Intent Mismatch Causes LLMs to Get Lost in Multi-Turn Conversation

Anonymous ACL submission

Abstract

Multi-turn conversation has emerged as the predominant interaction paradigm for Large Language Models (LLMs). Recent work reveals that LLMs suffer a ~30% performance drop in multi-turn settings compared to single-turn interactions, termed "Lost in Conversation" (LiC). We demonstrate that LiC is not solely a failure of model capability, but rather a breakdown in user-model interaction caused by an *intent alignment gap*. We analyze why scaling model size alone is unlikely to fully close this gap when user-specific signals are absent, since the gap arises from structural ambiguity in conversational context rather than only from representational limitations. As an empirical test of this explanation, we introduce a lightweight Mediator-Assistant architecture that decouples intent understanding from task execution. Experimental results across diverse LLMs show that our approach recovers about 60% of the performance lost in underspecified multi-turn interactions.

1 Introduction

Multi-turn dialogue has become the primary mode of interaction between users and large language models (LLMs). While LLMs perform remarkably well with clear, complete instructions in a single turn, real-world user behavior rarely conforms to this idealized setting. In practice, users frequently start with vague, underspecified, or even internally inconsistent goals, and only gradually clarify and refine their true needs through an iterative conversational process with the model (Zamfirescu-Pereira et al., 2023; Min et al., 2020). This incremental, exploratory nature of human problem formulation poses substantially greater challenges for LLMs than standard single-turn benchmarks: the model must not only understand and solve the current subtask, but also continually infer, update, and realign with a moving target of user intent across turns.

Recent research (Laban et al., 2025) presents a set of controlled experiments designed to simulate the instruction underspecification that frequently occurs in human conversation (Herlihy et al., 2024; Chao and Zipf, 1949). The study systematically compares performance under "single-turn, fully specified" (Full) versus "multi-turn, underspecified" (Sharded) interactions, revealing a substantial performance degradation of approximately 30% for all evaluated LLMs. The authors argue that under incomplete information, LLMs tend to make premature assumptions early in the dialogue and subsequently "lock in" these assumptions, causing the final responses to drift away from the user's true intent. They term this phenomenon "Lost in Conversation" (LiC) and primarily attribute it to the reduced reliability of LLMs in multi-turn dialogue. On this basis, they advocate that model builders should jointly optimize models' aptitude and reliability in iterative conversational settings. However, we argue that this attribution is incomplete: the observed reliability drop is better understood as a symptom of a deeper intent alignment gap between user expression and model interpretation. Without user-specific signals, scaling model capacity alone may improve absolute performance but is unlikely to eliminate this gap.

In this work, we revisit this phenomenon and offer a different explanatory perspective. We argue that: (1) **Making early assumptions and providing tentative answers is not simply erroneous behavior, but a rational strategy induced by the dominant training objective of being helpful** (Ouyang et al., 2022) and the penalty often associated with evasive responses in RLHF pipelines. Under conditions of incomplete information, the model is inclined to construct a plausible task formulation for a typical user and produce a provisional answer based on that formulation, instead of repeatedly refusing to answer or endlessly requesting additional information. (2) **The primary**

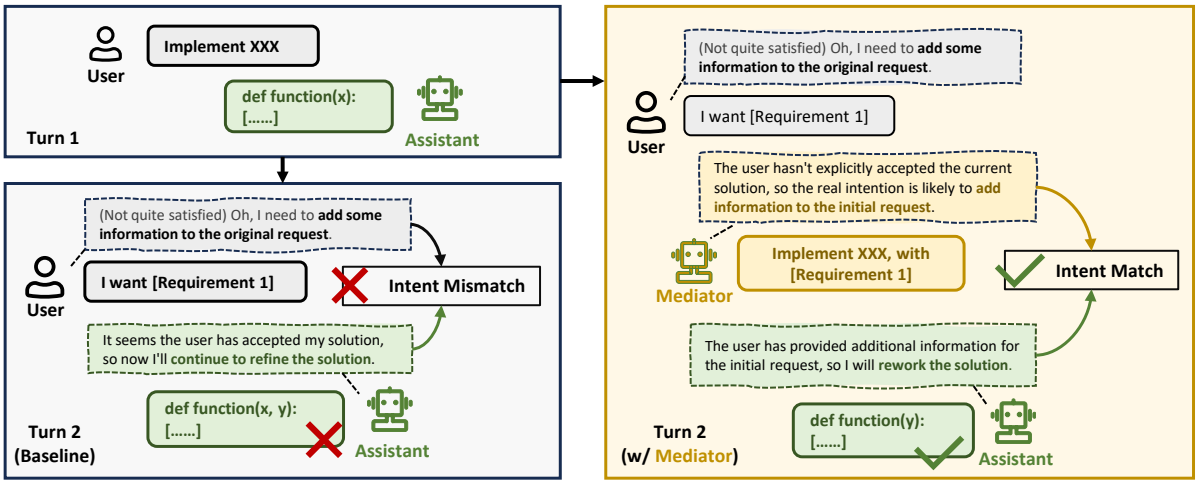

Figure 1: **Intent Mismatch in Multi-turn Dialogue.** (Left) The LiC benchmark simulates passive users who act as "lazy" interlocutors, omitting corrections for erroneous model assumptions. This behavior causes the Assistant's interpretation to progressively drift away from the user's true intent, leading to significant performance degradation. (Right) Our approach introduces a Mediator to bridge this interpretive gap by fundamentally decoupling intent inference from task execution. The Mediator aligns the Assistant with the user's true goals, effectively mitigating performance degradation.

bottleneck in failed multi-turn conversations is not a lack of model capacity or reasoning depth, but an intent alignment gap between user expression and model interpretation (Figure 1 left). Ambiguous utterances often admit multiple plausible interpretations, and the intended interpretation depends on the user's communicative style. The LiC benchmark simulates a particular style of user: a passive interlocutor who reveals instructions fragment by fragment and rarely corrects erroneous model assumptions. General-purpose LLMs, aligned to broad population-level priors, may fail to adapt to this simulated user style.

