Ask Precisely, Assume Wisely: Interactive Evaluation of Ambiguity Resolution in Large Language Models

Anonymous ACL submission

Abstract

Ambiguity is a core challenge in human–AI interaction, where models must decide when to make assumptions and when to seek clarification. Existing approaches to ambiguity evaluation largely focus on coverage over multiple interpretations using static metrics, offering limited insight into how efficiently a model converges toward a single user-intended interpretation through interaction. We introduce a preference-grounded simulated user framework for controlled, multi-turn evaluation of ambiguity resolution, grounding simulator responses strictly in user disambiguation preferences to avoid hallucination and over-disclosure. Alongside this, we define three observable dissatisfaction signals — incorrect assumptions, repeating clarifications, and irrelevant clarifications — that characterize interaction quality beyond task accuracy. Our analysis reveals a clear trade-off: models that frequently seek clarification tend to over-clarify and prolong interactions, while assumption-driven models are more prone to misaligning with user intent. We demonstrate that system prompts optimized on these dissatisfaction signals achieve a Pareto-optimal balance across all three metrics, reducing overall error rates by up to 7.65%, underscoring the value of our framework in designing agents that ask precisely and assume wisely.

1 Introduction

Ambiguity is a fundamental challenge in human–AI interaction. Real-world user queries are often under-specified, requiring language models (LMs) to infer missing intent or engage in clarification through interaction (Keyvan and Huang, 2022). While recent LLMs have demonstrated strong capabilities across reasoning and conversational tasks, effectively resolving ambiguity remains difficult. Models that aggressively assume a single interpretation risk misunderstanding user intent (Berant et al., 2025), whereas models that over-rely on clarification can prolong interactions through repetitive or irrelevant questions (Acikgoz et al., 2026; Andukuri et al., 2024). Designing agents that can *ask precisely and assume wisely* therefore remains an open problem in interactive language systems.

Despite growing interest in interactive evaluation, existing approaches remain limited in two important ways. First, evaluation frameworks largely measure final task success, overlooking the quality and efficiency of the conversational process used to arrive at the answer (Zhang and Choi, 2023; Min et al., 2020). Second, many interactive evaluations rely on LLM-only user simulators to facilitate multi-turn interaction. While flexible, these proxies are difficult to control and reproduce, often prone to hallucination and over-disclosure of information (Dou et al., 2025; Seshadri et al., 2026).

We address both limitations by introducing a **preference-grounded user simulator** for evaluating ambiguity resolution across multiple turns. We formalize each ambiguous query through a user disambiguation preference set as the minimal set of specifications required to uniquely identify the user's intended interpretation. This simulator restricts its responses strictly to these preferences, enabling reproducible and controlled evaluation without risk of information leakage. Derived from this are three observable **dissatisfaction signals** — incorrect assumptions, repeating clarifications, and irrelevant clarifications — that together provide a more nuanced view of how efficiently a model navigates ambiguity during interaction.

Our analysis spanning ambiguous question answering, semantic parsing and text-to-visualization benchmarks reveal a clear trade-off: assumption-driven strategies have higher frequency of wrong attempts while clarification-seeking models prolong conversations while asking ill-scoped questions. By prompt-optimizing on our dissatisfaction signals (DisPO), we achieve pareto optimal balance, leading to cleaner interactions without sacrificing task success.

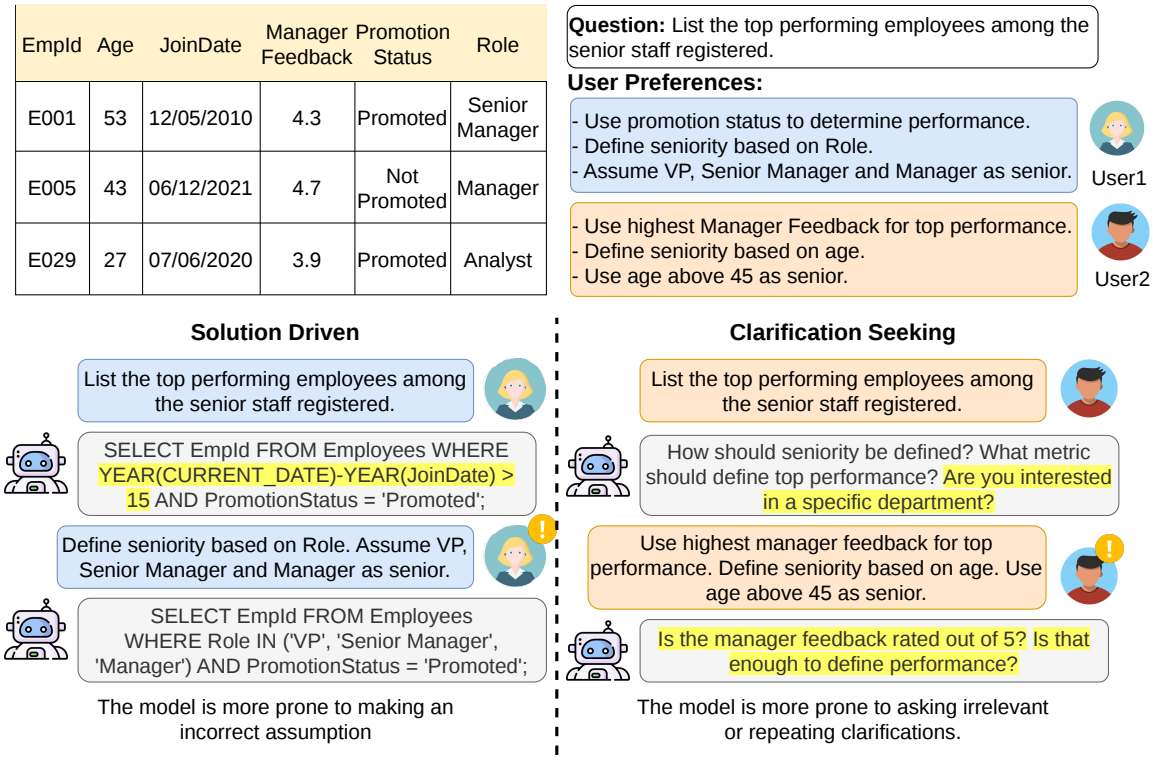

Figure 1: **Demonstrates different ambiguity resolution styles and signals of user dissatisfaction during interaction.** Users may choose different resolution strategies towards a given ambiguity. Solution-driven models often make incorrect assumptions while clarification-seeking models ask repeating or irrelevant clarifications.

Overall contributions can be summarized as:

- We propose a preference-grounded user simulator for controlled and reproducible evaluation towards ambiguity resolution in a multi-turn setting.
- We introduce three dissatisfaction signals that correlate to inefficient ambiguity handling in conversations.
- We demonstrate a clear trade-off between assumption-driven and clarification-seeking strategies, where optimization on our framework yields systems that ask only necessary and precise questions.

2 Problem Formulation

We formalize ambiguity and define how different disambiguation preferences correspond to different user interpretations. We then introduce our simulated user framework, which grounds interactions on these preferences, and describe how it enables measuring user dissatisfaction signals as indicators of ambiguity resolution effectiveness.

