

Thinking Is Not Telling: Information Disclosure in User-Service LLM Agents

Jiatong Li

University of Wisconsin-Madison

jli2947@cs.wisc.edu

Changdae Oh

University of Wisconsin-Madison

changdae@cs.wisc.edu

Hyeong Kyu Choi

University of Wisconsin-Madison

froilanchoi@cs.wisc.edu

Jindong Wang

William & Mary

jdw@wm.edu

Sharon Li

University of Wisconsin-Madison

sharonli@cs.wisc.edu

Reviewed on OpenReview: <https://openreview.net/forum?id=xngXTG9Jyj>

Abstract

User-engaged LLM agents increasingly operate in service scenarios where task success depends on coordination between the agent, the user, and a stateful environment. In such interactions, the agent often has access to rich information that the user does not observe, such as task policies, tool results, and environment states. This makes agent-user communication a central component of task completion. In this work, we study a communication-level failure mode of user-service agents: internal reasoning is counterproductive for the interactive information disclosure. We introduce *information disclosure* as a measurable communication mechanism, covering state corrections, available options, constraints, consequences, and proactively retrieved environment information. Motivated by recent advances in LLM reasoning, we evaluate and analyse information disclosure under a controlled setting of agents equipped with a “think” tool, which allows agents to reason internally before responding or acting. Across five models and three user-service scenarios, we first show that enforced thinking has unstable utility. We then analyse response scaling under different think efforts and find that more thinking does not reliably translate into richer information disclosure. Using evidence-grounded probing and turn-level counterfactual intervention, we show that injecting golden decision-relevant information significantly intervenes agents to correct action trajectories, indicating that missing disclosure can causally affect the agent’s immediate downstream environment-edit behaviour. We further find that failed trajectories most often lack state corrections and constraints, and that a lightweight disclosure-aware prompt effectively mitigates degradation from enforced thinking. These results suggest that the key challenge is not simply making agents think more, but ensuring that reasoning and environment information are externalised as timely, user-facing information disclosure.

1 Introduction

Large language model (LLM) agents are increasingly applied to user-service scenarios where task completion requires coordination among an agent, a user, and a stateful environment (Yao et al., 2023b; Gao et al., 2025). Unlike conventional reasoning benchmarks in which the full problem specification is given upfront,

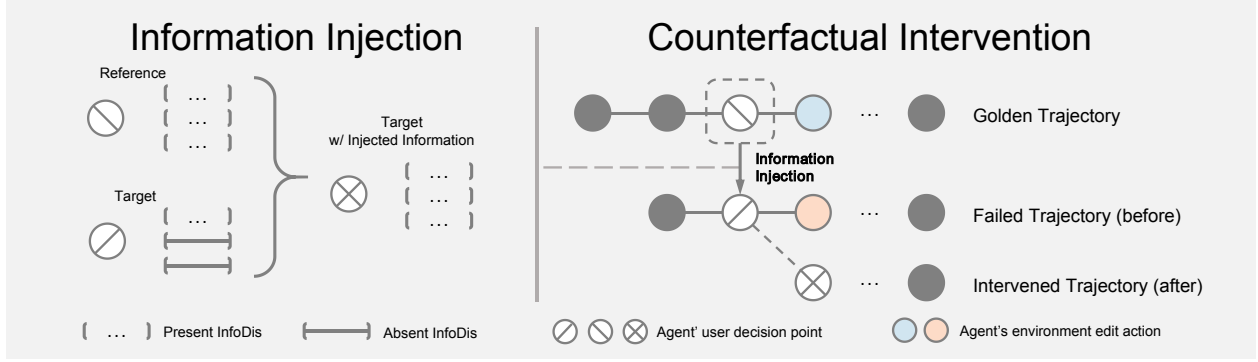


Figure 1: An overview of the counterfactual intervention design. To handle intervention in multi-turn dialogue processes, the method identifies the *last* agent’s user decision point before the *first* environment edit action. Next, the method injects the InfoDis evidence annotated from evidence-grounded probing.

user-engaged tasks often unfold as collaborative decision-making under asymmetric and partial information (Hemmer et al., 2025; Luo et al., 2025; Lu et al., 2025; Wang et al., 2024c; Qian et al., 2025). The agent may have access to rich hard-to-read information including task policies, tool interfaces, and environment states, while the user holds complicated intent: goals, preferences, or constraints. As a result, the agent must establish sufficient common ground with the user before consequential decisions are made.

In previous studies, a common practice in enhancing LLM agents is to equip them with higher reasoning efforts (Yao et al., 2024; Anthropic, 2025), which allows agents to handle complicated intent understanding, task planning and logical reasoning (Muennighoff et al., 2025; Wei et al., 2022). However, the interaction-aware characteristic of user-engaged scenarios makes the contribution of internal reasoning doubtful. In our large-scale validation across five models, three user-service scenarios, and two think modes, enforcing internal thinking does not reliably improve agent performance (see Table 1). This raises a mechanism-level research question: **How productive is thinking for incentivising agents’ user-service behaviour?**

We argue that the missing piece lies in the distinction between *external communication* and *internal reasoning*. In user-engaged tasks, the agent’s success is directly decided by whether it has sufficient communication with the user and whether it applies correct environment-aware actions during the communication. The agent’s internal reasoning is useful only insofar as it is converted into appropriate user-facing responses and environment actions. When the next state transition depends on the user’s decision – confirmation, selection, or preference (Williams & Young, 2007), the user must be informed of the relevant information: state, options, constraints, and consequences (Clark & Brennan, 1991). Although an internal think step may help the agent organise its own plan, it does not guarantee that decision-relevant information is either identified by the agent or disclosed to the user at the right time. This view is consistent with theories of shared collaborative plans (Grosz & Kraus, 1996), POMDP-based models of dialogue, and cooperative decision-making under uncertainty (Hadfield-Menell et al., 2016; Lin et al., 2024). As a result, a natural communication–reasoning mismatch exists: the user still makes decisions under incomplete information while the agent may privately process useful information.

To study this mismatch, we introduce **information disclosure** as a measurable communication mechanism in user-engaged LLM agents. We define information disclosure as the agent’s proactive communication of decision-relevant information that the user likely does not possess but needs for the next decision. This includes state corrections, available options, task constraints, decision consequences, and proactively retrieved environment information. This perspective complements prior work on information clarification (Luo et al., 2025; Zhang & Choi, 2025; Suri et al., 2026), where the agent elicits missing information from the user. In contrast, information disclosure studies the reverse direction of information flow: what the agent should reveal to help the user make an informed decision.

We conduct a multi-stage empirical study to examine how information disclosure and downstream task dynamics are affected by internal thinking efforts. Taking the “think” tool as a controlled and practically im-

portant instantiation, we first evaluate across multiple models and scenarios, showing that enforced thinking does not consistently improve user-service task success. Second, we analyse different levels of think effort and find that more internal thinking does not reliably translate to user-facing communication. Third, a motivating case study shows that an enforced-think agent can omit a critical task constraint before the user’s decision, misleading the user toward an invalid action. We then formalise this phenomenon through evidence-grounded probing of information disclosure and perform turn-level counterfactual interventions on failed trajectories (see Figure 1). Compared with verbosity and think-content controls, injecting golden decision-relevant information produces a more significant improvement in First Edit Correction Rate (FECR), indicating that *missing disclosure can directly affect the agent’s immediate downstream environment-edit behavior*. Finally, we show that a simple InfoDis-aware prompting strategy mitigates the degradation of enforced thinking and improves overall performance across models and domains. Overall, this study demonstrates that information disclosure is an interpretable and controllable factor of user-engaged LLM agents, providing insights for future development of relevant fields.

In summary, this study makes three contributions:

- We formulate missing decision-relevant information disclosure as a general communication bottleneck in user-service LLM agents and show that augmenting internal reasoning alone does not ensure that users receive the information required for informed decisions.
- We introduce a five-category taxonomy and an evidence-grounded framework for probing information disclosure in agent-user dialogues. Through turn-level counterfactual intervention, we demonstrate that missing disclosure is an interpretable and actionable factor in downstream agent failures.
- We evaluate disclosure-aware prompting as a lightweight, training-free intervention for both thinking and non-thinking agents, showing that explicitly optimizing user-facing disclosure can improve overall performance and, in particular, mitigate degradation under enforced thinking.