To validate this explanation, we introduce a Mediator-Assistant architecture that decouples intent reconstruction from task execution. The Mediator maps ambiguous dialogue contexts into explicit task instructions for the Assistant, reducing intent mismatch without updating the Assistant's weights (Figure 1 right). Across diverse LLMs, this intervention substantially recovers multi-turn performance, highlighting the importance of history-aware intent modeling for LiC. Our main contributions are:

- We re-interpret the Lost in Conversation phenomenon as an intent alignment failure: models often fail not because they cannot execute the task, but because they infer the wrong task from fragmented interaction.
- We provide an analytical account of why scaling model size alone may be insufficient under ambiguous contexts without user-specific signals.
- We instantiate this view with a Mediator-Assistant architecture that decouples intent understanding from task execution, demonstrating an average recovery of about 60% of the lost performance across diverse LLMs.

2 Related Works

Multi-turn Dialogue Evaluation. Recent benchmarks for multi-turn dialogue, such as MT-Bench (Zheng et al., 2023), MT-Bench-101 (Bai et al., 2024), and LOCOMO (Maharana et al., 2024), evaluate models in settings that require sequential reasoning, instruction following, or long-context retention. These benchmarks often assume that the user's intent is sufficiently specified within each turn or recoverable from the dialogue history. In contrast, our work focuses on *incremental intent revelation*, where the user's goal is expressed through fragmented follow-ups and may contradict earlier model assumptions. This setting is systematically studied in the "Lost in Conversation" (LiC) benchmark (Laban et al., 2025), which serves as the basis of our analysis.

Intent Understanding and Clarification. Inferring user intent from ambiguous or underspecified utterances has long been a central problem in dialogue systems (Traum and Larsson, 2003). One natural solution is clarification: recent work encourages LLMs to ask follow-up questions when

queries are ambiguous (Li, 2025; Herlihy et al., 2024). However, clarification shifts part of the burden back to the user and conflicts with the LiC setting, where users reveal constraints incrementally and often do not explicitly correct mistaken assumptions. Instead of requiring users to reformulate their intent, our approach resolves ambiguity upstream by rewriting the dialogue context into an explicit instruction.

Personalization and Memory in LLMs. LLM personalization has been explored through PEFT (Xu et al., 2023), user-specific adapters (Zhong et al., 2021), and memory-augmented systems such as Mem0 (Chhikara et al., 2025), A-Mem (Xu et al., 2025), and MemoryBank (Zhong et al., 2024). These methods primarily address factual personalization, such as remembering user preferences, past facts, or long-term context. Our focus is different: we study intent alignment, where historical interaction signals are used to interpret ambiguous utterances according to a recurring communicative style. In this sense, memory provides evidence, but the key challenge is deciding how that evidence should change the interpretation of the current turn.

3 Problem Analysis

We formulate multi-turn interaction between a user and an LLM assistant through a latent-intent perspective. Our goal is not to provide a formal impossibility proof, but to introduce an analytical lens that separates intent inference from task execution and clarifies where multi-turn degradation can arise.

Let $I_t \in \mathcal{I}$ denote the user's underlying intent at turn t, and let $T \in \mathcal{T}$ represent the user's expression habits and communicative patterns. At the t-th turn, the input to the LLM is the accumulated dialogue context C_t, consisting of previous user utterances and assistant responses:

$$C_t = (u_1, a_1, u_2, a_2, \ldots, u_t). \tag{1}$$

The user's current utterance u_t can be viewed as a surface realization of the underlying intent:

$$u_t \sim P_{\text{user}}(u \mid I_t, T, C_{t-1}). \tag{2}$$

This realization process is often lossy: complex intents are expressed through short, partial, or context-dependent utterances. As a result, the same surface form may correspond to different intents for different users or even for the same user at different moments.

The LLM, parameterized by θ, generates a response conditioned on the observed context:

$$R \sim P_\theta(R \mid C_t). \tag{3}$$

The key question is whether the observed context C_t contains enough evidence for the model to infer the intended task before executing it.

3.1 Performance Decomposition

We view multi-turn performance through two interacting components: *intent inference*, the ability to recover the user's intended task from the dialogue context, and *task execution*, the ability to solve the recovered task. This is an analytical abstraction rather than a claim that the two processes are fully independent inside an LLM.

For exposition, suppose that once the intended task I_t is specified, the remaining dialogue context C_t contributes little additional information for producing the correct response. Under this simplifying assumption, the response distribution can be written as:

$$P_\theta(R \mid C_t) = \sum_{I_t \in \mathcal{I}} \underbrace{P_\theta(R \mid I_t)}_{\text{Execution}} \cdot \underbrace{P_\theta(I_t \mid C_t)}_{\text{Inference}}. \tag{4}$$

This decomposition highlights a plausible mechanism behind performance degradation in multi-turn dialogue. The execution term $P_\theta(R \mid I_t)$ corresponds to the model's ability to solve a well-specified task. This capability is largely reflected by single-turn performance, where the user's intent is explicitly stated. In contrast, the inference term $P_\theta(I_t \mid C_t)$ captures the model's ability to identify the intended task from an ambiguous and evolving dialogue context.

In an ideal single-turn setting, users tend to provide self-contained instructions, making I_t relatively easy to infer. In multi-turn dialogue, however, users often communicate through fragments, references, corrections, and implicit constraints. Such behavior is natural under the principle of least effort in human communication, but it increases the burden on the model: the model must infer not only what has been said, but also how each new utterance should revise or constrain the previous context.

