WHERE DID WE GO WRONG? LOCALIZING ERRORS TO REPAIR MULTI-TURN CONVERSATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model (LLM) assistants are increasingly used in multi-turn conversations, where they must progressively infer a user's intent from partial information revealed over time. This makes them vulnerable to early misinterpretations that can rule out the true intent and persist across turns, ultimately causing the interaction to fail. Recovering effectively from such failures requires identifying which earlier interpretations need revision while preserving the information that remains valid. We introduce LOCI (Localization Of Corrupted Inferences), a framework for probabilistically localizing such errors and selectively repairing the conversational state. Given an observed failure, LOCI reasons over possible user intents and assigns each prior update a posterior probability of having excluded the true intent. These probabilities rank likely error locations and quantify confidence in the localization. We then use this confidence in a repair policy that decides whether to retain, verify, or remove suspected updates, enabling selective repair of the conversational state. To evaluate LOCI's ability to identify corrupted inferences, we first test it in a controlled setup with injected errors. We find that LOCI improves localization over the baselines, with its advantage becoming larger when multiple errors occur. We then evaluate it in free-form conversations, where errors arise naturally, and measure the additional interactions needed to complete the task after an initial failure. Using localization confidence to guide repair, LOCI reduces mean recovery cost by 46–79% relative to the strongest baseline across multiple domains.

1 INTRODUCTION

Many real-world applications require large language model (LLM) assistants to engage in a conversation: users rarely specify all relevant constraints, and the assistant must elicit and refine their intent progressively. Yet, while modern LLMs perform strongly on single-turn tasks, their performance has been observed to deteriorate in multi-turn interactions (Deshpande et al., 2025; Laban et al., 2026). A key failure mode is that an early misinterpretation becomes embedded in the conversation and continues to shape later responses, even as new information becomes available. Such errors can persist across turns and prevent successful task completion. This failure mode also appears in sequential decision problems where information is collected over time to infer an unknown state. In medical diagnosis, for example, a misinterpretation of a patient response or test result may affect later decisions; localizing that error can allow correction without discarding remaining information.

Prior efforts have focused on making LLMs more proactive in preventing misunderstandings, e.g., by planning when and how to gather information (Hu et al., 2024; Dong et al., 2026), or by teaching models to ask clarifying questions and collaborate with users (Chen et al., 2025; Wu et al., 2025). Such approaches can reduce errors, but do not explicitly address the problem of recovery once an error becomes apparent–often several turns after the original mistake, when its effects have already compounded. Some work on verification of intermediate reasoning steps (Zheng et al., 2025; Kim et al., 2026a) and self-correction (Huang et al., 2024) addresses related forms of error recovery, and recent approaches target multi-turn interactions by refactoring conversation history (Shen et al., 2026) or generating recovery plans (Kim et al., 2026b). Yet, with erroneous updates buried among otherwise useful context, how best to localize the failure and selectively repair the conversation remains open.

We introduce LOCI (Localization Of Corrupted Inferences), a framework for identifying erroneous interpretations in a multi-turn conversation and selectively repairing the conversational state. The key

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.

idea is that each candidate user intent implies a possible explanation of which earlier updates were incorrect: if it were the true intent, every update that excludes it would be wrong. LOCI weighs these explanations using the recorded updates and an observed failure (e.g., the user rejects a proposal) to assign each update a probability of having excluded the true intent. A candidate that violates several updates contributes to each of their corruption probabilities, allowing localization of multiple errors.

These probabilities serve as corruption scores: their ranking localizes likely sources of errors, while their magnitudes quantify the model's confidence in such attribution. Based on the insight that retaining invalid updates can prevent recovery, while deleting valid updates can require additional questions, LOCI uses the scores to guide an adaptive repair policy. The policy selectively repairs the context by deciding whether to discard or verify an update, or retain the current state and resume interaction. The policy repeats this decision to recover from the failure, balancing verification costs against the risks of retaining errors or discarding useful information.

To isolate localization from the stochasticity of LLMs, we first evaluate LOCI in a controlled environment where the user's target, the meaning of each update, and the injected errors are known exactly. LOCI reliably localizes corrupted updates across both single- and multi-error settings, with its largest gains in interactions containing multiple errors. At the highest tested error rate, LOCI ranks an invalid update first in 74.4% of failed interactions, compared with 62.8% for the strongest baseline. We then evaluate recovery in free-form LLM conversations, where errors arise naturally, and use confidence to guide repair. After an initial failure, LOCI reaches the correct guess in 46–79% fewer additional turns on average than the strongest evaluated baseline across four domains.

In summary, our contributions are as follows. We provide:

- **A formal model for retrospective error localization.** We represent a conversation as a sequence of potentially incorrect updates to the assistant's state and formulate the problem of identifying which updates excluded the user's true intent after an observed failure.
- **A posterior corruption score for localizing multiple errors.** We derive LOCI, which assigns each update a probability of having excluded the true intent under the model; these probabilities rank suspected errors and quantify localization confidence.
- **An adaptive policy for selective repair.** We use localization confidence to decide whether to retain the current state, remove a suspected update, or verify it, balancing the cost of checking an update against the risk of retaining errors or discarding useful information.
- **Empirical evidence of improved localization and recovery.** We evaluate LOCI in controlled interactions, free-form LLM conversations, and task-oriented dialogue, showing improved error localization and fewer recovery interactions than the evaluated baselines.

2 PROBLEM SETUP

Let $\mathcal{X}$ be a finite candidate set and let $X \in \mathcal{X}$ denote the user's latent target. At turn i, the assistant produces a message a_i and the user responds with u_i, giving history $h_i = (a_1, u_1, \ldots, a_i, u_i)$. The exchange (a_i, u_i), interpreted in context, induces an *intended* constraint $\ell_i : \mathcal{X} \to \{0, 1\}$, where $\ell_i(x) = 1$ means that the candidate x is consistent with the interaction. By construction, $\ell_i(X) = 1$. The assistant does not observe ℓ_i directly; instead, it forms an interpretation of the exchange, which we represent by an *applied* constraint $\tilde{\ell}_i : \mathcal{X} \to \{0, 1\}$. An applied update is valid when $\tilde{\ell}_i = \ell_i$ and misinterpreted otherwise, and we restrict our attention to invalid updates that exclude the true target (i.e., $\tilde{\ell}_i(X) = 0$ whenever $\tilde{\ell}_i \neq \ell_i$).

After T turns, we distinguish the raw interaction history h_T from the assistant's task-relevant state $C_T = (\tilde{\ell}_1, \ldots, \tilde{\ell}_T)$, which records how the preceding interactions have been interpreted. The corresponding set of feasible user intents is

$$\mathcal{X}_T = \{x \in \mathcal{X} : \tilde{\ell}_i(x) = 1 \; \forall i \leq T\}.$$

For example, let $\mathcal{X}$ be a set of possible flights, and suppose the assistant asks $a_i =$ "What departure time works?" and the user responds $u_i =$ "Not before 6 p.m." Let $t(x)$ denote the departure time of flight x. The intended constraint is $\ell_i(x) = \mathbf{1}\{t(x) \geq 18{:}00\}$, but the assistant instead interprets it as $\tilde{\ell}_i(x) = \mathbf{1}\{t(x) < 18{:}00\}$. The true flight X is therefore excluded from the current feasible set $\mathcal{X}_T$.

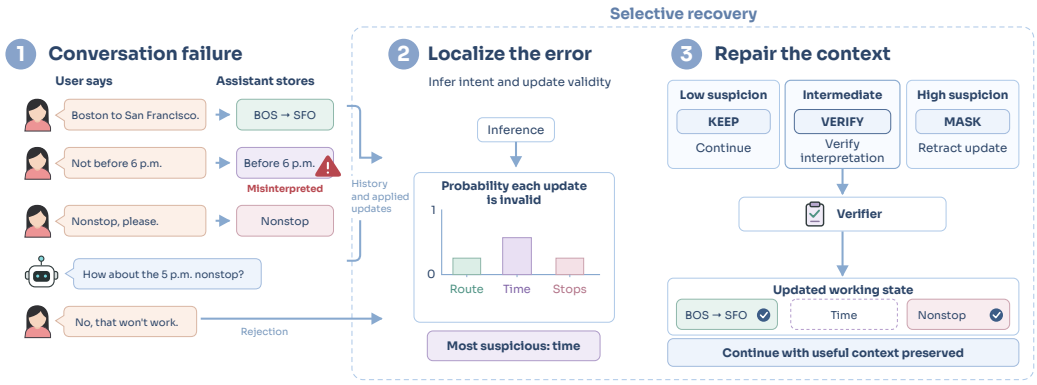

Figure 1: **Overview of LOCI.** (1) The assistant stores an incorrect interpretation of a user constraint, and the error leads to a proposal that the user rejects. (2) LOCI uses the rejection and the applied updates to infer the intended target and each update's probability of being invalid. (3) These scores are used to guide a repair policy to selectively fix the conversational state.

Let $M_i = \mathbf{1}\{\tilde{\ell}_i = \ell_i\} \in \{0, 1\}$ denote the latent validity of update i. Because $M_{1:T}$ is unobserved, the assistant does not know which earlier updates are invalid. It may commit to a candidate $\hat{x} \in \mathcal{X}_T$ or submit a complete requirement summary for validation. Let F denote the feedback returned by this validation step. Rejection shows that the submitted object was not accepted, but does not identify which update, if any, was invalid. At repair time, the available information is $O_T = (h_T, C_T, \hat{x}, F)$, while the target X, intended constraints $\ell_{1:T}$, and validity indicators $M_{1:T}$ remain latent.

3 LOCALIZATION OF CORRUPTED INFERENCES

In this section we introduce LOCI. The high level idea is that each candidate user intent implies a possible explanation of which earlier updates were incorrect, and LOCI weighs these explanations using the observed updates and failure feedback to assign each update a corruption probability (Section 3.1). These probabilities rank likely error locations and quantify localization confidence (Section 3.2). A repair policy then uses them to decide whether to verify an update, remove it, or resume interaction (Section 3.3). An overview of LOCI is provided in Figure 1.

3.1 POSTERIOR ERROR ATTRIBUTION

Each candidate intent explains which updates would be invalid: if it were the target, every update that excludes it would be wrong. Because the target is unknown, we weigh these explanations using the recorded updates and the rejection feedback. An update receives a high LOCI score when the intents it excludes collectively account for a large share of the posterior probability.

Definition 3.1 (LOCI score)**.** Let b be a posterior over candidate intents in $\mathcal{X}$, with probabilities denoted by $\Pr_b$. The LOCI score of update i is

$$c_i := \Pr_b(M_i = 0) = \sum_{x \in \mathcal{X}} b(x)\big[1 - \tilde{\ell}_i(x)\big]. \tag{1}$$

For each update i, we treat the assistant's question and its current recorded state as given. We assume that each candidate intent x then determines a unique intended constraint. We model the recorded update as correct with probability $\rho_i \in (0, 1)$. Conditional on an error, we specify a distribution over the possible incorrect updates for each candidate. Let $q_i(x)$ denote the probability of the recorded update under this distribution when $X = x$. Under the target-excluding error assumption in Section 2, update i is invalid exactly when $\tilde{\ell}_i(X) = 0$. This gives the following likelihood contribution

$$L_i(x) := \rho_i \tilde{\ell}_i(x) + (1 - \rho_i) q_i(x)\big[1 - \tilde{\ell}_i(x)\big]. \tag{2}$$

For a binary update with one possible incorrect outcome, $q_i(x) = 1$ whenever $\tilde{\ell}_i(x) = 0$. More generally, when $q_i(x)$ is constant across incompatible candidates, we denote it by κ_i.