2 Related Work

Agent-user communication. User-engaged agents have become an important class of LLM agents in both industrial and academic settings (Clarke et al., 2022; Deshpande et al., 2025). These agents communicate with users through natural language (Qian et al., 2024) while interacting with external environments through formatted actions or tool calls (Yao et al., 2023b; OpenAI, 2025a). Accordingly, recent benchmarks have moved beyond static multi-turn dialogue evaluation (Bai et al., 2024) toward interactive evaluation with simulated users and verifiable task outcomes, such as database-state matching in τ -Bench (Yao et al., 2024), tool-use feedback in MINT (Wang et al., 2024c), and milestone tracking in ToolSandbox (Lu et al., 2025). A key challenge in user-agent communication is asymmetric information. Existing studies mainly address this issue through *information clarification*, where the agent elicits missing or ambiguous information from the user. Prior works cover implicit user intention understanding (Qian et al., 2024), deciding when to ask clarification questions (Zhang & Choi, 2025), clarification-oriented dialogue evaluation (Luo et al., 2025), and training agents to reason, speak, and act interactively (Acikgoz et al., 2026). In contrast, our work studies the reverse direction of information flow: *information disclosure*, where the agent proactively communicates task-relevant information to help the user make informed decisions. This perspective complements clarification-based interaction and highlights a different failure mode: the agent may have access to information that the user needs, but fail to disclose it before the user’s next decision.

Reasoning of LLM agents. Reasoning has been widely studied as a core capability of LLMs and LLM agents. Chain-of-thought prompting improves mathematical and symbolic reasoning by eliciting intermediate reasoning traces (Wei et al., 2022; Kojima et al., 2022), and later methods extend this idea through structured search, planning, and reflection (Yao et al., 2023a; Shinn et al., 2023; Hao et al., 2023). In agent settings, ReAct interleaves reasoning traces with environment actions (Yao et al., 2023b), and subsequent agent frameworks explore reflection, memory, and planning for long-horizon tool use (Shinn et al., 2023; Wang et al., 2024b). However, explicit reasoning does not always improve user-engaged tool-calling agents. Previous works in reasoning have discovered the overthinking problem (Ghosal et al., 2025). Recent studies find

that chain-of-thought or reasoning traces can have marginal or inconsistent utility in interactive tool-use settings (Yao et al., 2024; Sprague et al., 2025; Zhang et al., 2026). Meanwhile, the private “think” tool has been introduced as a model-agnostic way for tool-calling agents to write intermediate reasoning into their own context without exposing it to users or modifying the environment (Anthropic, 2025). Existing explanations for its unstable utility often focus on insufficient training (Wang et al., 2024a), hallucinated thoughts (Yin et al., 2026; Baker et al., 2025), or mismatch between reasoning and tool-use objectives (Sprague et al., 2025). Our work instead examines a communication-level mechanism: in user-engaged tasks, internal reasoning may fail to help when decision-relevant information is not externalized to the user before the next decision point.

3 An Empirical Study of Think on Response Scaling

3.1 Preliminary Evaluation of the “Think” Tool

We consider a user-engaged tool-calling agent that interacts with a user through natural language and with an external environment through tool calls. Let $F = \{f_1, \dots, f_C\}$ denote the original tool set, and let f_{think} denote a internal reasoning tool. Following prior practice (Anthropic, 2025; Yao et al., 2024), f_{think} does not query the environment or modify its state; instead, it appends a free-form reasoning trace to the agent’s internal context. This trace is visible to the agent in subsequent steps but invisible to the user. We compare agents equipped with F against agents equipped with $F \cup \{f_{\text{think}}\}$. As a preliminary validation, we evaluate whether enforcing the “think” tool before each agent step improves task performance.

Our evaluation covers diverse model families and generations, including GPT-4o (Hurst et al., 2024), GPT-4.1 (OpenAI, 2025b), GPT-5 (OpenAI, 2025c), DeepSeek-V3.1 (DeepSeek, 2025), and gpt-oss-120b (Agarwal et al., 2025). We use three representative user-service scenarios: Retail and Airline from τ -Bench (Yao et al., 2024), and the multi-turn phone scenario from ToolSandbox, denoted as TS-Phone (Lu et al., 2025). The original ToolSandbox uses a process-aware milestone similarity in $[0, 1]$ as the evaluation metric. For our comparative evaluation, we label similarity > 0.9 as “passed” to get a binary outcome score. Each setting is evaluated with three random seeds, and we report pass^1 and pass^2 , where pass^k measures the probability that all k independent trials of a task succeed. Table 1 shows that enforced thinking does not consistently improve user-engaged tool-calling agents. While it helps in a few settings, such as gpt-oss-120b on Retail and GPT-4.1 on Airline, it also substantially degrades performance in others, such as GPT-4.1 on Retail, where pass^2 drops from 63.48% to 52.46%, and GPT-4o on Airline, where pass^2 drops from 30.00% to 19.33%. These results suggest that, unlike reasoning-centric benchmarks, where extended reasoning often improves performance, it does not necessarily improve user-service agents.

3.2 Performance and Response Scaling Comparison

To examine how the think tool affects user-service agents, we first study *response scaling* under different think efforts. In user-engaged settings, think content is private to the agent, while user-visible responses are the only communication channel available to the user. Thus, response length serves as a coarse surface proxy for how much information is communicated. Following prior work, we compare four settings: *no think*, *low think* (System 1), *medium think* (default), and *high think* (System 2) (Kahneman, 2011; Li et al., 2025). We append System 1/2 instructions to the agent system prompt to control low/high thinking efforts, with details in Appendix A. As shown in Figure 2, increasing internal think effort does not reliably improve performance or scale user-facing responses. In several settings, response length even decreases despite enforced thinking. This observation suggests that tool-based internal reasoning does not automatically translate into richer user-facing communication. We next examine this effect at the semantic level through a case study of agent outputs.

3.3 Case Study on Agent Output

Figure 3 shows a representative failure pattern before the first environment-edit action. Both agents face the same retail task that helps a user to return and cancel items. In the trajectory without the think tool, the agent explicitly discloses a key *task constraint* that the system can only cancel the entire pending order rather than an individual item. This constraint changes the user’s next decision: the user chooses to skip

Table 1: Performance comparison of user-engaged tool-calling agents under different think tool mode. Each setting is run for three trials with random seeds. The pass^k quantifies the chance that all k i.i.d. task trials are successful, averaged across tasks, representing the reliability and consistency of LLM agents.

Model	Think Mode	Retail		Airline		TS-Phone	
		Pass ¹	Pass ²	Pass ¹	Pass ²	Pass ¹	Pass ²
GPT-4o	w/o think	64.06	53.33	43.33	30.00	39.29	30.21
	Enforced	59.71	45.51	35.33	19.33	41.52	31.85
GPT-4.1	w/o think	73.33	63.48	44.00	31.33	46.73	37.65
	Enforced	65.51	52.46	52.00	40.67	45.24	35.57
GPT-5	w/o think	74.78	64.78	50.67	42.67	24.85	15.77
	Enforced	69.57	56.23	42.00	32.00	28.72	19.64
DeepSeek-V3.1	w/o think	71.01	59.71	50.00	34.67	41.52	30.80
	Enforced	63.48	46.09	49.33	35.33	41.82	29.91
gpt-oss-120b	w/o think	63.77	51.59	45.33	31.33	38.10	27.68
	Enforced	66.67	55.65	40.67	27.33	37.05	25.15

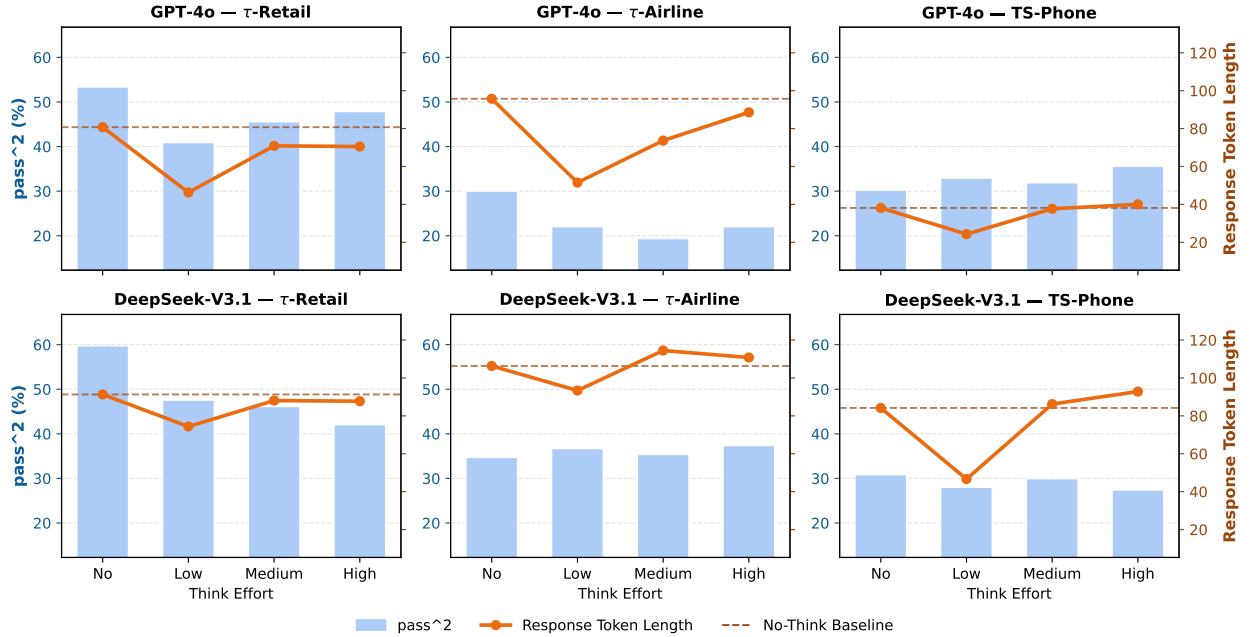


Figure 2: Response token length and pass² with respect to think effort.