From this perspective, the "Lost in Conversation" phenomenon need not indicate that the model has lost the ability to solve the task. Rather, the model may fail because it has inferred the wrong task. Once the inferred intent diverges from the user's true intent, even a capable execution model can produce an incorrect response.

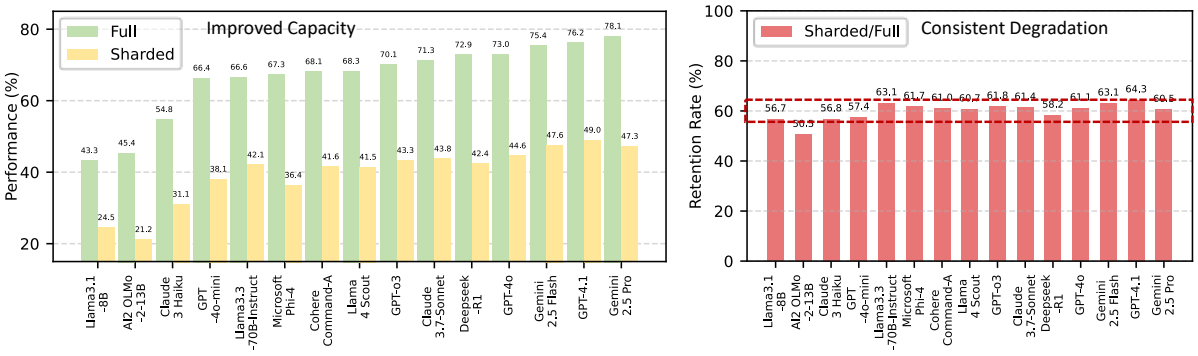

Figure 2: **Performance comparison across different LLMs on the LiC benchmark (Laban et al., 2025).** While absolute performance improves with model scale, the Sharded-to-Full performance ratio remains relatively stable (~60%). This relatively stable degradation is consistent with the hypothesis that alignment priors contribute to the bottleneck beyond model capacity.

3.2 The Information Bottleneck

The challenge of maximizing $P_\theta(I_t \mid C_t)$ is not merely a matter of model capacity; it also reflects an information bottleneck in the observed context. When critical constraints are omitted by the user, the current dialogue may not contain enough evidence to uniquely identify the intended task.

Under such uncertainty, a general-purpose LLM tends to rely on its learned priors. That is, when multiple intents are compatible with the same context, the model is likely to favor the intent that is most probable under its training and alignment distribution:

$$\arg\max_{I_t} P_\theta(I_t \mid C_t). \tag{5}$$

This prior may be reasonable for an average user, but it can be misaligned with a specific user's communicative habits. For example, a fragmentary continuation may be interpreted as confirmation of a previous assumption, whereas the user may intend it as a correction or refinement.

This interpretation is consistent with the empirical pattern in Figure 2. A re-analysis of results from Laban et al. (2025) shows that stronger models achieve higher absolute scores, but the relative degradation between fully specified and underspecified settings remains relatively stable across model families. This suggests that model scale improves execution and broad contextual competence, but does not by itself eliminate ambiguity when user-specific evidence is absent.

We hypothesize that this relatively stable degradation partly reflects the homogeneity of alignment priors across models. Although models differ in capacity, they are often trained and aligned on broadly similar data distributions, encouraging them to converge toward similar assumptions about typical user behavior. In ambiguous contexts, these shared priors may dominate intent inference:

$$\arg\max_{I_t} P_\theta(I_t \mid C_t) \approx \arg\max_{I_t} P_{\text{avg}}(I_t \mid C_t). \tag{6}$$

When the user's intended interpretation deviates from this average prior, increasing model capacity alone may improve the quality of the response to the inferred task, but it does not necessarily recover the intended task itself.

3.3 Reducing Ambiguity via History

The decomposition in Eq. (4) suggests that improving multi-turn performance requires sharpening the intent inference term $P_\theta(I_t \mid C_t)$. Since the current context C_t may be ambiguous, we introduce an auxiliary source of evidence: the user's generalized interaction history, denoted as $\mathcal{H}$.

We assume that while C_t may be insufficient on its own, the extended context $(C_t, \mathcal{H})$ can contain useful evidence about the user's communicative habits, prior corrections, and reformulation patterns. This assumption is essential: if neither the current context nor the history contains signals about the user's intended interpretation, no mediator can reliably recover the missing intent. However, when such signals are available, $\mathcal{H}$ can help disambiguate how the current utterance should be understood.

Our method introduces a Mediator M to approximate this inference process. Instead of asking the Assistant to respond directly to the ambiguous dialogue context, the Mediator reconstructs a more explicit instruction:

$$\hat{U} \sim P(U \mid C_t, \mathcal{H}). \tag{7}$$

Here, $\hat{U}$ is a clarified instruction that serves as a proxy for the user's latent intent I_t. The