2.1 User Disambiguation Preferences

User Disambiguation Preferences are resolution strategies that guide an ambiguous query toward a specific intended interpretation A query q is *ambiguous* if the cardinality of its solution space satisfies $|\mathcal{A}(q)| > 1$, where $\mathcal{A}(q)$ denotes the set of valid solutions consistent with the query. For example, in Figure 1, *seniority* could be interpreted in multiple ways: age, organizational role, or joining date, leading to multiple solutions. To resolve ambiguity toward a target solution $a \in \mathcal{A}(q)$, we introduce a user disambiguation preference set $P = \{p_1, p_2, \ldots, p_n\}$,such that $q \circ P$ admits a unique solution a. We require P to satisfy: (i) *completeness*, where removing any $p_i \in P$ reintroduces ambiguity; (ii) *non-redundancy*, where no specification p_i can be removed or changed without altering solution a; and (iii) *disentangled scope*, where each $p_i \in P$ resolves a distinct aspect of ambiguity, either independently or conditioned on previously resolved ambiguities. For example, in Figure 1, the user preferences resolve: (1) how *performance* should be evaluated; (2) what constitutes *seniority*; (3) the threshold or parameter associated with that

definition (e.g., age threshold or role hierarchy). Each of these are disjoint decision points when a solution is computed. Different assignments of P therefore model different user interpretation toward a specific task. The cardinality $|P|$ symbolizes the extent of uncertainty characterized by the number of decision points at which the model encounters multiple plausible interpretations.

2.2 Preference-Grounded User Simulator

We propose a user simulator that generates responses grounded on the disambiguation preference set P corresponding a given interpretation. Prior work on multi-turn evaluation commonly relies on a single LLM call to serve as user simulator. While they can produce natural conversations (Dou et al., 2025), the interactions remain difficult to control and reproduce consistently across evaluations, and are prone to hallucinated user behavior and unnecessary information disclosure beyond what is required to resolve the ambiguity.

To address these limitations, we design a simulator where the user vocabulary is restricted to the preference set P (see Figure 2). At each interaction turn t, the evaluated agent may either provide a candidate solution or seek clarification. If a solution is proposed, the framework checks whether the generated answer satisfies all specifications in the user preference set P. Any unresolved or incorrectly assumed ambiguity is rectified by responding with: $\{p \in P | p : violated\}$. If the agent instead asks a clarification question, the framework determines whether the question meaningfully targets a specific ambiguity dimension resolved by some p_i. In such cases, the simulated user returns the associated clarification response $\{p \in P | p : asked\}$; otherwise, the user responds with "*Please proceed with your best judgment*" (Vijayvargiya et al., 2025), if the set is empty. The conversation terminates when agent's solution aligns with P. Our goal is not to faithfully emulate real users, but to enable controlled information injection into interaction trajectories for systematically analyzing ambiguity handling by models (Vijayvargiya et al., 2025).

2.3 Measuring Ambiguity Resolution

Beyond task success, our framework measures how efficiently the model resolves ambiguity toward the user's intended interpretation, focusing on three dissatisfaction signals: incorrect assumptions, repeated clarifications, and irrelevant clarifications (Sandbank et al., 2018; Shea et al., 2026). We formalize these dissatisfaction signals through two indicator functions. The *Answer* indicator function $A(t, p) = \mathbb{I}[p \in P \wedge (p = \texttt{violated} \vee p = \texttt{asked})]$, takes the value 1 if the user reveals preference $p \in P$ at interaction turn t and 0 otherwise. Similarly, we define a *Clarification* indicator function $Q(t, p) = \mathbb{I}[p \in P \wedge p = \texttt{asked}]$, which takes the value 1 if the agent asks a clarification question corresponding to ambiguity dimension p at turn t and 0 otherwise. $\overline{P}$ denotes complement of P, suggesting those questions that resolve a preference $p \notin P$, i.e. irrelevant towards resolving ambiguity.

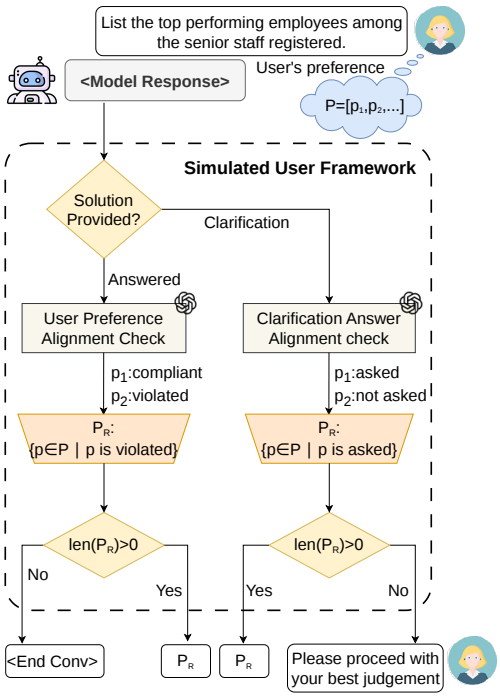

Figure 2: Preference-Grounded User Simulator that *rectifies* or *clarifies* based on preference set P.

Incorrect Assumption. We count the number of times the agent attempted a solution with assumptions different from the user's interpretation during the entire conversation. This is captured by the frequency of repair utterances from the user (Walker et al., 1997) when clarification is not asked.

$$\text{IncAssmp} = \sum_{t=1}^{\tau} \sum_{p \in P} A(t, p)\, \mathbf{1}\{Q(t, p) = 0\}$$

Repeating Clarification. We capture the number of times the agent asks a clarification question, whether relevant to P or not, that has already

been previously asked or resolved by the user in its response earlier, thereby introducing unnecessary conversational repetition and user frustration.

$$\text{RepClar} = \sum_{t=1}^{\tau} \sum_{p \in P \cup \overline{P}} Q(t,p)\, \mathbf{1}\Bigg\{ \sum_{s=1}^{t-1} \left(Q(s,p) + A(s,p)\right) > 0 \Bigg\}$$

Irrelevant Clarification. We measure how often the agent asks clarification questions unrelated to resolving the intended ambiguity. Formally, this corresponds to questions $Q(t,p)$ where $p \in \overline{P}$.

$$\text{IrrelClrf} = \sum_{t=1}^{\tau} \frac{\sum_{p \in \overline{P}} Q(t,p)}{\mathcal{I}(q)^{1-\frac{r(t)}{|P|}}}$$

We scale each irrelevant question detected by $\mathcal{I}(q)^{1-\frac{r(t)}{|P|}}$, where $\mathcal{I}(q)$ denotes the number of sensible interpretations (cardinality of assignment space of P) to ambiguous query q, and $r(t)$ is a monotonic function that stores the count of preferences that were either revealed by the user or the agent got them right at turn t: $r(t) := |P_{\text{revealed}}(t)|, \quad P_{\text{revealed}}(t) \subseteq P$. The scaling factor is added to allow agents room to ask clarification questions more freely earlier when the user intent is unclear, and be more restrictive when a certain amount of information has been revealed over the turns. During the course of the conversation as the simulated user reveals its preference, the solution space reduces by a multiplicative factor and so does the room for asking irrelevant clarifications. Consequently, the penalty increases exponentially over the course of the interaction.