the cancellation and proceed only with the return. In contrast, the agent with enforced think internally records relevant task context in its private thought, but its user-visible response omits the constraint. The agent lists the options and asks the user to confirm the cancellation reason, without informing the user of the cancellation constraint. Therefore, the user makes the next decision under incomplete information and confirms the cancellation. We refer to this missing user-facing communication as **information disclosure** (InfoDis). In user-engaged tasks, InfoDis denotes the agent’s proactive communication of decision-relevant information that the user likely does not possess but needs before making the next decision. Figure 3 presents a *constraint disclosure*. More broadly, information disclosure can include state correction, available options, constraints, consequences, and proactively retrieved environment information. In the next section, we formalize these dimensions and test their effects through turn-level counterfactual intervention.

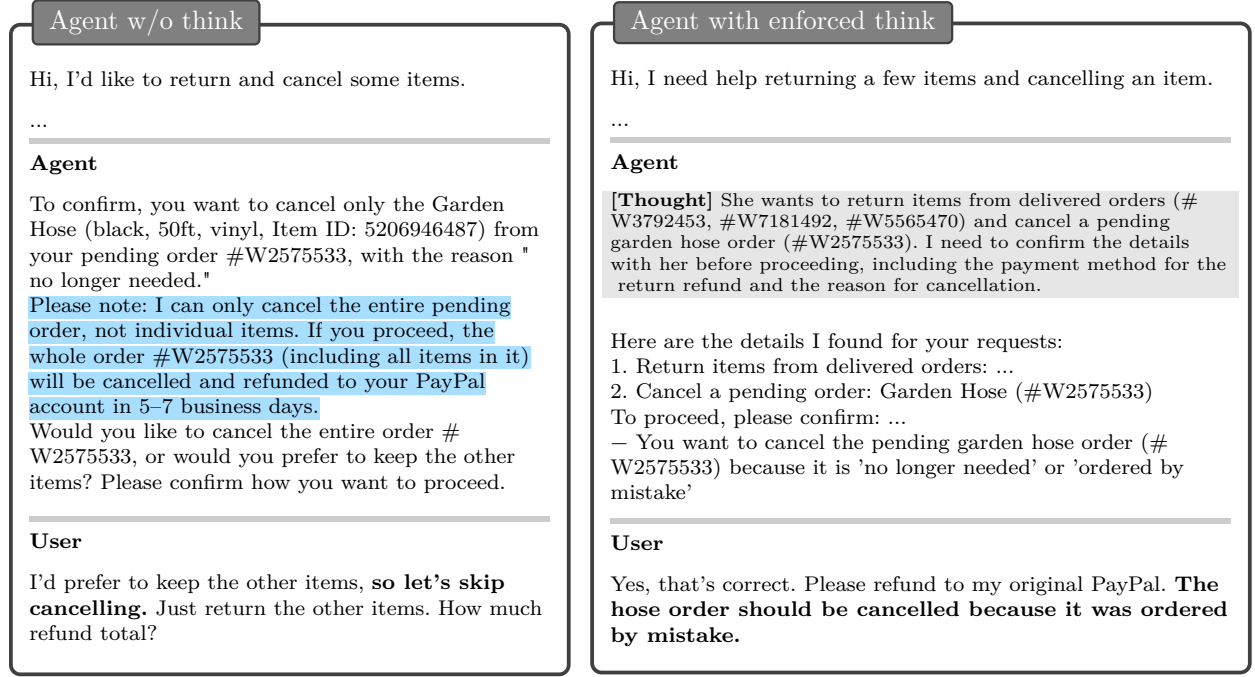


Figure 3: A case of agent-user interaction before the first environment edit action on a retail task (agent: GPT-4o; user: GPT-4.1). The blue highlights the information disclosure of task **constraint**, which intervenes the next user decision. The thought of agent with enforced think is invisible to the user.

Table 2: Information disclosure dimensions.

Category	Definition	Example
State Correction (SC)	Information that corrects a factual misunderstanding or false assumption the user has expressed.	The User says "my cancelled order" but agent clarifies "this order was delivered, not cancelled"
Option (OP)	The set of choices available to the user for their upcoming decision — including relevant variants, alternatives, or paths.	"Here are available options for replacement: - ... - ..."
Constraint (CN)	Rules, policies, eligibility conditions, or limits that affect what the user can or cannot do — and that the user likely does not already know.	"Modifications are only possible before shipment"
Consequence (CQ)	The concrete outcomes that will result from the user's decision, including financial impact, timing, and irreversible effects.	Refund amount, refund timeline, charge amount, order status change
Proactive Retrieval (PR)	Information the agent proactively fetched from its tools /database and disclosed even though the user did not possess and did not directly ask for.	Listing the user's orders with their contents, showing both current and new address for comparison

4 Counterfactual Intervention Pipeline of Information Disclosure

4.1 Evidence-Grounded Semantic Probing of Information Disclosure

To study information disclosure in agent-user interaction, we first need to make it observable and measurable. This is challenging because information disclosure is expressed through natural language rather than through a structured action space. The same response may contain multiple types of task-relevant information, and its meaning depends on the preceding user request, retrieved environment states, task policies, and the upcoming user decision. Moreover, response length alone is insufficient: a short response may disclose a critical constraint, while a long response may add verbosity without changing what the user knows. We therefore avoid assigning a scalar "informativeness" score and instead use an evidence-grounded probing procedure, where an information category is marked as present only when an exact text span in the user-

visible response supports the label. The bracketed information lists in the left part of Figure 1 illustrate this probing process.

Information Disclosure Rubrics. We operationalize information disclosure with five binary categories, summarized in Table 2: *State Correction (SC)*, *Option (OP)*, *Constraint (CN)*, *Consequence (CQ)*, and *Proactive Retrieval (PR)*. These categories capture common forms of decision-relevant information in user-service tasks: correcting the user’s false assumptions, presenting feasible choices, disclosing rules or eligibility conditions, explaining downstream outcomes, and revealing environment information proactively retrieved by the agent. For each agent response, every category is annotated as either present or absent. We use binary labels rather than graded scores to reduce ambiguity and to make trajectory-level information disclosure comparable across models, domains, and intervention conditions.

Evidence-Grounded Annotation. Given an agent-user trajectory, we perform annotation at the agent-turn level. To avoid exposing annotators to experimental conditions, we mask internal reasoning traces and tool calls, and present only the system prompt, user utterances, and user-visible agent responses. For each agent response, the annotator labels the five InfoDis categories. A category can be marked as present only if the annotator identifies an exact substring in the response that directly supports the label; otherwise, it is marked as absent. We also annotate whether each agent turn is a *user decision point*, defined as a turn after which the next user response can determine the subsequent task trajectory, such as confirming an action, selecting among options, providing preferences, or authorizing an environment edit. These annotations provide the basis for comparing successful reference trajectories with failed target trajectories and identifying missing disclosure evidence for counterfactual injection, as shown in Figure 1.

4.2 Counterfactual Information Injection

Figure 1 illustrates our turn-level counterfactual intervention pipeline. Given a failed target trajectory and a matched successful golden trajectory for the same task, we first compare their evidence-grounded InfoDis annotations. The goal is to identify decision-relevant information that is present in the golden response but absent from the failed response, and then inject this missing information into the failed trajectory before the agent makes its first task-critical environment edit. Formally, let τ be a failed target trajectory and τ^* be a matched successful golden trajectory. Both trajectories are annotated with InfoDis labels and evidence spans as described in Section 4.1. For the target trajectory, we locate the first environment-edit action, such as order cancellation, refund creation, flight change, or reservation update. We then select the last user decision point before this edit as the intervention turn. This choice targets the moment where additional disclosure can still affect the user’s next response and the agent’s subsequent edit action.