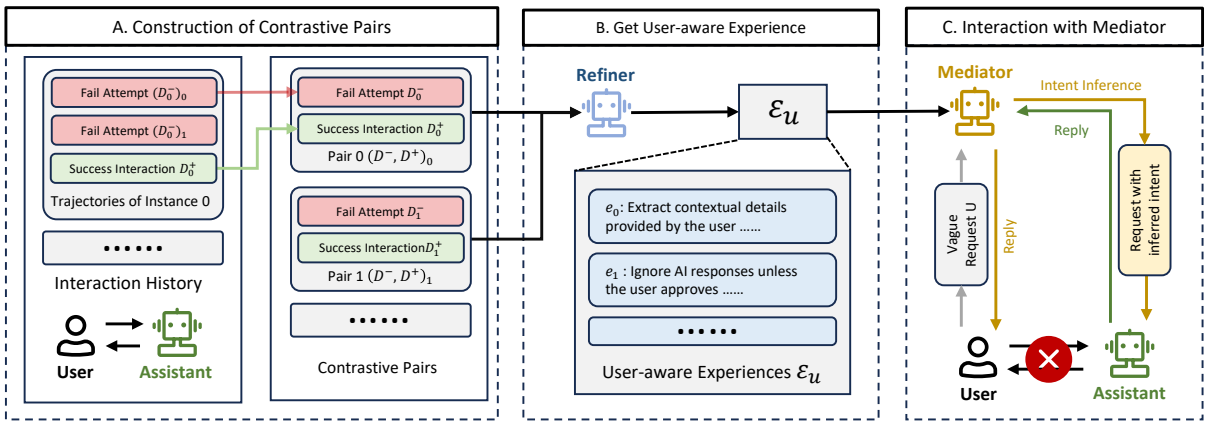

Figure 3: **Pipeline of the Mediator Framework.** We construct contrastive pairs by extracting a failed conversational trajectory D^- and the corresponding successful trajectory D^+ from held-out historical interactions. The Refiner distills these pairs into explicit communicative experiences $\mathcal{E}$, which guide the Mediator to explicate ambiguous user contexts into precise instructions for the Assistant. The Mediator operates as a transparent alignment layer, decoupling the user from the raw execution model while maintaining a seamless interaction flow.

downstream Assistant then operates on this lower-ambiguity input:

$$R \sim P_\theta(R \mid \hat{U}). \tag{8}$$

By conditioning on interaction history, the Mediator aims to reduce uncertainty in intent inference:

$$H(I_t \mid C_t, \mathcal{H}) < H(I_t \mid C_t), \tag{9}$$

when $\mathcal{H}$ contains relevant user-specific evidence. This formulation motivates our Mediator-Assistant architecture: the Mediator focuses on reconstructing the intended task, while the Assistant focuses on executing the clarified instruction.

4 Method

Building upon the analysis in §3, we introduce the Mediator-Assistant framework as an empirical test of our central hypothesis: if LiC is primarily driven by intent inference failure, then explicitly reconstructing the intended instruction before task execution should recover much of the lost performance without changing the Assistant's weights. Our goal is not to solve all forms of real-world user personalization, but to use the LiC benchmark as a controlled testbed for this mechanistic hypothesis. Since LiC induces a recurring fragmented and passive communication style, learning reusable rewriting rules from held-out trajectories allows us to test whether correcting intent interpretation is a viable intervention point for mitigating LiC.

Our current implementation realizes this Mediator through lightweight prompt-based rewriting. However, the framework is not tied to prompting: with sufficient interaction data, the same mediation layer could in principle be learned through supervised fine-tuning or reinforcement learning.

To bridge the information gap, we leverage the user's generalized history $\mathcal{H}$. In our implementation, $\mathcal{H}$ consists of raw contrastive interaction pairs (successful vs. failed trajectories). However, raw history could be noisy and high-dimensional. Therefore, we introduce a Refiner module (R) to distill $\mathcal{H}$ into a compact set of explicit Experiences ($\mathcal{E}$). These experiences serve as the "communicative profile" for the Mediator M. Our approach is training-free, learning from history through prompting rather than parameter updates. In realistic deployments, analogous signals may come from conversation history, such as reformulations, corrections, follow-up questions, task abandonment, or regeneration requests. We treat these signals as a natural extension beyond the controlled LiC setting, and leave validation with heterogeneous real users to future work.

4.1 The Mediator Framework

The core premise, as derived in Eq. (4), is that the raw context C_t acts as a lossy compression of the true intent I_t. Direct interaction fails because the generic Assistant (A) maximizes $P(R \mid C_t)$ based on population priors rather than individual intent. The Mediator acts as an alignment layer, approximating the inference distribution by conditioning on the distilled experiences: $P(I_t \mid C_t, \mathcal{E})$.

The Mediator takes the accumulated context C_t and experiences $\mathcal{E}$. It analyzes the discrepancy between the surface utterance and the latent intent, generating a reconstructed instruction $\hat{U}$. This $\hat{U}$ is an explicit, fully specified articulation of I_t. The

Assistant A then generates the response based on this low-entropy input:

$$R = A(\hat{U}), \quad \text{where } \hat{U} \approx I_t. \tag{10}$$

For the simulated user style in LiC, the distilled experiences are aggregated into a fixed knowledge base $\mathcal{E}_u$, which is injected into the Mediator as a system instruction. In real deployments, $\mathcal{E}_u$ could be constructed from a particular user's past interactions; in our benchmark, it captures the recurring behavior of the benchmark's fragmented, passive user simulator. Mathematically, the Mediator performs the mapping:

$$\hat{U} = M(C_t \mid \mathcal{E}_u). \tag{11}$$

During inference, the Mediator uses $\mathcal{E}_u$ to interpret C_t, detect omitted information, and synthesize the clarified prompt $\hat{U}$. Since $\mathcal{E}_u$ encodes user-specific communicative habits, the Assistant receives a low-entropy instruction rather than a noisy dialogue trace.

4.2 Experience Acquisition

This subsection details how we construct $\mathcal{H}$ and distill it into actionable experiences $\mathcal{E}$. Although the LiC benchmark does not contain multiple persistent real-user identities, it implicitly defines a consistent simulated user style: the user reveals constraints incrementally, provides fragmentary follow-ups, and often does not explicitly correct the Assistant's mistaken assumptions. We therefore treat held-out contrastive trajectories as historical evidence about this simulated communicative style. This setting evaluates whether a Mediator can adapt to a recurring pattern of intent revelation, rather than whether it can distinguish among many real users.