The user accepts a complete proposal exactly when it matches their target. Following rejection of a candidate $\hat{x}$, the remaining candidate set is $S := \mathcal{X} \setminus \{\hat{x}\}$. If $\hat{x} \notin \mathcal{X}$, then $S = \mathcal{X}$. Given a prior p_0 on X, the posterior under the specified update model is

$$b(x) := \frac{p_0(x) \prod_{i=1}^{T} L_i(x)}{\sum_{x' \in S} p_0(x') \prod_{i=1}^{T} L_i(x')}, \qquad x \in S, \tag{3}$$

provided the denominator is positive, with $b(x) = 0$ outside S. The resulting LOCI scores are posterior corruption probabilities under this model.

Continuing the example, suppose the update requires departure before 6 p.m. from BOS. There are two time categories, before 6 p.m. and 6 p.m. or later, and three airports: BOS, JFK, and SFO. Assume an error changes one uniformly chosen field to a different value. For a candidate departing at 7 p.m. from BOS, the recorded update requires changing the time field, so $q_i(x) = 1/2$. For a candidate departing at 5 p.m. from JFK, the airport field must be selected and its value changed to BOS among the alternatives BOS and SFO, giving $q_i(x) = (1/2)(1/2) = 1/4$. A candidate departing at 7 p.m. from JFK differs in both fields, so $q_i(x) = 0$ under this single-field error model.

3.2 LOCALIZING CORRUPTED UPDATES

We localize errors by ranking updates according to their corruption probabilities c_i, with larger values indicating a higher probability that an update excluded the intent. The top-ranked update is $\hat{\tau} \in \arg\max_{1 \le i \le T} c_i$. Under b, this rule maximizes the probability of selecting an invalid update. We use $c_{\hat{\tau}}$ as the model's confidence in that selection.

Each corruption probability c_i aggregates posterior mass over every candidate intent for which update i is invalid, including candidates that violate several applied constraints. A candidate that violates two updates contributes its posterior probability to both scores. Multiple updates may be invalid, so the corruption probabilities do not necessarily sum to one; their sum is the posterior expected number of invalid updates. Once b is computed, the T corruption probabilities require only summation, with no simulation or additional model calls.

The single-error case gives a simple interpretation of this localization rule. If exactly one update is invalid, removing it makes the target consistent with every remaining constraint. For each update i, count the previously excluded candidate intents made feasible by removing it:

$$N_i := \left| \left\{ x \in S : \tilde{\ell}_i(x) = 0,\ \tilde{\ell}_j(x) = 1 \text{ for every } j \neq i \right\} \right|.$$

The following lemma states when these counts determine the posterior over error locations.

Lemma 3.2 (Single-error localization)**.** *Let $E = \{\sum_{j=1}^{T}(1 - M_j) = 1\}$ denote the event that exactly one update is invalid, and let τ denote its position. Suppose $p_0(x)$ is constant and positive on S, $\rho_i = \rho \in (0, 1)$ for every update i, and there is a common $\kappa \in (0, 1]$ such that $q_i(x) = \kappa$ whenever $x \in S$ and $\tilde{\ell}_i(x) = 0$. If $N := \sum_{j=1}^{T} N_j > 0$, then $\Pr_b(\tau = i \mid E) = N_i / N$.*

Thus, under the lemma's assumptions, $\hat{\tau}_{\mathrm{SE}} \in \arg\max_i N_i$ is the maximum a posteriori (MAP) localization rule conditional on exactly one invalid update. We refer to this estimator as Counting in the experiments. We prove this result in Appendix A.1. Under the same assumptions, the normalized LOCI corruption probabilities $c_i / \sum_j c_j$ converge to the single-error posterior N_i / N as $1 - \rho$ approaches zero, without conditioning on exactly one invalid update (Appendix A.2).

3.3 REPAIRING THE CONTEXT

After a rejected commitment, retaining invalid updates can prevent recovery, while deleting valid updates can require additional questions. A repair policy π uses the available observations O_T to choose interventions on the working state of the downstream assistant μ. Let $N_{\mathrm{repair}}(\pi; \mu)$ denote the number of additional user interactions before μ commits to the correct target X, with $N_{\mathrm{repair}}(\pi; \mu) = \infty$ if recovery does not occur. For a fixed assistant μ, the repair objective is

$$\pi^{\star}(O_T) \in \arg\min_{\pi} \mathbb{E}[N_{\mathrm{repair}}(\pi; \mu) \mid O_T].$$

We consider three repair actions. KEEP preserves the current state and resumes the task. VERIFY(i) uses one interaction to check whether the target satisfies the constraint in update i, retaining it if valid and removing it otherwise. Verification, which we assume to be reliable, asks the user to confirm the constraint, but could alternatively defer to a stronger model or a task-specific tool. MASK(i) removes the constraint from the working state without a user interaction. Questions, verification checks, and rejected commitments each cost one interaction; masking and the final correct commitment cost zero. Masking can avoid checking an invalid update, verification avoids deleting valid ones.

The cost of removing an update depends on both its corruption probability and how μ continues after deletion. Updates with the same corruption probability can therefore warrant different repair actions. We restrict attention to a two-threshold policy based on LOCI scores. An update is unresolved if it has neither been deleted nor verified valid. At each repair step, we select the unresolved update $\hat{\tau}$ with the largest LOCI score. For thresholds $0 \leq \gamma_{\text{keep}} < \gamma_{\text{mask}} \leq 1$, the policy takes

$$\pi_\gamma = \begin{cases} \texttt{KEEP}, & c_{\hat{\tau}} < \gamma_{\text{keep}}, \\ \texttt{VERIFY}(\hat{\tau}), & \gamma_{\text{keep}} \leq c_{\hat{\tau}} < \gamma_{\text{mask}}, \\ \texttt{MASK}(\hat{\tau}), & c_{\hat{\tau}} \geq \gamma_{\text{mask}}. \end{cases}$$

Masking update i makes it ineligible for further selection. Its likelihood $L_i(x)$ remains in the inference model, so masking alone leaves the posterior and the other corruption scores unchanged. Positive verification keeps the constraint and conditions the posterior on $\tilde{\ell}_i(X) = 1$. Negative verification has the same effect as masking: its verdict is discarded, and $L_i(x)$ is retained.

After MASK or VERIFY, the controller applies the rule again to the remaining unresolved updates. KEEP resumes μ if the working candidate set is nonempty; otherwise, the controller verifies the selected update. It also resumes μ when no unresolved update remains. New updates and rejected commitments during continuation provide evidence for subsequent repair decisions.

4 RESULTS

We evaluate localization and repair in different settings. We start from a simplified, controlled setup with a fixed pool of binary questions and injected mistakes. Knowing which updates are incorrect lets us measure localization accuracy directly, while controlling the repair actions lets us separate the effects of localization and action selection on recovery cost. Then, we relax controlled setup simplifications and move to a free-form conversation experiment, where a language model generates questions and interprets responses, allowing mistakes to arise naturally. Finally, we extend the evaluation beyond binary questions to categorical requirement updates in task-oriented dialogues. Appendix B provides further experimental details and results.

4.1 CONTROLLED EXPERIMENTS

Setup. We start with a controlled environment adapted from the 20 Questions (20Q) game (Hu et al., 2024). The assistant aims to identify a hidden target by asking yes/no questions from a fixed pool. Each answer narrows the candidate set (e.g., answering "yes" to "Can it fly?" retains only candidates that can fly). We store the questions in a binary attribute matrix, where each row represents a candidate, each column represents a question, with each entry representing the candidate's truthful answer. The question–answer pair therefore defines the intended constraint ℓ_i, which retains candidates whose answers agree with the user's response. At each turn, with probability $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$, we simulate a misinterpretation by purposefully applying the truthful answer to a different question's predicate, producing an applied constraint $\tilde{\ell}_i$. We uniformly sample this replacement among predicates that exclude the target while retaining at least one current candidate. This gives us known invalid updates for evaluating localization and repair. The assistant then continues asking questions until it makes a wrong guess or a later update leaves no feasible candidates, revealing that an earlier interpretation was incorrect.

We evaluate this setup on four main domains: Animals, Food, Objects, and Places (Hu et al., 2024). To evaluate performance with a larger set of intents, we also use CUB-200-2011, a dataset of 200 bird species (Wah et al., 2011) (more details in Appendix B.1.3). Given the previous context, questions

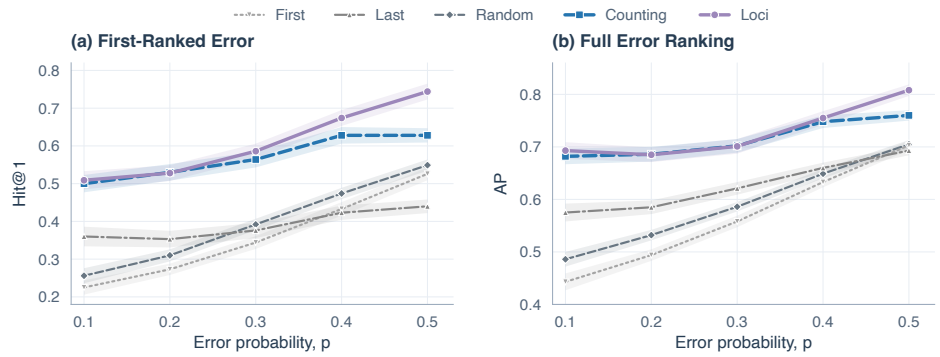

Figure 2: **Error localization in controlled 20Q setup.** We compare localization methods after a wrong guess for different error probabilities. (a) Hit@1 measures whether the top-ranked update is invalid. (b) Average precision evaluates the ranking of all invalid updates.

are selected under three policies. Random samples uniformly among unasked questions that split the remaining candidates. Value of information (VoI) selects the question with the largest expected information gain (Dong et al., 2026). Natural uses Llama 3.1 8B (Grattafiori et al., 2024) to select a question from the pool. Throughout this section, the reported results are obtained under the Natural policy. Results under different policies can be found in Appendix B.1.1 and Appendix B.1.2.

Localization. Figure 2 reports Hit@1 in panel (a), which measures whether the top-ranked update is invalid, and average precision (AP) in panel (b), which measures how well the ranking places invalid updates ahead of valid ones. We compare LOCI with Random, First, and Last, which rank updates randomly, earliest first, and latest first, respectively. We also compare it to Counting, which corresponds to the single-error MAP rule from Section 3.2, which ranks updates by the number of candidates made feasible by removing each update. LOCI and Counting perform similarly at low error rates. Their gap widens at higher error rates, where failed conversations contain more invalid updates: the mean number rises from 1.26 at $p = 0.1$ to 2.70 at $p = 0.5$. At $p = 0.5$, LOCI achieves 74.4% Hit@1 and 0.808 AP, compared with 62.8% and 0.760 for Counting. This advantage is consistent with the benefit of allowing multiple invalid updates in the posterior. The full localization results are reported in Table A1; Tables A2 and A3 report results under the alternative question policies.

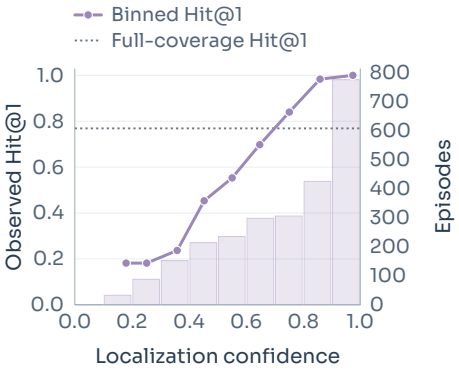

Figure 3: **Higher localization confidence corresponds to higher accuracy.** The curve shows Hit@1 within bins of $c_{\hat{\tau}}$. Bars show the number of episodes in each bin.