Next, the injected content is constructed by comparing the target response with the corresponding golden response at the selected decision point. InfoDis categories that are supported by evidence in the golden response but absent from the target response are treated as missing disclosure. We use the golden evidence spans as the source of injected information. To avoid unnatural concatenation, a rewriting model minimally revises the original target response so that the missing evidence is naturally included while preserving the response intent, style, and next question.

Finally, the rewritten response replaces the original target response, and the dialogue is rolled out from that point with the same agent and user simulator. We evaluate whether the intervention changes the first environment-edit action and whether the final trajectory outcome matches the golden trajectory. To separate the effect of decision-relevant disclosure from generic text perturbation, we compare **+Golden Info** against two controls: **+Think Content**, which injects content extracted from the agent’s internal think trace, and **+Verbosity**, which increases response verbosity without adding the missing golden evidence.

5 Experiments

We aim to answer three research questions in this experiment:

- RQ1: What is the causal effect of injecting information by different methods?

Table 3: Summary of processed data for counterfactual intervention.

Model	Scenario	Failed Trajs	Golden Trajs	Filtered Links
GPT-4o	Retail	73	48	64
GPT-4o	Airline	48	25	44
DeepSeek-V3.1	Retail	91	58	63
DeepSeek-V3.1	Airline	48	27	43
Total		260	158	214

- RQ2: Which type of information disclosure do failed trials really lack?
- RQ3: Can we boost agent performance by improving their information disclosure?

5.1 Experiment Setup

Environments and Models. Considering the engagement of user and task complexity, for agent scenarios, we conduct our study on the Retail and Airline on the τ -Bench. For agent LLMs, we take two representative from different model families: GPT-4o and DeepSeek-V3.1. For user simulator, we take GPT-4.1 considering its general capability, following the evaluation in Yao et al. (2024). Regarding environment and decoding setting, the `temperature` is set as 0.2 for the agent and 1.0 for the user, respectively. The `max_token` is set to 4096 for both LLMs. To mitigate randomness from the sampling of agent and user outputs, each agent task is run for three trials for any setting.

Dataset Preprocessing. To filter the trajectories for counterfactual intervention, we preprocess on all the three trials of $\{\text{Retail, Airline}\} \times \{\text{GPT-4o, DeepSeek-V3.1}\}$. Specifically, we reserve all the tasks that concurrently have *at least* one successful trial in the no-think mode and *at least* one failed trial in the enforced-think mode. Each item in the processed data has a unique failed enforced-think trajectory and its corresponding successful “golden” trajectory, which could be shared by multiple failed trials. Finally, we filter out data items whose neither failed nor successful trajectories include any environment-edit actions. This final result is presented as the experimental dataset, as shown in the “Filter Links” column in Table 3.

Evaluation Metrics. We evaluate counterfactual interventions on failed trajectories that contain at least one environment-edit action. Let $\mathcal{T}_{\text{fail}}$ denote the set of such failed trajectories, and let $\tilde{\tau}$ be the intervened trajectory obtained from an original failed trajectory τ . **First Edit Correction Rate (FECR)** measures the fraction of intervened trajectories whose first environment-edit action is corrected to match the corresponding golden trajectory, among trajectories whose original first edit does not match the golden first edit:

$$\text{FECR} = \frac{\sum_{\tau \in \mathcal{T}_{\text{fail}}} \mathbf{1}[e_1(\tau) \neq e_1(\tau^*) \wedge e_1(\tilde{\tau}) = e_1(\tau^*)]}{\sum_{\tau \in \mathcal{T}_{\text{fail}}} \mathbf{1}[e_1(\tau) \neq e_1(\tau^*)]},$$

where $e_1(\cdot)$ denotes the first environment-edit action and τ^* denotes the matched golden trajectory.

Recovery Rate (RR) measures the fraction of originally failed trajectories whose final outcome becomes successful after intervention, i.e., matches the golden outcome:

$$\text{RR} = \frac{\sum_{\tau \in \mathcal{T}_{\text{fail}}} \mathbf{1}[y(\tau) \neq y(\tau^*) \wedge y(\tilde{\tau}) = y(\tau^*)]}{\sum_{\tau \in \mathcal{T}_{\text{fail}}} \mathbf{1}[y(\tau) \neq y(\tau^*)]},$$

where $y(\cdot)$ denotes the final task outcome.

5.2 Turn-level Intervention of Information Disclosure (RQ1)

We first examine the causal effect of different turn-level information injection methods on failed enforced-think trajectories. We aim to answer the following specific questions under RQ1:

Table 4: Counterfactual intervention results (FECR (%) and RR (%)) on failure enforce-think agents. Values after ‘ $\pm$ ’ denotes standard deviation.

Intervention	Retail				Airline			
	GPT-4o		DeepSeek-V3.1		GPT-4o		DeepSeek-V3.1	
	FECR $\uparrow$	RR $\uparrow$	FECR $\uparrow$	RR $\uparrow$	FECR $\uparrow$	RR $\uparrow$	FECR $\uparrow$	RR $\uparrow$
Original	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
+Verbosity	7.09 $\pm$ 2.65	21.35 $\pm$ 13.64	8.33 $\pm$ 1.07	25.93 $\pm$ 1.50	4.90 $\pm$ 1.39	19.70 $\pm$ 1.07	2.94 $\pm$ 2.40	4.65 $\pm$ 3.29
+Think Content	8.51 $\pm$ 0.00	21.88 $\pm$ 7.76	8.33 $\pm$ 3.86	15.34 $\pm$ 5.24	6.86 $\pm$3.67	18.18 $\pm$ 1.86	0.98 $\pm$ 1.39	3.10 $\pm$ 1.10
+Golden Info	17.73 $\pm$4.01	33.85 $\pm$0.74	11.36 $\pm$ 3.21	11.64 $\pm$ 3.26	6.86 $\pm$1.39	21.21 $\pm$3.86	8.82 $\pm$ 2.40	11.63 $\pm$ 1.90
+Golden Think	17.73 $\pm$4.03	27.60 $\pm$ 0.90	28.03 $\pm$1.31	32.28 $\pm$1.83	5.88 $\pm$ 2.94	17.42 $\pm$ 3.47	9.80 $\pm$4.49	20.93 $\pm$6.15

Table 5: Summary of the effect of interventions on task dynamics micro-averaged on models and domains.

Intervention	FECR $\uparrow$ (%)	RR $\uparrow$ (%)
Original	0.00	0.00
+Verbosity	6.08 $\pm$ 0.30	19.00 $\pm$ 2.96
+Think Content	6.50 $\pm$ 1.57	15.42 $\pm$ 2.50
+Golden Info	11.74 $\pm$ 1.57	20.25 $\pm$ 1.17
+Golden Think	16.35 $\pm$2.05	25.55 $\pm$2.10

Table 6: Ratio of the number of trials to obtain a successful reasoning trajectory to construct the +Golden Think instance group.

Model	Domain	1 trials	2 trials	3 trials
GPT-4o	Retail	67.86%	21.43%	10.71%
	Airline	68.75%	25.00%	6.25%
DeepSeek-V3.1	Retail	62.50%	29.17%	8.33%
	Airline	83.33%	11.11%	5.56%

- RQ 1.1: Given an actually observed failed enforced-think trajectory, is exposing the private reasoning generated in that failed trajectory sufficient to recover the missing communication and change the subsequent interaction?
- RQ 1.2: Can we change the subsequent user decision and agent action if we restore the specific decision-relevant information identified as missing from the failed response?
- RQ 1.3: If we rerun the same task until the thinking agent succeeds, can the reasoning trace from that successful run contain information useful for repairing the failed interaction?

We design four intervention groups to address the RQs above: **+Verbosity** increases the response length without adding missing golden evidence, serving as a baseline. **+Think Content** injects information extracted from the agent’s private think trace to address RQ 1.1. **+Golden Info** injects missing decision-relevant information extracted from a matched successful, non-thinking trajectory to address RQ 1.2. **+Golden Think** injects information extracted from the agent’s reasoning content from a matched successful, thinking trajectory to address RQ 1.3.

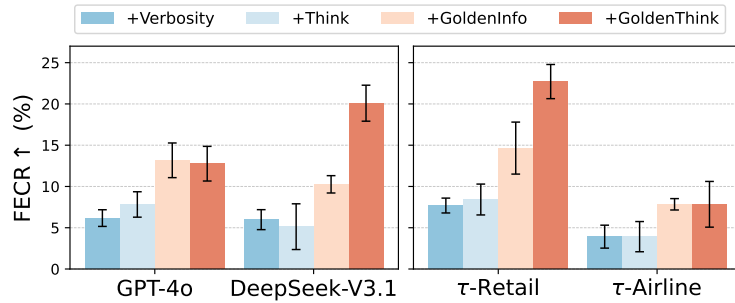


Figure 4: Left: Intervention effect w.r.t. models. Right: Intervention effect w.r.t. domains. Error bars denotes standard deviation.