Contrastive Pair Construction. We premise our data construction on a common behavioral pattern in human-computer interaction: iterative query reformulation (Huang and Efthimiadis, 2009; Odijk et al., 2015). When an initial ambiguous utterance fails to elicit the desired response, users often refine their instructions until the model successfully executes the task. This behavior naturally yields paired trajectories: the final successful turn can be viewed as an intent articulation (D^+), while the preceding failed interaction sequence provides the ambiguous context (D^-). The difference between D^- and D^+ reveals information omitted from the ambiguous context but needed for intent disambiguation.

In the LiC benchmark, we instantiate this idea synthetically. For instances where the model fails in the sharded setting but succeeds with the corresponding full instruction, we pair the failed trajectory D^- with the successful full input D^+. Although LiC does not contain multiple persistent real-user identities, it simulates a consistent fragmented and passive user style; we therefore treat held-out contrastive trajectories as historical evidence about this simulated communicative style. All mined instances are removed from evaluation, and the Refiner and Mediator never access the full prompt, gold solution, or failed trajectory of any test instance.

Experience Distillation via Refiner. To extract explicit inference rules from these interaction logs, we employ a Refiner (R). The Refiner performs an inductive analysis over contrastive pairs, identifying patterns that help recover the user's intended instruction from future ambiguous trajectories. It outputs structured textual guidelines $\mathcal{E} = \{e_0, e_1, ..., e_n\}$, which describe reusable rewriting strategies rather than task-specific content (e.g., "If the user has not explicitly approved the previous solution, he is not satisfied with it."). These distilled experiences serve as the context for the Mediator.

4.3 Training-Free Implementation

Our implementation involves no parameter updates. Both the Refiner and the Mediator are instantiated through prompting. Given a small held-out set of contrastive pairs $\{(D_i^-, D_i^+)\}_{i=1}^K$, the Refiner receives the failed multi-turn trajectory D_i^- and the corresponding successful full instruction D_i^+, and outputs a compact set of textual experiences $\mathcal{E}$. Each experience describes a reusable interpretation rule rather than task-specific content, such as how to treat fragmentary follow-up turns, corrections, or omitted constraints.

At test time, the Mediator receives the current sharded dialogue context C_t together with the distilled experiences $\mathcal{E}$. It rewrites the dialogue into a single explicit instruction $\hat{U}$ that preserves all constraints inferred from the conversation. The Assistant then receives only $\hat{U}$ as its task input and generates the final answer. This design prevents the Assistant from re-interpreting the ambiguous dialogue history and isolates intent reconstruction in the Mediator.

Unless otherwise stated, the same backbone model is used for the Assistant, the Mediator and the Refiner in each model setting. This ensures that improvements are not due to replacing the execu-

Model	Method	Code		Database		Actions		Math		Average	
		$\bar{P}$	R	$\bar{P}$	R	$\bar{P}$	R	$\bar{P}$	R	$\bar{P}$	R
GPT-4o-mini	Full	74.2	76.0	92.5	93.5	93.7	92.4	87.2	70.9	86.9	83.2
	Sharded	51.4	57.0	52.5	54.2	45.5	60.0	64.9	45.6	53.6	54.2
	w/ Ours	66.9	63.6	65.3	59.8	85.7	81.2	77.7	70.4	73.9	68.8
	Gain	+15.5	+6.6	+12.8	+5.6	+40.2	+21.2	+12.8	+24.8	+20.3	+14.6
GPT-5.2	Full	83.2	84.0	96.3	95.9	90.2	93.2	94.5	89.4	92.7	85.4
	Sharded	39.4	44.0	49.4	48.0	35.6	46.6	69.6	48.6	48.5	46.8
	w/ Ours	69.1	70.1	64.5	56.7	76.2	65.2	80.6	62.0	72.6	63.5
	Gain	+29.7	+26.1	+15.1	+8.7	+40.6	+18.6	+11.0	+13.4	+24.1	+16.7
DeepSeek-v3.2-Thinking	Full	98.3	95.9	94.4	88.8	92.2	88.6	94.0	81.6	94.7	88.7
	Sharded	78.4	65.7	43.6	54.2	42.3	48.6	78.8	56.3	60.8	56.2
	w/ Ours	86.1	84.8	67.3	55.9	88.0	71.6	86.3	67.3	81.9	69.9
	Gain	+7.7	+19.1	+23.7	+1.7	+45.7	+23.0	+7.5	+11.0	+21.1	+13.7

Table 1: **Main Results.** Comparison of average performance ($\bar{P}$) and reliability (R) across three LLM backbones. All experiments are averaged over 5 runs. R is calculated at the instance level as $1 - (S_{\text{max}} - S_{\text{min}})$, where scores are normalized to $[0, 1]$, and then averaged across instances and reported as a percentage. *Full* represents ideal instructions, while *Sharded* represents ambiguous user inputs. *w/ Ours* denotes the performance when using our Experience-Driven Mediator. The best improvements are highlighted in green.

tion model with a stronger model.

5 Experiments

To systematically evaluate the robustness of LLMs in multi-turn interactions and validate our proposed framework, we conduct extensive experiments leveraging the simulation benchmark introduced by Laban et al. (2025). We benchmark a diverse set of state-of-the-art models, including the widely-used GPT-4o-mini (Hurst et al., 2024), the highly capable GPT-5.2, and the reasoning-specialized DeepSeek-V3.2-Thinking (DeepSeek-AI, 2025). Details of experiment setup are available at §C. In this section, we present our main results, demonstrating the universality of the LiC phenomenon and the efficacy of our Mediator across different model architectures (§5.1). We also provide an in-depth ablation analysis to distinguish intent alignment from simple factual memory retrieval (§5.2).