On top of localization accuracy, we also study whether the score of the top-ranked update predicts how often that selection is correct. Figure 3 groups top-ranked updates by their predicted corruption probability $c_{\hat{\tau}}$ and reports Hit@1 within each bin. Observed accuracy increases monotonically across score bins, indicating that the score helps identify when the top-ranked update is more likely to be invalid. This trend supports interpreting LOCI scores as a measure for localization confidence, from which we can build a repair policy to selectively fix the conversational state.

Repair. Repair begins after a wrong guess or an empty candidate set. Errors continue at rate p during recovery, and LOCI uses the known reliability $\rho = 1 - p$. Recovery cost counts additional questions, verifications, and rejected guesses after the initial failure; masking and the final correct guess cost zero. For each domain and error rate, we choose the repair-policy thresholds γ_{keep} and γ_{mask} from a grid of candidate pairs, and we evaluate the policy using the threshold pair that gives the lowest mean recovery cost on a train set. To isolate the effect of localization, we first let each method select a suspect corrupt update and always apply `VERIFY` to that turn. This allows us to compare LOCI

	Controlled recovery		Free-form LLM recovery				
Method	20Q	CUB	Animals	Food	Objects	Places	20Q
First + VERIFY	15.66 ± 0.50	26.72 ± 0.44	22.29 ± 1.79	20.62 ± 1.38	25.33 ± 1.76	20.27 ± 1.77	22.13 ± 0.87
Last + VERIFY	13.84 ± 0.43	25.81 ± 0.42	19.71 ± 1.28	18.57 ± 1.12	23.67 ± 1.62	18.73 ± 1.57	20.17 ± 0.72
Random + VERIFY	14.81 ± 0.46	26.31 ± 0.42	20.67 ± 0.94	19.95 ± 0.94	26.00 ± 1.97	20.91 ± 1.50	21.88 ± 0.71
Counting + VERIFY	13.48 ± 0.44	23.82 ± 0.41	21.52 ± 2.34	18.62 ± 0.90	24.75 ± 2.02	19.00 ± 1.56	20.97 ± 0.91
Restart	63.76 ± 3.33	428.35 ± 12.51	43.67 ± 5.90	46.71 ± 6.88	56.88 ± 7.71	20.09 ± 6.47	41.84 ± 3.38
LOCI + VERIFY	12.04 ± 0.38	21.86 ± 0.36	12.05 ± 1.34	11.19 ± 1.03	15.29 ± 1.41	9.36 ± 1.22	11.97 ± 0.63
LOCI + MASK	30.18 ± 0.74	217.01 ± 3.30	12.52 ± 3.32	10.48 ± 2.64	15.42 ± 3.08	6.00 ± 2.23	11.10 ± 1.44
LOCI adaptive	$\mathbf{9.99 \pm 0.35}$	$\mathbf{20.51 \pm 0.35}$	$\mathbf{10.62 \pm 2.50}$	$\mathbf{7.10 \pm 1.40}$	$\mathbf{10.83 \pm 2.13}$	$\mathbf{4.00 \pm 1.48}$	$\mathbf{8.14 \pm 0.96}$

Table 1: **Fitted adaptive LOCI has the lowest mean recovery cost in both the controlled and free-form setup.** Mean additional interactions after an initial failure $\pm$ target-cluster bootstrap SE. First, Last, Random, and Counting use VERIFY. Free-form 20Q averages the four domains equally.

to the baselines under the same repair action. We then fix localization to LOCI and compare the always-VERIFY, always-MASK, and adaptive policies. The Restart baseline removes unresolved constraints and resumes questioning. Results are shown in Table 1. Under the shared always-VERIFY policy, LOCI requires 12.04 recovery interactions on average, compared with 13.48 for Counting, a reduction of 10.7%. Adaptive repair further reduces mean recovery cost to 9.99 interactions, a 17.1% reduction relative to always-VERIFY. Full repair results are reported in Table A4; Tables A5 and A6 report results under the alternative question policies. We also ablate whether masked updates remain evidence for posterior inference and when the controller resumes questioning (Appendix B.5.2).

4.2 REPAIR IN LANGUAGE-MODEL INTERACTIONS

So far, we have evaluated LOCI in a controlled sandbox. We now relax two main simplifications: in the free-form extension, LLMs generate the conversation, allowing errors to arise naturally; in task-oriented dialogue, updates express categorical requirements beyond binary answers.

4.2.1 FREE-FORM LANGUAGE-MODEL INTERACTIONS

Setup. Here, we evaluate LOCI on the same 20Q game introduced in Section 4.1, focusing on free-form conversations. We use Llama 3.1 8B to generate the assistant's questions and to simulate the user's responses. In this scenario, there are no fixed question pools or injected errors. To represent each exchange as a candidate-set update, the assistant model evaluates the question for every candidate and retains candidates whose predicted answers agree with the recorded response. These model-generated constraints define the working state and the updates evaluated by LOCI. Recovery begins after a wrong guess. After a repair action, the assistant continues generating questions and interpreting responses using the same procedure. Recovery costs follow Section 4.1, with a cap of 100 interactions. For these experiments, we fix the common update reliability of $\rho = 0.9$ across all four domains.

Localization. Free-form conversations lack ground-truth error labels, hence evaluating localization accuracy directly is difficult. An LLM judge can provide reference labels, but these depend on the judge's accuracy and interpretation of ambiguous questions (Appendix B.2.1). To avoid this, we therefore use recovery cost–conditional on a fixed policy–as a proxy for quality of localization. All methods start from the same failed conversations and reconstructed updates, and use the same always-VERIFY and continuation procedures. Under this setup, any change in recovery can be attributed to the localization quality. LOCI requires fewer interactions than all evaluated baselines in every domain, suggesting superior quality in the localization estimator. Relative to the strongest of these baselines in each domain, it reduces mean recovery cost by 35–50%. We report a separate localization evaluation using reference labels from an LLM judge in Appendix B.2.1.

Alternatively, we ask the interacting LLM which earlier updates need repair. We test Llama 3.1 8B as the error selector, with both methods using the same always-VERIFY controller. In this comparison, LOCI reduces mean recovery cost by 41.1% (Appendix B.2.2). This may reflect that the model's original misinterpretations also affect its ability to diagnose the problem: the same model constructs

the updates and later judges which ones were incorrect. This supports the value of a separate inference procedure for error attribution on top of the underlying model powering the assistant.

Repair. We next fix localization to LOCI and compare the always-VERIFY, always-MASK, and adaptive policies. We select adaptive thresholds using five target folds within each domain, choosing each fold's thresholds on the other four folds. The adaptive policy reduces mean recovery cost by 12–57% relative to always-VERIFY and has the lowest mean cost in all four domains. It avoids many verification checks while requiring fewer additional questions and rejected guesses than always-MASK. All three policies recover every failed conversation within the interaction budget. Per-domain results are reported in Table 1. We also observe that LOCI achieves larger relative reductions in recovery cost over the baselines in free-form conversations than in the controlled setting. The adaptive policy also provides larger savings over always-VERIFY.

To investigate this pattern, we study the recovery cost components in Appendix B.2.3. Verification accounts for a larger share of recovery cost under LOCI with always-VERIFY; reducing verification checks can therefore save a larger fraction of recovery cost in the free-form setting. Under this policy, LOCI reduces checks that confirm target-compatible constraints compared with the localization baselines. The adaptive policy avoids further checks through masking, although it requires more questions and rejected guesses than always-VERIFY. This suggests that LOCI may be more effective in longer conversations, where it provides the opportunity to save checks while retaining enough information to identify the target. We also repeat the evaluation with Qwen3-8B and observe similar recovery gains (Table A10) and ablate the simulated user's access to the history (Appendix B.5.3).

4.2.2 BEYOND BINARY QUESTIONS

Setup. We next relax the assumption that questions must receive binary answers. For this, we use MultiWOZ 2.4 (Ye et al., 2022), a dataset of task-oriented dialogues in which users communicate requirements to an assistant (e.g., finding a cheap hotel in the east). These requirements are annotated as categorical slot values, where each slot is a named requirement (e.g., price or area). Each exchange defines an intended constraint containing all requirements introduced in that exchange (e.g., `price = cheap` and `area = east`). The assistant's goal is to recover the complete requirement configuration from a finite set of configurations supported by the database. We exclude annotated configurations outside the candidate space, where successful recovery would be impossible, and require at least two requirement-bearing turns, so localization involves choosing among multiple updates. Similarly to the 20Q setup, we construct the updates from the annotations and inject errors independently with different probabilities. For each update, we randomly select one slot and replace its value with a uniformly sampled alternative (e.g., changing `area = east` to `area = west`). During repair, the model selects a slot and generates a question about it; the simulated user returns the slot's annotated value. Appendix B.3.1 details the benchmark, setup, and instantiation of Loci.

Localization. As in Section 4.1, we evaluate localization using Hit@1 and AP over invalid updates. LOCI achieves higher Hit@1 and AP than all baselines (Table 2). Compared with Counting, the strongest baseline, it improves Hit@1 from 65.5% to 75.8% and AP from 0.817 to 0.871. These results average across error rates, while the full results can be found in Table A11.

	Localization		Repair
Method	Hit@1 ↑	AP ↑	Interactions ↓
First	0.530 ± 0.011	0.745 ± 0.006	3.51 ± 0.05
Last	0.513 ± 0.012	0.734 ± 0.007	3.54 ± 0.06
Random	0.528 ± 0.005	0.741 ± 0.004	3.53 ± 0.06
Counting	0.655 ± 0.012	0.817 ± 0.006	3.33 ± 0.05
Restart	–	–	3.17 ± 0.06
LOCI	$\mathbf{0.758 \pm 0.011}$	$\mathbf{0.871 \pm 0.006}$	$\mathbf{2.63 \pm 0.06}$

Table 2: **LOCI improves localization and reduces recovery cost on MultiWOZ.** Baselines use the always-VERIFY policy, LOCI the adaptive one.

Repair. We compare adaptive LOCI against the localization baselines with always-VERIFY. Adaptive LOCI requires 2.63 recovery interactions on average, versus 3.17 for the strongest baseline (Table 2). With localization fixed to LOCI, always-MASK requires fewer interactions than adaptive repair (Table A12). Reliable, targeted slot questions likely make discarded information cheap to recover, so verification checks cost more than they save in questions and wrong guesses. However, adaptive repair achieves the lowest mean cost in the free-form experiments, suggesting that the preferred intervention depends on localization confidence but also the cost of recovering discarded information, which is lower in this scenario (see Appendix B.3.3 for more details).

Generalization across interaction structures. Our experiments progressively relax the structure of the interaction while retaining the underlying problem of identifying corrupted interpretations. The controlled setting isolates localization under known errors, the free-form setting introduces naturally occurring model misinterpretations without a predefined question pool, and MultiWOZ extends the formulation to categorical and multi-slot requirements. Across these settings, LOCI operates on the same abstraction: a sequence of uncertain updates to the assistant's representation of user intent.

