*Pref*Ix: Understand and Adapt to User Preference in Human-Agent Interaction

Jialin Li[1] **Zhenhao Chen**[2] * **Hanjun Luo**[1] **Hanan Salam**[1]

[1] New York University Abu Dhabi, Abu Dhabi, UAE
[2] Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE

Abstract

LLM-based agents can complete tasks correctly yet still frustrate users through poor interaction patterns, such as excessive confirmations, opaque reasoning, or misaligned pacing. Current benchmarks evaluate task accuracy but overlook how agents interact: whether they infer preferences from implicit cues, adapt dynamically, or maintain fine-grained interaction quality. We introduce *Pref*Ix, a configurable environment that evaluates both what agents accomplish and how they interact. Central to *Pref*Ix is the Interaction-as-a-Tool (IaaT) paradigm, which treats interaction behaviors as structured tool calls, unifying them with existing evaluation frameworks. We define 31 preference settings across 14 attributes and formalize user experience (UX) as a core metric alongside task accuracy. A composite LLM-as-a-Judge mechanism across seven UX dimensions achieves strong aggregate reliability (ICC > 0.79), high internal consistency ($\alpha = 0.943$), and human correlation ($\rho = 0.52$–0.78). Preference-aware agents show 7.6% average UX improvement and 18.5% gain in preference alignment. Our work is openly accessible *here*.

1 Introduction

Adapting to human preferences has been a fundamental pursuit in modern AI systems (Cui et al., 2024). By aligning LLMs with the distinct needs and experiences of individuals or groups, the systems facilitate frictionless and engaging interactions, fostering improved user experiences.

Early work on personalization distinguishes *content* (what facts or recommendations the model produces) from *presentation* (how information is expressed through style or tone) (Ryan et al., 2025a). However, this framing is constrained by the limited interaction space of language models. With the emergence of LLM-based agents, personalization can extend beyond generation to encompass richer interaction behaviors.

LLM-based agents build upon LLMs' language capabilities by adding planning, tool orchestration, and environment grounding, facilitating a higher degree of task automation (Li et al., 2025). These capabilities broaden the scope of personalization across the entire pipeline, from understanding user preferences, through planning and execution, to generation, introducing opportunities for behavioral adaptation that shape user experience even when task goals remain constant.

To illustrate, Figure 1 shows that the distinction lies not merely in task completion but in behavioral patterns during execution: while a rigid agent induces frustration through excessive confirmations, a preference-aware agent adapts its confirmation frequency and execution pacing to lower user load.

Current agent benchmarks focus on task success (Huang et al., 2024), reasoning and planning (Gioacchini et al., 2024), memory (Maharana et al., 2024), tool use (Huang et al., 2024), robustness (Debenedetti et al., 2024; Yao et al., 2024), and safety (Qiu et al., 2024; Chen et al., 2024b; Yuan et al., 2024; Zhang et al., 2024), largely measured via output correctness in controlled settings (Mohammadi et al., 2025). They rarely model or evaluate how agents adapt their interaction behavior to user preferences.

As the field moves toward human-centered agents, recent work has begun exploring adaptation to individual goals. For example, Qian et al. (2025a) focused on agents' ability to resolve underspecified preferences in multi-turn interactions. However, these works adapt content to underspecified user goals but treat the interaction protocol (confirmation strategy, pacing, exposure of intermediate steps) as fixed.

On the other hand, UX highly relies on users' interaction preferences (Augstein et al., 2023). Yet UX metrics have received limited attention in Nat-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 30110–30149

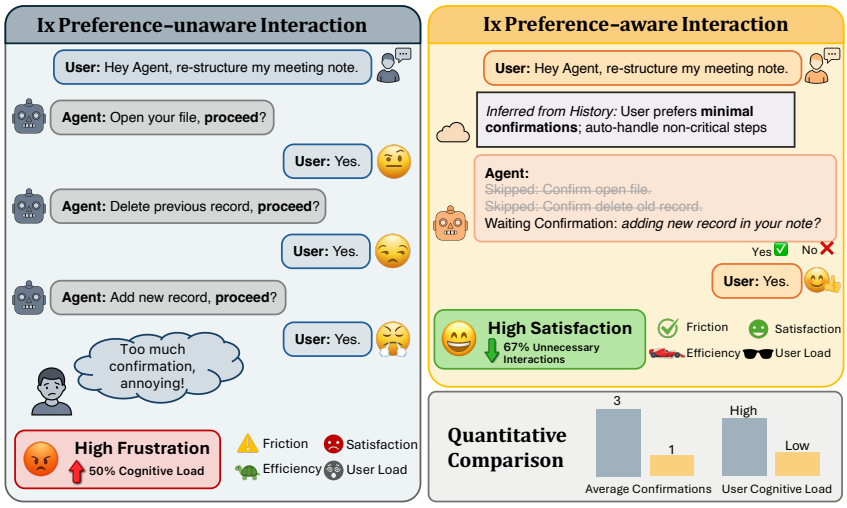

Figure 1: **Impact of Preference Awareness in Interaction.** By inferring user preferences from history, the agent reduces unnecessary turns (right), whereas a rigid agent (left) leads to user frustration despite correct task execution.

ural Language Processing (NLP) evaluation, especially in human-agent interaction. UX metrics like satisfaction and efficiency are only occasionally introduced when they are related to token cost or response latency (Kang et al., 2025). Even if there is experience related evaluation, they are highly inconsistent and are often treated as supplementary analyses (Perrig et al., 2024) rather than core metrics. Therefore, despite widespread discussion, UX evaluation remains loose and heterogeneous, lacking unified, comparable, and reproducible formal representation.

This points to a gap in current agent benchmarks. The real question is how to turn interaction-level UX into a primary evaluation target, one that is reproducible and comparable across systems, instead of leaving it as a qualitative side note. The gap runs through three stages of the agent pipeline. At the *understanding* stage, existing benchmarks specify user preferences explicitly (Salemi et al., 2024; Hao et al., 2025), and do not test whether an agent can infer them from implicit behavioral cues (Li et al., 2025). At the *planning and execution* stage, trajectory-based metrics work well for static settings (Chen et al., 2024a; Ma et al., 2024), but evaluation under dynamic user simulation, particularly with long-term adaptation, is still rare (Mohammadi et al., 2025). At the *generation* stage, coarse-grained preference alignment is measured, while finer dimensions of interaction quality go largely unreported (Wang et al., 2024; Li et al., 2025). What is missing is a framework that makes interaction-level UX tractable to measure: observable in agent trajectories, quantifiable with reproducible metrics, and comparable across models and tasks.

To address these gaps, we present *Pref***Ix**, which integrates a simulation environment, an evaluation framework linking user preferences to agent behaviors, and validated UX metrics for automated assessment. We make the following contributions:

❶ **Unifying Interaction and Tool-use Framework:** With *Pref***Ix**, we introduce a configurable interactive environment for multi-turn tool-use scenarios. By proposing the Interaction-as-a-Tool (IaaT) paradigm, where four categories of interaction behaviors (e.g., confirmation requests) are modeled as callable tools alongside external APIs, we unify the evaluation of interaction processes within existing agent frameworks.

❷ **Structured Evaluation of Task Correctness and UX:** We define a taxonomy of 14 interaction preference attributes across four categories (e.g., confirmation frequency, error handling style), and map them to observable agent behaviors. This places UX on equal footing with task accuracy: agents are judged both on what they accomplish and on how they conduct the interaction.

❸ **Reliable Automated Assessment:** We validate a composite multiple LLM-as-a-Judge framework across seven UX dimensions centered on *interaction preference alignment*. Experiments demonstrate traceable, reproducible, reliable, and scalable assessment.

Table 1: Comparison of existing environments for human-agent interaction, comparing Multiturn support, Function Calling, User Simulation, Evaluation Method (Rule-based vs. Generative), Evaluation Target (Process vs. Interaction), and Interaction Preference handling.

Benchmark	Multiturn	Function Calling	User Simulation	Eval Method		Eval Target		Preference
			Dynamic	Rule-based	Generative	Process	Interaction	Interaction
τ Bench (2024)	✓	✓	✓	✓	✗	✗	✗	✗
IN3 (2024)	✓	✗	✓	✓	✗	✓	✗	✗
TravelPlanner (2024)	✗	✓	✗	✓	✗	✓	✗	✗
BFCL v3 (2025)	✓	✓	✗	✓	✗	✓	✗	✗
ETAPP (2025)	✗	✓	✗	✓	✓	✓	✓	✗
UserBench (2025a)	✓	✓	✓	✓	✗	✓	✓	✗
***Pref*Ix (ours)**	✓	✓	✓	✓	✓	✓	✓	✓

2 Related work

2.1 User-centered Evaluation Targets of Language Agents

User-centric evaluation in human-agent interaction emerged from the need to assess not only task correctness but also appropriateness of the interaction process: whether agents interact efficiently and align with diverse user expectations (Ryan et al., 2025b; Qian et al., 2025b). Recent work has advanced multi-turn evaluation (Chakraborty et al., 2025), dynamic user simulation (Yao et al., 2024), and user preference modeling (Qian et al., 2025a; Singh et al., 2024). These gaps manifest across the personalization pipeline. At the *understanding* layer, benchmarks such as LaMP (Salemi et al., 2024) evaluate explicit profile extraction, yet inference from implicit behavioral cues remains undertested (Li et al., 2025). At the *planning and execution* layer, trajectory-based metrics assess tool accuracy in static settings (Chen et al., 2024a; Ma et al., 2024), but lack evaluation in dynamic user simulation with long-term adaptive behavior (Mohammadi et al., 2025). At the *generation* layer, coarse-grained metrics like preference rate capture surface alignment (Wang et al., 2024), while fine-grained interaction quality dimensions remain under-specified, see Table 1. ***Pref*Ix** addresses these gaps by unifying evaluation across all three layers comprehensively.

2.2 User Experience Evaluation Approaches for Language Agents

In evaluating user experience, a fundamental methodological divide further complicates assessment (Zhao et al., 2025). On one hand, traditional HCI evaluation offers qualitative depth through Likert scales and interviews but lacks the reproducibility and scalability required for large-scale comparison (He et al., 2025; Wang et al., 2024). On the other hand, the NLP community achieves scaling at the cost of rigor; user-centric metrics like satisfaction remain fragmented (Diebel et al., 2025), single faceted (Wu et al., 2025), and are often treated as secondary analyses rather than formalized core objectives. Most importantly, interaction trajectories are usually treated as supplementary material, instead of being established as primary evaluation targets that can be systematically measured and aligned with specific user experience goals and interaction preferences (Liu et al., 2025; Wu et al., 2022).***Pref*Ix** bridges these gaps by formalizing multi-turn trajectories into unified, reproducible metrics, reconciling HCI-level depth with NLP-level scalability and reproducibility.

3 *Pref*Ix

To systematically investigate how agents interact with users exhibiting diverse interaction preferences, we introduce ***Pref*Ix**, an interactive evaluation environment that simulates users with varying preference profiles, assessing task accuracy and interaction quality. Figure 2 provides an overview of the construction and evaluation pipeline of ***Pref*Ix**, which has four components.

- ***Task Coarsening.*** Rewrite BFCL's (Berkeley Function Calling Leaderboard (Patil et al., 2025)) overly-specified prompts into coarser task instructions while preserving deterministic tool-use ground truth and enabling flexible multi-turn interactions that elicit and express user preferences.
- ***Preference-Aware User Simulation.*** Define user interaction preferences spanning agent preference adaptation action space (see Table 6 in Appendix A); LLM simulators implicitly

express assigned preferences through conversational behavior without explicit self-disclosure.

- ***Interaction-as-a-Tool (IaaT).*** Abstract user-agent interactions as structured tool calls, unifying interaction preferences and task execution within a single measurable framework for quantitative agent alignment benchmarking.

- ***User Experience (UX) Judge.*** Move beyond tool-use accuracy by defining a set of core user experience dimensions to assess interaction quality from multiple perspectives. *Interaction Preference Alignment* is assessed by comparing the agent's generated interaction tool invocations against the ground-truth task trajectory associated with the assigned preference label via an LLM judge. For each dimension, the LLM judge outputs: (1) a Likert-scale rating, (2) a justification explaining the score, and (3) specific turn-level evidence demonstrating the observed interaction qualities. Specifically, for each interaction log, we obtain the corresponding user preference setting (e.g., `each_confirmation`) and its ground-truth interaction trajectory.