5.1 Main Results

Table 1 presents the comprehensive evaluation results across four domains: Code, Database, Actions, and Math. We compare three settings: (1) **Full**: The idealized upper bound, where users provide complete, unambiguous instructions in a single turn. (2) **Sharded**: The baseline setting, representing simulated multi-turn conversations where context is fragmented and ambiguous. (3) **w/ Ours**: The proposed framework, where the Experience-Driven Mediator reconstructs the *Sharded* input into a specified instruction before passing it to the Assistant. To rigorously assess model robustness against the stochasticity inherent in multi-turn generation, we conducted five independent runs for every experimental instance. We report two key metrics: (1) Average Performance ($\bar{P}$): The mean score across the five runs. (2) Reliability (R): A consistency metric calculated at the instance level. For each instance, we first normalize the five run-level scores to $[0, 1]$ and measure the spread between the best and worst outcomes, $S_{\text{max}} - S_{\text{min}}$. The instance-level reliability is defined as $1 - (S_{\text{max}} - S_{\text{min}})$, and the reported R is the average reliability across all instances, multiplied by 100.

The Persistence of LiC. Comparing the *Full* and *Sharded* settings reveals a severe alignment gap that transcends model architectures. First, standard instruction-tuned models exhibit dramatic degradation; for instance, **GPT-5.2** drops from a near-perfect 92.7% to 48.5%. This suggests that scaling model capabilities alone cannot resolve the ambiguity problem. In the absence of explicit intent, even the most capable models often resort to ungrounded guessing to fill the information gap, resulting in performance collapses similar to those observed in smaller baselines. Crucially, even reasoning-enhanced models like **DeepSeek-v3.2-Thinking** fail to overcome this hurdle. Contrary to the expectation that intrinsic Chain-of-Thought (CoT) might infer missing context, DeepSeek's performance in the *Sharded* setting remains severely compromised.

This result exposes a critical limitation: reasoning capabilities are ineffective against information ambiguity. While CoT excels at logical deduction when premises are clear, it cannot "reason" its way out of an information vacuum. Without the explicit context reconstruction provided by our Mediator, the model is forced into ungrounded speculation, regardless of its reasoning depth.

Efficacy of Mediator. Implementing our proposed Mediator yields substantial and consistent improvements across all backbones and domains. On average, our method recovers performance by approximately 20% in $\bar{P}$ and 15% points in R. Notably, even for the reasoning-enhanced DeepSeek-v3.2-Thinking, our approach secures a 21.1% increase, demonstrating that the Mediator provides value complementary to advanced internal reasoning. By reconstructing ambiguous contexts into explicit, self-contained instructions ($\hat{U}$), it collapses the uncertainty space for each individual instance. This ensures that the Assistant receives a low-entropy input, leading to consistently high-quality execution regardless of the random seed, effectively converting a probabilistic guessing game into a deterministic reasoning task.

Reliability and Performance. The improvements in R generally track the improvements in $\bar{P}$, suggesting that instability in the Sharded setting is largely driven by intent ambiguity rather than generation stochasticity alone. Once the Mediator reconstructs a clearer instruction, repeated runs become more consistent. Since R is bounded above by 100, gains are naturally smaller for models that are already relatively reliable.

5.2 Intent Alignment vs. Factual Memory

To identify the root cause of the performance degradation in multi-turn dialogues, we compare our approach against two representative context utilization strategies (Table 2): a naive summarization method (**w/ Sum**), which concatenates all prior turns into a single context window without any structural processing, and a mainstream RAG-based memory framework **Mem0** to persist user facts.

Memory is not Understanding. w/ Mem0 yields only marginal gains over Sharded (53.6% $\rightarrow$ 56.5%), and **w/ Sum** improves little on average while degrading some domains. This suggests that the main failure is not simply forgetting prior facts. In the *Sharded* setting, the relevant information may appear somewhere in the dialogue, but the model must determine how the latest user utterance should revise, constrain, or reinterpret that information. For example, a retrieved memory may preserve an earlier constraint, but the Assistant still has to decide whether the user's next fragment confirms that constraint, replaces it, or adds a new exception. By explicitly reconstructing an actionable instruction, our Mediator bridges the gap between raw context and intended use, outperforming Mem0 by 17.4%.

Method	C	D	A	M	Avg.
Full	74.2	92.5	93.7	87.2	86.9
Sharded	51.4	52.5	45.5	64.9	53.6
w/ Sum	57.1	44.9	54.6	62.0	54.7
w/ Mem0	52.8	56.9	49.4	66.9	56.5
w/ Ours	66.9	65.3	85.7	77.7	73.9

Table 2: **Comparative Analysis.** We compare our method against Summarization (Sum) and a RAG-based memory framework (Mem0) on GPT-4o-mini. The marginal gains of Mem0 indicate that simply recalling factual context is insufficient for multi-turn robustness. In contrast, our method succeeds by explicitly reconstructing the user's intent.

Why Refiner? Appendix B compares our distilled experiences with Direct ICL using raw contrastive pairs. Direct ICL reaches comparable accuracy but consumes $3.6\times$ more tokens, indicating that the Refiner preserves the mediation benefit while substantially improving inference efficiency.

6 Conclusions

In this paper, we revisited the "Lost in Conversation" (LiC) phenomenon, demonstrating that the common attribution to model unreliability is incomplete. Instead, LiC can be largely explained by an intent alignment gap between user expression and model understanding, a structural issue that capacity scaling alone is unlikely to resolve without additional historical disambiguation signals, such as real user histories or recurring interaction patterns from a simulated user style. We tested this view with a Mediator-Assistant framework that decouples intent understanding from task execution, demonstrating significant performance recovery across diverse LLMs. Our current implementation relies on limited historical interaction signals; when such signals are unavailable, the method gracefully degrades to generic guidance.