5 RELATED WORK

Multi-turn failure of LLMs. LLMs have been observed to often fail in multi-turn conversations. Laban et al. (2026) show that LLMs commit to early incorrect interpretations of underspecified requests, struggling to recover as the conversation unfolds. Similar weaknesses are documented by Deshpande et al. (2025). Tack et al. (2026) show that performance can also degrade when the user revises their intent over successive turns, a failure mode closely related to the explicit corrections documented in Chen et al. (2026). Huang et al. (2026a) show that all the assistant's own prior turns can sometimes be omitted from context without degrading the conversation quality. Recent work examines the causal effects of earlier assistant turns (Li et al., 2026) and organizes nonlinear dialogue histories to better manage context (Hu et al., 2026). Overall, these studies suggest that LLMs may over-condition on past (wrong) turns and that revisiting conversation history may help.

Error localization and repair. Recovering from a failed conversation requires identifying the earlier updates responsible for the failure. LLMs however struggle to self-correct without external feedback (Huang et al., 2024), but can often correct an error when given its location (Tyen et al., 2024). In stepwise reasoning, process reward models are used to score intermediate steps (Lightman et al., 2024); closer to us, Kim et al. (2026a) treat step correctness as latent and infer it from the reasoning trace. In a conversation, the bad update may be the interpretation of an earlier exchange. Xie et al. (2022) study this problem in dialogue states; Shen et al. (2026) refactors conversation history to mitigate state drift. Other methods plan recovery or repair known memory faults (Kim et al., 2026b; Yu et al., 2026). LOCI first localizes the mistake and then makes repair selective by using failure feedback to estimate the validity of each past update.

Clarification and information acquisition. Clarification works study what type of information the assistant should aim to collect, and when asking the user is worth the effort. Rao & Daumé III (2018) rank questions by their expected information value, and Hu et al. (2024) plan information-seeking questions under uncertainty. Deciding when to ask also benefits from learned policies over the conversation and models of future turns (Chen et al., 2025; Zhang et al., 2025). Wu et al. (2025) frame eliciting user intent as part of long-term collaboration, while Dong et al. (2026) explicitly weighs the benefit of asking against user effort. Other works test whether models acquire the information needed to determine an answer (Li et al., 2025; Huang et al., 2026b). Our controller weighs the cost of checking a past interpretation against the risks of keeping or removing it.

6 CONCLUSION

Early misinterpretations can persist in the conversation and shape later interactions. LOCI treats these interpretations as uncertain, and uses posterior inference over the possible user intents to identify likely sources of errors. This uncertainty guides a repair policy that selectively discards misleading context while preserving useful information, reducing the interactions needed for recovery. Comparing LOCI with the interacting language model as an error selector further supports the value of an external controller for diagnosing these failures. In future work, LOCI scores could be supplied to the model to inform its own revision and clarification decisions, although this integration remains untested. Localization accuracy depends on the assumed update reliability. In the free-form recovery study, paired confidence intervals for cost changes include zero, although this does not establish equivalence (Appendix B.5.1). Moreover, our evaluation assumes a fixed user intent within a given finite candidate space containing the target. Extending LOCI to open-ended tasks requires constructing candidate hypotheses with sufficient coverage and possibly appropriate weights. As we discuss in Appendix B.4, candidate generation offers a possible extension.

AI USE STATEMENT

Generative AI tools were used for assistance in code and paper writing. More specifically, Codex was used to assist with writing, debugging, and running experimental code, together with assistance in refining plot styles. Moreover, ChatGPT was used to polish the writing of author-written text in some parts of the paper and to verify the proofs. We have not used generative AI tools to draft sections of the paper. We have reviewed all AI-assisted work. That is, any language and code written by an AI was re-read by a human and checked for factual inaccuracies, bugs, or plagiarism by at least one of the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We do not believe our submission raises any questions regarding the code of ethics. The evaluations use controlled simulations, LLM-generated conversations, and existing publicly available benchmarks. We did not recruit human participants or conduct a user study. Our method aims to identify mistaken interpretations and selectively repair an assistant's conversational state. Incorrect localization may remove valid constraints or retain mistaken ones. Deployment should account for the consequences of these errors when choosing between automatic removal and verification with the user.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide extensive methodological and experimental details in the appendix. We also plan to release the full codebase used to run our experiments in the near future.

REFERENCES

Max Chen, Ruoxi Sun, Tomas Pfister, and Sercan Arik. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. 2025. URL https://openreview.net/pdf?id=SIE6VFps9x.

Xingwu Chen, Zhanqiu Zhang, Steven Y. Guo, and Difan Zou. Breaking contextual inertia: Reinforcement learning with single-turn anchors for stable multi-turn interaction. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 6302–6318, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.313. URL https://aclanthology.org/2026.findings-acl.313/.

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/.

Yijiang River Dong, Tiancheng Hu, Zheng Hui, Caiqi Zhang, Ivan Vulić, Andreea Bobu, and Nigel Collier. Value of information: A framework for human–agent communication. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42879–42896, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1987. URL https://aclanthology.org/2026.acl-long.1987/.

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

Junan Hu, Shudan Guo, Wenqi Liu, Jianhua Yin, and Yinwei Wei. Context-agent: Dynamic discourse trees for non-linear dialogue. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 29449–29462, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1472. URL https://aclanthology.org/2026.findings-acl.1472/.

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei Koh, and Bryan Hooi. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in llms. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 24181–24215. Curran Associates, Inc., 2024. doi: 10.52202/079017-0762. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2b0e14abd8128e6bf98b6b0bec1cfcbf-Paper-Conference.pdf.

Jenny Y Huang, Leshem Choshen, Wei Sun, Omar Khattab, Ramón Fernandez Astudillo, Mehul Damani, Tamara Broderick, and Jacob Andreas. Do llms benefit from their own words? *arXiv preprint arXiv:2602.24287*, 2026a.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32808–32824, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8b4add8b0aa8749d80a34ca5d941c355-Paper-Conference.pdf.

Yepeng Huang, Jiawen Zhang, Michelle Dai, Xiaorui Su, Shanghua Gao, Zi Wang, and Marinka Zitnik. Do llms know what to ask and when? evaluating multi-turn information seeking. *arXiv preprint arXiv:2608.14808*, 2026b.

Minsu Kim, Jean-Pierre R. Falet, Oliver Ethan Richardson, Xiaoyin Chen, Moksh Jain, Sungjin Ahn, Sungsoo Ahn, and Yoshua Bengio. Latent veracity inference for identifying errors in stepwise reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=eux1cp8GqC.

Takyoung Kim, Jinseok Nam, Chandrayee Basu, Xing Fan, Chengyuan Ma, Heng Ji, Gokhan Tur, and Dilek Hakkani-Tür. Rein: Conversational error recovery with reasoning inception. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=4J3kkHI6m5.

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.

Belinda Li, Been Kim, and Zi Wang. Questbench: Can llms ask the right question to acquire information in reasoning tasks? In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-4503. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/c42c8d51556fabb4b57fc86d3d3d0d09-Paper-Datasets_and_Benchmarks_Track.pdf.

Jinnan Li, Zheren Fu, Yue Wang, Jinzhe Li, Yuan Wu, and Yi Chang. What if llms ate their words: Causal history effects in multi-turn interaction. *arXiv preprint arXiv:2609.05882*, 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*, volume 2024, pp. 39578–39601, 2024.

Sudha Rao and Hal Daumé III. Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pp. 2737–2746, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1255. URL https://aclanthology.org/P18-1255/.

Jiawei Shen, Jia Zhu, Hanghui Guo, Weijie Shi, Yue Cui, Qingyu Niu, Guoqing Ma, Jingjiang Liu, Yidan Liang, Yilin Wang, Shimin Di, and Jiajie Xu. ACR: Adaptive context refactoring via context refactoring operators for multi-turn dialogue. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3149–3167, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.155. URL https://aclanthology.org/2026.findings-acl.155/.

Jihoon Tack, Philippe Laban, and Jennifer Neville. Llms get lost in evolving user intent. *arXiv preprint arXiv:2607.20734*, 2026.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. LLMs cannot find reasoning errors, but can correct them given the error location. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13894–13908, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.826. URL https://aclanthology.org/2024.findings-acl.826/.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. CollabLLM: From passive responders to active collaborators. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67260–67283. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wu25i.html.

Hongyan Xie, Haoxiang Su, Shuangyong Song, Hao Huang, Bo Zou, Kun Deng, Jianghua Lin, Zhihui Zhang, and Xiaodong He. Correctable-DST: Mitigating historical context mismatch between training and inference for improved dialogue state tracking. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 876–889, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.56. URL https://aclanthology.org/2022.emnlp-main.56/.

Fanghua Ye, Jarana Manotumruksa, and Emine Yilmaz. Multiwoz 2.4: A multi-domain task-oriented dialogue dataset with essential annotation corrections to improve state tracking evaluation. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 351–360, 2022.

Caili Yu, Yiqi Wang, Jiaqi Zhang, Yiqun Duan, Mingkai Zheng, Zhangkai Wu, Kaize Shi, and Taotao Cai. From faulty memories to corrected actions: Dependency-guided rollback repair for memory-augmented agents, 2026. URL https://arxiv.org/abs/2608.10502.

Michael Zhang, W. Bradley Knox, and Eunsol Choi. Modeling future conversation turns to teach llms to ask clarifying questions. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 60722–60742, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/97e2df4bb8b2f1913657344a693166a2-Paper-Conference.pdf.

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/.

A THEORETICAL RESULTS

Throughout this section, let $S := \mathcal{X} \setminus \{\hat{x}\}$.

A.1 SINGLE-ERROR LOCALIZATION

Proof of Theorem 3.2. For each update i, define its one-flip witness set

$$W_i := \{x \in S : \tilde{\ell}_i(x) = 0,\ \tilde{\ell}_j(x) = 1 \text{ for every } j \neq i\}.$$

Thus, $|W_i| = N_i$. Under the error model of Section 3.1, $M_j = \tilde{\ell}_j(X)$, so E holds exactly when X violates one applied constraint and satisfies the rest. Since $\hat{x} \in \mathcal{X}_T$ satisfies every applied constraint, E forces $X \neq \hat{x}$, and E is equivalent to $X \in \bigcup_{j=1}^{T} W_j$. The sets $W_1, \ldots, W_T$ are disjoint, and conditional on E, the event $\tau = i$ is equivalent to $X \in W_i$.

For every $x \in W_i$, Equation (3) assigns the unnormalized posterior weight

$$p_0(x)(1-\rho)\kappa\rho^{T-1}.$$

This weight is the same positive constant for every one-flip witness because p_0 is uniform on S. Therefore,

$$\Pr(\tau = i \mid O_T, E) = \frac{|W_i|}{\sum_{j=1}^{T} |W_j|} = \frac{N_i}{N},$$

where the denominator is positive by assumption. □

A.2 RARE-ERROR EXTENSION

The preceding proof conditions on exactly one invalid update. We now consider the unrestricted posterior as the prior error probability approaches zero.

Proposition A.1 (Rare-error limit)**.** *Suppose p_0 is uniform on S. Fix $\kappa > 0$, and for each $p \in (0,1)$ let $\rho_i = 1 - p$ and $\kappa_i = \kappa$ for every update i. Let $c_i(p)$ denote the corruption probability in Equation* (1) *under these parameters, and assume $N := \sum_{j=1}^{T} N_j > 0$. Then $\sum_{j=1}^{T} c_j(p) > 0$, and*

$$\bar{c}_i(p) := \frac{c_i(p)}{\sum_{j=1}^{T} c_j(p)} = \frac{N_i}{N} + O(p) \qquad \text{as } p \downarrow 0$$

for every update i. Moreover, for every pair i, j with $N_i > N_j$,

$$c_i(p) > c_j(p) \qquad \text{whenever} \qquad 0 < p < \frac{1}{1 + \kappa|S|}.$$

Proof. For $x \in S$, let

$$d(x) := \sum_{t=1}^{T} \big(1 - \tilde{\ell}_t(x)\big), \qquad q_p := \frac{p\kappa}{1-p}.$$

A satisfied constraint contributes likelihood $1 - p$, while a violated constraint contributes likelihood $p\kappa$. Since p_0 is uniform on S, Equation (3) becomes

$$b_p(x) = \frac{q_p^{d(x)}}{Z_p}, \qquad Z_p := \sum_{x' \in S} q_p^{d(x')}.$$

The assumption $N > 0$ implies that S is nonempty, so $Z_p > 0$.

