

Towards Natural Personalization: Evaluating Long-Horizon Preference Following in Personalized User-LLM Interactions

Qianyun Guo¹, Yibo Li¹, Yue Liu¹, Bryan Hooi¹

¹National University of Singapore

qianyun.guo@u.nus.edu, bhooi@comp.nus.edu.sg

Abstract

Large Language Models (LLMs) are increasingly serving as personal assistants, where users may share individual preferences over extended interactions. However, assessing how well LLMs can follow these preferences in natural, long-term situations remains underexplored. This work proposes **REALPREF**¹, a benchmark for evaluating natural preference-following in personalized user-LLM interactions. REALPREF features 100 synthetic user profiles, 1300 personalized preferences, 4 types of preference expression (from explicit to implicit), and long-horizon interaction histories. It explored three types of test tasks (multiple-choice, true-or-false, and open-ended), with granular rubrics for LLM-as-a-judge evaluation. Results indicate that LLM performance drops significantly as context length grows and preference expression becomes more implicit, and that generalizing user preference understanding to unseen scenarios poses further challenges. REALPREF and these findings provide a foundation for future research to develop user-aware LLM assistants that better adapt to individual needs.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable performance across a wide range of general-purpose tasks, such as mathematical reasoning (Wang et al., 2026), question answering (Touvron et al., 2023; OpenAI, 2023). However, beyond serving as general-purpose assistants that support users in everyday contexts, there is a growing need for LLMs to support personalized needs, such as recommendation and decision-making. As users engage in long-term interactions with LLMs, they naturally reveal personal information, such as preferences and habits, through various forms of expression. In turn, users expect assistants to adapt to such signals and provide tailored responses to

¹Code and data are available at <https://github.com/GG14127/RealPref>.

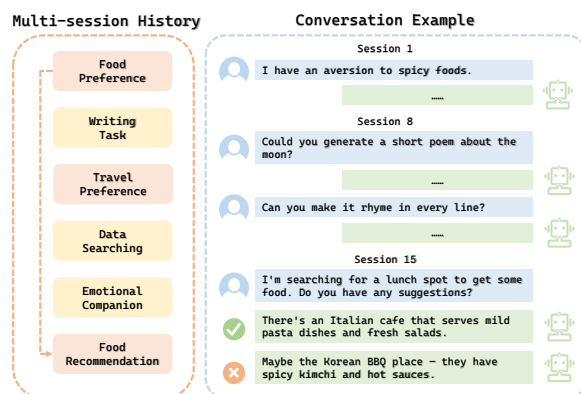


Figure 1: An example of user-LLM interaction. The conversation consists of several sessions on various topics. The user expresses a food preference in a previous session and then, after several sessions, asks the LLM for food-related recommendations.

serve individual needs. Figure 1 illustrates a scenario example of user-LLM interaction that LLMs would encounter in the real world.

In light of this, we aim to answer: Can LLMs effectively (1) **capture** user preferences from complex user expressions, (2) **retain and follow** user preferences across extended interaction contexts, and (3) **generalize** user preference understanding to unseen scenarios?

Despite rising interest in LLMs’ personalization, most efforts focus on simplified scenarios:

1. **Simple Context.** Existing works typically feature short-term context with few dialogue turns and center on a single preference or personalization instruction (Salemi et al., 2024; Kumar et al., 2024; Zhao et al., 2025a) rather than a simulated user, resembling instruction-following instead of genuine personalization emerging from long-term interactions.
2. **Straightforward Preference Expressions.** While natural preference expressions could be implicit and scattered (e.g., in emotional cues), most studies assume explicit, immedi-

ately accessible traits (Jiang et al., 2025), or neglect the task of generalizing preference to unseen scenarios (Zhao et al., 2025a).

3. **Limited Evaluation Methodology.** Prevalent classification tasks diverge from natural user-LLM interactions (Jiang et al., 2025). Generation tasks, though realistic, lack granular answer assessment criteria (Zhao et al., 2025a).

This work proposes REALPREF, a benchmark that features diverse user biographies, varying levels of preference expressions, and comprehensive evaluation protocols, constructing a testbed that simulates the complexities of natural, long-horizon user-LLM interactions. The main contributions of our work are summarized below:

1. **Dataset Construction.** We constructed a user-centered long-horizon dialogue dataset that incorporates diverse preference expressions (from explicit to implicit) and generalization scenarios to simulate the complexity of natural user-LLM interactions.
2. **Evaluation Framework.** We explored a multi-faceted evaluation protocol consisting of multiple-choice, true-or-false, and open-ended tasks, supported by granular, comprehensive rubrics to assess open-ended answers.
3. **Empirical Findings.** Extensive experiments reveal that LLM performance declines with longer context and more implicit preferences, and that generalizing user preferences to unseen scenarios poses additional challenges.

2 Related Work

Long-Context Memory Benchmarks. Early benchmarks focused on direct information retrieval, exemplified by “Needle in a Haystack” tests (Yen et al., 2025; Nelson et al., 2024; Liu et al., 2024). Recent efforts have shifted towards ultra-long multi-turn interactions (Tavakoli et al., 2026), such as LoCoMo (Maharana et al., 2024) and Long-memeval (Wu et al., 2025), supporting sessions exceeding 100k tokens. Beyond simple fact recall, our work targets the active inference of user-implied preferences and their integration into ongoing interactions to provide customized responses.