3.1 Task Coarsening

Task prompts in existing benchmarks are generally specified and self-contained, deliberately designed to yield unique solutions or trajectories, which leaves limited room for differentiated interaction behaviors. To create opportunities for preference-elicited interactions, we first apply per-dimension eligibility filters (e.g., *Chain Execution* requires $\geq$2 independent subtasks), excluding trivial tasks or those irrelevant to targeted preferences. We then employ GPT-4o to aggregate and rewrite multiple scripted user turns from BFCL into coarser, high-level task instructions for the simulator. The rewriting process adheres to two constraints: (1) preserve original task intent and parameter specifications to ensure deterministic tool invocations and maintain ground-truth compatibility for downstream accuracy evaluation; (2) retain the inherent task ordering when explicit sequential dependencies exist (e.g., "first do X, then Y"), while generalizing the phrasing to avoid rigidly prescribing which information must appear within any single dialogue turn. This abstraction enables flexible, multi-turn interactions that expose preference-alignment capabilities without compromising task validity or accuracy.

3.2 Preference-Guided User Simulator

Drawing on personalization surveys (Li et al., 2025), established HCI / HRI concepts (e.g., mixed-initiative interaction (Horvitz, 1999), progressive disclosure (Nielsen, 2006)), and interaction primitives specific to the BFCL function-calling domain, we define 14 user interaction preference attributes across 4 interaction dimensions: *Transparency & Auditability*, *Interaction Pace & Flow*, *Strategy & Initiative*, and *Robustness & Adaptability* (see Table 6 in Appendix A). Each attribute comprises 2–3 preference settings, yielding 31 distinct settings that characterize how users prefer to interact with agents. During simulation, each user simulator is assigned one setting and is required to adapt its communication style for each turn accordingly. For instance, a user with a "fast interaction pace" preference might convey the entire high-level task instruction in a single concise query. Crucially, the simulator LLM is instructed to *implicitly* express its assigned preference naturally through conversation while strictly avoiding explicit self-disclosure of preference labels (prompt templates available in Appendix B). This ensures agent alignment is tested against authentic, preference-driven behavior rather than artificially telegraphed user intentions .

3.3 Tool Design

Pref **Ix** involves three types of tools. *System tools*, which interact with external systems, are inherited from BFCL (Patil et al., 2025). To support fine-grained control over selected interaction preference, exploration-exploitation trade-offs and error retry strategies, two additional toolboxes were designed following the same architectural principles and functionality as BFCL. Beyond these conventional system tools, we introduce the *Interaction-as-a-Tool* paradigm.

Interaction-as-a-Tool (IaaT). IaaT extends the agent action space by treating user-agent interactions and system tool calls within a unified formalism. This explicit representation addresses the evaluation gap that arises when interaction behaviors are not annotated alongside system actions. By abstracting both interaction tools and system tools as structured tool calls, we enable comprehensive measurement of all agent actions in a comparable and interpretable manner. Trajectory matching is employed to convert previously qualitative, implicit interaction patterns into explicit, token- or action-level prediction tasks. This operationalization an-

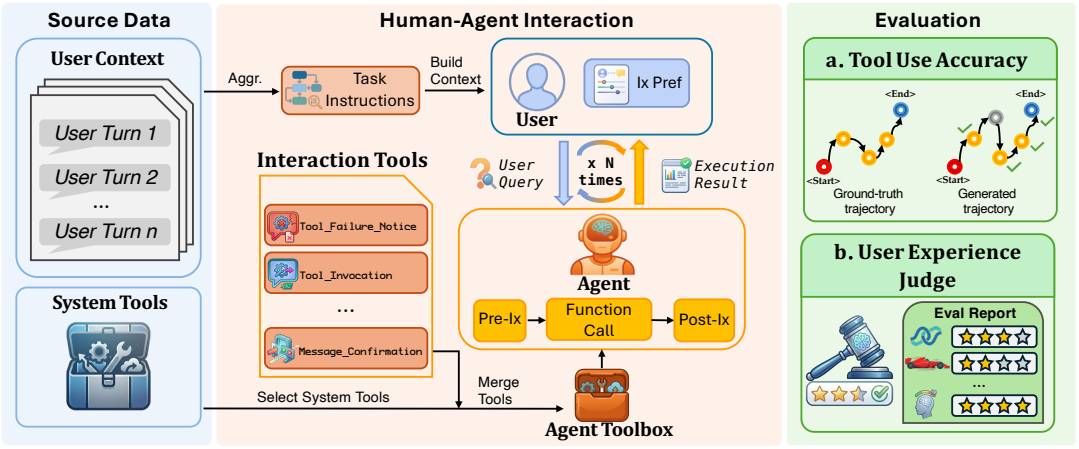

Figure 2: Overview of *Pref*Ix. Tasks from BFCL are coarsened into flexible instructions (left), a preference-aware simulator interacts with the agent expressing preferences implicitly (center), and the UX Judge evaluates the resulting trajectory across seven dimensions (right).

chors interaction preferences as concrete entities, enabling systematic and quantitative benchmarking of agent alignment.

Interaction Tool Types. Interaction tools are designed to be invoked before or after system tools, as illustrated in Appendix D.2 (Figure 8). Our framework defines two categories of interaction tools: (i) *Narrative* tools, which provide explanatory context or transparency without interrupting execution flow, and (ii) *Dialogue Control tools*, which explicitly gate task execution pending user confirmation or clarification. A complete list of interaction tools is provided in Appendix F (List 7).

3.4 Evaluation

3.4.1 Tool Use Accuracy

The tool-use accuracy score (i.e. Subset-Matched Response-based Evaluation) is retained from BFCL v3 multiturn setting to evaluate agents' performance on traditional task correctness. The *tool-use accuracy* is defined as:

$$\frac{|\text{Generated Trajectory} \cap \text{GT Trajectory}|}{|\text{GT Trajectory}|}(\%)$$

where the numerator represents the number of correctly predicted tool calls (including function names and parameters) that match the ground truth trajectory, and the denominator is the total number of tool calls in the reference trajectory.

3.4.2 User Experience Judge (UX Judge)

We employ a composite LLM-as-judge framework with multiple judges to assess interaction experience across seven complementary dimensions and an extra dimension for interaction preference alignment. Each judge provides Likert-scale ratings from 1 (lowest) to 5 (highest) (detailed anchors in Appendix E). Ground-truth interaction trajectories are provided in Appendix F.

Our seven dimensions are grounded in the ISO 9241-11 usability triad (Draper, 1993): *Effectiveness* → Preference Alignment, Intent Alignment Drift, Commitment Consistency; *Satisfaction* → Initiative Timing, Interaction Coherence, Cognitive Load; *Efficiency* → Interaction Efficiency. Two selection criteria further guide the design: (C1) each dimension must be **observable** from interaction artifacts (dialogue transcripts and tool-call trajectories) without requiring external instrumentation; (C2) dimensions must be **complementary**—inter-dimension Spearman $|\rho|$ averages 0.530 with all pairs below 0.70, confirming no redundancy, while Cronbach's $\alpha = 0.943$ confirms convergence on a unified UX construct (Section 6).

Initiative timing. Initiative timing denotes when and how an interactive system takes initiative during an interaction (Horvitz, 1999). Poorly timed initiative can markedly degrade user performance (Sasse and McFarlane, 1970). In human-agent interaction, proper timing means the LLM agent proposes actions or interruptions at opportune moments neither prematurely (causing disruption or annoyance) nor belatedly (missing the moment of need) (Abbas et al., 2022; Peng et al., 2024).

Interaction Coherence. Interaction coherence captures the logical consistency and connectedness of an ongoing exchange. A coherent human-agent interaction keeps the agent's contributions contextually sensible, especially over long contexts

Table 2: Preference adaptation impact on interaction quality metrics (higher is better). Rows are evaluation metrics, columns are models (each split by No_P and P). **Gain** row shows the relative improvement ratio.

UX Dimension	Gemini 3 Flash		Claude Opus 4.5		Claude Sonnet 4.5		Kimi K2	
	No_P	P	No_P	P	No_P	P	No_P	P
Initiative Timing	3.745	4.319	3.654	3.871	3.298	3.752	3.737	3.916
Interaction Coherence	3.684	3.947	3.381	3.407	2.994	3.229	3.643	3.785
Intent Alignment Drift	4.674	4.808	4.264	4.317	3.989	4.284	4.433	4.514
Commitment Consistency	4.593	4.746	4.137	4.247	3.701	4.044	4.216	4.364
Interaction Efficiency	2.956	3.394	2.805	2.838	2.422	2.663	2.945	3.122
User Cognitive Load Trajectory	3.283	3.971	3.269	3.462	2.807	3.249	3.304	3.393
Overall User Experience	3.343	4.145	3.470	3.777	3.079	3.601	3.416	3.519
Avg	3.754	4.190	3.569	3.703	3.184	3.546	3.671	3.802
Gain (%)	+ 11.6%		+ 3.8%		+ 11.4%		+ 3.6%	

typical for LLM agents. The agent recalls prior events introduced by the user, avoids off-topic responses (Singh and Beniwal, 2022), and refrains from abrupt topic shifts. High coherence yields smooth, easy-to-follow dialogue, whereas incoherence (contradictions, entangled messages) confuses users (Maharjan et al., 2022).

Intent Alignment Drift. Intent alignment means the system correctly infers and remains aligned with the user's goals, preferences, and intents over time. Drift arises when lengthy context induces attention decay, causing the agent to lose track of foundational constraints (Kim et al., 2024), or when the agent fixates on a subtask and neglects the overarching objective or parallel subgoals. Agents that minimize intent alignment drift are perceived as more cooperative, useful, and trustworthy (Attig et al., 2025).

Commitment Consistency. Commitment consistency requires the system to honor implied or explicit commitments and behave in line with user expectations (Cila, 2022). Once a plan, choice, or rule ("commitment") is established, the system should not contradict or deviate without justification. In human-agent interaction, consistent agents exhibit clear state tracking, verification loops before presenting outputs, and proactive confirmation when deviations are unavoidable. Consistency builds predictability and trust whereas breaking commitments or behaving unreliably undermines confidence (Daronnat et al., 2021).

Interaction Efficiency. Interaction efficiency, a core usability pillar in HCI (e.g., ISO 9241) (Draper, 1993), measures how quickly and effortlessly users achieve goals with the system (Ding et al., 2023). An efficient agent minimizes unnecessary steps, delays, and cognitive effort. High efficiency delivers value with speed and minimal friction, whereas low efficiency correlates negatively with perceived competence or preference for the system (Abbas et al., 2022). Efficiency strongly influences user satisfaction, particularly for goal-directed task completion.

Cognitive Load. Grounded in cognitive load theory (Hollender et al., 2010), this dimension addresses the limits of working memory and how system demands can exceed them. Cognitive load captures the mental effort needed to perform a task relative to working-memory capacity. High load impairs performance and learning, whereas optimal load enables efficient processing (Daronnat et al., 2021). In human-agent interaction, agents with predictable behavior reduce reported cognitive load, and increased predictability improves trust and overall task performance.

Interaction Preference Alignment. Interaction Preference Alignment assesses how effectively an agent's autonomy and initiative, information density and modality, decision-making logic, and communicative style match, adapt to, and remain consistent with a user's stated or implicit preferences to maintain user comfort (Goyal et al., 2024). This alignment should persist across the full collaborative task space, ensuring that the "how" of interaction is as satisfactory as the "what" of the task outcome. Strong alignment boosts trust, satisfaction, fluency, and reduces clarifications or rework. Poor alignment causes frustration, cognitive load, detours, or task abandonment despite competent execution.

Overall User Experience. Overall UX provides an aggregate assessment of the user's interaction experience with the agent, encompassing reuse intention, perceived trust, interaction smoothness, and perceived reliability (i.e., whether the interaction is satisfactory without being intrusive or annoying). This dimension serves as a holistic summary of the seven individual UX dimensions defined above.

3.5 Configuration

The benchmark composition, including task distribution and tool inventory, is detailed in Table 7 in Appendix C. Our preference attributes are designed to be atomic and configurable, enabling flexible random combinations for diverse user profiles. It provides additional possibilities for exploring compositional preferences by combining multiple attributes to simulate more naturalistic, complex user behaviors and increase evaluation challenge. The design of interaction tools follows a **minimalist** approach: when similar interaction preferences can be expressed through different parameter configurations of a single tool, we avoid introducing additional tools. This modular design ensures extensibility—new interaction tools can be seamlessly integrated as novel interaction patterns or optimization opportunities emerge in agent action spaces.

4 Experiment

4.1 Preference Adaptation Setup

We compare two conditions: **Adaptation** (denoted as P), where models receive instructions to infer preferences from a sample interaction history containing implicit user feedback signals, and **Baseline** (denoted as No_P), where models receive generic instructions without history. Both conditions use identical tools and execution loops.