Define

$$A_i(q) := \sum_{x \in S:\, \tilde{\ell}_i(x) = 0} q^{d(x)}.$$

Then Equation (1) gives $c_i(p) = A_i(q_p)/Z_p$. The coefficient of q in $A_i(q)$ is N_i because the corresponding candidates violate update i and no other update. Since S is finite,

$$A_i(q) = N_i q + O(q^2).$$

Moreover,

$$D(q) := \sum_{j=1}^{T} A_j(q) = \sum_{x \in S} d(x) q^{d(x)} = Nq + O(q^2).$$

For $q > 0$, the assumption $N > 0$ gives $D(q) > 0$. Hence $\sum_j c_j(p) = D(q_p)/Z_p > 0$, and

$$\bar{c}_i(p) = \frac{A_i(q_p)}{D(q_p)} = \frac{N_i + O(q_p)}{N + O(q_p)} = \frac{N_i}{N} + O(q_p).$$

Finally, $q_p = p\kappa/(1-p) = O(p)$ as $p \downarrow 0$, which proves the first claim.

For the ranking claim, fix i, j with $N_i > N_j$. We have

$$Z_p\big(c_i(p) - c_j(p)\big) = (N_i - N_j) q_p + R_{ij}(q_p),$$

where

$$R_{ij}(q) := \sum_{x \in S:\, d(x) \geq 2} \Big(\mathbf{1}\{\tilde{\ell}_i(x) = 0\} - \mathbf{1}\{\tilde{\ell}_j(x) = 0\} \Big) q^{d(x)}.$$

If $0 < q < 1$, then $|R_{ij}(q)| \leq |S| q^2$. Suppose $q_p < 1/|S|$. Since S is nonempty, $1/|S| \leq 1$, so the bound on R_{ij} applies at $q = q_p$, and $N_i - N_j \geq 1$ gives

$$Z_p\big(c_i(p) - c_j(p)\big) \geq q_p - |S| q_p^2 = q_p\big(1 - |S| q_p\big) > 0.$$

The condition $q_p < 1/|S|$ is equivalent to

$$p < \frac{1}{1 + \kappa |S|}.$$

Since $Z_p > 0$, the ranking claim follows. The normalized scores $\bar{c}_i(p)$ preserve the same ordering because they divide every $c_i(p)$ by the common positive factor $\sum_{j=1}^{T} c_j(p)$. □

B Additional Results and Discussion

B.1 Controlled Localization and Repair

Candidate sets come from the 20Q benchmark of Hu et al. (2024), which provides four domains: Animals (27 candidates), Food (26), Objects (37), and Places (21). For each domain we wrote a pool of yes/no questions and encoded each question as a binary predicate over the candidates, giving an attribute matrix with one row per candidate and one column per question, where entry (x, q) is the true answer to question q for candidate x. The matrix determines the truthful answers and the candidate-set updates and each target can be uniquely identified through the questions.

Under the Natural question policy of Section 4.1, a language model uses the question text to select among the available predicates. In this section we report a series of robustness checks to our main results. In particular, we report localization and recovery results across different question-selection policies, evaluate recovery with a larger candidate set, and examine sensitivity to the assumed update reliability and variants of the repair controller.

Throughout this section, evaluation is conditioned on an initial failure. For localization, Hit@1 is the fraction of scored histories whose top-ranked update is invalid. Average precision (AP) is computed for each history by averaging the precision at the ranks occupied by invalid updates. Equal scores are ordered by earlier turn. For repair, we measure the mean number of additional interactions required to recover: each question, verification and subsequent wrong guess costs one, while masking and the final correct guess cost zero. Costs exclude the preceding interaction and the initial rejection.

B.1.1 Localization Across Question Policies

Tables A1 to A3 report localization results under three question policies: Natural, VoI, and Random. The assistant asks predicates from the fixed pool. Under VoI it picks the predicate with the largest expected information gain over its current candidate set; under Random it picks uniformly among

	1 error	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
Scored failures	4,275	565	822	970	1,038	1,066
Mean T	4.78	6.60	6.28	5.90	5.36	5.24
Mean K	1.00	1.26	1.63	1.97	2.26	2.70
Hit@1						
First	0.313 ± 0.016	0.225 ± 0.019	0.273 ± 0.015	0.344 ± 0.017	0.433 ± 0.018	0.526 ± 0.017
Last	0.352 ± 0.011	0.360 ± 0.026	0.353 ± 0.022	0.376 ± 0.020	0.423 ± 0.019	0.440 ± 0.018
Random	0.296 ± 0.009	0.256 ± 0.020	0.310 ± 0.016	0.392 ± 0.015	0.474 ± 0.015	0.549 ± 0.015
Counting	$\mathbf{0.737 \pm 0.020}$	0.500 ± 0.023	$\mathbf{0.530 \pm 0.022}$	0.564 ± 0.021	0.628 ± 0.022	0.628 ± 0.019
LOCI	0.719 ± 0.019	$\mathbf{0.509 \pm 0.024}$	0.528 ± 0.021	$\mathbf{0.586 \pm 0.020}$	$\mathbf{0.674 \pm 0.021}$	$\mathbf{0.744 \pm 0.021}$
AP						
First	0.545 ± 0.011	0.443 ± 0.016	0.494 ± 0.011	0.558 ± 0.011	0.633 ± 0.010	0.703 ± 0.009
Last	0.583 ± 0.007	0.575 ± 0.017	0.585 ± 0.013	0.621 ± 0.012	0.660 ± 0.010	0.693 ± 0.009
Random	0.538 ± 0.008	0.486 ± 0.014	0.532 ± 0.010	0.586 ± 0.009	0.649 ± 0.009	0.705 ± 0.008
Counting	$\mathbf{0.849 \pm 0.012}$	0.682 ± 0.015	$\mathbf{0.686 \pm 0.014}$	$\mathbf{0.702 \pm 0.013}$	0.748 ± 0.012	0.760 ± 0.010
LOCI	0.837 ± 0.012	$\mathbf{0.693 \pm 0.016}$	0.685 ± 0.014	0.701 ± 0.014	$\mathbf{0.755 \pm 0.013}$	$\mathbf{0.808 \pm 0.011}$

Table A1: Localization under the Natural question policy (mean $\pm$ SE). Higher is better; bold marks the highest mean for each metric in each column.

	1 error	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
Scored failures	2,527	2,013	3,351	4,274	4,824	5,178
Mean T	3.77	4.70	4.69	4.69	4.69	4.71
Mean K	1.00	1.21	1.46	1.77	2.11	2.51
Hit@1						
First	0.396 ± 0.020	0.283 ± 0.012	0.328 ± 0.009	0.396 ± 0.009	0.468 ± 0.009	0.539 ± 0.008
Last	0.345 ± 0.011	0.383 ± 0.016	0.420 ± 0.014	0.472 ± 0.012	0.520 ± 0.010	0.601 ± 0.010
Random	0.377 ± 0.010	0.346 ± 0.012	0.368 ± 0.009	0.427 ± 0.008	0.492 ± 0.007	0.572 ± 0.008
Counting	$\mathbf{0.783 \pm 0.015}$	0.555 ± 0.023	0.571 ± 0.021	0.588 ± 0.019	0.608 ± 0.017	0.636 ± 0.017
LOCI	0.773 ± 0.016	$\mathbf{0.569 \pm 0.023}$	$\mathbf{0.593 \pm 0.022}$	$\mathbf{0.626 \pm 0.020}$	$\mathbf{0.671 \pm 0.018}$	$\mathbf{0.744 \pm 0.017}$
AP						
First	0.636 ± 0.012	0.530 ± 0.009	0.563 ± 0.007	0.608 ± 0.006	0.657 ± 0.006	0.704 ± 0.005
Last	0.572 ± 0.008	0.587 ± 0.011	0.617 ± 0.010	0.656 ± 0.008	0.693 ± 0.007	0.743 ± 0.006
Random	0.608 ± 0.008	0.567 ± 0.009	0.586 ± 0.007	0.626 ± 0.006	0.668 ± 0.005	0.721 ± 0.005
Counting	$\mathbf{0.869 \pm 0.010}$	$\mathbf{0.707 \pm 0.015}$	$\mathbf{0.706 \pm 0.014}$	0.712 ± 0.013	0.728 ± 0.011	0.752 ± 0.010
LOCI	0.855 ± 0.011	0.702 ± 0.017	0.703 ± 0.016	$\mathbf{0.720 \pm 0.014}$	$\mathbf{0.757 \pm 0.012}$	$\mathbf{0.819 \pm 0.010}$

Table A2: Localization under the VoI question policy (mean $\pm$ SE). Higher is better; bold marks the highest mean for each metric in each column.

unasked predicates that still split its candidate set; under Natural, the language model picks the next question given the question history and the candidates that survive its current state.

In every case, the user answers truthfully from the attribute matrix, and a grounding error at rate p per turn replaces the applied predicate by one that excludes the true target while at least one candidate survives, so every failure is a rejected wrong guess. Invalid updates are the turns whose applied constraint excludes the target; K is their number and T the conversation length at failure.

B.1.2 RECOVERY ACROSS QUESTION POLICIES

Tables A4 to A6 report repair results under the three question policies. The tables share the same repair methods and error-rate scenarios. All the baselines that support localization (i.e., all but Restart) are evaluated under the always-`VERIFY` policy. For LOCI scores, as they support uncertainty quantification, we also evaluate the adaptive policy, alongside the always-`MASK` policy.

Questions, verifications and later wrong guesses cost one; masks and the correct guess cost zero. For each domain and error-rate scenario, we fit 26 threshold pairs separately for each adaptive variant using five target folds. Every repetition of a target stays in the same fold, and we select the lowest training mean cost.

	1 error	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
Scored failures	3,416	2,616	3,915	4,665	5,038	5,267
Mean T	6.94	6.87	6.41	6.04	5.78	5.62
Mean K	1.00	1.32	1.67	2.05	2.47	2.89
Hit@1						
First	0.221 ± 0.018	0.221 ± 0.008	0.282 ± 0.007	0.363 ± 0.008	0.448 ± 0.007	0.530 ± 0.007
Last	0.229 ± 0.010	0.295 ± 0.009	0.340 ± 0.009	0.401 ± 0.009	0.474 ± 0.010	0.551 ± 0.011
Random	0.228 ± 0.012	0.249 ± 0.009	0.311 ± 0.007	0.388 ± 0.008	0.473 ± 0.009	0.559 ± 0.008
Counting	$\mathbf{0.697 \pm 0.016}$	$\mathbf{0.538 \pm 0.014}$	0.549 ± 0.013	0.562 ± 0.012	0.581 ± 0.013	0.605 ± 0.014
LOCI	0.646 ± 0.016	0.527 ± 0.016	$\mathbf{0.554 \pm 0.014}$	$\mathbf{0.593 \pm 0.014}$	$\mathbf{0.652 \pm 0.013}$	$\mathbf{0.768 \pm 0.014}$
AP						
First	0.456 ± 0.014	0.442 ± 0.006	0.496 ± 0.005	0.559 ± 0.005	0.623 ± 0.004	0.683 ± 0.004
Last	0.458 ± 0.009	0.510 ± 0.007	0.549 ± 0.006	0.598 ± 0.005	0.653 ± 0.006	0.708 ± 0.006
Random	0.460 ± 0.010	0.469 ± 0.006	0.520 ± 0.005	0.576 ± 0.005	0.640 ± 0.005	0.696 ± 0.004
Counting	$\mathbf{0.827 \pm 0.009}$	$\mathbf{0.693 \pm 0.009}$	0.677 ± 0.008	0.673 ± 0.008	0.686 ± 0.008	0.709 ± 0.007
LOCI	0.789 ± 0.010	0.686 ± 0.010	$\mathbf{0.685 \pm 0.009}$	$\mathbf{0.705 \pm 0.009}$	$\mathbf{0.750 \pm 0.009}$	$\mathbf{0.827 \pm 0.008}$

Table A3: Localization under the Random question policy (mean $\pm$ SE). Higher is better; bold marks the highest mean for each metric in each column.