These metrics are not directly comparable, as they arise from different decision contexts and opportunity spaces. Incorrect assumptions occur when the agent chooses to act, while clarification-based errors occur when the agent chooses to ask. Repeating clarification depends on interaction history, whereas irrelevant clarification depends on the remaining ambiguity space. This creates a turn-wise asymmetry: early interactions allow many irrelevant clarifications but few repetitions, while later turns increase repetition likelihood and space for irrelevant queries shrink. To account for this, we normalize irrelevant clarification with lower penalties in early turns and higher penalties in later turns, approximately aligning it with repeated clarification. For example, in a query with two interpretations and a single ambiguity $|P| = 1$, two early irrelevant clarifications may roughly correspond to one repeated clarification, though this alignment is only approximate.

3 Dataset and Metrics

AmbigQA covers ambiguity in open-domain question answering tasks (Min et al., 2020). Built on top of NaturalQuestions (Kwiatkowski et al., 2019), the dataset provides annotated disambiguation variants for each ambiguous query, which we leverage to construct user disambiguation preference sets. We split 1,142 unique queries into 3,649 variants, with an average of 3.2 interpretations per query and an $|P|_{avg} = 1.26$. Given the open-ended nature of responses, we employ GPT-5 (reasoning) as LLM-as-a-judge (Zheng et al., 2023) to evaluate, instead of relying on lexical similarity metrics such as EDIT-F1 or BLEU (Papineni et al., 2002), which are overly strict and prone to higher false positive rates.

Ambrosia covers ambiguity in text-to-SQL semantic parsing tasks (Saparina and Lapata, 2024). The dataset captures diverse forms of ambiguity in tabular queries, including scope ambiguity, attachment ambiguity, and vagueness, along with their corresponding unambiguous variants, which are used to construct the preference set. From 1,141 unique ambiguous queries, we build a total interaction set of 2,689 instances, with an average of 2.36 interpretations per query and $|P|_{avg} = 1.16$. For evaluation, we measure execution accuracy by comparing the execution result of the predicted SQL query with that of the gold query.

nvBench2.0 covers ambiguous queries in *Text2VIS* systems, where models generate visualizations from tabular data (Luo et al., 2026). It introduces ambiguity on existing table-query pairs from nvBench (Luo et al., 2021) and BIRD (Li et al., 2023), and provides stepwise disambiguation reasoning paths, which we leverage to construct user preference sets. From 747 unique ambiguous queries, we derive 2,251 interaction instances, with an average of 3.01 interpretations per query and $|P|_{avg} = 1.55$. Following (Luo et al., 2022, 2021), we evaluate predictions by comparing the generated Vega-Lite specifications with the corresponding gold.

We expand each query into all possible interpretations, enabling ambiguity resolution to be evaluated over the complete interpretation space rather than being biased toward specific user preferences. Additionally In addition to task success, we report the Clean Interaction Rate (CIR%), which measures the percentage of benchmark queries whose conversations contain no *Incorrect Assumption*, *Repeating Clarification* and *Irrelevant Clarification*.

4 Evaluation

We attempt to answer the following questions:

RQ1 How different models and strategies effectively resolve ambiguity?

RQ2 Can dissatisfaction signals reliably guide when agents should ask clarifying questions versus make assumptions?

RQ3 How resolution strategies change with increasing ambiguity?

RQ4 Can our user simulator overcome limitations of existing LLM-only baselines?

4.1 Ambiguity Resolution in LLMs (RQ1)

We evaluate LLMs and different clarification strategies by measuring their ability to resolve ambiguity through interactions with our user simulator.

4.1.1 Evaluating Models

Framework. We evaluate both open-weight: Phi-4 (Microsoft Research, 2024), Deepseek-R1-Distill-Qwen-32B (abbrv. as Deepseek-R1-Distill) (DeepSeek-AI, 2025); and proprietary models: Claude-Opus-4.6 (Anthropic, 2026) and GPT-5.4 (OpenAI, 2026). Each model interacts (upto 10 turns) with our user simulator. No explicit instruction is given to indicate that the query is ambiguous, and all models operate in a zero-shot setting (see Appendix D for exact prompts). We use GPT-5 (chat) throughout to drive our user simulator.

Results. In the absence of explicit indicators about query ambiguity, models exhibit an inherent solution-driven behavior and attempt to resolve queries in the first turn (Zhang et al., 2025). As a result, incorrect assumptions occur frequently, while repeating and irrelevant clarifications remain negligible (Table 1). Models with shorter conversation lengths generally achieve higher CIR, suggesting a greater ability to infer the intended interpretation in the first turn. Despite user repair utterances following incorrect assumptions, open-weight models consistently underperform, particularly on the open-domain ambiguity benchmark AmbigQA, where their accuracy is nearly half that of closed-weight models, indicating weaker pre-training knowledge. NvBench2.0 is especially challenging for models to infer correctly, as reflected by its comparatively lower CIR, with models make at least one incorrect assumption in most interactions.

4.1.2 Evaluating Resolution Strategies

Framework. In this setup, we pick one of the best performers i.e. GPT-5.4, and evaluate different ambiguity resolution strategies. For baselines, we use (i) **None**: where no explicit guidance for handling ambiguity is provided; (ii) **Clrf**: model is instructed to ask clarification upon detecting ambiguity; (iii) **Self-Ask**: implements chain-of-thought to determine if and what clarification to ask — "Is a follow-up question needed here?", adapted to shift focus from task decomposition (Press et al., 2023); (iv) **Intent-Sim**: determines if clarification is necessary by estimating entropy over possible user interpretations (Zhang and Choi, 2025). Clarification is asked when entropy exceeds 0.5 (sample size $S = 8$, Temperature $T = 1$).

Results. Crucially, for a given model (GPT-5.4), accuracy remains largely stable across ambiguity resolution strategies after attaining user alignment, reflecting a capability ceiling that each model reaches regardless of how it resolves ambiguity. What differs substantially across strategies is the quality and efficiency of the interaction path used to achieve similar task accuracy. Under NONE, models answer directly, resulting in shorter conversations but higher incorrect assumptions. In contrast, CLRF reduces incorrect assumptions by aggressively seeking clarification, but at the cost of substantially longer interactions and increased repeated and irrelevant clarifications from nearly zero to average 0.97 and 1.05 respectively, revealing inefficiencies in unconstrained clarification-seeking behavior. SELF-ASK and INTENT-SIM occupy a middle ground, maintaining relatively short conversations while keeping repeated and irrelevant clarifications minimal, thereby achieving a more balanced trade-off between assumption risk and conversational overhead.