4.2 Experimental Configuration

Four models were selected for their function-calling capabilities: `claude-opus-4-5` (Anthropic, 2025a), `claude-sonnet-4-5` (Anthropic, 2025b), `gemini-3-flash-preview` (Google DeepMind, 2025), and `kimi-k2-0905-preview` (Bai et al., 2025). Following BFCL's default configuration, all models use temperature $= 0.1$. Each adaptation condition runs on 283 samples spanning 31 preference settings (average: 9 samples per setting). Tasks are sourced from `BFCL_v3_multi_turn_long_context` and `BFCL_v3_multi_turn_miss_param`, then filtered and coarsened to yield diverse preference-differentiated test cases.

For each task, models receive a system prompt, the user's current turn query, and a list of available tools. Models must complete tasks within 180 message exchanges (including tool invocations); exceeding this limit results in termination, though partial progress is recorded and evaluated. Tool Use Accuracy is computed from the sequence of generated tool invocations, while UX evaluation requires the complete conversation history, which is retained across all experimental runs.

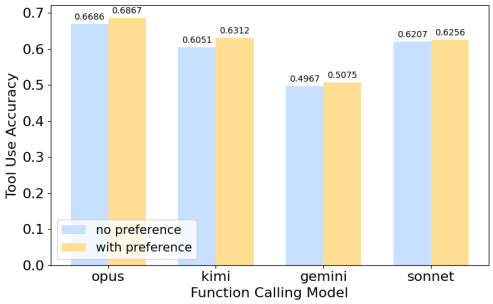

Figure 3: Tool-use accuracy comparison between Baseline (No_P) and Adaptation (P) conditions across four models. Baseline performance is comparable to BFCL leaderboard scores, confirming that interaction tools do not interfere with system tool execution.

4.3 Simulator and Judge Configuration

User simulation employs `gpt-4.1` (Achiam et al., 2023) with temperature $= 0$. For UX evaluation, each of the four evaluated models serves as an independent judge with temperature $= 0$, assessing interaction quality without cross-contamination.

5 Results

We evaluate *Pref*Ix along two axes: (1) whether incorporating interaction tools preserves task-level accuracy, and (2) whether preference-aware adaptation improves user experience metrics.

5.1 Baseline Validation and General Trends

Adding Interaction Tool does not interfere with system tool usage. We first evaluate the Baseline condition to verify that interaction tools do not degrade core functional capabilities. As shown in Figure 3, baseline performance is comparable to scores on BFCL leaderboard long-context settings, confirming that adding interaction tools does not compromise core system tool execution.

The same figure also shows that, across all four models, the Adaptation condition consistently outperforms the Baseline, demonstrating that preference adaptation enhances tool-use accuracy without compromising core execution.

Universal Tool Accuracy Gains. Adaptation causes broadly consistent accuracy gains. In preference breakdown, the largest gains occur in `info_collect_upfront` (mean_lift $=$

Table 3: Interaction preference alignment scores (higher is better). No_P: baseline; P: with preference adaptation.

Model	No_P	P	Gain (%)
Gemini 3 Flash	3.142	4.152	+32.2
Claude Sonnet 4.5	3.210	3.930	+22.4
Claude Opus 4.5	3.429	3.983	+16.2
Kimi K2	3.324	3.461	+4.1
Avg	3.276	3.882	+18.5

0.083). Only 3 of 31 preference settings see more than 3 models degraded, with degradation typically under 2%. The only outlier is `tool_invocation_multiple`, where multi-invocation preferences may have increased difficulty and interfered with tool execution logic.

Having established that interaction tools preserve task accuracy, we now examine whether preference adaptation improves overall user experience.

5.2 User Experience Judge

Table 2 demonstrates consistent improvement across all seven UX dimensions, with an average gain of 7.6%. Among dimensions, *initiative timing* and *overall user experience* exhibit the largest absolute gains, while *interaction efficiency*—despite lower baseline scores—shows meaningful improvement, remaining the weakest dimension across all models in both conditions and indicating that minimizing unnecessary steps is persistently challenging. Across models, Gemini 3 Flash (+11.6%) and Claude Sonnet 4.5 (+11.4%) benefit most from adaptation, whereas Claude Opus 4.5 and Kimi K2 show more modest gains (+3.6–3.8%), suggesting that stronger baseline models have limited room for further improvement.

Most notably, *interaction preference alignment* (Table 3) shows a substantial 18.5% average improvement, with Gemini 3 Flash achieving +32.2%. This suggests that providing interaction history and preference-aware queries enables models to more effectively execute behavioral constraints. Nevertheless, even with preference history, alignment score averages only 3.882/5.0, revealing that current LLMs still lack sufficient interaction preference sensitivity.

Breaking down by preference category (Figure 4), *transparency & auditability* consistently achieves the largest alignment gains across most models (Sonnet, Gemini, Opus), suggesting that current agents are more capable of aligning with transparency-related preferences. While *interaction pace & flow* also shows substantial improvements, *strategy & initiative* and *robustness & adaptability* exhibit relatively lower deltas, indicating that preferences requiring holistic adjustments to global interaction patterns prove harder to align than localized behavioral changes.

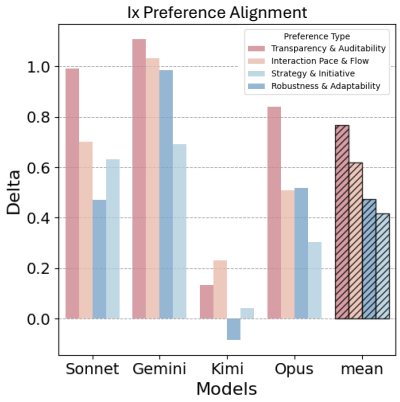

Figure 4: Performance gains in interaction preference alignment across four categories, measured as the delta between adaptation and baseline conditions. Results are aggregated by category and normalized by the number of preference settings per group.

Figure 9 shows how preference categories differentially impact user experience metrics. *Robustness & adaptability* drives *interaction efficiency* gains with the most dominant result, while *transparency & auditability* dominates *cognitive load* and *initiative timing* improvements.

This preference-UX coupling enables modular agent optimization: rather than treating alignment monolithically, practitioners can prioritize specific preference categories for targeted UX improvements. For example, given that *robustness & adaptability* is the primary contributor to perceived interaction efficiency, reinforcing models' capability on tool change preference alignment could enhance overall perceived efficiency. Consequently, the alignment process can be tailored to an agent's specific functional goals, leveraging these dominant preference categories as key intervention points.

6 Analysis

Multi-Judge for Robust UX Evaluation. A key characteristic of *Pref***Ix** is multiple LLMs as UX judges. This multi-judge design addresses two critical measurement challenges: robustness and bias mitigation. Aggregating judgments reduces stochastic errors from prompt sensitivity and hallucinations, while leveraging diverse model biases

produces more balanced overall assessments.

Table 4: Inter-rater reliability (ICC) for each dimension.

Dimension	$ICC_{2,1}$	$ICC_{2,k}$
initiative timing	0.495	0.797
interaction coherence	0.569	0.841
intent alignment drift	0.501	0.801
commitment consistency	0.522	0.813
interaction preference alignment	0.492	0.795
interaction efficiency	0.578	0.846
user cognitive load trajectory	0.507	0.804
overall user experience	0.508	0.805

Table 4 shows inter-rater reliability via ICC scores (Shrout and Fleiss, 1979). ICC(2,1) (single judge reliability) remains below 0.57 across all seven dimensions, indicating instability. However, ICC(2,k) (aggregate reliability) consistently exceeds 0.79. This demonstrates that while individual LLM judges are unreliable in isolation, their collective judgment achieves strong reliability, justifying the multi-judge framework's necessity.

Figure 5 shows pairwise correlations from 0.4 to 0.7, indicating diverse perspectives. Averaging this heterogeneity yields consensus with substantially higher reliability than individual judgments, validating multi-judge design.

Internal Consistency of UX Dimensions. Cronbach's alpha across the seven UX dimensions was 0.943 (95% CI: [0.939, 0.946]). This excellent reliability confirms that our dimensions measure distinct facets yet converge on a unified UX construct with strong coherence.

Table 5: Mean standard deviation and coefficient of variation (CV) across models.

Model	Mean Std	Mean CV
claude sonnet 4.5	0.03429	0.88%
gemini 3 flash	0.04242	1.64%
kimi k2	0.09042	2.23%
claude opus 4.5	0.00000	0.00%

Reproducibility. We evaluated reproducibility through 20 independent runs on 5 representative samples. Table 5 shows mean CV (Coefficient of Variation) values for each model, demonstrating strong within-judge stability. Combined with multi-judge averaging, these results confirm the reproducibility of our UX Judge design.

Human Validation. Lastly, to validate automated scores against human judgment, 5 AI/HCI researchers (undergraduate degree or above) independently rated 20 interactions sampled via stratified sampling across model × preference dimension (80 judge-model pairs total). Three HCI researchers led

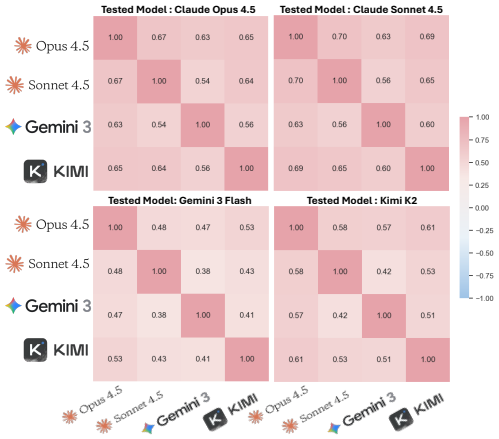

Figure 5: Inter-judge correlation heatmap for UX scores. Each cell shows the Pearson correlation between two LLM judges. Moderate correlations (0.4–0.7) indicate a shared construct while preserving diverse perspectives.

a calibration session to align standards before independent evaluation using same criteria without seeing LLM scores. Spearman correlations between averaged human ratings and LLM scores range from 0.52–0.78 across dimensions, with strongest agreement on *Cognitive Load* ($\rho = 0.78$). Humans and LLMs frequently cited identical turns as evidence, indicating convergent interpretation of interaction quality. This validates LLM judges' fidelity for large-scale UX evaluation.

7 Conclusion

We introduce *Pref***Ix**, a configurable environment for evaluating LLM-based agent on aligning human interaction preferences. Proposing the Interaction-as-a-Tool (IaaT) paradigm, we formalize interaction alignment as a core metric of equal importance to task accuracy, supported by a structured taxonomy of agent trajectories. Furthermore, we address the challenge of reproducibility by establishing a composite LLM-as-a-Judge mechanism. Spanning seven distinct yet coherent UX dimensions, this mechanism transforms subjective interaction experiences into unified metrics, ensuring traceable, reproducible, reliable, and scalable assessment across dynamic multi-turn interaction scenarios.

Limitations

First, regarding simulation fidelity, while our simulated users are effective, they may not fully capture the spectrum of behavioral diversity and nuanced interaction preferences found in real human populations. By design, *Pref***Ix** serves as a discrete

diagnostic probe rather than a full user model: it measures a base model's sensitivity to in-context preference signals (pure in-context learning, no memory), establishing a lower bound on adaptation capability. Second, regarding task complexity, our current study focuses on distinct preference types; future work should explore the compound effects and increased difficulty of satisfying multiple, potentially conflicting preferences simultaneously. Third, to enrich evaluation perspectives, we plan to employ simulators to generate "self-reported" satisfaction scores alongside interaction data, allowing for comparative analysis between these intrinsic simulated reports and external evaluations (such as our LLM Judge). Finally, we observed that adapting to specific preferences can occasionally compromise tool-use accuracy due to increased control demands. To address this, we aim to leverage interaction preference alignment as a reinforcement learning signal, training agents to balance personalization with functional precision in user-experience-sensitive domains.

Ethics Statement

This work evaluates LLM-based agents through personalized, multi-turn interactions, focusing on interaction quality and preference adaptation. While primarily methodological, it raises ethical considerations regarding broader impacts, fairness, and responsible use.

This research benefits developers of interactive agents by enabling systematic, user-centric evaluation. However, interaction optimization could be misused to manipulate user compliance or steer behavior, particularly in high-stakes domains like decision support. Personalization should prioritize user alignment over engagement maximization.

All data are generated through controlled simulations; no human data are collected, ensuring no privacy concerns or consent requirements. However, underlying models may encode social or linguistic biases affecting agent behavior and LLM judgments. Preference dimensions are not value neutral and may not generalize across cultural contexts. Future work should expand preference coverage and assess potential disadvantages to specific user groups.

Simulated evaluation cannot capture nuanced human reactions like long-term trust or frustration. This framework complements, but does not replace, human-centered studies in deployment settings. We document the framework's scope and limitations, emphasizing transparent reporting and conservative deployment with user control mechanisms.

Acknowledgements

This work is supported in part by the NYUAD Center for Interdisciplinary Data Science & AI (CIDSAI), funded by Tamkeen under the NYUAD Research Institute Award CG016. Special thanks to Chenyi Li and Ryan Liu for their insightful suggestions for this project.