	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
First + VERIFY	10.12 ± 0.48	12.28 ± 0.49	15.48 ± 0.51	18.29 ± 0.68	22.11 ± 0.72
Last + VERIFY	8.51 ± 0.38	10.69 ± 0.41	13.42 ± 0.44	16.56 ± 0.61	20.04 ± 0.70
Random + VERIFY	9.22 ± 0.41	11.48 ± 0.45	14.64 ± 0.49	17.56 ± 0.65	21.17 ± 0.74
Counting + VERIFY	7.36 ± 0.36	10.01 ± 0.42	13.25 ± 0.47	16.18 ± 0.62	20.60 ± 0.69
Restart	16.36 ± 0.93	31.87 ± 2.43	57.54 ± 3.98	91.00 ± 5.37	122.01 ± 6.07
LOCI + VERIFY	6.98 ± 0.34	9.06 ± 0.35	11.64 ± 0.40	14.49 ± 0.53	18.03 ± 0.57
LOCI + MASK	7.36 ± 0.40	12.19 ± 0.54	23.02 ± 0.83	40.07 ± 1.41	$68.26 \pm 2.29^{*}$
LOCI adaptive	$\mathbf{6.31 \pm 0.34}$	$\mathbf{7.66 \pm 0.34}$	$\mathbf{9.50 \pm 0.37}$	$\mathbf{11.69 \pm 0.47}$	$\mathbf{14.77 \pm 0.52}$

Table A4: Recovery under the Natural question policy (mean $\pm$ SE). Lower is better; bold marks the lowest mean in each column. *The MASK mean at $p = 0.5$ excludes 12 episodes that reached the action limit.

B.1.3 LARGER CANDIDATE SETS

To evaluate repair with a larger candidate set, we use the 200 bird species in CUB-200-2011 (Wah et al., 2011). We represent each species by yes/no attributes, marking an attribute as present when it occurs in at least 50% of that species' annotated examples. Removing uninformative and redundant attributes leaves 169 possible questions. Prior reliability is fixed at $\rho = 1 - p$. Verification and interaction costs follow Section 4.1.

	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
First + VERIFY	7.67 ± 0.30	9.66 ± 0.35	12.49 ± 0.38	15.83 ± 0.48	19.88 ± 0.51
Last + VERIFY	7.29 ± 0.29	9.23 ± 0.35	12.30 ± 0.39	15.51 ± 0.43	19.51 ± 0.47
Random + VERIFY	7.58 ± 0.29	9.44 ± 0.36	12.49 ± 0.38	15.50 ± 0.44	19.82 ± 0.47
Counting + VERIFY	5.88 ± 0.25	8.32 ± 0.34	11.95 ± 0.41	15.77 ± 0.46	19.83 ± 0.49
Restart	12.46 ± 0.66	21.84 ± 1.14	34.72 ± 1.70	50.47 ± 1.87	60.39 ± 1.62
LOCI + VERIFY	5.74 ± 0.25	8.03 ± 0.32	11.24 ± 0.36	14.80 ± 0.43	18.13 ± 0.45
LOCI + MASK	9.12 ± 0.52	16.81 ± 0.67	24.81 ± 0.71	32.75 ± 0.69	32.85 ± 0.84
LOCI adaptive	$\mathbf{5.38 \pm 0.24}$	$\mathbf{7.22 \pm 0.31}$	$\mathbf{9.84 \pm 0.34}$	$\mathbf{12.90 \pm 0.42}$	$\mathbf{16.99 \pm 0.46}$

Table A5: Recovery under the Random question policy (mean $\pm$ SE). Lower is better; bold marks the lowest mean in each column.

	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
First + VERIFY	5.72 ± 0.23	7.65 ± 0.25	9.68 ± 0.29	12.21 ± 0.35	16.03 ± 0.44
Last + VERIFY	5.77 ± 0.25	7.27 ± 0.26	9.33 ± 0.31	11.84 ± 0.37	15.66 ± 0.45
Random + VERIFY	5.75 ± 0.23	7.35 ± 0.23	9.35 ± 0.28	11.91 ± 0.36	15.69 ± 0.45
Counting + VERIFY	5.07 ± 0.23	6.98 ± 0.26	9.40 ± 0.33	12.22 ± 0.39	16.31 ± 0.49
Restart	8.34 ± 0.47	15.35 ± 0.85	24.17 ± 1.01	35.85 ± 1.35	48.17 ± 1.38
LOCI + VERIFY	5.14 ± 0.23	6.95 ± 0.25	9.23 ± 0.32	11.89 ± 0.37	15.44 ± 0.46
LOCI + MASK	8.38 ± 0.60	15.95 ± 0.88	23.97 ± 1.04	29.77 ± 0.98	30.42 ± 0.90
LOCI adaptive	$\mathbf{4.53 \pm 0.21}$	$\mathbf{6.19 \pm 0.23}$	$\mathbf{8.08 \pm 0.28}$	$\mathbf{10.31 \pm 0.33}$	$\mathbf{14.32 \pm 0.42}$

Table A6: Recovery under the VoI question policy (mean $\pm$ SE). Lower is better; bold marks the lowest mean in each column.

Method	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
First + VERIFY	14.71 ± 0.39	19.92 ± 0.45	25.73 ± 0.57	32.17 ± 0.67	41.09 ± 0.77
Last + VERIFY	13.72 ± 0.37	19.00 ± 0.43	24.66 ± 0.56	31.20 ± 0.63	40.48 ± 0.74
Random + VERIFY	14.42 ± 0.38	19.46 ± 0.42	25.28 ± 0.56	31.51 ± 0.62	40.86 ± 0.76
Counting + VERIFY	9.72 ± 0.30	15.92 ± 0.41	22.78 ± 0.55	30.30 ± 0.65	40.41 ± 0.79
Restart	46.62 ± 2.33	128.69 ± 6.56	337.77 ± 15.35	622.74 ± 22.75	1005.93 ± 28.17
LOCI + VERIFY	9.45 ± 0.27	14.96 ± 0.36	21.16 ± 0.50	27.71 ± 0.57	36.01 ± 0.66
LOCI + MASK	21.18 ± 0.90	65.00 ± 2.62	223.92 ± 6.56	401.02 ± 3.77	373.95 ± 7.54
LOCI adaptive	$\mathbf{9.11 \pm 0.27}$	$\mathbf{14.14 \pm 0.35}$	$\mathbf{19.29 \pm 0.47}$	$\mathbf{25.37 \pm 0.56}$	$\mathbf{34.68 \pm 0.68}$

Table A7: Recovery on CUB (mean $\pm$ SE). Lower is better; bold marks the lowest mean in each column.

Adaptive thresholds are fitted on pilot histories and held fixed for evaluation on new histories. Table A7 reports mean additional interactions conditional on initial failure.

B.2 FREE-FORM LANGUAGE-MODEL INTERACTIONS

We generate free-form conversations on the 20Q game from Hu et al. (2024) with Llama 3.1 8B playing both roles. The assistant writes its own yes/no questions and is not restricted to the predicate pool; the simulated user answers each question for the hidden target, which belongs to the fixed candidate set for each domain. Each conversation allows up to 20 questions, and a conversation counts as failed when the assistant does not identify the target within this budget.

B.2.1 LOCALIZATION RESULTS

Free-form conversations do not provide ground-truth labels for invalid updates, hence localization accuracy is hard to assess. For this, we use a separate LLM judge to construct reference labels for localization. These labels depend on the judge's accuracy and interpretation of ambiguous questions. This motivates the controlled evaluation with injected mistakes in Section 4.1 and the recovery comparison under an always-VERIFY policy in Section 4.2.1. Nevertheless, here we present some results to assess the localization quality using an LLM judge.

We use Fable 5 to separately annotate the answers for each question, given the known target. We label a turn invalid when the recorded user answer contradicts the judge's unambiguous answer. If it labels a question as ambiguous, then we do not mark it as invalid. We score the 20 failed conversations with at least one judge-labeled invalid turn; one further failure has none and is excluded. We use common reliability $\rho = 0.6$. Table A8 reports Hit@1 and AP against these reference labels.

B.2.2 LLM-BASED LOCALIZATION

We test whether the interacting language model can identify which earlier updates need repair. We compare LOCI with Llama 3.1 8B as an update selector on the same free-form conversations of

	Hit@1	AP
First	0.200 ± 0.089	0.353 ± 0.037
Last	0.450 ± 0.108	0.582 ± 0.052
Random	0.350 ± 0.106	0.446 ± 0.038
Counting	0.200 ± 0.089	0.353 ± 0.038
LOCI	$\mathbf{0.750 \pm 0.098}$	$\mathbf{0.643 \pm 0.037}$

Table A8: Localization on 20 failed conversations in the Animals domain using reference labels from an LLM judge (mean $\pm$ SE).

Section B.2.3. The LLM receives the recorded questions and answers, the candidate sets retained by each update, rejected guesses, and each update's repair status, but does not see the hidden target, reference error labels, or LOCI scores. It ranks unresolved updates by how likely they are to have excluded the target. The ranking is reused during consecutive verification checks, and a new failure triggers a new ranking. Here, the methods use the same always-VERIFY controller to ensure fairness in the comparison.

Selector	Average	Animals	Food	Objects	Places
Last	20.96 ± 1.06	20.00 ± 1.56	18.10 ± 0.85	27.00 ± 3.43	18.73 ± 1.57
Counting	21.31 ± 1.10	20.76 ± 1.90	18.24 ± 0.68	27.25 ± 3.46	19.00 ± 1.56
Llama 3.1 8B	21.19 ± 1.20	21.00 ± 2.20	18.48 ± 1.36	26.67 ± 3.49	18.64 ± 1.74
LOCI	$\mathbf{12.47 \pm 0.79}$	$\mathbf{12.48 \pm 1.59}$	$\mathbf{11.10 \pm 1.00}$	$\mathbf{16.96 \pm 2.24}$	$\mathbf{9.36 \pm 1.22}$

Table A9: Free-form recovery under different update selectors with always-VERIFY, measured by interaction score (mean $\pm$ SE). Lower is better; bold marks the lowest mean in each column.

In Table A9, LOCI has the lowest mean cost in every domain. On average it required 12.47 interactions, compared with 21.19 for Llama, a 41.1% reduction. Most of the saving comes from fewer verification checks: Llama requires 18.47 checks on average and LOCI requires 10.96. We note that, though this comparison evaluates localization through recovery cost, it also provides a proxy for error-location accuracy given the fixed policy.