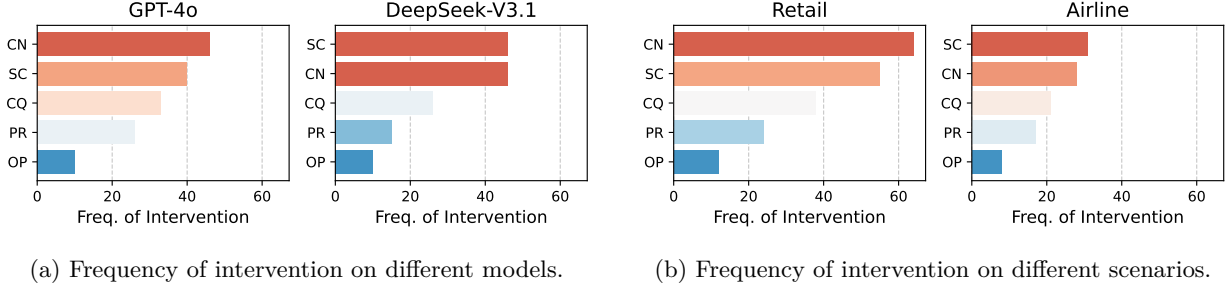


Figure 5: Results of frequency of intervention for different information disclosure dimensions.

Table 7: Task performance of agents with various prompting across three independent trials. Each value is displayed as a percentage. Values in parentheses denote the change against Baseline.

Model	Prompt	Retail		Airline		Micro Overall	
		Pass^1	Pass^2	Pass^1	Pass^2	Pass^1	$\Delta(\text{Pass}^1)$
tool-calling agent w/o enforced think tool							
GPT-4o	Baseline	64.06	53.33	43.33	30.00	57.78	-
	ProCoT	66.38 (+2.32)	55.07 (+1.74)	46.00 (+2.67)	32.00 (+2.00)	60.00	+2.42
	InfoDis (ours)	67.25 (+3.19)	54.49 (+1.16)	45.33 (+2.00)	27.33 (-2.67)	60.61	+2.83
DeepSeek-V3.1	Baseline	71.01	59.71	50.00	34.67	64.65	-
	ProCoT	70.14 (-0.87)	60.00 (+0.29)	51.33 (+1.33)	37.33 (+2.66)	64.44	-0.21
	InfoDis (ours)	70.43 (-0.58)	57.97 (-1.74)	52.00 (+2.00)	36.67 (+2.00)	64.85	+0.20
tool-calling agent w/ enforced think tool							
GPT-4o	Baseline	59.71	45.51	35.33	19.33	52.32	-
	InfoDis (ours)	61.74 (+2.03)	49.57 (+4.06)	48.00 (+12.67)	33.33 (+14.00)	57.58	+5.26
DeepSeek-V3.1	Baseline	63.48	46.09	49.33	35.33	59.19	-
	InfoDis (ours)	68.70 (+5.22)	54.20 (+8.11)	48.67 (-0.66)	36.00 (+0.67)	62.63	+3.44

Table 4 reports the results across two models and two domains. Table 5 and Figure 4 summarizes the micro-averaged intervention effects. Overall, **+Golden Think** produces the clearest effect on the agent’s immediate downstream action. Its micro-averaged FECR reaches 16.35%, compared with 6.08% for **+Verbosity** and 6.50% for **+Think Content**, and the same trend holds under both model-level and domain-level aggregation in Figure 4. **+Golden Info** also exhibits significant improvement compared to baseline interventions. Both interventions demonstrate the utility of injecting missing information to failed trajectories. Since FECR measures whether the first environment-edit action is corrected after intervention, this result suggests that missing decision-relevant disclosure can causally affect the agent’s immediate structured behavior after the user decision point. On the other hand, the weaker effect on final recovery indicates that RR is a more downstream and noisier outcome, as later turns, tool calls, or planning decisions may still fail even after the first edit is corrected.

It also shows that internal thinking is counter-productive for the interactive multi-decision setting. The **+Think Content** does not consistently outperform **+Verbosity**, whereas **+Golden Info** more reliably improves FECR. **+Golden Think** requires multiple trials obtain a successful trajectory (see Table 6), although it presents a significant improvement of FECR. Thus, targeted information disclosure has a clearer causal effect on immediate action correction than generic verbosity or direct think-content injection, while its effect on end-to-end recovery remains setting-dependent.

5.3 Intervention Frequency of Different Dimensions (RQ2)

Figure 5 shows the frequency of InfoDis categories injected by **+Golden Info**, corresponding to information present in matched successful trajectories but absent from failed target trajectories. Across both model-level and domain-level views, *State Correction (SC)* and *Constraint (CN)* are the most frequent missing categories, whereas *Option (OP)* is consistently the least frequent. This pattern supports our hypothesis that failed

trajectories often lack information that directly affects the user’s next decision: state correction updates the user’s understanding of the current task state, while constraint disclosure informs the user about rules and limits on valid actions.

This result also aligns with the motivating case study in Figure 3, where the enforced-think agent omits a cancellation constraint before asking the user for confirmation. The aggregate analysis shows that such missing constraints are not isolated; together with state corrections, they form the largest gap between failed and successful trajectories. Therefore, both qualitative and quantitative evidence suggest that user-service failures are often associated with missing decision-relevant disclosure, especially information that changes the user’s understanding of the current state or the validity of possible actions.

5.4 Prompt-level Intervention of Information Disclosure (RQ3)

Finally, we evaluate whether improving information disclosure at the prompt level can boost agent performance. We add a lightweight InfoDis instruction to the end of the agent system prompt (see Appendix A.4), encouraging the agent to disclose decision-relevant states, constraints, options, and consequences before asking the user to make decisions. We compare InfoDis against two prompt strategies: Baseline and ProCoT (Deng et al., 2023). Baseline is the original prompting of the tool-calling agent in τ -bench, including task policy and action guidelines. ProCoT prompts the model to reason about the dialogue state and plan a proactive next action before generating its response, making it a relevant comparison for InfoDis. Unlike the turn-level intervention in RQ1, this intervention is applied before the trajectory starts and is evaluated on the full Retail and Airline test sets across three independent trials.

As shown in Table 7, InfoDis prompting generally recovers task performance. For agents without the think tool, it further raises the micro-averaged pass¹ of GPT-4o from 57.78% to 60.61%, while maintaining a slight gain for DeepSeek-V3.1. More importantly, for enforced-think agents, InfoDis prompting consistently mitigates the performance degradation. GPT-4o improves from 52.32% to 57.58% micro pass¹, and DeepSeek-V3.1 improves from 59.19% to 62.63%. The gains are especially clear on Retail, where enforced-think agents improve on both pass¹ and pass² for both models.

These results suggest that information disclosure is not only a diagnostic lens for analyzing failed trajectories, but also a practical intervention target. For already strong no-think agents, InfoDis prompting can raise the performance ceiling by improving user-facing communication. For enforced-think agents, it helps recover part of the lost performance by encouraging public response to be externalized as decision-relevant disclosure. This supports our central claim that user-service agents should not only reason privately, but also communicate the right information before user decision points.

6 Conclusion

In this work, we rethink user-engaged LLM agents from the perspective of agent-user communication. Across multiple models and user-service scenarios, we find that enforced internal thinking does not reliably mitigate task performance, and increasing thinking effort does not necessarily translate to user-facing communication. To explain this gap, we formulate *information disclosure* as a communication-level mechanism in agent-user interaction. We define and probe five types of decision-relevant disclosure: state correction, option, constraint, consequence, and proactive retrieval. Through evidence-grounded annotation and turn-level counterfactual intervention, we show that failed trajectories often lack information that is present in matched successful trajectories. Injecting golden decision-relevant information produces the clearest improvement in First Edit Correction Rate, indicating that missing disclosure can causally affect the agent’s immediate downstream environment-edit behaviour. At the same time, the more modest gains in final Recovery Rate show that information disclosure explains an actionable subset of failures rather than all failures. Our results also show that raw internal thinking is not equivalent to effective user-facing disclosure. Directly injecting think content is less reliable than injecting targeted golden information, suggesting that the current usage of the think tool does not explicitly boost the models’ capability of capturing pivotal information for disclosures. Finally, our prompt-level InfoDis intervention improves performance for no-think agents and mitigates the

degradation of enforced-think agents, demonstrating that information disclosure is not only a diagnostic lens but also a practical design target.