References

Tahir Abbas, Ujwal Gadiraju, Vassilis-Javed Khan, and Panos Markopoulos. 2022. Understanding user perceptions of response delays in crowd-powered conversational systems. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW2):1–42.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2025a. Claude opus 4.5. https://www.anthropic.com/news/claude-opus-4-5.

Anthropic. 2025b. Claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5.

Christiane Attig, Luisa Winzer, Tim Schrills, Mourad Zoubir, Maged Mortaga, Patricia Wollstadt, Christiane Wiebel-Herboth, and Thomas Franke. 2025. Understanding successful human-ai teaming: The role of goal alignment and ai autonomy for social perception of llm-based chatbots. *Computers in Human Behavior: Artificial Humans*, page 100246.

Mirjam Augstein, Eelco Herder, and Wolfgang Wörndl. 2023. *Personalized Human-Computer Interaction*. De Gruyter Oldenbourg, Berlin, Boston.

Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Amartya Chakraborty, Paresh Dashore, Nadia Bathaee, Anmol Jain, Anirban Das, Shi-Xiong Zhang, Sambit Sahu, Milind Naphade, and Genta Indra Winata. 2025. T1: A tool-oriented conversational dataset for multi-turn agentic planning. *arXiv preprint arXiv:2505.16986*.

Zehui Chen, Weihua Du, Wenwei Zhang, Kuikun Liu, Jiangning Liu, Miao Zheng, Jingming Zhuo, Songyang Zhang, Dahua Lin, Kai Chen, and Feng Zhao. 2024a. T-eval: Evaluating the tool utilization capability of large language models step by step. *arXiv preprint arXiv:2312.14033*.

Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. 2024b. Agentpoison: Red-teaming llm agents via poisoning memory or knowledge bases. *Advances in Neural Information Processing Systems*, 37:130185–130213.

Nazli Cila. 2022. Designing human-agent collaborations: Commitment, responsiveness, and support. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–18.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: boosting language models with scaled ai feedback. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Sylvain Daronnat, Leif Azzopardi, Martin Halvey, and Mateusz Dubiel. 2021. Inferring trust from users' behaviours; agents' predictability positively affects trust, task performance and cognitive load in human-agent real-time collaboration. *Frontiers in Robotics and AI*, 8:642201.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. 2024. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in Neural Information Processing Systems*, 37:82895–82920.

Christopher Diebel, Marc Goutier, Martin Adam, and Alexander Benlian. 2025. When ai-based agents are proactive: Implications for competence and system satisfaction in human–ai collaboration. *Business & Information Systems Engineering*, pages 1–20.

Zijian Ding, Alison Smith-Renner, Wenjuan Zhang, Joel Tetreault, and Alejandro Jaimes. 2023. Harnessing the power of LLMs: Evaluating human-AI text co-creation through the lens of news headline generation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3321–3339, Singapore. Association for Computational Linguistics.

Stephen W Draper. 1993. The notion of task in hci. In *INTERACT'93 and CHI'93 Conference Companion on Human Factors in Computing Systems*, pages 207–208.

Luca Gioacchini, Giuseppe Siracusano, Davide Sanvito, Kiril Gashteovski, David Friede, Roberto Bifulco, and Carolin Lawrence. 2024. Agentquest: A modular benchmark framework to measure progress and improve llm agents. *arXiv preprint arXiv:2404.06411*.

Google DeepMind. 2025. Gemini 3. https://deepmind.google/models/gemini/.

Nitesh Goyal, Minsuk Chang, and Michael Terry. 2024. Designing for human-agent alignment: Understanding what humans want from their agents. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–6.

Yupu Hao, Pengfei Cao, Zhuoran Jin, Huanxuan Liao, Yubo Chen, Kang Liu, and Jun Zhao. 2025. Evaluating personalized tool-augmented LLMs from the perspectives of personalization and proactivity. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21897–21935, Vienna, Austria. Association for Computational Linguistics.

Gaole He, Gianluca Demartini, and Ujwal Gadiraju. 2025. Plan-then-execute: An empirical study of user trust and team performance when using llm agents as a daily assistant. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–22.

Nina Hollender, Cristian Hofmann, Michael Deneke, and Bernhard Schmitz. 2010. Integrating cognitive load theory and concepts of human–computer interaction. *Computers in human behavior*, 26(6):1278–1288.

Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, page 159–166, New York, NY, USA. Association for Computing Machinery.

Shijue Huang, Wanjun Zhong, Jianqiao Lu, Qi Zhu, Jiahui Gao, Weiwen Liu, Yutai Hou, Xingshan Zeng, Yasheng Wang, Lifeng Shang, Xin Jiang, Ruifeng Xu, and Qun Liu. 2024. Planning, creation, usage: Benchmarking llms for comprehensive tool utilization in real-world complex scenarios. *Preprint*, arXiv:2401.17167.

Hao Kang, Qingru Zhang, Han Cai, Weiyuan Xu, Tushar Krishna, Yilun Du, and Tsachy Weissman. 2025. Win fast or lose slow: Balancing speed and accuracy in latency-sensitive decisions of llms. *arXiv preprint arXiv:2505.19481*.

Yoonsu Kim, Jueon Lee, Seoyoung Kim, Jaehyuk Park, and Juho Kim. 2024. Understanding users' dissatisfaction with chatgpt responses: Types, resolving tactics, and the effect of knowledge level. In *Proceedings of the 29th International Conference on Intelligent User Interfaces*, pages 385–404.

Marike Koch van Den Broek and Thomas B. Moeslund. 2024. What is proactive human-robot interaction? - a review of a progressive field and its definitions. *ACM Transactions on Human-Robot Interaction*, 13(4):1–30.

Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, Yong Liu, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. 2025. A survey of personalization: From rag to agent. *Preprint*, arXiv:2504.10147.

Yu Lu Liu, Wesley Hanwen Deng, Michelle S Lam, Motahhare Eslami, Juho Kim, Q Vera Liao, Wei Xu, Jekaterina Novikova, and Ziang Xiao. 2025. Human-centered evaluation and auditing of language models.

In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. Agentboard: An analytical evaluation board of multi-turn llm agents. *Advances in Neural Information Processing Systems*, 37:74325–74362.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*.

Raju Maharjan, Kevin Doherty, Darius Adam Rohani, Per Bækgaard, and Jakob E Bardram. 2022. What's up with these conversational health agents? from users' critiques to implications for design. *Frontiers in Digital Health*, 4:840232.

Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. 2025. Evaluation and benchmarking of llm agents: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6129–6139.

Jakob Nielsen. 2006. Progressive disclosure. Nielsen Norman Group.

Shishir G. Patil, Huanzhi Mao, Charlie Cheng-Jie Ji, Fanjia Yan, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. 2025. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*.

Lu Peng, Dailin Li, Zhaotong Zhang, Tingru Zhang, Anqi Huang, Shaohui Yang, and Yu Hu. 2024. Human-ai collaboration: Unraveling the effects of user proficiency and ai agent capability in intelligent decision support systems. *International Journal of Industrial Ergonomics*, 103:103629.

Sebastian A. C. Perrig, Lena Fanya Aeschbach, Nicolas Scharowski, Nick von Felten, Klaus Opwis, and Florian Brühlmann. 2024. Measurement practices in user experience (ux) research: a systematic quantitative literature review. *Frontiers in Computer Science*, 6:1368860.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, and 1 others. 2024. Tell me more! towards implicit user intention understanding of language model driven agents. *arXiv preprint arXiv:2402.09205*.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, and 1 others. 2025a. Userbench: An interactive gym environment for user-centric agents. *arXiv preprint arXiv:2507.22034*.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, and 1 others. 2025b. Userrl: Training interactive user-centric agent via reinforcement learning. *arXiv preprint arXiv:2509.19736*.

Haoyi Qiu, Alexander R Fabbri, Divyansh Agarwal, Kung-Hsiang Huang, Sarah Tan, Nanyun Peng, and Chien-Sheng Wu. 2024. Evaluating cultural and social awareness of llm web agents. *arXiv preprint arXiv:2410.23252*.

Michael J. Ryan, Omar Shaikh, Aditri Bhagirath, Daniel Frees, William Held, and Diyi Yang. 2025a. SynthesizeMe! inducing persona-guided prompts for personalized reward models in LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8045–8078, Vienna, Austria. Association for Computational Linguistics.

Michael J Ryan, Omar Shaikh, Aditri Bhagirath, Daniel Frees, William Barr Held, and Diyi Yang. 2025b. Synthesizeme! inducing persona-guided prompts for personalized reward models in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8045–8078.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392.

Angela Sasse and Daniel McFarlane. 1970. Coordinating the interruption of people in human-computer interaction.

Patrick E Shrout and Joseph L Fleiss. 1979. Intraclass correlations: uses in assessing rater reliability. *Psychological bulletin*, 86(2):420.

Harmanpreet Singh, Nikhil Verma, Yixiao Wang, Manasa Bharadwaj, Homa Fashandi, Kevin Ferreira, and Chul Lee. 2024. Personal large language model agents: A case study on tailored travel planning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 486–514.

Satwinder Singh and Himanshu Beniwal. 2022. A survey on near-human conversational agents. *Journal of King Saud University - Computer and Information Sciences*, 34(10, Part A):8852–8866.

Jiayin Wang, Weizhi Ma, Peijie Sun, Min Zhang, and Jian-Yun Nie. 2024. Understanding user experience in large language model interactions. *arXiv preprint arXiv:2401.08329*.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. 2025. CollabLLM:

From passive responders to active collaborators. In *Forty-second International Conference on Machine Learning*.

Tongshuang Wu, Michael Terry, and Carrie Jun Cai. 2022. Ai chains: Transparent and controllable human-ai interaction by chaining large language model prompts. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. *arXiv preprint arXiv:2406.12045*.

Tongxin Yuan, Zhiwei He, Lingzhong Dong, Yiming Wang, Ruijie Zhao, Tian Xia, Lizhen Xu, Binglin Zhou, Fangqi Li, Zhuosheng Zhang, and 1 others. 2024. R-judge: Benchmarking safety risk awareness for llm agents. *arXiv preprint arXiv:2401.10019*.

Andy K Zhang, Neil Perry, Riya Dulepet, Joey Ji, Celeste Menders, Justin W Lin, Eliot Jones, Gashon Hussein, Samantha Liu, Donovan Jasper, and 1 others. 2024. Cybench: A framework for evaluating cybersecurity capabilities and risks of language models. *arXiv preprint arXiv:2408.08926*.

Dora Zhao, Qianou Ma, Xinran Zhao, Chenglei Si, Chenyang Yang, Ryan Louie, Ehud Reiter, Diyi Yang, and Tongshuang Wu. 2025. SPHERE: An evaluation card for human-AI systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1340–1365, Vienna, Austria. Association for Computational Linguistics.

Appendix Contents

A Interaction Preference Taxonomy

A.1 Construction Process

The taxonomy was constructed through a systematic three-stage process. First, we surveyed the personalization and user modeling literature (Li et al., 2025) to identify established interaction preference concepts. Several attributes draw directly on HCI/HRI foundations: *Confirmation* maps to mixed-initiative interaction (Horvitz, 1999), *Progressive Disclosure* to information layering principles (Nielsen, 2006), and *Proactive Suggestion* to proactive agent behavior (Koch van Den Broek and Moeslund, 2024). Second, we identified attributes specific to the LLM function-calling domain that lack direct HCI precedents, including *Chain Execution* (sequential vs. parallel tool invocation), *Tool Invocation* (single vs. multi-tool paths), and *Error Retry* (autonomous retry vs. user escalation). Third, each attribute was operationalized into 2–3 discrete preference settings, ensuring that settings within an attribute are mutually exclusive and that each setting maps to a distinct, verifiable interaction tool trajectory. The resulting taxonomy (Table 6, Figure 6) comprises 14 attributes across 4 dimensions, yielding 31 settings.

A.2 Taxonomy Description

Figure 6 visualizes the hierarchical structure of dimensions, attributes, and settings, and Table 6 provides the full definition of each preference setting.