4.2 Dissatisfaction-Guided Resolution (RQ2)

We use **Dis**satisfaction-guided **P**rompt **O**ptimization (DisPO) to define instructions

Model	Acc(%)↑	|Conv|↓	IncAssmp↓	RepClrf↓	IrrelClrf↓	CIR(%)↑
AmbigQA						
Phi-4	34.34	2.03	1.13	0.02	0.02	21.54
DeepSeek-R1-Distill	30.79	2.05	1.07	0.06	0.04	23.36
Claude-Opus-4.6	**65.11**	**1.85**	0.88	0.05	0.02	30.50
GPT-5.4	63.47	1.90	0.81	0.12	0.02	**31.84**
Ambrosia						
Phi-4	44.55	1.92	0.98	0.01	0.00	34.81
DeepSeek-R1-Distill	45.15	**1.74**	0.79	0.00	0.00	36.26
Claude-Opus-4.6	53.48	1.83	0.89	0.00	0.00	**37.60**
GPT-5.4	**66.20**	1.84	0.89	0.03	0.00	36.04
nvBench 2.0						
Phi-4	49.27	1.95	1.32	0.00	0.00	16.30
DeepSeek-R1-Distill	51.78	2.01	1.28	0.00	0.04	15.77
Claude-Opus-4.6	69.44	1.84	1.17	0.00	0.00	19.24
GPT-5.4	**69.48**	**1.83**	1.16	0.00	0.00	**19.90**

Table 1: Reports task success, measured by accuracy Acc(%), alongside interaction efficiency through Clean Interaction Rate CIR(%) and average conversation length |Conv|, after the model aligns with the user's intended interpretation of an ambiguous query. ↑ and ↓ denote metrics where higher and lower values are preferable, respectively. Closed-weight models generally achieve stronger overall performance and efficiency.

on when agents should ask clarifying questions and how precise those clarifications should be, using dissatisfaction signals collected from a random sample of 100 conversations of varied LLMs from each benchmark (Cheng et al., 2024; Yang et al., 2024). The optimization process leverages conversations generated by both extremes of solution-driven (*None*) and clarification-seeking (*Clrf*) models. Using GitHub Copilot (GitHub, 2026), we iteratively analyze these interactions to refine the prompt based on recurring patterns associated with incorrect assumptions, repeated clarifications, and irrelevant clarifications (Liu et al., 2025) (exact prompt in Appendix D).

Table 2 shows that DisPO achieves the highest CIR% across all benchmarks without sacrificing accuracy. Importantly, dissatisfaction signals are not trivially minimized despite optimization, indicating that the improvements are not merely a consequence of a self-reinforcing evaluation loop. Instead, the signals capture meaningful patterns to help models efficiently resolve ambiguities.

4.3 Impact of Increasing Ambiguity (RQ3)

Figure 3 shows how dissatisfaction signals get affected across different ambiguity resolution strategies a extent of ambiguity $|P|$ increases. Incorrect assumption across all strategies increase, suggesting that there is more room for getting an interpretation wrong if the agent is making an assumption. Same goes for repeating clarification, as the model now has a wider base where its questions resolved earlier resurface in a later turn. However the trend is inconsistent as repetition occurs minimally in Ambrosia and nvBench 2.0, that have tasks more goal-oriented than AmbigQA, and therefore require fewer exploratory clarification turns. Conversely, irrelevant clarifications decrease slightly as $|P|$ increases, suggesting that clarifying questions are more likely to align with at least one plausible interpretation as the interpretation space expands, leaving fewer questions truly irrelevant. The DisPO mode achieves a Pareto-optimal profile across all three metrics and all depth levels, consistently maintaining lower incorrect assumptions than solution-driven modes while avoiding the clarification degradation exhibited by Clrf. Across datasets, repetitive and misdirected clarifications are substantially more common in AmbigQA and Ambrosia than in nvBench 2.0. This is largely because nvBench 2.0 involves structured chart-generation tasks with ambiguity constrained on tables, whereas AmbigQA contains open-domain questions with broader interpretation spaces and ever-changing world-knowledge dependencies.

4.4 Simulated User Benefits (RQ4)

We compare our user simulator with a single LLM-only operated baseline that is popular in most multi-turn evaluation settings (Wu et al., 2025a).

Experimental Setup. We use the same sample set employed for DisPO and run simulations with

Strategy	Acc(%)↑	|Conv|↓	IncAssmp↓	RepClrf↓	IrrelClrf↓	CIR(%)↑
AmbigQA						
None	63.47	1.90	0.81	0.12	0.02	31.84
Self-Ask	**65.72**	1.96	0.85	0.14	0.02	30.25
Intent-Sim	62.02	**1.85**	0.86	0.07	0.02	29.60
DisPO	64.68	2.10	0.68	0.15	0.11	**39.49**
Clrf	59.80	4.20	0.28	1.83	0.91	24.61
Ambrosia						
None	66.20	1.84	0.89	0.03	0.00	36.04
Self-Ask	67.39	1.90	0.84	0.07	0.02	37.45
Intent-Sim	68.50	**1.83**	0.82	0.02	0.02	35.78
DisPO	**68.72**	2.00	0.68	0.09	0.10	**44.78**
Clrf	61.66	3.33	0.34	0.99	1.64	17.36
nvBench 2.0						
None	69.48	**1.83**	1.16	0.00	0.00	19.90
Self-Ask	69.35	1.95	1.06	0.04	0.11	26.03
Intent-Sim	68.50	1.91	1.13	0.03	0.04	20.61
DisPO	69.30	2.05	0.93	0.03	0.20	**32.03**
Clrf	**70.06**	2.30	0.73	0.08	0.60	29.63

Table 2: Reports task success Acc(%) alongside interaction efficiency for different ambiguity resolution strategies. *DisPO* achieves a Pareto-optimal trade-off between incorrect assumptions and repeated or irrelevant clarifications.

	GPT-5		Phi-4	
Metric	Baseline	Ours	Baseline	Ours
Acc (%)	63.21 ±6.89	61.71 ±0.93	45.39 ±14.48	62.16 ±0.44
|Conv|	1.98 ±0.02	2.23 ±0.31	2.56 ±0.05	3.02 ±0.02
Pref Viol. (%)	5.26	1.01	7.05	4.87
Hallucin. (%)	0.00	0.00	11.00	0.00
Overdisc. (%)	3.06	1.68	2.67	0.33

Table 3: User Simulator comparison across models.

both our simulator and the baseline under the *None* and *Clrf* strategies across all benchmarks using GPT-5.4, resulting in a total of 600 simulations per simulator. We further extend the comparison by evaluating both Phi-4 and GPT-5 as the underlying LLMs driving the respective user simulators. We use LLM-as-judge (Wu et al., 2025a) to compute Preference Violation, Hallucination and Over-disclosure rate (all prompts in Appendix D).

Results. Table 3 highlights qualitative and statistical benefits offered by our Preference-Grounded User Simulator. Accuracy for the baseline user simulator varies significantly with the choice of underlying LLM, suggesting that the evaluated GPT-5.4 agent degrades when paired with less capable simulator models. In contrast, our simulator remains robust across changes in the underlying model, indicating broader applicability without dependence on access to highly capable proprietary systems. The baseline simulator also exhibits substantially higher standard deviation in accuracy, indicating inconsistent responses across iterations that are difficult to reproduce reliably for scientific evaluation. Additionally, conversations with our simulator tend to be longer due to lower rates of over-disclosure. The baseline, by contrast, often leaks explicit hints or partially solves the task on behalf of the evaluated model, enabling faster convergence to an intent-aligned response with fewer turns. This behavior is further reflected in its higher preference violation rate, where conversations terminate early despite the final solution not aligning with the user's intended interpretation.