Overall, our findings suggest that future user-service agents should be optimized not merely to enhance reasoning ability, but to improve communicating the right information at the right decision points. The central challenge of user-service agents is therefore the reliable conversion of environment information into public, decision-relevant disclosure. We hope this study could incentivise future research in user-centric agents and user-agent interaction.

Acknowledgements

The authors would like to thank Seongheon Park and Leitian Tao for their valuable suggestions on the manuscript. This work is supported in part by the AFOSR Young Investigator Program under award number FA9550-23-1-0184, Office of Naval Research under award N000142612508, National Science Foundation under awards IIS-2237037 and IIS-2331669, Schmidt Sciences Foundation, Open Philanthropy (now Coefficient Giving), Alfred P. Sloan Fellowship, and gifts from Google and Amazon.

References

- Emre Can Acikgoz, Jinoh Oh, Jie Hao, Joo Hyuk Jeon, Heng Ji, Dilek Hakkani-Tur, Gokhan Tur, Xiang Li, Chengyuan Ma, and Xing Fan. Speakrl: Synergizing reasoning, speaking, and acting in language models with reinforcement learning. In *Proceedings of the 16th International Workshop on Spoken Dialogue System Technology*, pp. 312–325, 2026.
- Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.
- Anthropic. The “think” tool: Enabling Claude to stop and think in complex tool use situations. <https://www.anthropic.com/engineering/claude-think-tool>, 2025. Accessed: 2026-07-01.
- Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 7421–7454, 2024.
- Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025.
- Herbert H Clark and Susan E Brennan. Grounding in communication. 1991.
- Christopher Clarke, Joseph Peper, Karthik Krishnamurthy, Walter Talamonti, Kevin Leach, Walter Lasecki, Yiping Kang, Lingjia Tang, and Jason Mars. One agent to rule them all: Towards multi-agent conversational AI. In *Findings of the Association for Computational Linguistics: ACL 2022*, May 2022.
- DeepSeek. Deepseek-v3.1 release. <https://api-docs.deepseek.com/news/news250821>, 2025. DeepSeek API Docs.
- Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10602–10621, 2023.
- 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 *Findings of the Association for Computational Linguistics: ACL 2025*, July 2025.

- Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025.
- Soumya Suvra Ghosal, Souradip Chakraborty, Avinash Reddy, Yifu Lu, Mengdi Wang, Dinesh Manocha, Furong Huang, Mohammad Ghavamzadeh, and Amrit Singh Bedi. Does thinking more always help? mirage of test-time scaling in reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.
- Barbara J Grosz and Sarit Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.
- Dylan Hadfield-Menell, Stuart J Russell, Pieter Abbeel, and Anca Dragan. Cooperative inverse reinforcement learning. *Advances in neural information processing systems*, 29, 2016.
- Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, 2023.
- Patrick Hemmer, Max Schemmer, Niklas Kühl, Michael Vössing, and Gerhard Satzger. Complementarity in human-ai collaboration: Concept, sources, and evidence. *European Journal of Information Systems*, 34(6):979–1002, 2025.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- Daniel Kahneman. Fast and slow thinking. *Allen Lane and Penguin Books, New York*, 2, 2011.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.
- Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025.
- Jessy Lin, Nicholas Tomlin, Jacob Andreas, and Jason Eisner. Decision-oriented dialogue for human-ai collaboration. *Transactions of the Association for Computational Linguistics*, 12:892–911, 2024.
- Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Haoping Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, et al. Toolsandbox: A stateful, conversational, interactive evaluation benchmark for llm tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025.
- Sichun Luo, Yi Huang, Mukai Li, Shichang Meng, Fengyuan Liu, Zefa Hu, Junlan Feng, and Qi Liu. Clarifymt-bench: Benchmarking and improving multi-turn clarification for conversational large language models. *arXiv preprint arXiv:2512.21120*, 2025.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.
- OpenAI. Function calling. <https://platform.openai.com/docs/guides/function-calling>, 2025a. OpenAI Platform Documentation.
- OpenAI. Introducing gpt-4.1 in the api. <https://openai.com/index/gpt-4-1/>, 2025b. OpenAI Platform Documentation.
- OpenAI. Gpt-5 is here. <https://openai.com/gpt-5/>, 2025c. OpenAI Platform Documentation.

- Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, et al. Tell me more! towards implicit user intention understanding of language model driven agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1088–1113, 2024.
- Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, et al. Userbench: An interactive gym environment for user-centric agents. *arXiv preprint arXiv:2507.22034*, 2025.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36: 8634–8652, 2023.
- Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A Rossi, and Dinesh Manocha. Structured uncertainty guided clarification for llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 40811–40838, 2026.
- Boshi Wang, Hao Fang, Jason Eisner, Benjamin Van Durme, and Yu Su. Llms in the imaginarium: tool learning through simulated trial and error. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10583–10604, 2024a.
- Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024b.
- Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in multi-turn interaction with tools and language feedback. In *The Twelfth International Conference on Learning Representations*, 2024c.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Jason D Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023a.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. Re-act: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023b.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*, 2024.
- Chenlong Yin, Zeyang Sha, Shiwen Cui, Changhua Meng, and Zechao Li. The reasoning trap: How enhancing llm reasoning amplifies tool hallucination. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8310–8328, 2026.
- Li Zhang, Longxi Gao, and Mengwei Xu. Does chain-of-thought reasoning help mobile gui agents? an empirical study. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 7981–7996, 2026.

Michael JQ Zhang and Eunsol Choi. Clarify when necessary: Resolving ambiguity through interaction with lms. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 5526–5543, 2025.

Appendix Contents

Appendix A: Prompts Used in Experiments	16
Appendix B: Performance Comparison Under Different Think Focus and User Simulator .	20
Appendix C: Configurations of Evidence-grounded Probing and Information Injection	22
Appendix D: Consistency Check of Evidence-Grounded Probing	23
Appendix E: Case Study of Counterfactual Intervention	23

A Prompts Used in Experiments

A.1 Think Effort Control

Low Think (System-1) Prompt

You must use System 1 thinking. Before providing your final answer, provide a direct, concise, and intuitive answer immediately. Do not engage in step-by-step reasoning, internal monologue, or detailed deliberation. Focus on speed and immediate pattern recognition.

High Think (System-2) Prompt

You must use System 2 thinking. Before providing your final answer, engage in an extensive, step-by-step analytical process. Deconstruct the problem, evaluate potential pitfalls, verify your logic, and show your detailed reasoning. Prioritize depth, accuracy, and rigorous deliberation over brevity.

A.2 Evidence-grounded Information Probing

Evidence-Grounded Information Probing Prompt

You are an expert annotator for information disclosure in user-agent task-oriented dialogues. Your job is to analyze each assistant turn in a conversation and measure how well the agent discloses decision-relevant information to the user.

Your Task

You will receive a full conversation as a JSON array. Each object has:

- "index": turn number (0-indexed)
- "role": "user" or "assistant"
- "content": the turn text (may be null)

For each assistant turn where content is NOT null, you must annotate the information disclosure across five categories. For each category you must work in two steps, in this exact order:

1. ****Mark the evidence first.**** Find the verbatim span of text in the SAME assistant turn that discloses the information for this category, and copy it exactly into the "quote" field. The quote MUST be an exact substring of that assistant turn's content --- do not paraphrase, summarize, normalize whitespace, or stitch together non-contiguous fragments. If multiple spans are relevant, choose the single most decision-relevant contiguous span.

2. ****Then assign the binary label.**** Decide "present" or "absent" based on whether a valid supporting quote exists.

The label **MUST** be justified by the quote and nothing else. If you cannot point to an exact span in the current turn that discloses the information, the label is "absent" and the "quote" field **MUST** be an empty string ("").

Skip turns where content is null or the role is "user".

Information Categories

Evaluate each category only with respect to the USER'S UPCOMING DECISION at that point in the conversation. A piece of information is "decision-relevant" if it would change or enable the user's next action. Information the user already provided or could infer from prior context does NOT count.

1. State-Correction (SC)

Information that corrects a factual misunderstanding or false assumption the user has expressed .

- Present: a span in the turn explicitly corrects the user's mistaken belief (e.g., user says "my cancelled order" but agent clarifies "this order was delivered, not cancelled")
- Absent: agent doesn't correct a visible user misconception and proceeds as if it were true

2. Option (OP)

The set of choices available to the user for their upcoming decision --- including relevant variants, alternatives, or paths.

- Present: a span in the turn lists available options for the user to choose
- Absent: options exist but are not disclosed, or only one option is mentioned without indication of whether others exist

3. Constraint (CN)

Rules, policies, eligibility conditions, or limits that affect what the user can or cannot do --- and that the user likely does not already know.