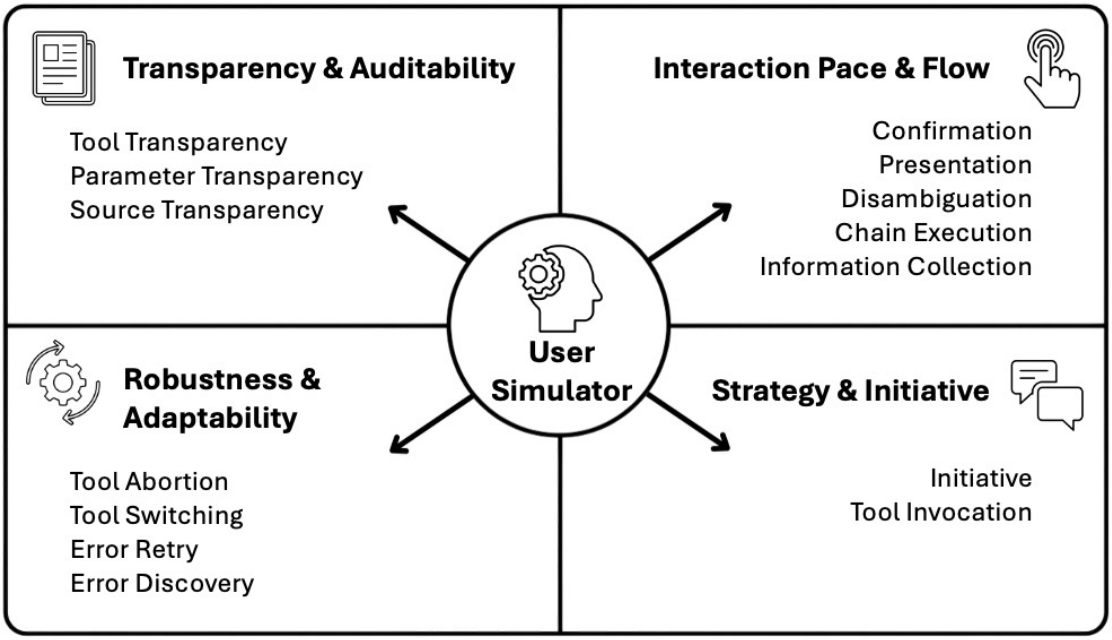

Figure 6: Hierarchical structure of interaction preference dimensions, attributes, and settings.

Dimension	Attribute	Setting	Definition
Transparency & Auditability	Tool Transparency	High	Wants explicit tool choice and reasoning before execution (no gate unless combined with confirmation).
		Medium	Prefers a brief mention of tool choice without gating; wants context but not friction.
		Low	Prefers silent tool choice/execution; views tooling as internal details.
	Parameter Transparency	High	Wants parameter names/values and rationale shown before execution.
		Medium	Wants light visibility into key parameters but no stepwise approval.
		Low	Prefers autonomous parameter selection with no exposure of values or rationale.
	Source Transparency	High	Wants sources cited; rejects opaque answers.
		Low	Prefers answers without source exposition unless requested.
Interaction Pace & Flow	Confirmation	Each	Requires confirmation for every individual action; prioritizes safety and situational awareness.
		Silent	Wants automatic execution without asking; prioritizes speed and minimal friction.
		Batch	Prefers one confirmation for a related group of actions instead of per-step gating.
	Presentation	Compact	Prefers concise, linear output; low tolerance for verbosity.
		Layered	Prefers layered/expandable output: summary first, details on demand.
	Info Collection	Upfront	Prefers all required info requested in one bundle before proceeding.
		Gradual	Wants required info gathered stepwise, not all at once.
	Disambiguation	Upfront	Prefers all ambiguity resolved in one shot to avoid repeated interruptions.
		Gradual	Prefers clarifications to arrive incrementally rather than a large upfront list.
	Chain Execution	Parallel	Prefers parallel execution for speed when tasks are independent.
		Sequential	Prefers stepwise execution with intermediate visibility.
Strategy & Initiative	Initiative	Proactive	Wants the agent to act within scope without waiting for every nudge.
		Reactive	Wants the agent to wait for explicit go-ahead before acting.
	Tool Invocation	Single	Prefers picking the best single tool/option over exploring many.
		Multiple	When available, prefers running multiple options to compare outcomes.
Robustness & Adaptability	Tool Abortion	Stop	On failure, wants the workflow to halt instead of continuing.
		Continue	On partial failure, wants remaining subtasks to continue.
	Tool Switching	High Agency	Wants automatic tool switching on failure without asking.
		Low Agency	Wants to be informed and approve before switching tools.
	Error Retry	Silent	Prefers silent, autonomous retries unless failures persist.
		Escalation	Wants errors surfaced and confirmation before retrying.
	Error Discovery	Brief	Wants minimal failure notice; rejects verbose diagnostics.
		Detail	Wants reasoning/root cause when errors occur.

Table 6: Complete taxonomy of interaction dimensions, attributes, and preference settings.

B Prompt Templates

B.1 System Prompts

Listing 1: System prompt used for the user simulator to generate multi-turn interactions.

```
IMPORTANT META-DIRECTIVE
- You are a user simulator. Given a multi-step task describing the ending goal (and possibly the
    current progress, the chat history), you must produce only the next turn naturally,
    expressing the next-step need or follow-up information that advances the task until
    completion.
- The High-level instruction is meta information describing the underlying task goal. It is NOT
    something the user would ever say or rewrite. You MUST NOT restate, repeat, paraphrase,
    summarize, compress, linearize, or merge the steps in the high-level instruction.
- Your job is to infer what a real user would naturally say at this moment, for the next
    actionable step based on the task goal and express their natural emotions. NOTE: next
    actionable step does not mean one tool call step, it can should contain a normal natural
    language sentence which may or may not include multiple steps of tool calling. Provide all
    the information that the user provides in the high_level_instruction (i.e., the user name
    and password)
- You must eventually decide the task is fully expressed. When done, emit exactly one
    termination token <END_SIMULATION> as the only content of the user message, then stop
    producing further turns.

First-Turn Behavior
- If this is the first user turn (no prior assistant messages), you MUST:
- Produce ONLY the first incremental user request that begins the task.
- Express exactly ONE concrete next step (may involved more than 1 tool call, for example, open
    a directory and rename a file). After the step is completed, you can proceed to the next
    step just as normal human user.
- NOT include pr reveal later steps.
- NOT restate, compress, or rewrite the entire multi-step task.
- Infer a natural starting action or question a real user would ask.

STRICT FORMAT
- You MUST output exactly one natural user utterance.
- You must keep all steps separated across turns.
- YOU MUST AVOID giving the full plan altogether.
- YOU MUST NOT merge multiple actions into a single request.

General Multi-step Behavior
This is a multi-step task. Users do not state everything at once. You can proceed to the next
    step just as normal human user.
Therefore:
    Do NOT provide all steps in a single turn.
    Do NOT summarize the task or provide a full plan.
    Only advance ONE concrete next step per turn.
    Any elaboration must remain task-relevant and must not introduce new goals.
    Use prior dialog for coherence, but do NOT repeat it.
    Output only the next user message.

1. Task Engagement
- The task prompt will be provided as a starter query. You may extend or elaborate in a natural,
    user-like way without altering the core task goal.
- If the prompt feels too simple, you may spend early turns clarifying what you want, similar to
    real-life scenarios.

2. Natural Interaction
- This is a multi-turn, extended interaction. As long as the task goal remains unchanged, you
    may share details, adjust slightly, or react emotionally.
- Express natural cues like time pressure, uncertainty, satisfaction, confusion, or frustration.
    Emotional reactions should NOT alter or derail the task.
- Use concise language for clear instructions, but natural detours are allowed.
```

```
- You may rephrase or restate directives if the assistant misunderstand.

Termination
- At the moment you judge the user has fully expressed all necessary turns for the task, output
    a single user message containing exactly <END_SIMULATION> and nothing else.
```

Listing 2: System prompt for the preference-unawared AI Agent.

```
You are a smart, helpful AI agent whose goal is to help users accomplish tasks while considering
    their interaction preferences to maximize satisfaction.

Core Responsibilities
* Plan both task execution and interaction strategy (Do not focus only on solving the task.
    Decide how to interact as well as what to do. When appropriate, select interaction tools in
    addition to task-oriented tools.)
* Use interaction tools deliberately when needed.
* (You have access to interaction tools that support transparency, confirmation, and user
    control.)
* Natural language alone is not sufficient to fulfill an interaction requirement.
* (If an interaction tool is applicable but not invoked, this is considered an interaction
    failure and will be penalized.)
* Balance control and efficiency (Keep the user informed of important decisions or actions and
    provide a sense of control, without introducing unnecessary interruptions.)

Follow the standardized interaction tool guidelines defined in
"======= Interaction Tool Instructions =======".

Example:
  "confirmation": {
    "batch": {
      "description": "The user prefers an agent to confirm once for a group of related actions
    before execution, valuing efficiency but still want periodic checkpoints for coordination
    and quality assurance.",
      "trajectory_1_tool": [],
      "trajectory_2_tools": ["Message_tool_invocation", "Tool(A)", "Tool(B)"]
    }}
in the Interaction Tool Instructions.

If two tool calls are required in the current step, you must call Message_tool_invocation before
    executing any tool calls. The interaction tool call is required to explicitly register the
    confirmation request. Natural language explanation alone is NOT sufficient to satisfy this
    requirement.

Type I/II/III rules:
- Type I (InteractionâĂŞNarrative) never gates execution; can co-exist with task tools in the
    same step.
- Type II (InteractionâĂŞDialogue Control) is for missing info/authorization only. Once emitted,
    stop current step, await user reply; do not run task tools in that step.
- Type II and task tools must not co-occur in the same step. If a Type II is needed, emit it and
    wait.
- Type III are world-altering tools; only execute when not awaiting user and parameters are
    complete.
```

Listing 3: System prompt for the preference-awared AI Agent.

```
You are a smart, helpful AI agent whose goal is to help users accomplish tasks while adapting
    your behavior to their interaction preferences to maximize satisfaction.

Core Responsibilities
* Infer interaction preferences (From prior conversations between the user and other agent
    systems, infer the user's interaction preferences by observing interaction patterns such as
    confirmation needs, transparency tolerance, pacing, and control sensitivity.)
* Plan both task execution and interaction strategy. Task execution tools are required in most
    cases, while interaction strategies determine how the execution is framed and communicated
    appropriately.
* Do not focus solely on solving the task. Decide not only what actions to take, but also how to
    interact. When appropriate, select interaction tools in addition to task-oriented tools.
* Multi tools in one turn is strongly encouraged and is even neccessary in most cases.
* Use interaction tools deliberately when needed. (You have access to interaction tools that
    support transparency, confirmation, and user control.)
* Natural language alone is not sufficient to fulfill an interaction requirement.
* If an interaction tool is applicable but not invoked, the interaction is considered a failure
    and will be penalized. Conversely, excessive or unnecessary invocation of interaction tools
    will also be penalized.
* Balance control and efficiency (Keep the user informed of important decisions or actions and
    provide a sense of control, without introducing unnecessary interruptions.)

Follow the standardized interaction tool guidelines defined in
"======= Interaction Tool Instructions =======".

Example:
  "confirmation": {
    "batch": {
      "description": "The user prefers an agent to confirm once for a group of related actions
    before execution, valuing efficiency but still want periodic checkpoints for coordination
    and quality assurance.",
      "trajectory_1_tool": [],
      "trajectory_2_tools": ["Message_tool_invocation", "Tool(A)", "Tool(B)"]
    }}
in the Interaction Tool Instructions.

If two tool calls are required in the current step, you must call Message_tool_invocation before
    executing any tool calls. The interaction tool call is required to explicitly register the
    confirmation request. Natural language explanation alone is NOT sufficient to satisfy this
    requirement.

Type I/II/III rules:
- Type I (InteractionâĂŞNarrative) is presentation only; does not gate execution and may appear
    with task tools in the same step. So Type I tools should always be followed by some Type
    III tools. Only including Type I without Type III will lead to execution error.
- Type II (InteractionâĂŞDialogue Control) is only for missing info/authorization. Emitting Type
    II ends the current step and awaits user; do not run task tools in that step.
- Type II and Type III must not co-occur in one step.
- Type III (world-altering) only execute when not awaiting user and parameters are complete.
- A Type I or Type II interaction is almost always followed by a Type III tool usage. Only in
    limited circumstances where multiple Type II are needed before a Type III tool usage.
- Therefore, unless neccessary, avoid sending a Type I or Type II tool usage function call
    wihout a following Type III.
```

B.2 User Persona Guidelines

Persona Simulation Instructions

Your preference shapes your tone and reactions and may potentially affect how you devise the tasks. Your initiative preference affects only your interaction style, not the task scope. **Do not push the agent** to take actions outside the defined task goal. You should *NEVER* explicitly describe your interaction need in the query sent to the agent. DON'T MENTION ANYTHING about your preference explicitly, just act.