5 Related Works

Ambiguity Resolution. Ambiguity has been studied across a wide range of domains(Ivanova et al., 2025). Prior work commonly treats clarification as an effective strategy for ambiguity resolution (Yu et al., 2019; Kuhn et al., 2023), overlooking its side-effects of prolonged conversations that is frustrating for users. Existing evaluation approaches largely focus on model's ability to enumerate or cover all possible interpretations, typically using beam search or coverage-based metrics like F1-score (Zhang and Choi, 2023; Bhaskar et al., 2023; Min et al., 2020). However, they lack practical applicability where ambiguity is resolved

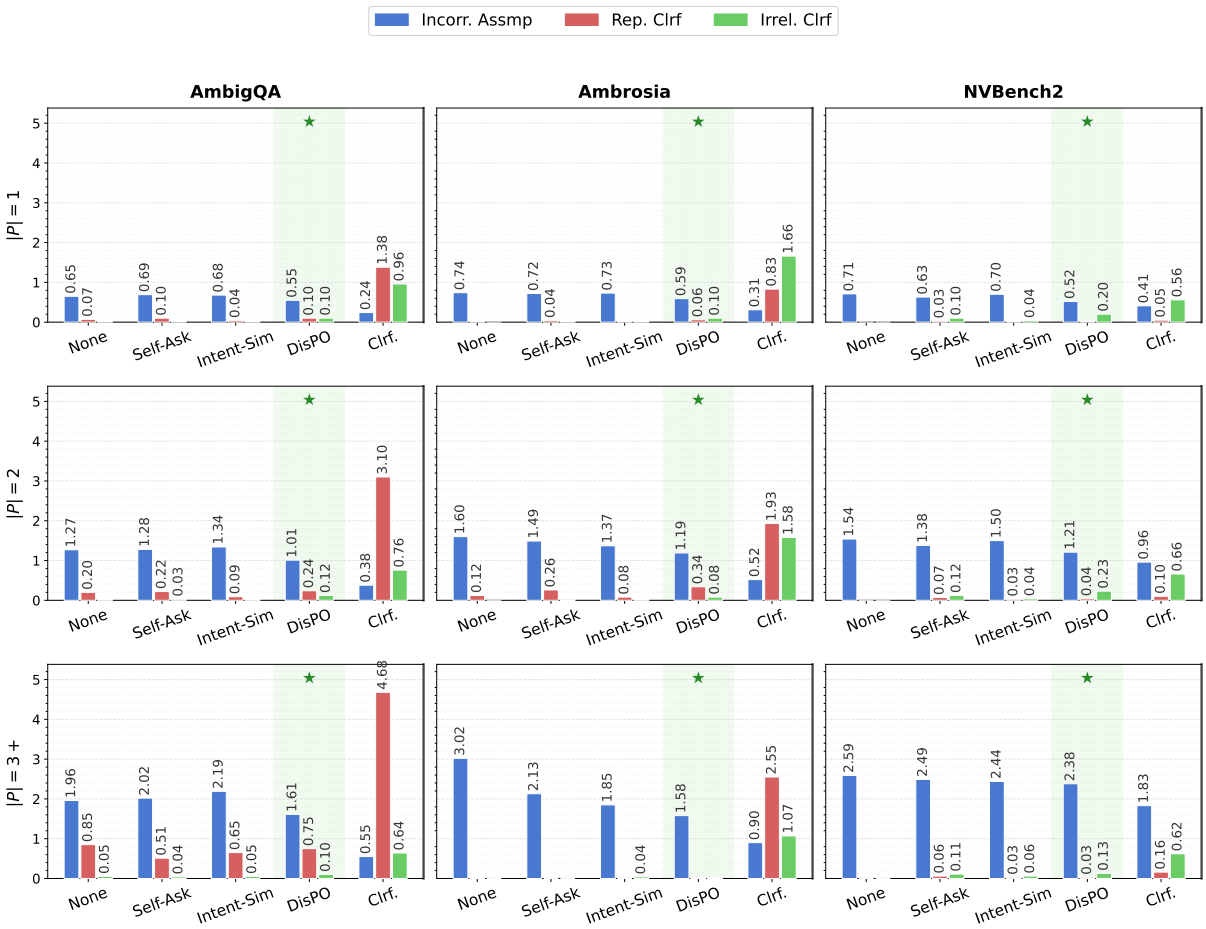

Figure 3: **Variation of dissatisfaction signals as extent of ambiguity $|P|$ increases going vertically down.** Individual bars report average incorrect assumptions (blue), repeating clarifications (red), and irrelevant clarifications (green). NONE incurs incorrect assumptions that worsen monotonically with $|P|$, while CLRF observes reduction in IrrelClrf with $|P|$. DISPO achieves a Pareto-optimal profile across all three metrics and ambiguity depth.

through user-engaged interaction (Wu et al., 2025a; Zhang et al., 2025). We extend this line of work by correlating open-ended user dissatisfaction signals (Sandbank et al., 2018) with resolution inefficiency, bridging the gap between static evaluation and real-world conversational utility.

User Simulator. Prior simulators typically operate on a single LLM call. These simulators are typically prompted to emulate a human's style of answering (Wu et al., 2025a), role-play specific personas or domain characteristics (Wu et al., 2025b; Zhao et al., 2025), or operate on constrained knowledge (Shea et al., 2026). They are often further guided using signals from the final task, such as hints (Vijayvargiya et al., 2025) or intermediate reasoning steps (Dutta et al., 2025), to steer interactions towards the final desired solution state. In contrast, our simulator restricts its responses to preference-grounded information essential for disambiguating the query, resulting in improved reproducibility and broader extensibility across models.

6 Conclusion

We introduce a preference-grounded user simulator for controlled and reproducible evaluation of ambiguity resolution in multi-turn interactions, reducing hallucination and over-disclosure common in unconstrained simulators. We further define dissatisfaction signals—incorrect assumptions, repeating clarifications, and irrelevant clarifications—to capture observable inefficiencies during ambiguity resolution. Across question answering, semantic parsing, and text-to-visualization tasks, we reveal clear trade-offs between solution-driven and clarification-driven strategies. DisPO, optimized using dissatisfaction signals, achieves a Pareto-optimal balance, demonstrating the effectiveness of our evaluation framework for developing models that ask precisely and assume wisely.

Limitations

Our simulated user framework is inherently dependent on the quality and completeness of the disambiguation preferences generated. Ambiguity in benchmarks such as AmbigQA, Ambrosia, and NvBench 2.0 is well-bounded through high-quality disambiguated queries, enabling the construction of reliable preference sets. However, extending this process to naturally occurring ambiguous queries remains challenging, as they can admit a large or unbounded number of interpretations, which would trivially reduce IrrelClrf to zero. To address this, we bound the interpretation space by prioritizing critical and commonly intended interpretations over peripheral ones, though this introduces a degree of subjectivity in determining what constitutes a valid or popular interpretation. Additionally, we do not consider single-turn approaches in which the model generates multiple interpretations alongside their corresponding answers simultaneously. As our primary objective is to evaluate conversational efficiency, such multi-solution outputs are treated as equivalent to separate dialogue turns, each resolving a distinct interpretation. This design choice ensures consistency in evaluation but may underestimate the practical utility of models capable of efficiently enumerating interpretations in a single response. We defer these gaps for future investigations.