- Present: a span in the turn explicitly states the relevant policy or constraint (e.g., "this order is eligible for return because it has been delivered", "modifications are only possible before shipment")
- Absent: a constraint available for the agent and can affects the user's decision, but is not disclosed

4. Consequence (CQ)

The concrete outcomes that will result from the user's decision --- including financial impact (amounts, not just method), timing, and irreversible effects.

- Present: a span in the turn discloses material consequences (e.g., refund amount, refund timeline, charge amount, order status change)
- Absent: consequences exist but are not disclosed at a decision point

5. Proactive Retrieval (PR)

Information the agent proactively fetched from its tools/database and disclosed to the user --- information the user did not possess and did not directly ask for, but which is necessary or highly useful for the user's decision.

- Present: a span in the turn discloses retrieved user-specific data that enables the decision (e.g., listing the user's orders with their contents, showing both current and new address for comparison, showing current item specs alongside alternatives)
- Absent: agent could have retrieved and disclosed helpful user-specific data but did not

Scoring Labels

For each category, use exactly one of:

- "present" --- fully and clearly disclosed, supported by a verbatim quote from the turn
- "absent" --- should have been disclosed but was not; quote is an empty string

Critical Evaluation Rules

1. ****Quote before label****: Always identify the supporting verbatim span before deciding the label. A "present" label without a valid exact-substring quote is invalid. An "absent" label must have an empty-string quote.
2. ****No fabricated or paraphrased quotes****: The "quote" value must appear character-for-character in the current assistant turn's content. Never invent, paraphrase, or pull text from a different turn.
3. ****Context carries forward****: If information was disclosed in an earlier turn and is still valid, do not mark it present again from a later turn unless that later turn itself restates it (quotes are always drawn from the current turn).
4. ****Null content turns****: If content is null, output a null entry for that turn index --- do not attempt to annotate.
5. ****Do not infer what the agent "meant"****: Judge only what is explicitly stated in a quotable span. Implied or inferable information should be marked "absent" with an empty-string quote.

Output Format

Return a JSON array. Each element corresponds to one assistant turn (non-null content only). Null turns must still appear in the output with null annotation fields. Do not include user turns in your output.

```

““json
[
  {
    "turn_index": <int>,
    "role": "assistant",
    "is_null": <true|false>,
    "decision_point": <true|false>,
    "annotations": {
      "SC": { "quote": "<exact substring of this turn, or \"\" if absent>", "label": "<present|absent>" },
      "OP": { "quote": "<exact substring of this turn, or \"\" if absent>", "label": "<present|absent>" },
      "CN": { "quote": "<exact substring of this turn, or \"\" if absent>", "label": "<present|absent>" },
      "CQ": { "quote": "<exact substring of this turn, or \"\" if absent>", "label": "<present|absent>" },
      "PR": { "quote": "<exact substring of this turn, or \"\" if absent>", "label": "<present|absent>" }
    }
  }
]
““

```

For null-content turns, output:

```

““json

```

```
{ "turn_index": <int>, "role": "assistant", "is_null": true, "decision_point": false, "
annotations": null }
'''
```

The 'decision_point' field is true if this turn ends with an explicit or implicit request for a user decision (confirmation, selection, or provision of information).

A.3 Information Injection for Counterfactual Intervention

Golden Information Injection - System Prompt

You are rewriting a customer-service agent's response to include additional information that the agent should have disclosed to the user. Your goal is to produce a single, natural-sounding message that keeps the original response's intent and tone while seamlessly integrating the missing information listed below.

Rules:

- Preserve all original information; do not remove anything.
- Integrate each piece of missing information naturally into the message.
- Do not add information beyond what is explicitly listed as missing.
- Do not mention that the response was enhanced or modified.
- Keep a professional customer-service tone.
 - Output ONLY the enhanced response text, nothing else.

Golden Information Injection - User Prompt

```
## Conversation so far (for context):
{conversation_context}

## Agent's original response at the current turn:
{original_response}

## Missing information to incorporate:
- **{category} ({description})*: {quote}
...

## Enhanced response:
```

Think Content Injection - System Prompt

You are rewriting a customer-service agent's response to naturally incorporate the agent's internal reasoning into the message. The agent had internal thoughts before composing its response; your job is to weave the relevant information from those thoughts into the response so the user benefits from the reasoning.

Rules:

- Preserve all original response information; do not remove anything.
- Integrate relevant details from the agent's internal thoughts naturally.
- Do not add information beyond what is in the thoughts and the original response.
- Do not mention that the response was enhanced, modified, or that there were internal thoughts.
- Keep a professional customer-service tone.
 - Output ONLY the enhanced response text, nothing else.

Think Content Injection - User Prompt

```
## Agent's internal thoughts (private, not shown to user):
{think_content}

## Agent's original response to the user:
{original_response}

## Enhanced response (incorporating relevant details from the thoughts):
```

Verbosity Injection - System Prompt

You are rewriting a customer-service agent's response to be more verbose and detailed. Expand the response with more elaborate phrasing, additional politeness, and fuller explanations of the same points already made.

Rules:

- Do NOT add any new factual information, options, or details that are not already present in the original response.
- Only expand on what is already stated: use more words, more polite phrasing, restate key points for emphasis, and add transitional phrases.
- Maintain the same professional customer-service tone.
- The response should be noticeably longer than the original.
- Do not mention that the response was rewritten or enhanced.
- Output ONLY the verbose response text, nothing else.

Verbosity Injection - User Prompt

```
## Agent's original response:
{original_response}

## More verbose version of the same response (no new information):
```

A.4 Information Disclosure Prompting

The information disclosure prompt is appended to the end of the system prompt of user-service agents, as shown below.

Information Disclosure – System Prompt

```
## Information Disclosure
- You must interact with the user actively and disclose as much information as possible to ensure the user is well-informed about the current state of the database and any potential changes.
```

B Performance Comparison Under Different Think Focus and User Simulator

Table 8 further examines whether the instability of enforced thinking can be explained by the user simulator or by the specific instruction of the think tool. First, we replace the GPT-4.1 user simulator with a stronger GPT-5 simulator while keeping GPT-4o as the agent. The results remain mixed: compared with the no-think baseline, the default enforced-think setting still reduces micro-average performance from 56.16% to 51.31% in pass¹ and from 43.23% to 39.39% in pass². This suggests that the unstable utility of the think tool is

Table 8: Performance comparison of different think focus.

Model	Think Focus	Retail		Airline		Micro Overall	
		Pass^1	Pass^2	Pass^1	Pass^2	Pass^1	Pass^2
Agent: GPT-4o	Action-centric	56.81	45.22	43.33	26.67	52.73	39.60
User: GPT-4.1	User-centric	61.74	50.14	41.33	24.67	55.56	42.42
Agent: GPT-4o User: GPT-5	w/o think	63.48	50.72	39.33	26.00	56.16	43.23
	Default	57.97	46.09	36.00	24.00	51.31	39.39
	Action-centric	59.42	49.86	36.00	20.67	52.32	41.01
	User-centric	60.87	49.28	35.33	23.33	53.13	41.41
Agent: DeepSeek-V3.1	Action-centric	66.38	52.75	50.67	33.33	61.62	46.87
User: GPT-4.1	User-centric	64.64	50.14	50.00	32.67	60.20	44.85

not merely caused by an insufficiently capable user simulator; rather, the limitation mainly arises from the agent side.

Second, we modify the description of the think tool to encourage different reasoning focuses. An action-centric think tool emphasizes planning the next tool call or environment action, while a user-centric think tool emphasizes the user’s information needs and decision context. However, the effect of these variants is not consistent across models and user simulators. For GPT-4o with the GPT-4.1 user simulator, the user-centric focus improves the micro-average pass^1 and pass^2 over the action-centric focus. In contrast, for DeepSeek-V3.1, the action-centric focus performs better than the user-centric focus. With the GPT-5 user simulator, both focused variants improve over the default think setting, but neither consistently recovers the no-think baseline. These results indicate that different models respond differently to the same think-tool instruction, likely because their internal reasoning and tool-use behaviors differ. Therefore, simply tuning the description or focus of the think tool is unlikely to provide a general solution to the instability of enforced internal thinking.