The bad localization performance of Llama may be due to the fact that the LLM selector shares some of the errors that produced the failed interaction. In fact, the same model constructs the candidate-set updates and later tries to diagnose them, so a mistaken belief or interpretation may persist during diagnosis. If the model failed during the conversation, in principle, it may also be unable to diagnose that failure. This motivates the external repair controller approach that we propose in this paper. In fact, we see LOCI performing better. Moreover, we note that a stronger or different model may improve diagnosis, and providing it with LOCI scores may combine the benefits of both approaches. These possibilities remain untested.

B.2.3 RECOVERY RESULTS

We follow the free-form setup in Section 4.2.1. Table 1 uses the failed conversations from the Animals, Food, Objects, and Places domains. Verification returns a reliable answer to whether the target belongs to the selected constraint's candidate set. Masking removes the corresponding exchange from the assistant's context; the simulated user retains the full history. Adaptive thresholds are selected from $\gamma_{\text{keep}} \in \{0.1, 0.2, 0.4\}$ and $\gamma_{\text{mask}} \in \{0.5, 2/3, 0.9\}$, using the target folds described in Section 4.2.1.

Table 1 reports recovery costs, and Figure A1 separates them into questions, verification checks, and rejected guesses. Verification accounts for 90% of the always-VERIFY cost in free-form conversations, compared with 42% in the controlled setting. Adaptive repair saves 7.17 checks on average while adding 3.34 questions and rejected guesses, yielding a lower total cost. Compared with always-MASK, it requires fewer questions and rejected guesses, although it incurs verification costs. Uncertainty conditions on the selected thresholds and one continuation seed. Restart uses earlier runs on the same failed conversations.

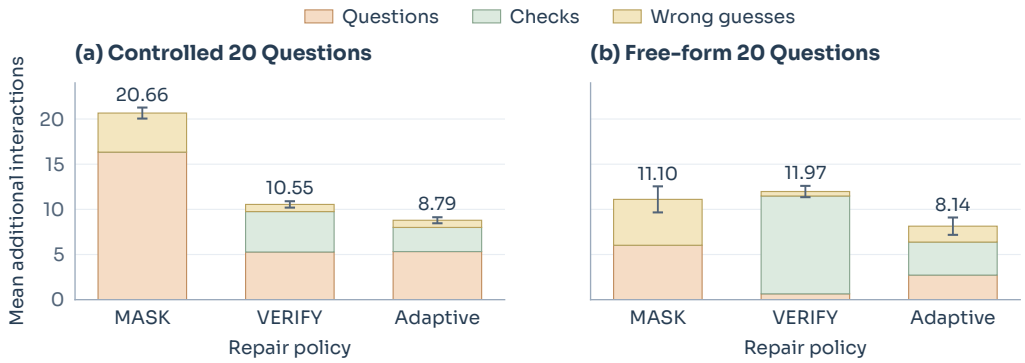

Figure A1: **Recovery cost under different LOCI repair policies.** Stacked bars show additional questions, verification checks, and rejected guesses for always-MASK, always-VERIFY, and adaptive repair in (a) controlled 20Q and (b) free-form conversations. Means weight the four domains equally.

For robustness of our result, we repeat the recovery evaluation with Qwen3-8B (thinking disabled). The user retains the conversation history. We use the same four domains, reliability $\rho = 0.9$, interaction costs, and budget of 100. Each model generates its own failure cohort, so we compare methods within each model.

Method	Average	Animals	Food	Objects	Places
First + VERIFY	21.22 ± 0.84	21.39 ± 1.57	20.45 ± 1.66	25.30 ± 1.50	17.71 ± 1.88
Last + VERIFY	20.91 ± 1.13	19.00 ± 1.48	18.91 ± 1.36	24.52 ± 1.69	21.21 ± 3.56
Random + VERIFY	22.17 ± 1.03	21.22 ± 1.36	19.95 ± 1.75	25.94 ± 1.64	21.57 ± 2.98
Counting + VERIFY	20.18 ± 1.03	20.61 ± 1.45	19.23 ± 1.93	24.52 ± 1.47	16.36 ± 2.92
Restart	47.57 ± 3.36	43.11 ± 7.03	43.18 ± 6.55	70.79 ± 6.46	33.21 ± 7.04
LOCI + VERIFY	13.26 ± 0.87	12.78 ± 1.48	9.86 ± 1.21	20.12 ± 2.08	10.29 ± 1.92
LOCI + MASK	16.23 ± 1.75	15.39 ± 3.82	10.14 ± 2.38	28.91 ± 4.56	10.50 ± 3.14
LOCI adaptive	$\mathbf{9.86 \pm 1.03}$	$\mathbf{6.94 \pm 1.33}$	$\mathbf{6.27 \pm 1.08}$	$\mathbf{18.15 \pm 3.07}$	$\mathbf{8.07 \pm 2.14}$

Table A10: Free-form recovery with Qwen3-8B, measured by interaction score (mean $\pm$ SE). Lower is better; bold marks the lowest mean in each column. Capped recoveries receive score 100.

Table A10 shows that LOCI with always-VERIFY reduces mean recovery cost by 34.3% relative to Counting with the same action. Adaptive repair has the lowest mean cost in each domain and reduces the equal-domain mean by 51.1% relative to Counting. All always-VERIFY methods and adaptive LOCI recover all 87 failures; always-MASK recovers 86 and Restart recovers 69. The remaining runs reach the interaction cap and remain in the reported means.

Table A11: Localization on MultiWOZ (mean ± SE). Higher is better; bold marks the highest mean for each metric in each column. The Average column weights the five error rates equally.

	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$	Average
Failed draws	247	427	600	740	844	2,858
Mean T	2.58	2.56	2.55	2.51	2.49	2.54
Mean K	1.13	1.15	1.31	1.36	1.53	1.29
Hit@1						
First	0.457 ± 0.032	0.518 ± 0.023	0.523 ± 0.021	0.545 ± 0.019	0.609 ± 0.016	0.530 ± 0.011
Last	0.449 ± 0.032	0.436 ± 0.023	0.520 ± 0.021	0.549 ± 0.019	0.614 ± 0.018	0.513 ± 0.012
Random	0.459 ± 0.010	0.473 ± 0.009	0.529 ± 0.009	0.556 ± 0.008	0.625 ± 0.009	0.528 ± 0.005
Counting	0.608 ± 0.031	0.630 ± 0.022	0.647 ± 0.020	0.668 ± 0.018	0.723 ± 0.015	0.655 ± 0.012
LOCI	$\mathbf{0.673 \pm 0.030}$	$\mathbf{0.730 \pm 0.022}$	$\mathbf{0.772 \pm 0.018}$	$\mathbf{0.792 \pm 0.016}$	$\mathbf{0.825 \pm 0.013}$	$\mathbf{0.758 \pm 0.011}$
AP						
First	0.700 ± 0.018	0.738 ± 0.013	0.742 ± 0.011	0.753 ± 0.010	0.793 ± 0.008	0.745 ± 0.006
Last	0.699 ± 0.018	0.688 ± 0.013	0.737 ± 0.012	0.756 ± 0.011	0.793 ± 0.009	0.734 ± 0.007
Random	0.701 ± 0.007	0.709 ± 0.006	0.741 ± 0.006	0.757 ± 0.005	0.796 ± 0.005	0.741 ± 0.004
Counting	0.795 ± 0.016	0.803 ± 0.012	0.811 ± 0.010	0.824 ± 0.009	0.851 ± 0.008	0.817 ± 0.006
LOCI	$\mathbf{0.833 \pm 0.015}$	$\mathbf{0.856 \pm 0.011}$	$\mathbf{0.877 \pm 0.009}$	$\mathbf{0.888 \pm 0.008}$	$\mathbf{0.901 \pm 0.007}$	$\mathbf{0.871 \pm 0.006}$

B.3 TASK-ORIENTED DIALOGUE

B.3.1 SETUP

We use the hotel, restaurant, and attraction search requirements in MultiWOZ 2.4. Each slot is a named requirement, such as hotel area or price range, and each candidate intent $x \in \mathcal{X}$ assigns values to the specified slots; unspecified or unrestricted slots do not impose any constraints. The set of specified slots is known. We construct $\mathcal{X}$ by combining configurations that are supported by the database across domains. Each exchange contributes one constraint containing its newly specified values, e.g. "a cheap hotel in the east" gives `hotel.price = cheap` and `hotel.area = east`. We stop each dialogue before the first revision or removal of a requirement to preserve the fixed-target assumption. We retain dialogues with at least two requirement-bearing turns so that localization involves a choice among updates. These restrictions retain 346 test dialogues.

Each original update is corrupted independently with probability p. For a corrupted update, we choose one of its slots uniformly and replace its value with a uniformly chosen alternative from that slot's database vocabulary. Let k_i be the number of slots in update i and m_s the vocabulary size of slot s. For a candidate x incompatible with the recorded update, the error likelihood is

$$q_i(x) = \begin{cases} 1/[k_i(m_s - 1)], & \text{if only slot } s \text{ differs,} \\ 0, & \text{if multiple slots differ.} \end{cases}$$

We use prior reliabilities $\rho_i = 1 - p$ and a uniform prior over $\mathcal{X}$, conditioning on rejection of the recorded requirement configuration.

We follow the controlled evaluation in Section 4.1, injecting errors independently with probability $p \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and using reliability $\rho_i = 1 - p$ for inference. The errors change slot values rather than the binary answers, as described in the previous paragraph.

B.3.2 LOCALIZATION RESULTS

We evaluate localization before repair using Hit@1 and AP on the same failed histories used for recovery. Table A11 reports the results. LOCI consistently outperforms the baselines across the different error-rate scenarios.

B.3.3 RECOVERY RESULTS

Recovery follows the repair policies and interaction costs described in Section 3.3. We select adaptive thresholds on development dialogues to minimize recovery cost, then fix them for test evaluation. This threshold search followed inspection of test outcomes. Table A12 reports recovery costs on the same histories used for localization. All methods recover every history within the interaction budget.

Method	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$	Average
Failed draws	247	427	600	740	844	2,858
First + VERIFY	3.07 ± 0.09	3.10 ± 0.08	3.56 ± 0.08	3.68 ± 0.08	4.11 ± 0.09	3.51 ± 0.05
Last + VERIFY	3.07 ± 0.11	3.23 ± 0.08	3.63 ± 0.08	3.68 ± 0.08	4.11 ± 0.09	3.54 ± 0.06
Random + VERIFY	3.12 ± 0.10	3.13 ± 0.07	3.59 ± 0.08	3.71 ± 0.08	4.12 ± 0.09	3.53 ± 0.06
Counting + VERIFY	2.81 ± 0.08	2.90 ± 0.07	3.40 ± 0.08	3.50 ± 0.08	4.01 ± 0.09	3.33 ± 0.05
Restart	3.15 ± 0.10	3.15 ± 0.08	3.18 ± 0.07	3.18 ± 0.07	3.19 ± 0.07	3.17 ± 0.06
LOCI + VERIFY	2.70 ± 0.09	2.80 ± 0.08	3.29 ± 0.08	3.38 ± 0.08	3.92 ± 0.09	3.22 ± 0.05
LOCI + MASK	$\mathbf{1.86 \pm 0.09}$	$\mathbf{1.88 \pm 0.08}$	$\mathbf{2.15 \pm 0.07}$	$\mathbf{2.21 \pm 0.08}$	$\mathbf{2.56 \pm 0.08}$	$\mathbf{2.13 \pm 0.05}$
LOCI adaptive	2.16 ± 0.09	2.31 ± 0.08	2.76 ± 0.08	2.80 ± 0.07	3.11 ± 0.07	2.63 ± 0.06

Table A12: Recovery on MultiWOZ, measured by interaction score (mean $\pm$ SE). Lower is better; bold marks the lowest mean in each column. The Average column weights the five error rates equally.