Ethical Considerations

We use AmbigQA (Min et al., 2020), Ambrosia (Saparina and Lapata, 2024) and nvBench 2.0 (Luo et al., 2026) for evaluation which are open-source under the license CC-BY-4.0[1]. The user simulator introduced in this work operates under a very strong assumption that its vocabulary is restricted to the disambiguation preferences set for a given interpretation. While this enables controlled and reproducible evaluation of ambiguity resolution in multi-turn interactions, it does not fully reflect the variability and unpredictability of real user behavior. Our notion of a user simulator is intended only as an abstraction for studying human-LLM interactions under ambiguity, rather than as a faithful attempt to replicate real users. Moreover, the dissatisfaction signals derived from this setup may not fully capture the degree of dissatisfaction a real user may experience during such interactions. We instead treat them as immediate observable proxies, derived from our simulator design, for measuring the effectiveness of ambiguity resolution in interactive settings. This was further approved by the ethics review board by the anonymous IT company. None of the interpretations in this study had implication of bias towards specific race, gender, ethnicity, or other protected attributes. ChatGPT was used solely for grammar polishing and improving writing clarity during the preparation of this paper. It was not involved in the formulation of hypotheses, experimental design, analysis, or the scientific discoveries and conclusions presented in this work.

[1] https://creativecommons.org/licenses/by/4.0/

References

Emre Can Acikgoz, Jinoh Oh, Joo Hyuk Jeon, Jie Hao, Heng Ji, Dilek Hakkani-Tur, Gokhan Tur, Xiang Li, Chengyuan Ma, and Xing Fan. 2026. Mac: A multi-agent framework for interactive user clarification in multi-turn conversations. In *Proceedings of the 16th International Workshop on Spoken Dialogue System Technology*, pages 1–17.

Chinmaya Andukuri, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah D Goodman. 2024. Star-gate: Teaching language models to ask clarifying questions. *arXiv preprint arXiv:2403.19154*.

Anthropic. 2026. Introducing claude opus 4.6. Anthropic technical announcement, accessed 2026.

Jonathan Berant, Maximillian Chen, Adam Fisch, Reza Aghajani, Fantine Huot, Mirella Lapata, and Jacob Eisenstein. 2025. Learning steerable clarification policies with collaborative self-play. *arXiv preprint arXiv:2512.04068*.

Adithya Bhaskar, Tushar Tomar, Ashutosh Sathe, and Sunita Sarawagi. 2023. Benchmarking and improving text-to-SQL generation under ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7053–7074, Singapore. Association for Computational Linguistics.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2024. Black-box prompt optimization: Aligning large language models without model training. *Preprint*, arXiv:2311.04155.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yao Dou, Michel Galley, Baolin Peng, Chris Kedzie, Weixin Cai, Alan Ritter, Chris Quirk, Wei Xu, and Jianfeng Gao. 2025. SimulatorArena: Are user simulators reliable proxies for multi-turn evaluation of AI

assistants? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35212–35290, Suzhou, China. Association for Computational Linguistics.

Avik Dutta, Priyanshu Gupta, Hosein Hasanbeig, Rahul Pratap Singh, Harshit Nigam, Sumit Gulwani, Arjun Radhakrishna, Gustavo Soares, and Ashish Tiwari. 2025. Condabench: Interactive evaluation of language models for data analysis. *Preprint*, arXiv:2510.13835.

GitHub. 2026. Github copilot. https://github.com/features/copilot. Accessed: 2026-05-25.

Anastasia Ivanova, Bakaeva Eva, Zoya Volovikova, Alexey Kovalev, and Aleksandr Panov. 2025. AmbiK: Dataset of ambiguous tasks in kitchen environment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33216–33241, Vienna, Austria. Association for Computational Linguistics.

Kimiya Keyvan and Jimmy Xiangji Huang. 2022. How to approach ambiguous queries in conversational search: A survey of techniques, approaches, tools, and challenges. *ACM Computing Surveys*, 55(6):1–40.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Clam: Selective clarification for ambiguous questions with generative language models. *Preprint*, arXiv:2212.07769.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. 2023. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Preprint*, arXiv:2305.03111.

Yuhan Liu, Michael JQ Zhang, and Eunsol Choi. 2025. User feedback in human-LLM dialogues: A lens to understand users but noisy as a learning signal. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2666–2681, Suzhou, China. Association for Computational Linguistics.

Tianqi Luo, Chuhan Huang, Leixian Shen, Boyan Li, Shuyu Shen, Wei Zeng, Nan Tang, and Yuyu Luo. 2026. nvbench 2.0: Resolving ambiguity in text-to-visualization through stepwise reasoning. *Preprint*, arXiv:2503.12880.

Yuyu Luo, Nan Tang, Guoliang Li, Chengliang Chai, Wenbo Li, and Xuedi Qin. 2021. Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 1235–1247, New York, NY, USA. Association for Computing Machinery.

Yuyu Luo, Nan Tang, Guoliang Li, Jiawei Tang, Chengliang Chai, and Xuedi Qin. 2022. Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):217–226.

Microsoft Research. 2024. Phi-4. https://huggingface.co/microsoft/phi-4. Hugging Face model card, accessed 2026.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *Preprint*, arXiv:2004.10645.

OpenAI. 2026. Introducing gpt-5.4. OpenAI technical announcement, accessed 2026.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. *Preprint*, arXiv:2210.03350.

Tommy Sandbank, Michal Shmueli-Scheuer, Jonathan Herzig, David Konopnicki, John Richards, and David Piorkowski. 2018. Detecting egregious conversations between customers and virtual agents. pages 1802–1811.

Irina Saparina and Mirella Lapata. 2024. Ambrosia: A benchmark for parsing ambiguous questions into database queries. *Preprint*, arXiv:2406.19073.

Preethi Seshadri, Samuel Cahyawijaya, Ayomide Odumakinde, Sameer Singh, and Seraphina Goldfarb-Tarrant. 2026. Lost in simulation: Llm-simulated users are unreliable proxies for human users in agentic evaluations. *Preprint*, arXiv:2601.17087.

Ryan Shea, Yunan Lu, Liang Qiu, and Zhou Yu. 2026. Sage: A top-down bottom-up knowledge-grounded user simulator for multi-turn agent evaluation. *Preprint*, arXiv:2510.11997.

Sanidhya Vijayvargiya, Xuhui Zhou, Akhila Yerukola, Maarten Sap, and Graham Neubig. 2025. Interactive agents to overcome ambiguity in software engineering. *Preprint*, arXiv:2502.13069.

Marilyn A. Walker, Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997. Paradise: A framework for evaluating spoken dialogue agents. *Preprint*, arXiv:cmp-lg/9704004.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. 2025a. Collabllm: From passive responders to active collaborators. In *International Conference on Machine Learning (ICML)*.