7 Limitations

Constrained by the limited scale of existing benchmarks, our current Refiner operates in a few-shot, non-parametric manner, extracting explicit and heuristic-level guidelines. While this design ensures efficiency, it captures only coarse-grained interaction patterns. Future work could leverage larger-scale datasets to transition towards parameterized training, enabling the Mediator to internalize more detailed alignment strategies via fine-tuning rather than relying solely on in-context summaries. Furthermore, the multi-turn settings in current benchmarks exhibit relatively homogeneous user logic. Developing more comprehensive benchmarks that mirror complex user behaviors remains a critical direction to further validate and evolve our framework. One might ask whether our method remains effective when user-specific information is unavailable. We note that the required information is not additional data, but rather the natural byproduct of user-LLM interaction. In scenarios where even this is inaccessible (e.g., first-time users with no history), the Mediator can operate with generic guidelines derived from population-level patterns, gracefully degrading to baseline performance.

References

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and 1 others. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7421–7454.

Yuen Ren Chao and George Kingsley Zipf. 1949. *Human behavior and the principle of least effort: An introduction to human ecology*. Addison-Wesley Press.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

DeepSeek-AI. 2025. Deepseek-v3.2-exp: Boosting long-context efficiency with deepseek sparse attention.

Christine Herlihy, Jennifer Neville, Tobias Schnabel, and Adith Swaminathan. 2024. On overcoming miscalibrated conversational priors in llm-based chatbots. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, pages 1599–1620.

Jeff Huang and Efthimis N Efthimiadis. 2009. Analyzing and evaluating query reformulation strategies in web search logs. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 77–86.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*.

Ming Li. 2025. Verifiable accuracy and abstention rewards in curriculum rl to alleviate lost-in-conversation. *arXiv preprint arXiv:2510.18731*.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797.

Daan Odijk, Ryen W White, Ahmed Hassan Awadallah, and Susan T Dumais. 2015. Struggling and success in web search. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1551–1560.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

David R Traum and Staffan Larsson. 2003. The information state approach to dialogue management. In *Current and new directions in discourse and dialogue*, pages 325–353. Springer.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *Preprint*, arXiv:2312.12148.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.

J Diego Zamfirescu-Pereira, Richmond Y Wong, Bjoern Hartmann, and Qian Yang. 2023. Why johnny can't prompt: how non-ai experts try (and fail) to design

llm prompts. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–21.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

Wanjun Zhong, Duyu Tang, Jiahai Wang, Jian Yin, and Nan Duan. 2021. Useradapter: Few-shot user learning in sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1484–1488.

A Additional Discussion

Clarification vs. Mediation. One might suggest that LLMs should actively prompt users to clarify their intentions when faced with ambiguity, rather than relying on an external Mediator. However, this approach is fundamentally at odds with the LiC problem setting. The LiC benchmark explicitly models users who communicate incrementally, starting with vague or underspecified goals and only gradually revealing their true needs through iterative interaction. Asking such users to "clarify their intent" is, in practice, equivalent to asking them to provide fully specified instructions from the outset, which defeats the purpose of studying multi-turn conversation. Furthermore, requiring explicit clarification shifts the burden of intent alignment from the system to the user, degrading the user experience. A well-designed dialogue system should adapt to the user's natural communication style, not the other way around. Our approach preserves the natural flow of conversation by resolving ambiguity upstream through the Mediator, without requiring users to modify how they express themselves.

Using Successful Turns as Signals. A related concern is whether treating the final successful turn as ground-truth user intent (D^+) is a reliable assumption. We clarify that D^+ is not assumed to reveal a user's private intent perfectly; rather, it provides a practical feedback signal indicating which formulation successfully resolves the task. In the LiC benchmark, the user simulator follows a recurring fragmented and passive communication style: constraints are revealed incrementally, and mistaken assistant assumptions are often not explicitly corrected. Under this setting, contrastive pairs (D^-, D^+) provide historical evidence about how ambiguous fragmented contexts should be rewritten into complete task instructions.

This evidence is necessary because, without additional historical disambiguation signals, the same ambiguous dialogue context may admit multiple plausible interpretations. The role of D^+ is therefore not to serve as an oracle for hidden mental states, but to provide a minimal supervision signal for learning reusable rewriting patterns. In real-world deployments, analogous signals can be obtained from user behaviors such as query reformulation, follow-up questions, task abandonment, regeneration requests, or explicit feedback. These signals need not be perfect annotations; they only need to indicate which interpretations were more likely to satisfy the user.

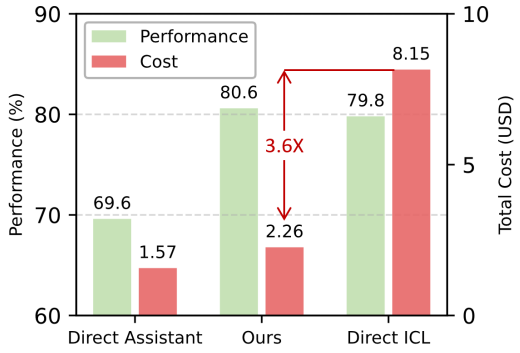

Figure 4: **Comparison with In-Context Learning.** We compare our method against a Direct Assistant baseline without mediation and a Direct ICL approach. Our method delivers a substantial performance boost over the no-mediator baseline with only a marginal cost increase. While ICL achieves comparable accuracy to Ours, it consumes 3.6× more tokens.

B Refiner Efficiency Analysis

We investigate the necessity of the Refiner by comparing our method with two baselines. The Direct Assistant baseline sends the sharded dialogue directly to the Assistant without any Mediator. The Direct ICL baseline embeds raw contrastive pairs directly into the Mediator prompt, without distilling them into reusable experiences.

As shown in Figure 4, adding a Mediator with distilled experiences substantially improves over the Direct Assistant baseline while incurring only a marginal increase in token usage. Direct ICL

achieves comparable accuracy to our method, but it requires $3.6\times$ more tokens because the raw contrastive trajectories are verbose. This demonstrates that the Refiner primarily serves as an efficiency mechanism: it compresses historical interaction logs into concise guidelines while preserving most of the benefit of mediation.