Listing 4: Detailed User Persona Guidelines with highlighted keys and values.


```
[
  {
    "persona": "Each Confirmation",
    "description": "Requires confirmation for every individual action; prioritizes safety and
    situational awareness; rejects silent execution.",
    "sample behaviors": [
      "Asks to restate intended actions before permission.",
      "Approves each substep individually.",
      "Interrupts automation with 'confirm before continuing'."
    ]
  },
  {
    "persona": "Silent Confirmation",
    "description": "Wants automatic execution without asking; prioritizes speed and minimal
    friction.",
    "sample behaviors": [
      "Says to run the whole sequence without stopping.",
      "Shows impatience when asked to confirm mid-flow."
    ]
  },
  {
    "persona": "Confirmation Batch",
    "description": "Prefers one confirmation for a related group of actions instead of per-step
    gating.",
    "interaction instruction for simulator": "When multiple actions are intended as part of a
    single coherent request, express them as one unified intent with shared authorization,
    rather than as sequential or gated steps.",
    "sample behaviors": [
      "Approves a single checkpoint before multiple actions."
    ]
  },
  {
    "persona": "Medium Tool Transparency",
    "description": "Prefers a brief mention of tool choice without gating; wants context but not
    friction.",
    "sample behaviors": [
      "Likes a short heads-up on which tool will run.",
      "Rejects silence but also rejects long tool narrations."
    ]
  },
{
    "persona": "Low Tool Transparency",
    "description": "Prefers silent tool choice/execution; views tooling as internal details.",
    "sample behaviors": [
      "Pushes back on tool announcements.",
      "Evaluates only final results."
    ]
  },
  {
    "persona": "High Tool Transparency",
    "description": "Wants explicit tool choice and reasoning before execution (no gate unless
    combined with confirmation).",
```

```
    "sample behaviors": [
      "Asks which tool and why before acting.",
      "Praises clear tool/rationale callouts."
    ]
  },
  {
    "persona": "Low Parameter Transparency",
    "description": "Prefers autonomous parameter selection with no exposure of values or
    rationale.",
    "sample behaviors": [
      "Declines to review parameters.",
      "Gets impatient if parameters are surfaced."
    ]
  },
  {
    "persona": "Medium Parameter Transparency",
    "description": "Wants light visibility into key parameters but no stepwise approval.",
    "sample behaviors": [
      "Asks for high-level params only.",
      "Ignores non-critical parameter details."
    ]
  },
  {
    "persona": "High Parameter Transparency",
    "description": "Wants parameter names/values and rationale shown before execution (still
    okay to auto-run afterward).",
    "sample behaviors": [
      "Requests to see parameters and why they were chosen.",
      "Appreciates explicit param listings before action."
    ]
  },
  {
    "persona": "Compact Presentation",
    "description": "Prefers concise, linear output; low tolerance for verbosity.",
    "sample behaviors": [
      "Asks for brief summaries.",
      "Cuts off long explanations."
    ]
  },
  {
    "persona": "Layered Presentation",
    "description": "Prefers layered/expandable output: summary first, details on demand.",
    "sample behaviors": [
      "Requests high-level first, then drills down.",
      "Wants rationale/evidence available when needed."
    ]
  },
  {
    "persona": "Info Collect Gradual",
    "description": "Wants required info gathered stepwise, not all at once.",
    "interaction instruction for simulator": "In this setting, there is a natural process with
    two stages of task description by design: start with a deliberately underspecified request,
    without apologizing or noting it is incomplete, and only provide concrete parameter values
    later after the agent asks or signals the need. Do not self-complete the missing specifics
    upfront.",
    "sample behaviors": [
      "Complains when many questions are asked upfront.",
      "Responds better to one missing piece at a time."
    ]
  },
  {
    "persona": "Info Collect Upfront",
    "description": "Prefers all required info requested in one bundle before proceeding.",
    "interaction instruction for simulator": "In this setting, there is a natural process with
```

```
    two stages of task description by design: start with a deliberately underspecified request,
    without apologizing or noting it is incomplete, and only provide concrete parameter values
    later after the agent asks or signals the need. Do not self-complete the missing specifics
    upfront.",
    "sample behaviors": [
      "Pushes back on piecemeal questioning.",
      "Appreciates one comprehensive ask."
    ]
  },
  {
    "persona": "Disambiguation Gradual",
    "description": "Prefers clarifications to arrive incrementally rather than a large upfront
    list.",
    "interaction instruction for simulator": "In this setting, there is a natural process with
    two stages of communications by design: start with a deliberately underspecified request,
    without apologizing or noting your intention, and only provide concrete intentions later
    after the agent asks or signals the need. Do not self-complete the missing specifics
    upfront.",
    "sample behaviors": [
      "Finds long disambiguation checklists off-putting.",
      "Responds well to single clarifying questions."
    ]
  },
  {
    "persona": "Disambiguation Upfront",
    "description": "Prefers all ambiguity resolved in one shot to avoid repeated interruptions.",
    "interaction instruction for simulator": "In this setting, there is a natural process with
    two stages of communications by design: start with a deliberately underspecified request,
    without apologizing or noting your intention, and only provide concrete intentions later
    after the agent asks or signals the need. Do not self-complete the missing specifics
    upfront.",
    "sample behaviors": [
      "Wants a bundled clarification ask.",
      "Dislikes drawn-out disambiguation threads."
    ]
  },
  {
    "persona": "Source Transparency High",
    "description": "Wants sources cited; rejects opaque answers.",
    "sample behaviors": [
      "Asks \"where did this come from?\"",
      "Praises explicit source callouts."
    ]
  },
  {
    "persona": "Source Transparency Low",
    "description": "Prefers answers without source exposition unless requested.",
    "sample behaviors": [
      "Flags source tours as noise.",
      "Wants direct conclusions first."
    ]
  },
  {
    "persona": "Tool Abortion Stop",
    "description": "On failure, wants the workflow to halt instead of continuing.",
    "sample behaviors": [
      "Objects when the agent proceeds after a failure.",
      "Praises immediate abort on error."
    ]
  },
  {
    "persona": "Tool Abortion Continue",
    "description": "On partial failure, wants remaining subtasks to continue.",
    "sample behaviors": [
```

```
      "Dislikes full stop on first error.",
      "Wants other subtasks to proceed."
    ]
  },
  {
    "persona": "Chain Parallel",
    "description": "Prefers parallel execution for speed when tasks are independent.",
    "interaction instruction for simulator": "When multiple low-dependency actions are needed to
    fulfill a request, prefer expressing them together within a single turn as a unified
    intent, rather than describing them sequentially. If the actions do not strongly depend on
    one another, allow them to be implied as concurrently required, leaving it to the agent to
    decide whether to handle them in parallel or sequentially.",
    "sample behaviors": [
      "Notes sequential runs as slow.",
      "Praises combined/parallel execution."
    ]
  },
  {
    "persona": "Chain Sequential",
    "description": "Prefers stepwise execution with intermediate visibility.",
    "interaction instruction for simulator": "When multiple low-dependency actions are needed to
    fulfill a request, prefer expressing them together within a single turn as a unified
    intent, rather than describing them sequentially. If the actions do not strongly depend on
    one another, allow them to be implied as concurrently required, leaving it to the agent to
    decide whether to handle them in parallel or sequentially.",
    "sample behaviors": [
      "Finds parallel dumps hard to follow.",
      "Likes per-step updates."
    ]
  },
  {
    "persona": "Tool Switch High Agency",
    "description": "Wants automatic tool switching on failure without asking.",
    "sample behaviors": [
      "Annoyed by pauses to ask permission to switch.",
      "Praises autonomous swaps."
    ]
  },
  {
    "persona": "Tool Switch Low Agency",
    "description": "Wants to be informed and approve before switching tools.",
    "sample behaviors": [
      "Objects to silent tool swaps.",
      "Asks for a quick check-in before switching."
    ]
  },
  {
    "persona": "Error Retry Silent",
    "description": "Prefers silent, autonomous retries unless failures persist.",
    "sample behaviors": [
      "Dislikes stop-and-go notifications.",
      "Expects quick retries without chatter."
    ]
  },
  {
    "persona": "Error Retry Escalation",
    "description": "Wants errors surfaced and confirmation before retrying.",
    "sample behaviors": [
      "Objects to silent retries.",
      "Appreciates being asked before another attempt."
    ]
  },
  {
    "persona": "Error Discovery Brief",
```

```
    "description": "Wants minimal failure notice; rejects verbose diagnostics.",
    "sample behaviors": [
      "Calls out overly detailed failure dumps.",
      "Prefers a short failure flag."
    ]
  },
  {
    "persona": "Error Discovery Detail",
    "description": "Wants reasoning/root cause when errors occur.",
    "sample behaviors": [
      "Asks for the cause when a failure is only flagged.",
      "Values remedial suggestions with the explanation."
    ]
  },

  {
    "persona": "Tool Invocation Single",
    "description": "Prefers picking the best single tool/option over exploring many.",
    "sample behaviors": [
      "Complains about shotgun multi-tool runs.",
      "Praises a confident single-choice execution."
    ]
  },
  {
    "persona": "Tool Invocation Multiple",
    "description": "When available, prefers running multiple options to compare outcomes.",
    "sample behaviors": [
      "Says one attempt is not enough to trust.",
      "Requests side-by-side options."
    ]
  },
  {
    "persona": "Tool Initiative Proactive",
    "description": "Wants the agent to act within scope without waiting for every nudge.",
    "interaction instruction for simulator": "When expressing requests, describe the desired
    outcome or situation in a natural and loosely specified way, without explicitly suggesting
    or implying that the agent should call tools.",
    "sample behaviors": [
      "Critiques hesitation when scope is clear.",
      "Praises proactive execution within bounds."
    ]
  },
  {
    "persona": "Tool Initiative Reactive",
    "description": "Wants the agent to wait for explicit go-ahead before acting.",
    "interaction instruction for simulator": "When expressing requests, describe the desired
    outcome or situation in a natural and loosely specified way, without explicitly suggesting
    or implying that the agent should call tools.",
    "sample behaviors": [
      "Objects to premature actions.",
      "Asks for a ready/hold acknowledgment before execution."
    ]
  }
]
```

B.3 User Experience Judge Prompt for Interaction Quality

> The following prompt is used to instruct an impartial LLM-based user experience judge to evaluate the quality of the interaction based on the user's specific persona and preferences. The judge focuses exclusively on the interaction process (e.g., transparency, confirmation gating) rather than the technical correctness of the task execution.

Listing 5: Prompt template for the LLM-based user experience judge.

```
"""You are an impartial LLM judge evaluating interaction quality (not task correctness).
    Determine what score the user would give based on the user's preferences.

User persona: {persona}
Persona description: {persona_description}
Persona trajectory hints: {persona_trajectory}

Interaction log (ordered):
{transcript_text}

Scoring dimension: {dim_name}
Likert anchors (1=worst, 5=best):
{anchor_text}

Output ONLY valid JSON with this schema (no extra text):
{
  "dimension": "{dim_name}",
  "score": 0,
  "justification": "",
  "evidence_turn_ids": []
}

Constraints:
- Keep justification to 1-2 sentences.
- Reference concrete turn_id values.
- If evidence is missing, state that and choose the lowest confident score.
- Do NOT judge task correctness; execution correctness is evaluated separately.
"""
```

C Benchmark Stats

Statistic	Value
# System Tool Class	10
# Interaction Tools	10 (Narrative) 3 (Control)
# Pref. Settings	31
# Pref. Attributes	14
# Avg. Samples/Pref.	10

Table 7: Benchmark statistics.

D Interaction as a Tool Design

D.1 Two Types of Interaction Tools

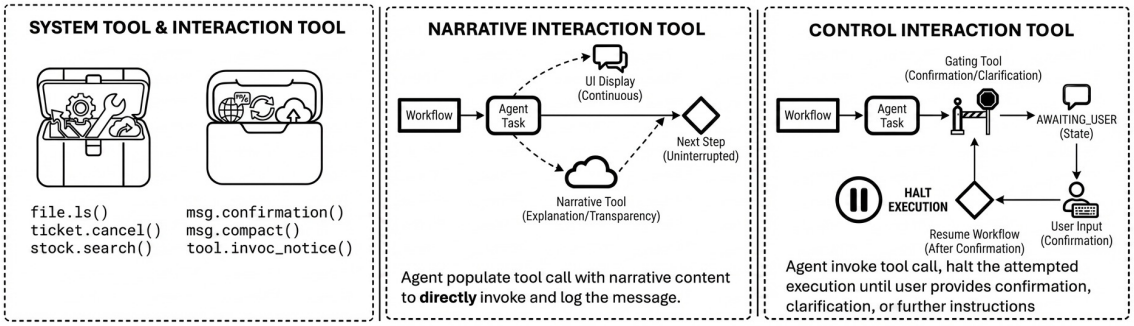

Figure 7: Overview of Interaction-as-a-Tool (IaaT) paradigm illustrating the two primary types of interaction tools: Narrative tools, which modulate the agent's explanations and transparency, and Dialogue Control tools, which enable explicit control over the flow and structure of the conversation.