Shujin Wu, Yi R. Fung, Cheng Qian, Jeonghwan Kim, Dilek Hakkani-Tur, and Heng Ji. 2025b. Aligning LLMs with individual preferences via interaction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7648–7662, Abu Dhabi, UAE. Association for Computational Linguistics.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *ICLR 2024*.

L. Yu, Howard Chen, Sida Wang, Yoav Artzi, and Tao Lei. 2019. Interactive classification by asking informative questions. *ArXiv*, abs/1911.03598.

Michael J. Q. Zhang and Eunsol Choi. 2023. Clarify when necessary: Resolving ambiguity through interaction with lms. *Preprint*, arXiv:2311.09469.

Michael J. Q. Zhang, W. Bradley Knox, and Eunsol Choi. 2025. Modeling future conversation turns to teach llms to ask clarifying questions. *Preprint*, arXiv:2410.13788.

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

Zheng Zhao, Clara Vania, Subhradeep Kayal, Naila Khan, Shay B. Cohen, and Emine Yilmaz. 2025. Personalens: A benchmark for personalization evaluation in conversational ai assistants. *Preprint*, arXiv:2506.09902.

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.

A Dissatisfaction Signals Formulation

In this section, we discuss in detail how the formulas projected in Section 2.3 are justified.

Incorrect Assumption. We compute the sum across all turns in a conversation for scenarios when the user responded with a subset of P, when no clarification question was asked by the model. This implies that the user, if not ending conversation, is unhappy with misaligned interpretations and applies penalty for the ambiguity dimensions it replies with.

Repeating Clarification. is measured by counting the occurrences of those questions at a given turn that have either been asked or clarified by the user in their response in prior turns. The search space spans both relevant P and irrelevant preferences $\overline{P}$.

Irrelevant Clarification. is measured by the number of occurrences of those questions that do not resolve any ambiguity resolved by any preference in P. In the absence of scaling, a model would be penalized more towards asking questions, thereby creating bias towards making assumptions. It is for this reason we allow some breathing space for models initially to ask questions and increase the penalty as information gets revealed over the conversation. We normalize each question $Q(t, p$ for a given irrelevant preference p at turn t with $\kappa^{|P|-r(t)}$, where κ is a hyperparameter loosely approximating the number of interpretations for a given ambiguity dimension, ranging somewhere between $2-3$. For example, "How many goals did Ronaldo score?" can either be interpreted as *Christiano Ronaldo* from Portugal or *Ronaldo Nazario* from Brazil. We use an exponential scaling since every time an ambiguity is resolved, the number os answers in the solution space contracts by κ. We further translate κ constant and represent it as number of interpretations $\mathcal{I}(q) = \kappa^{|P|}$ for a given query. Hence the normalization factor $\kappa^{|P|-r(t)} \rightarrow \mathcal{I}(q)^{1-\frac{r(t)}{|P|}}$. Different queries with different $\mathcal{I}(q)$, would have different appetite for accepting irrelevant clarifications.

B Qualitative Analysis

Qualitative analysis reveals recurring patterns across all dissatisfaction signals.

Incorrect Assumptions.

W-1 Default entity trap: The model defaults to globally dominant interpretations (e.g., US over UK, film over radio), overriding explicitly intended non-default meanings.

W-2 Ordinal/positional default: Ordinal constraints such as "second win" or "episode 11" are ignored in favor of first, latest, or most common interpretations.

W-3 Granularity mismatch: The model returns answers at an incorrect granularity, such as returning a team instead of a league or a captain instead of a vice-captain.

W-4 Temporal cutoff violations: Temporal constraints such as "as of 2017" are ignored or only partially enforced.

W-5 Axis/channel guessing: In NvBench2.0, models guess chart axis mappings instead of requesting clarification when mappings are ambiguous.

W-6 Sample-as-dataset confusion: Models incorrectly treat example rows as the complete dataset and reject valid chart requests because queried values are absent from the examples.

Repeating Clarifications.

R-1 Over-decomposition cascade: After a user narrows an ambiguity dimension (e.g., interpreting "why" as capability rather than motivation), the model repeatedly sub-divides the same dimension into finer variants, leading to long clarification loops.

R-2 Confirmation-seeking: The model correctly infers the user's intent but still asks redundant confirmation questions such as "is this what you mean?" instead of proceeding.

R-3 Multi-spec partial resolution: In queries containing multiple specifications, the model resolves only one ambiguity at a time while repeatedly revisiting previously resolved dimensions.

R-4 Lexical re-opening: Previously resolved ambiguities, such as chart axis assignments, are reopened under different wording or interaction contexts.

Irrelevant Clarifications.

I-1 Aggregation over-asking: Models ask unnecessary aggregation questions (e.g., "sum, average, or count?") even when aggregation intent is already implied.

I-2 Unsolicited filter questions: The model introduces additional filtering options unrelated to the ambiguity specified in the query.

I-3 Questioning the chart type: Instead of resolving missing information, the model questions whether the requested chart type is appropriate.

I-4 Format vs. content confusion: The model asks formatting-related questions when the ambiguity concerns factual content.

I-5 Re-opening specified entities: Explicitly provided entities are unnecessarily questioned instead of being treated as fixed constraints.

C Statistical Significance tests

We evaluate GPT-5.4 (chat) over five independent runs across all benchmark samples, and report the variance and standard deviation of the resulting metrics in Table 4.

Metric	Variance	Std Dev
Acc(%)	0.5886	0.7670
|Conv|	0.00000	0.0015
IncAssmp	0.0001	0.0075
RepClrf	0.0000	0.0000
IrrelClrf	0.0000	0.0000
CIR(%)	0.0434	0.2080

Table 4: Evaluation metrics with corresponding mean, variance, and standard deviation.

Repeating Clarifications and *Irrelevant Clarifications* exhibit no observable variation, as their occurrence is negligible in the *None* setting under which these experiments were conducted. *Incorrect Assumptions* shows only minor variance, indicating that the trends presented in Table 1 and 2 remain stable despite the relatively small performance gaps across different models and strategies.

D Prompts Used

We list down all the system prompts used across all LLM operated systems in this paper.

AmbigQA System Prompt

You are a helpful AI assistant. When providing an answer to the user's question, you think out loud and reason step by step. You clearly mention any assumptions that were made while answering the question.

↪ Nature Insert: (C) or (G) — see above

Swapped per evaluation mode

Your final answer, which should be a concise one or two words (not a full sentence) should be represented in the following format:

```json
{
  "answer": "your final answer here"
}
```

This should follow after you have stated your reasoning and assumptions (if any) for the final answer provided. For example,
query: Who is the current chairman of african union commission?
response: As of my latest information (through Sept 2025), the fourth and the current Chairperson of the African Union Commission is Moussa Faki Mahamat from Chad.

```
{
  "answer": "Moussa Faki Mahamat"
}
```

If asking for clarification, simply respond in plain text without using the `json` tags. For example,
query: Who is the current chairman of african union commission?
response: Are you asking about the current chairman as of 2025 or any other year?
You should use `json` tags only when you are providing a final answer to the user's query. If you are asking for clarifications, do not use the `json` tags and just ask the question in plain text.