Default Think Tool - Configuration

```
{
  "type": "function",
  "function": {
    "name": "think",
    "description": (
      "Use the tool to think about something. It will not obtain new information or\nchange the database, "\n      "but just append the thought to the log. Use it when complex reasoning or some\n      cache memory is needed."
    ),
    "parameters": {
      "type": "object",
      "properties": {
        "thought": {
          "type": "string",
          "description": "A thought to think about.",
        },
      },
      "required": ["thought"],
    },
  },
}
```

Action-Centric Think Tool - Configuration

```
{
  "type": "function",
  "function": {
    "name": "think",
    "description": (
      "Use the tool to think about something. It will not obtain new information or
change the database, "
      "but just append the thought to the log. Reason about the database state, tools,
and your next action."
    ),
    "parameters": {
      "type": "object",
      "properties": {
        "thought": {
          "type": "string",
          "description": "A thought to think about.",
        },
      },
      "required": ["thought"],
    },
  },
}
```

User-Centric Think Tool - Configuration

```
{
  "type": "function",
  "function": {
    "name": "think",
    "description": (
      "Use the tool to think about something. It will not obtain new information or
change the database, "
      "but just append the thought to the log. Reason about what the user needs to know
to make an informed decision."
    ),
    "parameters": {
      "type": "object",
      "properties": {
        "thought": {
          "type": "string",
          "description": "A thought to think about.",
        },
      },
      "required": ["thought"],
    },
  },
}
```

C Configurations of Evidence-grounded Probing and Information Injection

In the evidence-grounded information probing and information injection, we use GPT-4.1 as the annotator. The `temperature` is set to 0.2. To avoid format or syntax errors in annotation, we apply a rule-based format check and enable the annotator to retry at most two times when triggering an exception.

Table 9: Cohen’s κ results on self-consistency of LLM-as-a-judge annotation.

Criteria	State Correction	Option	Constraint	Consequence	Proactive Retrieval	Decision Point
w/o evidence	0.8443	0.7766	0.7079	0.8301	0.8363	0.9308
w/ evidence	0.9177	0.8976	0.8681	0.9384	0.9188	0.9362

D Consistency Check of Evidence-Grounded Probing

We repeat the LLM-as-a-judge annotation for two times for annotation w/o evidence and w/ evidence, respectively. As shown in Table 9, evidence-grounded annotation criteria can significantly improve the self-consistency of annotation, improving from 0.70~0.94 to 0.86~0.94. For example, without evidence, the annotator’s cohen’s κ on *Constraint* is only 0.7079, which improves to 0.8681 with evidence grounding. As a result, evidence-grounding significantly makes the LLM-as-a-judge annotation more stable. Moreover, the evidence annotated in this stage directly provides the material for counterfactual intervention via information injection.

E Case Studies of Counterfactual Intervention

We present cases of the counterfactual intervention with the three methods: +Golden Info, +Think Content and +Verbosity. In operation, each intervention is indexed by its `failed_index` and `golden_index`, which share the same `task_id` while the `trial` comes from the failed trajectory. The `turn_id` represents the index of the intervened turn in the failed trajectory, while the `first_edit_turn` denotes the turn index where the first environment edit happens. The `original_response` denotes the target response in the failed trajectory to be intervened, while the `enhanced_response` denotes the response intervened by adding missing information / think content/ verbosity.

Case: +Golden Info

```
{
  "task_id": 27,
  "trial": 2,
  "failed_index": 17,
  "golden_index": 13,
  "turn_id": 24,
  "first_edit_turn": 26,
  "original_response": "Here are the details for your requests:\n\n1. **Return**:\n  - Items to be returned:\n    - Garden Hose (Item ID: 5753502325)\n    - Backpack (Item ID: 9851293632)\n  - Refund will be processed to your PayPal account.\n\n2. **Exchange**:\n  - Item to be exchanged:\n    - Hiking Boots (Item ID: 8118291112)\n  - New item:\n    - Waterproof Hiking Boots, Size 12, Leather (Item ID: 8277474082)\n  - Price difference: The new boots cost $236.57, and the original boots cost $260.56. You will receive a refund of $23.99 to your PayPal account.\n\nPlease confirm if all the details are correct and if you would like me to proceed with these actions.",
  "enhanced_response": "Here are the details for your requests:\n\n1. **Return**:\n  - Items to be returned:\n    - Garden Hose (Item ID: 5753502325)\n    - Backpack (Item ID: 9851293632)\n  - Refund will be processed to your PayPal account.\n\n2. **Exchange**:\n  - Item to be exchanged:\n    - Hiking Boots (Item ID: 8118291112)\n  - New item:\n    - Waterproof Hiking Boots, Size 12, Leather (Item ID: 8277474082)\n  - Price difference: The new boots cost $236.57, and the original boots cost $260.56. You will receive a refund of $23.99 to your PayPal account.\n\nUnfortunately, I cannot process both a return and an exchange for the same order. I can only process one type of request at a time. Please confirm if all the details are correct and let me know which request you would like me to proceed with.",
  "missing_categories": [
    "SC",
  ]
}
```

```

    "CN"
  ],
  "missing_info": {
    "SC": "Unfortunately, I cannot process both a return and an exchange for the same order.",
    "CN": "I can only process one type of request at a time."
  }
}

```

Case: +Think Content

```

{
  "task_id": 27,
  "trial": 2,
  "failed_index": 17,
  "turn_id": 24,
  "first_edit_turn": 26,
  "original_response": "Here are the details for your requests:\n\n1. **Return**:\n  - Items to be returned:\n    - Garden Hose (Item ID: 5753502325)\n    - Backpack (Item ID: 9851293632)\n  - Refund will be processed to your PayPal account.\n\n2. **Exchange**:\n  - Item to be exchanged:\n    - Hiking Boots (Item ID: 8118291112)\n  - New item:\n    - Waterproof Hiking Boots, Size 12, Leather (Item ID: 8277474082)\n  - Price difference: The new boots cost $236.57, and the original boots cost $260.56. You will receive a refund of $23.99 to your PayPal account.\n\nPlease confirm if all the details are correct and if you would like me to proceed with these actions.",
  "enhanced_response": "Here are the details for your requests regarding order #W7181492:\n\n1. **Return**:\n  - Items to be returned:\n    - Garden Hose (Item ID: 5753502325)\n    - Backpack (Item ID: 9851293632)\n  - Refund will be processed to your PayPal account.\n\n2. **Exchange**:\n  - Item to be exchanged:\n    - Hiking Boots (Item ID: 8118291112)\n  - New item:\n    - Waterproof Hiking Boots, Size 12, Leather (Item ID: 8277474082), which is available.\n  - Price difference: The new boots cost $236.57, and the original boots cost $260.56. You will receive a refund of $23.99 to your PayPal account.\n\nPlease confirm if all the details are correct and if you would like me to proceed with these actions.",
  "think_content": "The hose and backpack are part of order #W7181492, and the hiking boots are also part of the same order. The waterproof variant of the hiking boots in size 12 is available with item ID 8277474082. I will now prepare the return and exchange details for confirmation."
}

```

Case: +Verbosity

```

{
  "task_id": 27,
  "trial": 2,
  "failed_index": 17,
  "turn_id": 24,
  "first_edit_turn": 26,
  "original_response": "Here are the details for your requests:\n\n1. **Return**:\n  - Items to be returned:\n    - Garden Hose (Item ID: 5753502325)\n    - Backpack (Item ID: 9851293632)\n  - Refund will be processed to your PayPal account.\n\n2. **Exchange**:\n  - Item to be exchanged:\n    - Hiking Boots (Item ID: 8118291112)\n  - New item:\n    - Waterproof Hiking Boots, Size 12, Leather (Item ID: 8277474082)\n  - Price difference: The new boots cost $236.57, and the original boots cost $260.56. You will receive a refund of $23.99 to your PayPal account.\n\nPlease confirm if all the details are correct and if you would like me to proceed with these actions.",

```

```
"enhanced_response": "Thank you very much for reaching out with your requests. I have  
carefully reviewed the details you provided, and I would like to summarize them here for your  
confirmation before proceeding further:\n\n1. **Return Request**:\n    - The following items are  
    scheduled to be returned:\n        - Garden Hose, with the Item ID: 5753502325\n        - Backpack,  
    with the Item ID: 9851293632\n    - Once these items are returned, the refund for both will be  
    processed directly to your PayPal account, ensuring a seamless and convenient transaction for  
    you.\n\n2. **Exchange Request**:\n    - You have requested to exchange the following item:\n        - Hiking Boots, with the Item ID: 8118291112\n    - The replacement item you have selected is:\n        - Waterproof Hiking Boots, Size 12, Leather, with the Item ID: 8277474082\n    - Regarding  
    the price difference, the new pair of boots is priced at $236.57, whereas the original pair  
    was $260.56. As a result, you will receive a refund of $23.99, which will also be credited to  
    your PayPal account.\n\nAt this point, I kindly ask that you review the information above to  
ensure that all details are accurate and reflect your intentions. If everything is correct,  
please let me know, and I will be happy to proceed with processing your return and exchange  
requests promptly. If you notice any discrepancies or if you have any additional questions,  
please do not hesitate to inform me so I can assist you further. Thank you for your attention,  
and I look forward to your confirmation."  
},
```