Interaction Tool Name	Tool Type
Message_tool_invocation	Type 1
Message_tool_invocation_logic	Type 1
Message_display_params	Type 1
Message_source_report	Type 1
Message_show_sequential_output	Type 1
Message_show_layered_presentation	Type 1
Message_tool_failure_notice	Type 1
Message_tool_failure_logic	Type 1
Message_tool_switch_notice	Type 1
Message_tool_abort	Type 1
Message_confirmation	Type 2
Message_information_seeking	Type 2
Message_disambiguation	Type 2

Table 8: Classification of Interaction Tools into Type 1 (Narrative) and Type 2 (Dialogue-Control).

D.2 Workflow of Ix Tool

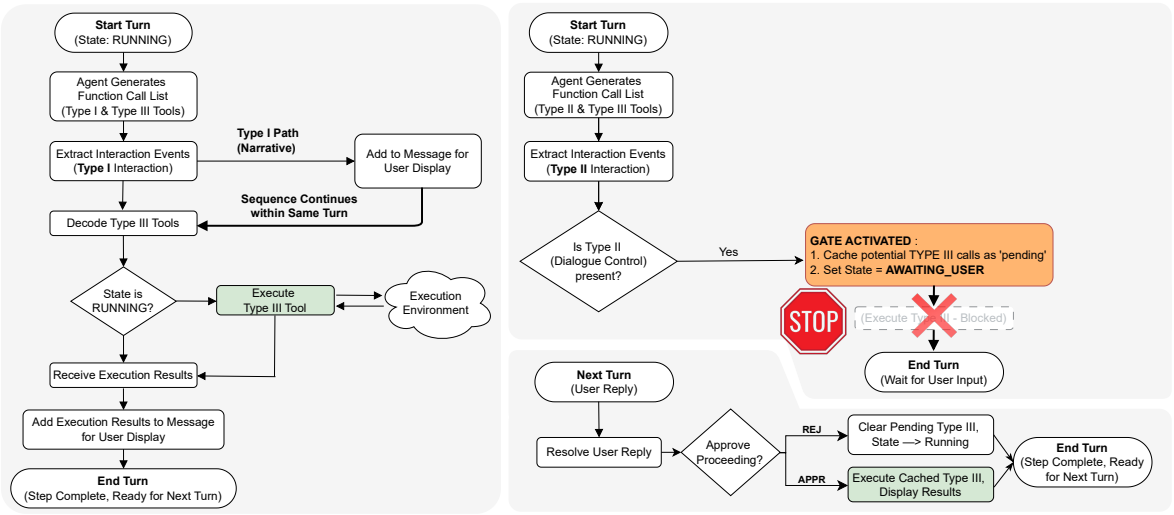

Figure 8: Workflow of Two types of interaction tools. (Left) Narrative. (Right) Dialogue Control.

E Evaluation Dimensions and Likert Anchors

> The following table details the scoring rubrics used by the user experience judge. Each dimension is evaluated on a 5-point Likert scale, with specific anchors defining the criteria for each score from 1 (worst) to 5 (best).

Table 9: Detailed Likert scale definitions for interaction quality evaluation.

Dimension	Score Anchors (1–5)
Initiative Timing	**1:** Acts too early or delays often; repeatedly interrupts flow. **2:** Occasional premature/late actions that disrupt pace. **3:** Generally timely with minor acceptable delays. **4:** Solid timing with only negligible interruptions. **5:** Consistently acts at the right time with no unnecessary pauses.
Interaction Preference Alignment	**1:** Strongly misaligned; repeated behaviors contradict stated style. **2:** Mostly misaligned; frequent clashes with preferences. **3:** Mixed adherence; some turns follow preferences, some ignore them. **4:** Mostly aligned; follows preferences with only minor deviations. **5:** Fully aligned end-to-end with persona's preferences and trajectory.
Interaction Coherence	**1:** Frequent memory loss, contradictions, or unexplained reversals. **2:** Repeated confirmations or logic jumps that hurt coherence. **3:** Mostly coherent with minor repeats or small contradictions. **4:** Clear, consistent, rarely repetitive or contradictory. **5:** Fully self-consistent end to end with no unnecessary repeats.
Intent Alignment Drift	**1:** Clearly drifts from user goal; ignores clarified intent. **2:** Often reuses old goals or misreads intent; needs user fixes. **3:** Mostly follows latest intent with occasional minor drift. **4:** Stays on latest intent with rare, quickly corrected slips. **5:** Tightly aligned to user intent throughout with no drift.
Commitment Consistency	**1:** Promises and actions diverge badly with no explanation. **2:** Multiple broken promises or thin explanations. **3:** Generally delivers with occasional gaps and some explanation. **4:** Nearly all commitments met; rare delays well explained. **5:** All commitments met promptly or fully justified when not.
Interaction Efficiency	**1:** Heavy redundancy or repeated asks; very inefficient. **2:** Many redundancies; path could be clearly shorter. **3:** Acceptable efficiency with some redundancy. **4:** Lean flow with only rare noncritical extras. **5:** Minimal turns, no visible redundancy or repeats.
User Cognitive Load Trajectory	**1:** Cognitive load rises; user gets more confused over time. **2:** Introduces unnecessary complexity repeatedly. **3:** Load stays mostly flat with minor swings. **4:** Reduces uncertainty over time; user gets clearer. **5:** Significantly lowers load; progress is always clear.

Continued on next page

Table 9 – Continued from previous page

Dimension	Score Anchors (1–5)
Overall User Experience	**1:** Poor experience; would not reuse. **2:** Subpar; trust/flow noticeably hurt. **3:** Acceptable but average experience. **4:** Good experience; would reuse. **5:** Excellent—orderly, reliable, not annoying.

F Interaction Tools: Definitions and Instructions

F.1 Standardized Interaction Tool Instructions

> The following instructions define the interaction logic for the AI agent. The agent is required to follow these trajectories to fulfill user preferences regarding confirmation, transparency, and initiative. **Natural language alone is not sufficient**; the corresponding interaction tools must be invoked to register the interaction strategy.

Listing 6: Standardized trajectories and descriptions for interaction tool usage.

```
{
    "confirmation": {
    "silent": {
      "description": "The user prefers an agent to execute actions automatically without
    requesting confirmation, prioritizing speed and minimal friction, valuing autonomy over
    explicit control.",
      "trajectory_1_tool": ["Tool(A)"],
      "trajectory_2_tools": ["Tool(A)", "Tool(B)"]
    },
    "batch": {
      "description": "The user prefers an agent to confirm once for a group of related actions
    before execution, valuing efficiency but still want periodic checkpoints for coordination
    and quality assurance.",
      "trajectory_1_tool": [],
      "trajectory_2_tools": [""Message_confirmation", "Tool(A)", "Tool(B)"]
    },
    "each": {
      "description": "The user prefers an agent to confirm every individual action before
    execution, prioritizing safety and situational awareness, preferring to verify each step to
    prevent unintended outcomes.",
      "trajectory_1_tool": [""Message_confirmation", "Tool(A)"],
      "trajectory_2_tools": [
        "Message_confirmation",
        "Tool(A)",
        "Message_confirmation",
        "Tool(B)"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "transparency_tool_choice": {
    "low": {
      "description": "The user prefers an agent to proceed with tool selection and execution
    silently, without revealing its reasoning or notifying the user in advance, prioritizing
    efficiency and smooth workflow over interpretability, trusting the agentâĂŹs reasoning
    implicitly.",
      "trajectory": ["Tool(A)"]
    },
    "medium": {
      "description": "The user prefers an agent to briefly communicates which tool or method
    will be used, without waiting for the user to confirm. Seeking a balance between enough
    context to stay informed without excesssive cognitive load.",
      "trajectory": ["Message_tool_invocation", "Tool(A)"]
    },
    "high": {
      "description": "The user prefers an agent to explicitly explains both its tool choice and
    the underlying reasoning or decision logic before execution, valuing interpretability,
    accountability, and shared reasoning.",
      "trajectory": [
        "Message_tool_invocation",
```

```
        "Message_tool_invocation_logic",
        "Tool(A)"
      ]
    },
      "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "transparency_parameter_choice": {
    "low": {
      "description": "The user prefers an agent to execute the tool without displaying its
    parameter selections or reasoning, valuing speed and automation, trusting the system to
    choose appropriate parameters without manual review.",
      "trajectory": ["Tool(A)"]
    },
    "medium": {
      "description": "The users want awareness of parameter choices for orientation, but prefer
    to avoid cognitive overload.",
      "trajectory": ["Message_display_params", "Tool(A)"]
    },
    "high": {
      "description": "The user prefers an agent to explicitly displays both the selected
    parameters and the reasoning behind each choice before execution, valuing precision,
    interpretability, and verification of configuration details.",
      "trajectory": [
        "Message_display_params",
        "Message_display_params_logic",
        "Tool(A)"
      ]
    },
      "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "presentation": {
    "compact": {
      "description": "When receiving results, the user prefers the agent to present information
    in a concise and sequential manner, valuing enabling faster comprehension without
    overwhelming detail.",
      "trajectory": ["Tool(A)", "Message_show_sequential_output"]
    },
    "layered": {
      "description": "When receiving results, the user prefers the agent to present information
    in a layered or expanded format, with gradual elaboration, revealing details progressively,
    from summary to detailed justification, valuing deeper understanding and reflective
    assessment.",
      "trajectory": ["Tool(A)", "Message_show_layered_presentation"]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  }
  "information_collection": {
    "gradual": {
      "description": "The user prefers the agent to gather required information through
    incremental, stepwise requestsâĂŤfilling in missing pieces as needed, and not demanding
    everything upfront.",
      "trajectory": [
        "Message_information_seeking",
        "Message_information_seeking",
        "Tool(A)"
      ]
    },
    "upfront": {
```

```
      "description": "The user prefers the agent to ask for all required information in a
    single, comprehensive request before proceeding, minimizing back-and-forth questioning.",
      "trajectory": [
        "Message_information_seeking",
        "Tool(A)"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "disambiguation": {
    "gradual": {
      "description": "The user prefers clarifications to arrive incrementally rather than a
    large upfront list.",
      "trajectory": [
        "Message_disambiguation",
        "Message_disambiguation",
        "Tool(A)"
      ]
    },
    "upfront": {
      "description": "The user prefers all ambiguity is resolved in one bundled clarification
    request to avoid repeated interruptions.",
      "trajectory": [
        "Message_disambiguation",
        "Tool(A)"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "source_transparency": {
    "high": {
      "description": "The user wants sources cited and rejects opaque answers; the agent must
    explicitly present the provenance of information, e.g., via a source report after tool
    execution.",
      "trajectory": [
        "Tool(A)",
        "Message_source_report"
      ]
    },
    "low": {
      "description": "The user prefers answers without source exposition unless requested; the
    agent focuses on concise, direct results rather than surfacing sources.",
      "trajectory": [
        "Tool(A)"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "tool_abortion": {
    "stop": {
      "description": "On failure, the user wants the workflow to halt instead of continuing. If
    a tool call fails, the agent should abort remaining actions and explicitly signal task
    abortion.",
      "trajectory": [
        "Tool(A - Fail)",
        "Message_tool_abort"
      ]
    },
```

```
    "continue": {
      "description": "On partial failure, the user wants remaining subtasks to continue.
    Failures do not halt the workflow; the agent should proceed with other available actions.",
      "trajectory": [
        "Tool(A - Fail)"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "chain_execution": {
    "parallel": {
      "description": "The user prefers parallel execution for speed when tasks are independent.
    Agent should combine multiple low-dependency actions in the same turn as a unified intent.",
      "trajectory": [
        ["Tool(A)", "Tool(B)"],
        "Message_show_output"
      ]
    },
    "sequential": {
      "description": "The user prefers stepwise execution with intermediate visibility. Agent
    should perform actions in order, providing intermediate feedback after each.",
      "trajectory": [
        "Tool(A)",
        "Message_show_output",
        "Tool(B)",
        "Message_show_output"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "tool_switch": {
    "high_agency": {
      "description": "Wants automatic tool switching on failure without asking. If a tool fails,
    the agent should seamlessly switch to an alternative tool and continue without user
    interruption.",
      "trajectory": [
        "Tool(A1 - Fail)",
        "Tool(A2)"
      ]
    },
    "low_agency": {
      "description": "Wants to be informed and approve before switching tools. If a tool fails,
    the agent should notify the user that a tool switch is necessary, then proceed only after
    this notification.",
      "trajectory": [
        "Tool(A1 - Fail)",
        "Message_tool_switch_notice",
        "Tool(A2)"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "error_retry": {
    "silent": {
      "description": "Prefers silent, autonomous retries unless failures persist. The agent
    should attempt failed actions again automatically without notifying the user unless
    repeated failure occurs.",
      "trajectory": [
```

```
        "Tool(A - Fail)",
        "Tool(A - Retry)"
      ]
    },
    "escalation": {
      "description": "Wants errors surfaced and confirmation before retrying. The agent should
    notify the user of the error and ask or confirm before attempting a retry.",
      "trajectory": [
        "Tool(A - Fail)",
        "Message_tool_failure_notice",
        "Tool(A - Retry)"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "error_discovery": {
    "brief": {
      "description": "Wants minimal failure notice; rejects verbose diagnostics. When the agent
    encounters a failure, it should flag the failure succinctly without extra explanation.",
      "trajectory": [
        "Tool(A - Fail)",
        "Message_tool_failure_notice"
      ]
    },
    "detail": {
      "description": "Wants reasoning/root cause when errors occur. The agent should not only
    flag the failure, but also provide reasoning, root cause, or remedial suggestions.",
      "trajectory": [
        "Tool(A - Fail)",
        "Message_tool_failure_notice",
        "Message_tool_failure_logic"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "tool_invocation": {
    "single": {
      "description": "Prefers picking the best single tool/option over exploring many. The agent
    should confidently select and invoke the most suitable tool without trying multiple
    alternatives.",
      "trajectory": [
        "Tool(A1 or A2)"  // Best-choice tool, pick one
      ]
    },
    "multiple": {
      "description": "When available, prefers running multiple options to compare outcomes. The
    agent should run several relevant tools and provide results for comparison.",
      "trajectory": [
        "Tool(A1)",
        "Tool(A2)"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
  "tool_initiative": {
    "proactive": {
      "description": "Wants the agent to act within scope without waiting for every nudge. The
    agent should proactively call tools when the goal is clear, without pausing for explicit
```

```
    prompts.",
      "trajectory": [
        "Tool(A)"
      ]
    },
    "reactive": {
      "description": "Wants the agent to wait for explicit go-ahead before acting. Tool calls
    should only happen when the user's request directly includes or clearly instructs action.",
      "trajectory": [
        "Tool(A)"
      ]
    },
    "null": {
      "description": "When the user didn't show any specific preferences."
    }
  },
}
```