LOCI beats the baselines across all the error rates, with always-MASK achieving the lowest mean cost. Table A13 shows that adaptive repair saves questions and wrong guesses, but these savings are smaller than its verification costs. This is because reliable slot answers may make discarded information relatively cheap to recover here. Always-MASK also resumes questioning once candidates remain, while adaptive repair can continue checking updates. This suggests that the value of verification depends on both corruption confidence and the cost of recovering information through the subsequent turns of the conversation.

Policy	Questions	Verifications	Wrong guesses	Total
Always-VERIFY	1.29 ± 0.03	1.59 ± 0.02	0.34 ± 0.01	3.22 ± 0.05
Always-MASK	1.66 ± 0.04	0.00 ± 0.00	0.47 ± 0.02	$\mathbf{2.13 \pm 0.05}$
Adaptive	1.45 ± 0.03	0.94 ± 0.03	0.24 ± 0.01	2.63 ± 0.06

Table A13: Recovery-cost components for LOCI (mean $\pm$ source-dialogue-cluster bootstrap SE). Means weight error rates equally.

B.4 CANDIDATE INTENT SPACES

LOCI attributes errors by averaging over possible user intents. The attribution rule uses a posterior over these candidates and their compatibility with each recorded update. In our experiments, we use finite candidate spaces containing the target, which permit exact computation of the scores under the specified model. For tasks where the candidate space is unknown, retrieval or generation could construct a working set during the interaction.

Open-set UoT provides a precedent for this approach: an LLM proposes a finite candidate set using the conversation history and refreshes it after each interaction (Hu et al., 2024). An analogous extension for LOCI would propose alternative intents using the original exchanges and rejection feedback, evaluate the recorded constraints on these candidates, and assign weights approximating the intent posterior. The same attribution rule would then apply to this representation. Candidate generation must allow alternatives that contradict the assistant's recorded interpretations, since these alternatives explain how an earlier update could have excluded the target.

Localization can also precede identification of the exact intent. By Equation (1), candidates with the same pattern of compatibility across recorded updates can have their posterior probabilities combined without changing any corruption score. For example, several flights departing after 6 p.m. can all provide evidence against an incorrectly recorded requirement to depart before 6 p.m., even while the intended flight remains uncertain. A candidate representation that preserves the posterior mass assigned to each compatibility pattern therefore preserves the localization scores. Constructing such a representation requires accounting for the weights of the alternatives it represents.

This suggests a way to combine LOCI with candidate generation in tasks without a predefined intent catalogue. The quality of the resulting scores would depend on candidate coverage, weighting, and accurate constraint evaluation. This is left for future work.

B.5 Ablations

B.5.1 Sensitivity to Assumed Reliability

The update reliability ρ in Equation (2) may be unknown in practice. We vary the common assumed reliability over $\{0.5, 0.7, 0.9, 0.95\}$ while holding the initial histories fixed. We test its effect on controlled localization and on free-form recovery with fixed repair thresholds. For localization, we reuse the Natural-policy histories and scoring protocol of Section B.1.1, with only the assumed reliability changing. Table A14 includes Counting and the nominal reference $\rho = 1 - p$.

	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$	$p = 0.5$
Scored failures	565	822	970	1,038	1,066
Hit@1					
Counting	0.500 ± 0.023	0.530 ± 0.022	0.564 ± 0.021	0.628 ± 0.022	0.628 ± 0.019
LOCI, $\rho = 1 - p$	0.509 ± 0.024	0.528 ± 0.021	0.586 ± 0.020	0.674 ± 0.021	0.744 ± 0.021
LOCI, $\rho = 0.5$	0.462 ± 0.026	0.534 ± 0.023	0.611 ± 0.022	0.710 ± 0.020	0.744 ± 0.020
LOCI, $\rho = 0.7$	0.491 ± 0.027	0.532 ± 0.022	0.586 ± 0.021	0.660 ± 0.020	0.672 ± 0.020
LOCI, $\rho = 0.9$	0.509 ± 0.026	0.535 ± 0.022	0.574 ± 0.021	0.634 ± 0.019	0.643 ± 0.019
LOCI, $\rho = 0.95$	0.509 ± 0.026	0.535 ± 0.022	0.576 ± 0.021	0.631 ± 0.020	0.640 ± 0.019
AP					
Counting	0.682 ± 0.015	0.686 ± 0.014	0.702 ± 0.013	0.748 ± 0.012	0.760 ± 0.010
LOCI, $\rho = 1 - p$	0.693 ± 0.016	0.685 ± 0.014	0.701 ± 0.014	0.755 ± 0.013	0.808 ± 0.011
LOCI, $\rho = 0.5$	0.609 ± 0.019	0.660 ± 0.016	0.710 ± 0.015	0.776 ± 0.012	0.808 ± 0.011
LOCI, $\rho = 0.7$	0.650 ± 0.018	0.672 ± 0.015	0.701 ± 0.014	0.752 ± 0.013	0.769 ± 0.012
LOCI, $\rho = 0.9$	0.694 ± 0.017	0.696 ± 0.014	0.710 ± 0.014	0.749 ± 0.012	0.760 ± 0.011
LOCI, $\rho = 0.95$	0.694 ± 0.017	0.697 ± 0.014	0.711 ± 0.014	0.749 ± 0.012	0.759 ± 0.011

Table A14: Controlled localization under different assumed reliabilities (mean $\pm$ SE). Higher is better. Each column uses the same failed histories for every row.

Localization depends on the assumed reliability. At $p = 0.1$, reducing ρ from 0.9 to 0.5 lowers AP from 0.694 to 0.609. At $p = 0.5$, increasing ρ from 0.5 to 0.9 lowers Hit@1 from 0.744 to 0.643 and AP from 0.808 to 0.760. The latter AP matches Counting to three decimals. No fixed reliability performs best throughout the grid.

The likelihood explains why high reliability can hurt when many updates are invalid. With binary constraints and $\kappa_i = 1$, each additional violated constraint multiplies a candidate's relative likelihood by $(1 - \rho)/\rho$. At $\rho = 0.9$, two additional violations incur an 81-fold penalty. A true target excluded by several corrupted updates can therefore receive little posterior mass, reducing its contribution to their corruption scores. Lower reliability weakens this penalty, but also weakens the information supplied by valid updates.

For recovery, we reuse the failed Llama 3.1 8B conversations and recorded candidate classifications from Section B.2.3, with the simulated user retaining its history. We rerun LOCI + `VERIFY` and adaptive repair at each reliability. The adaptive thresholds selected on other target folds at $\rho = 0.9$ remain fixed throughout. Results are in Table A15.

Mean always-`VERIFY` cost ranges from 12.17 to 12.32 interactions, and adaptive cost from 8.36 to 9.79. Every paired 95% bootstrap interval for a cost change relative to $\rho = 0.9$ includes zero, so these data establish neither a cost difference nor equivalence. With always-`VERIFY`, changing the order of checks need not change their number. Changing ρ from 0.9 to 0.5 changes the first selected update in $> 50\%$ histories, but every tested reliability eventually checks the same initially invalid updates. The mean number of checks of initially valid updates also stays between 1.56 and 1.72. Adaptive repair uses the scores to choose whether to verify or mask, so changing ρ also changes its actions.

B.5.2 Controller Ablations

Table A16 compares evidence retention and repair stopping rules on the same Natural-policy histories. Adaptive retain keeps masked updates as evidence for posterior inference; adaptive forget discards them. Feasibility-stop retains this evidence and resumes questioning when masking leaves a nonempty

Assumed reliability	LOCI + VERIFY	LOCI adaptive
0.5	12.17 ± 0.64	9.79 ± 0.82
0.7	12.27 ± 0.69	9.01 ± 0.76
0.9	12.32 ± 0.68	8.36 ± 0.98
0.95	12.32 ± 0.68	8.86 ± 1.02

Table A15: Free-form recovery under different assumed reliabilities, measured by additional interactions (mean $\pm$ target-cluster bootstrap SE). Means weight domains equally.

Controller	$p = 0.1$	$p = 0.2$	$p = 0.3$	$p = 0.4$
Adaptive retain	6.31 ± 0.34	7.66 ± 0.34	9.50 ± 0.37	11.69 ± 0.47
Adaptive forget	6.35 ± 0.34	7.75 ± 0.33	9.51 ± 0.37	11.64 ± 0.47
Feasibility-stop	6.30 ± 0.34	7.61 ± 0.34	9.50 ± 0.37	11.61 ± 0.47

Table A16: Recovery under the Natural question policy with different evidence retention and stopping rules (mean $\pm$ SE).

candidate set or verification confirms an update. Each variant uses thresholds fitted separately on the same target folds. Mean recovery costs are similar across the three variants at each error rate.

B.5.3 EFFECT OF USER CONVERSATION HISTORY

Table A17 compares recovery with and without user history. With history, the simulated user sees the conversation; without it, the user sees only the target and current question. The assistant retains its unmasked conversation in both settings. LOCI adaptive has the lowest mean recovery cost in all four domains with user history and in two of four without it.

Method	Animals	Food	Objects	Places
User with conversation history				
First + VERIFY	22.29 ± 1.79	20.62 ± 1.38	25.33 ± 1.76	20.27 ± 1.77
Last + VERIFY	19.71 ± 1.28	18.57 ± 1.12	23.67 ± 1.62	18.73 ± 1.57
Random + VERIFY	20.67 ± 0.94	19.95 ± 0.94	26.00 ± 1.97	20.91 ± 1.50
Counting + VERIFY	21.52 ± 2.34	18.62 ± 0.90	24.75 ± 2.02	19.00 ± 1.56
Restart	43.67 ± 5.90	46.71 ± 6.88	56.88 ± 7.71	20.09 ± 6.47
LOCI + VERIFY	12.05 ± 1.34	11.19 ± 1.03	15.29 ± 1.41	9.36 ± 1.22
LOCI + MASK	12.52 ± 3.32	10.48 ± 2.64	15.42 ± 3.08	6.00 ± 2.23
LOCI adaptive	$\mathbf{10.62 \pm 2.50}$	$\mathbf{7.10 \pm 1.40}$	$\mathbf{10.83 \pm 2.13}$	$\mathbf{4.00 \pm 1.48}$
User without conversation history				
First + VERIFY	17.52 ± 0.96	16.86 ± 1.83	18.68 ± 1.06	34.00 ± 10.49
Last + VERIFY	16.00 ± 0.80	18.43 ± 1.31	18.95 ± 1.12	36.00 ± 10.00
Random + VERIFY	17.29 ± 0.85	18.93 ± 1.17	18.11 ± 0.96	34.20 ± 10.36
Counting + VERIFY	15.38 ± 1.13	15.07 ± 1.51	14.68 ± 1.14	29.10 ± 10.23
Restart	47.76 ± 7.58	43.29 ± 7.06	48.42 ± 9.25	33.10 ± 10.46
LOCI + VERIFY	7.10 ± 0.79	7.29 ± 1.22	$\mathbf{7.42 \pm 1.08}$	19.80 ± 9.49
LOCI + MASK	7.14 ± 2.20	4.36 ± 1.77	15.58 ± 4.08	$\mathbf{17.30 \pm 9.30}$
LOCI adaptive	$\mathbf{4.38 \pm 1.54}$	$\mathbf{3.93 \pm 1.30}$	8.21 ± 1.61	17.90 ± 9.84

Table A17: Recovery on LLM-generated conversations with and without user history, measured by interaction score (mean $\pm$ SE). Lower is better; bold marks the lowest mean in each column within each history setting.