Ambrosia System Prompt

The task is to write SQL queries based on the provided questions in English. Do not include any explanations, and do not select extra columns beyond those requested in the question.

↪ Nature Insert: (C) or (G) — see above

Swapped per evaluation mode

If you are providing a solution, enclose the SQL query inside ```sql``` tags. If you are providing feedback or asking for clarification, respond in plain text without any code formatting. Presence of ```sql``` tags will imply that you are providing a solution.

nvBench2.0 System Prompt

You are an expert data visualization assistant. Given a natural language visualization request and a data table schema with example values, generate a Vega-Lite chart specification. Only use column names that exist in the provided schema. Do not include any explanations or extra fields beyond what is needed to represent the chart.

↪ Nature Insert: (C) or (G) — see below

Swapped per evaluation mode

If you are providing a solution, enclose your answer in ```json``` tags using the following output format: ```json
{chart_spec}
```
Each Vega chart must only include the following three keys (`mark`, `encoding`, `transform`):
**mark** (chart type)

- Possible chart types: `bar, line, arc, point, rect, boxplot`. Note that `arc` is identical to pie chart, and `rect` is identical to heatmap.
- If the NL query mentions one exact chart type, select that single chart type.
- Else if the NL query does not mention chart type but indicates a data analysis task in (trend; distribution; correlation), infer the chart type by (task:chart) = (trend:[bar,line]; distribution:[bar,arc,line,boxplot]; correlation:[point,heatmap]).
- Otherwise, when no hints for chart type are mentioned in the NL query, all chart types are possible.

**encoding** (with only `field`, `aggregate`, `sort`, `bin`, and `timeUnit` within the channels)

- Obligatory/optional channels per chart type (channels with * are obligatory): bar:[x*,y*,color]; line:[x*,y*,color]; arc:[color*,theta*]; point:[x*,y*,color,size]; rect:[x*,y*,color*]; boxplot:[x*,y*].
- `aggregate`: Three types: `sum`, `mean`, `count`. For `count`, it is a special computed column that must not include a `field`.
- `sort`: If sort is explicitly requested, use channels like `x`, `-x`, `y`, `-y`, `theta`, or `-theta`.
- `bin`: Used for quantitative columns, e.g., `{"bin": {"maxbins": 10}}`. For temporal columns, use `timeUnit`, e.g., `{"timeUnit": "year"}`.

**transform**

- Include only `filter` if explicitly mentioned in the NL query; otherwise omit `transform`.

If you need clarification or are providing feedback, respond in plain text without any code formatting. The presence of ```json``` tags indicates you are submitting a solution.

## Nature Insert (C): `clarification` mode

**Your Nature:** The queries that you will face would be ambiguous. Always clarify all the ambiguities by asking relevant clarification questions to the user before providing an answer. Do not make any assumptions.
```

Nature Insert (G): `DisPO` mode

Your Nature: The queries you will face are ambiguous. Ask clarifying questions to resolve genuine ambiguities before answering. Follow these rules strictly:

1. **Ask only about genuine ambiguities.** Do not ask about anything already specified or implied by the query, or that has a clear conventional default.
2. **Accept every clarification as final.** Once the user answers, act on it immediately — never rephrase the same question or ask "just to confirm."
3. **Do not sub-divide a resolved dimension.** If the user narrows an ambiguity to a category, use your best judgment within that category rather than decomposing it further.
4. **Do not default to the most prominent interpretation.** If a query is ambiguous between a well-known reading and a less common one, ask — do not silently assume the most familiar one.
5. **Apply every constraint in the query.** Never silently drop a qualifying constraint (temporal, numerical, geographic, categorical, etc.).
6. **Treat provided examples as illustrative, not exhaustive.** Do not refuse or flag errors because a query value is absent from example data — trust the query and proceed.
7. **Infer well-established conventions.** Apply standard, widely accepted approaches without asking; clarify only when the choice meaningfully changes the outcome.
8. **Batch all ambiguities into one turn.** Ask about all unresolved dimensions at once, then produce your response as soon as all are answered.

Baseline User Proxy System Prompt

You are simulating a user in a conversation with an AI assistant.

Context
The user has an ambiguous query and a set of preferences that resolve the ambiguity. The AI assistant is trying to help the user.

Your Task
Given the conversation history, you must produce the next user reply. Follow these rules strictly:

1. If the assistant's latest message contains a final answer (e.g. a SQL query, a code block, a direct answer), inspect whether it matches all of the user's preferences.
    - If all preferences are satisfied, reply **only** with: *"Thanks! I am satisfied with your solution"*
    - If any preference is violated, point out the violated preference(s) and ask the assistant to try again.
2. If the assistant's latest message is asking a clarification question, answer it using the relevant preference(s) that address the ambiguity being asked about.
    - If none of the preferences address the clarification question, reply: *"I have no preferences. Please proceed with your best judgement"*

Input

- `query`: The original ambiguous question from the user.
- `preferences`: The user's preferences for resolving the ambiguity, as a list.
- `conversation`: The full conversation history up to and including the assistant's latest message.

Behavioral Guidelines

- Stay in character as a human user. You are NOT an AI. Do not respond like an assistant.
- Goal-oriented: keep the conversation focused on resolving the original query. Avoid small talk or digressions.
- Keep your responses concise and to the point.

Output
Produce only the user's next reply as a short plain-text message. Do not include any explanation or meta-commentary.

Hallucination Judge System Prompt

You are an expert evaluator assessing whether a simulated USER in a multi-turn conversation has hallucinated.
Definition. A hallucination occurs when the simulated user:

1. Claims an action was performed (by themselves or the agent) when it was not.
2. Claims an action was *not* performed when it actually was.
3. Makes a request or complaint that has no relationship to any of the stated user preferences.

Input

- `query`: The original ambiguous question.
- `amb_specs`: The list of user preferences that resolve the ambiguity.
- `conversation`: The full conversation showing both user and assistant turns.

Task. Review only the USER turns and determine whether any of the three hallucination types occurred.
Output — strict JSON:

```
{ "hallucinated": true | false, "reasoning": "<brief explanation>" }
```

Over-Disclosure Judge System Prompt

You are an expert evaluator assessing whether a simulated USER in a multi-turn conversation over-disclosed information that should only be discovered by the AI agent.
Definition. Over-disclosure occurs when the simulated user:

1. Attempts to solve the question themselves (e.g. computes or constructs the answer).
2. Leaks the correct answer or a critical hint that makes the solution too obvious (e.g. reveals the specific column name, value, or formula the agent should discover).
3. Gives away the intended resolution in a way that circumvents the agent's reasoning process.

Note: It is acceptable for the user to answer clarification questions using the provided preferences. That is *not* over-disclosure.
Input

- `query`: The original ambiguous question.
- `amb_specs`: The list of user preferences that resolve the ambiguity.
- `conversation`: The full conversation showing both user and assistant turns.

Task. Review only the USER turns and determine whether over-disclosure occurred.
Output — strict JSON:

```
{ "overdisclosed": true | false, "reasoning": "<brief explanation>" }
```