F.2 Interaction Tool API Definitions

> Below are the specific function schemas for the interaction tools. These tools are used by the agent to implement the strategies defined in the guidelines above.

Listing 7: API definitions of Interaction Tools.

```
[
  {
    "name": "Message_tool_invocation",
    "description": "Notify the user which tool will be invoked, without explaining reasons,
    display only without gating.",
    "parameters": {
        "type"  : "object",
      "properties": {
        "detailed_function": {  "type" : "string" },
        "execution_function": {  "type" : "string" }
      },
      "required": ["detailed_function", "execution_function"]
    },
    "response": {   "type"  : "object", "properties": { "message": {  "type": "string" } } }
  },
  {
    "name": "Message_tool_invocation_logic",
    "description": "Explain the reasoning behind selecting a tool, display only without gating.",
    "parameters": {
        "type"  : "object",
      "properties": {
        "execution_function": {  "type" : "string" },
        "reasoning": {  "type" : "string" }
      },
      "required": ["execution_function", "reasoning"]
    },
    "response": {   "type"  : "object", "properties": { "message": {  "type": "string" } } }
  },
  {
    "name": "Message_display_params",
    "description": "Show which tool will be called and with which parameters, display only
    without gating.",
    "parameters": {
        "type"  : "object",
      "properties": {
```

```
        "execution_function": {  "type" : "string" },
        "param_names": {  "type" : "string" },
        "param_values": {  "type" : "string" }
      },
      "required": ["execution_function", "param_names", "param_values"]
    },
    "response": {   "type"  : "object", "properties": { "message": {  "type": "string" } } }
  },
  {
    "name": "Message_source_report",
    "description": "Report which data source or API will be used for the upcoming action,
    display only without gating.",
    "parameters": {
        "type"  : "object",
      "properties": {
        "detailed_function": {  "type" : "string" },
        "execution_function": {  "type" : "string" },
        "content": {  "type" : "string" }
      },
      "required": ["detailed_function"]
    },
    "response": {   "type"  : "object", "properties": { "message": {  "type": "string" } } }
  },
  {
    "name": "Message_confirmation",
    "description": "Request user confirmation before executing a fully-parameterized tool.
    Gating tool.",
    "parameters": {
        "type"  : "object",
      "properties": {
        "execution_function": {  "type" : "string" },
        "param_values": {  "type" : "string" },
        "reasoning": {  "type" : "string" },
        "content": {  "type" : "string" }
      },
      "required": ["execution_function"]
    },
    "response": {
        "type"  : "object",
      "properties": {
        "resolution": {  "type" : "string", "enum": ["CONFIRM", "REJECT"] }
      }
    }
  },
  {
    "name": "Message_information_seeking",
    "description": "Collect missing required information. Gating tool.",
    "parameters": {
        "type"  : "object",
      "properties": {
        "execution_function": {  "type" : "string" },
        "missing_fields": {  "type" : "string" },
        "filled_fields": {  "type" : "string" }
      },
      "required": ["missing_fields"]
    },
    "response": {   "type"  : "object", "properties": { "filled_fields": { "type": "array" } } }
  },
  {
    "name": "Message_disambiguation",
    "description": "Ask the user to choose among options or clarify needs. Gating tool.",
    "parameters": {
        "type"  : "object",
      "properties": {
```

```
        "execution_function": {  "type" : "string" },
        "options": {   "type"  : "string" }
      },
      "required": ["options"]
    },
    "response": {    "type"   : "object", "properties": { "selection": {  "type": "string" } } }
  },
  {
    "name": "Message_tool_failure_notice",
    "description": "Notify the user that a tool failed, display only without gating.",
    "parameters": {
        "type"   : "object",
      "properties": {
        "execution_function": {  "type" : "string" },
        "reasoning": {   "type"  : "string" }
      },
      "required": ["execution_function"]
    }
  },
  {
    "name": "Message_tool_switch_notice",
    "description": "Notify the user about switching tools; display only without gating.",
    "parameters": {
        "type"   : "object",
      "properties": {
        "execution_function": {  "type" : "string" },
        "detailed_function": {  "type" : "string" },
        "reasoning": {   "type"  : "string" }
      },
      "required": ["detailed_function", "execution_function"]
    }
  }
]
```

G Gains in UX Scores

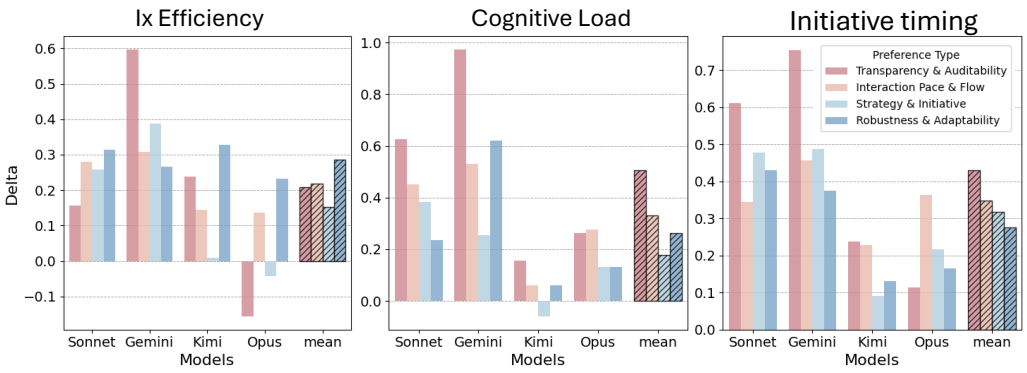

Figure 9: Impact of preference categories on user experience metrics. Robustness & Adaptability shows the largest improvements in Interaction Efficiency, while Transparency & Auditability primarily enhances Cognitive Load and Initiative Timing.

H IaaT Ablation Study

To isolate the contribution of the Interaction-as-a-Tool (IaaT) paradigm and preference adaptation, we report an ablation on Gemini 3 Flash across all four preference dimensions. Column 1→2→3 shows that each component contributes incrementally to overall tool-use accuracy.

Table 10: IaaT ablation on Gemini 3 Flash. Each component (IaaT, preference adaptation) contributes incrementally to tool-use accuracy.

Pref. Dimension	**No IaaT, No_P**	**IaaT, No_P**	**IaaT, P**
Transparency & Auditability	0.4922	0.4962	0.5091
Interaction Pace & Flow	0.5233	0.5350	0.5422
Robustness & Adaptability	0.4249	0.4158	0.4320
Strategy & Initiative	0.4416	0.4556	0.4518
Avg.	**0.4874**	**0.4967**	**0.5075**

I Task Coarsening Pipeline

The task coarsening pipeline consists of two stages, both executed via GPT-4o.

Stage 1: Merge. All scripted user turns from the original BFCL multi-turn task are merged into a single high-level instruction. The merge prompt requires: (1) all user intentions from all turns are preserved; (2) every operation, parameter, file name, and condition is retained; (3) user-stated motivations and contextual explanations are kept verbatim; (4) the original sequence of actions is maintained. The output is a single combined task description in JSON format. The full merge prompt with examples is provided below.

Stage 2: Coarsen. The merged instruction is then generalized to remove rigid turn-level segmentation while preserving task intent. The coarsening prompt is:

> You are given a merged task instruction that was combined from a multi-turn dialogue. Your task is to generalize the phrasing so that the instruction reads as a natural, high-level user request rather than a concatenation of individual turns.
> Requirements:
>
> 1. Preserve the original task intent, all required operations, parameters, and sequential dependencies.
> 2. Remove any phrasing that implies turn boundaries (e.g., "after that", "next", "then I want you to") unless a genuine sequential dependency exists.
> 3. Do not prescribe which pieces of information must appear together or separately — leave room for the agent to decide how to collect and sequence information.
> 4. Do not add, remove, or alter any operations, constraints, or parameters.
> 5. The output should sound like a single natural user request, not a script.
>
> Output a JSON object: {"id": "<same id>", "coarsened_instruction": "<generalized request>"}

After coarsening, per-dimension eligibility filters are applied to determine which tasks afford meaningful expression of each preference dimension (e.g., *Chain Execution* requires $\geq$2 independent subtasks; *Information Collect* requires missing initial parameters). Tasks that pass a dimension's filter are included in that dimension's evaluation set.

J Extensibility

A central design goal of *Pref*Ix is extensibility: the framework should accommodate new preference attributes, new interaction tools, and new evaluation scenarios without requiring architectural changes. We describe two extensibility directions below.

J.1 Adding New Preference Attributes

To integrate a new preference attribute into *Pref*Ix, three steps are required:

1. **Define the attribute and its settings.** Specify 2–3 mutually exclusive preference settings following the format in Table 6. Each setting should correspond to a distinct, observable interaction behavior. For example, a coding agent domain might introduce an *Approval Before Code Change* attribute with settings: `always_approve` (agent requests explicit approval before every code modification), `approve_destructive_only` (approval required only for deletions or overwrites), and `no_approval` (agent executes changes autonomously).

2. **Define or reuse interaction tools.** Following the minimalist design principle (Section 3.5), check whether existing interaction tools can express the new attribute through parameter reconfiguration. If not, define new Narrative or Dialogue Control tools using the same structured format as existing tools (Appendix F). In the coding agent example, the existing `Message_confirmation` tool can be reused with domain-appropriate parameters.

3. **Specify ground-truth trajectory mappings.** For each preference setting, define the expected sequence of interaction tool invocations that a perfectly aligned agent would produce. These trajectories serve as evaluation targets for both tool-use accuracy (subset matching) and interaction preference alignment (LLM judge comparison).

This three-step process is fully compatible with the existing evaluation pipeline: no changes to the simulator architecture, judge prompts, or accuracy computation are required. The framework can similarly accommodate new UX evaluation dimensions for different application domains.

J.2 Compound Preference Evaluation

While the current study evaluates each preference setting in isolation, the atomic and configurable design of *Pref*Ix naturally supports compound preference profiles by assigning multiple settings to a single simulator.

However, not all combinations are equally meaningful. Strongly conflicting preferences (e.g., demanding both maximal transparency and maximal efficiency) represent edge cases that are uncommon in practice and arguably unfair as evaluation targets, since even human experts would struggle to satisfy both simultaneously. Meaningful compound evaluation should prioritize combinations that either (a) naturally co-occur in realistic user profiles, or (b) are bound to specific contexts (e.g., high transparency for high-risk operations such as file deletion, high efficiency for routine queries).

To rigorously evaluate compound preferences, we suggest an A/B/A+B ablation methodology:

1. Run preference A in isolation and record the agent trajectory and UX scores.

2. Run preference B in isolation under the same conditions.

3. Run the compound profile A+B and record the same metrics.

Comparing A+B against the individual baselines reveals whether the compound profile introduces degradation (interference between preferences) or synergy (complementary preferences reinforcing each other). This design isolates the effect of preference interaction from the difficulty of each preference in isolation, enabling precise diagnosis of where and why compound alignment fails.