishinskaya, Boaz Barak, Stephanie Lin, Sam Toyer, Yaodong Yu, Rachel Dias, Eric Wallace, Kai Xiao, Johannes Heidecke, and Amelia Glaese. Trading Inference-Time Compute for Adversarial Robustness, January 2025. URL http://arxiv.org/abs/2501.18841. arXiv:2501.18841 [cs].

Chen Zhang, Xinyi Dai, Yaxiong Wu, Qu Yang, Yasheng Wang, Ruiming Tang, and Yong Liu. A Survey on Multi-Turn Interaction Capabilities of Large Language Models, January 2025a. URL http://arxiv.org/abs/2501.09959. arXiv:2501.09959 [cs].

Guibin Zhang, Yanwei Yue, Zhixun Li, Sukwon Yun, Guancheng Wan, Kun Wang, Dawei Cheng, Jeffrey Xu Yu, and Tianlong Chen. Cut the Crap: An Economical Communication Pipeline for LLM-based Multi-Agent Systems. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=LkzuPorQ5L.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying Process Errors in Mathematical Reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.50. URL https://aclanthology.org/2025.acl-long.50/.