C Benchmark Details

C.1 Dataset Details

We utilize the pre-constructed conversational data provided by Laban et al. (2025). The original benchmark encompassed six diverse tasks, which can be categorized into two distinct types based on their evaluation criteria. The first category, Binary Correctness Tasks, consists of Code, Database, Actions, and Math. For these tasks, success is defined by strict execution accuracy or exact constraint satisfaction. The second category, Refinement Tasks, comprises Data-to-text and Summarization, where the performance is evaluated based on generation quality rather than a single correct answer.

To transform these standard single-turn benchmarks into multi-turn conversational trajectories, the authors employed a semi-automated "sharding" pipeline. This process involved using a teacher LLM to decompose a fully specified instruction into a sequence of atomic, self-contained constraints (shards), which were subsequently rewritten to mimic a user progressively clarifying requirements. These shards were rigorously validated via human annotation to ensure semantic equivalence to the original instruction.

C.2 Experimental Adaptations

In this work, we introduce two critical adaptations to the original experimental setup to strictly isolate conversational reasoning failures from other confounding factors.

Task Selection. First, we refine the scope of evaluation by focusing exclusively on the Binary Correctness Tasks (Code, Database, Actions, and Math). We explicitly exclude the Refinement Tasks (Data-to-text and Summarization) from our experiments. Our preliminary analysis indicated that LLMs often exhibit floor-level performance on these open-ended tasks even in single-turn baselines. Furthermore, the high variance inherent in the evaluation metrics for text generation complicates the analysis, making it difficult to distinguish between multi-turn context failures and general generation capability issues.

Sequential Sharding Strategy. Second, we introduce a strict control regarding information ordering. The methodology in Laban et al. (2025) randomly shuffled the order in which shards were revealed to the model. We argue that randomizing constraints often introduces artificial logical incoherence (e.g., modifying a variable before it is defined), which conflates the model's reasoning capability with its ability to handle ill-posed logical puzzles. To mitigate this, we enforce a sequential sharding strategy where we re-order the provided shards to follow a natural, logical progression (e.g., Problem Definition followed by specific constraints). This ensures that any observed performance degradation is strictly attributable to the model's inability to manage multi-turn context, rather than confusion caused by disordered inputs.

Few Shot Learning Set. Because the benchmark does not contain persistent user identities, we randomly select 5 held-out instances from each task to construct historical contrastive dialogues. These histories approximate recurring patterns of underspecification and correction produced by the benchmark's simulated fragmented user, rather than long-term records of multiple real users. These instances are removed from the test set before evaluation and are used only for constructing contrastive histories and distilled experiences. No test instance contributes its full instruction, gold answer, failed trajectory, or successful single-turn prompt to the Refiner or Mediator. This separation ensures that improvements reflect reusable intent-alignment rules rather than instance-level leakage.

D Prompts

Prompt For Mediator

You are an intermediary AI mediator responsible for analyzing and rewriting the request based on the "Conversation Transcript between the User and the AI assistant".
Your task is to produce a clear and complete user request that can be sent to the AI assistant to obtain user's desired answer.

[Important Rules]
[[EXPS]]

[Output Format]
Output only the rewritten user request in first-person form, without any extra commentary. It should be ready to be copy-pasted and sent to another AI assistant exactly as if it came directly from the User.

[Conversation Transcript]
[[CONVERSATION]]

Please output:

Prompt For Refiner

You are a meta-analysis AI assistant responsible for extracting effective rewriting rules and best practices from examples of Conversation Transcripts between a User and an AI assistant, together with the Ground Truth rewritten request.
Your goal is to produce a clear, actionable set of RULES that can guide another "Request Rewriting Mediator" to accurately transform a given Conversation Transcript into a clear, first-person user request.
The RULES you produce must be:
- Few in number, focusing only on the most critical factors that determine correct rewriting.
- Direct, actionable, and applicable to any future Conversation Transcript.

[Input You Will Receive]
A list of examples, each containing:
1. Conversation Transcript (dialogue between the User and the AI assistant).
2. Ground Truth Rewritten Request (ideal output for that conversation).

[Your Tasks]
1. For each example:
  - Compare the Conversation Transcript with its Ground Truth request.
  - Note what transformations were applied (e.g., summarization, rephrasing, deletion of irrelevant parts, conversion to first-person, clarification of ambiguous points, tone adjustment).
2. Identify common patterns and shared decision rules across all examples.
3. Merge these insights into one unified, concise set of numbered RULES:
  - RULES must work for *new, unseen conversations*.
  - RULES must be clear, direct, and applicable without additional explanation.

[Output Format]
Output only the RULES section as:
[[RULES]]
1. ...
2. ...
3. ...
[[END]]

Do not include any commentary outside the [[RULES]] section.

[Now process the following data]
[[CONVERSATIONS]]

Example of Generated Experiences

1. Extract and consolidate all numerical values, relationships, and contextual details provided exclusively by the user across the entire conversation, ignoring AI responses and speculative examples.
2. Structure the rewritten request by:
- Starting with initial conditions or subjects
- Adding events/relationships in chronological or logical order
- Ending with the core question unchanged
3. Convert all references to the user's query subject (e.g., \"they,\" \"Andy,\" \"the hotel\") into third-person perspective and clarify ambiguous terms using precise language (e.g., \"proportional\" instead of \"go up based on,\" \"elected for\" instead of \"opted\").
4. Remove conversational artifacts (e.g., \"Oh,\" \"actually,\" \"by the way\") and partial confirmations, retaining only finalized user statements.
5. Explicitly include all mathematical relationships (e.g., \"twice as many,\" \"5% raise,\" \"three times\") and temporal conditions (e.g., \"on the first day\") even if implied in the dialogue.