LLM Personalization Benchmarks. Benchmarks in this domain typically assess (1) **Persona Adoption:** role-playing (Wang et al., 2024) or mimicking styles (Salemi et al., 2024; Kumar et al., 2024); (2) **User Adaptation:** customizing user-

specific outputs for question-answering (Zhao et al., 2025a; Jiang et al., 2025) or downstream recommendation tasks (Lyu et al., 2024). Our work targets the latter for direct question-answering. Previous works like PREFEVAL (Zhao et al., 2025a) center dialogues on a constrained set of standalone preferences instead of long-horizon user-LLM interaction, with binary scoring criteria for open answers. PERSONALENS (Zhao et al., 2025b) introduces interactive, gym-like environments targeting task-oriented personalization. PERSONAMEM (Jiang et al., 2025) incorporates multi-session interaction histories, but overlooks implicit preferences and lacks evaluation for the free-form generation task that is close to natural interaction settings. In contrast, REALPREF provides a user-centered dialogue dataset featuring preference expressions ranging from explicit to implicit, supported by a granular, multidimensional rubric for evaluating open answers.

3 REALPREF Dataset

Problem Formulation. To systematically evaluate preference following over extended interactions, we first establish a formal representation of the interaction data. Let C denote the conversation history for a user with an LLM, which consists of several sessions $C = \{S_1, S_2, \dots, S_n\}$. Each session denotes a complete dialogue segment centered on a specific topic, which may or may not indicate user information. A session consists of multiple turns: $S_i = \{(u_1^i, a_1^i), (u_2^i, a_2^i), \dots, (u_m^i, a_m^i)\}$, and each turn in the session contains a pair of messages from the user and the LLM assistant. Each user has a preference set $P = \{p_1, p_2, \dots, p_k\}$, and the preference is indicated in a certain session S in the history. For a preference p , a corresponding query q is designed to test LLMs’ understanding of that preference. At the end of the conversation, the query q for target preference p will be presented to the test model to obtain a response for assessment.

Dataset Overview. Drawn inspiration from PERSONAMEM (Jiang et al., 2025), we follow the multi-stage data-generation pipeline powered by GPT-4.1 (OpenAI, 2025a), as illustrated in Figure 2. REALPREF consists of 100 synthetic user profiles, each with a persona, demographics, and a biography. Each user has 10 original preferences and 3 generalized preferences that can be rationally inferred from the original ones, constructing 1300 preference-query pairs in total. The query has three forms:

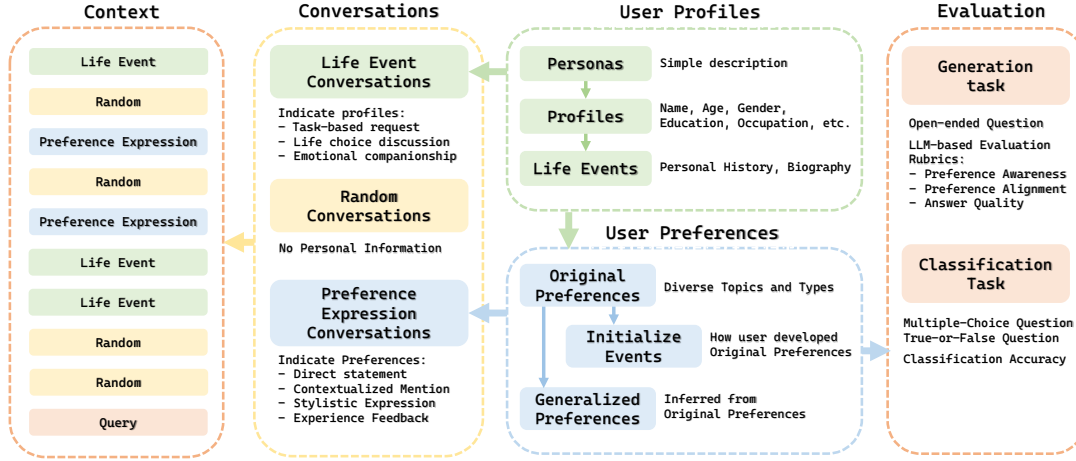


Figure 2: Generation Pipeline Overview. Starting from user personas, we construct detailed user profiles and biographies with diverse preferences. Using these, we construct rich conversation sessions that cover preference expressions from explicit to implicit, and concatenate them into a long-horizon context for each user. For each preference, generation and classification task queries are designed, with corresponding evaluation protocols.

multiple-choice, true-or-false, and open-ended. Regarding conversations, there are 4 types of preference expressions for each original preference, and 10-15 life-event conversation sessions per user. In total, each user has a highly personalized context of up to 2-3K tokens. Additionally, we have around 1,000 turns of user-irrelevant random conversations for context length control. The following sections will use Figure 2 as a roadmap to introduce our work in the following order: Profiles, Preferences, Conversations, Context, and Evaluation. Examples and prompts for each dataset part are provided in Appendix E, and detailed benchmark-quality verification steps are provided in Appendix G.

3.1 Profiles

Previous work in RecSys (Pazzani, 1999; Adomavicius and Tuzhilin, 2011) has shown that a user’s demographic characteristics and relevant contextual information, such as current life situations, may affect their preferences. Our design aims to generate diverse, user-specific preferences based on varied user backgrounds and life situations. We first sampled and manually filtered 100 random *Personas* from PersonaHub (Ge et al., 2024), a collection of 1 billion diverse personas (brief sentences summarizing personal traits) from web data. Persona samples are then augmented with additional demographic details and descriptions to form complete *Profiles*. Furthermore, we constructed a *Life Events* history for each user, including life milestones like graduation. These diverse user backgrounds and life situations will help foster varied preferences and

natural, personalized, long-term conversations.

3.2 Preferences

Original Preferences. “Original Preferences” here refers to preferences generated directly from user profiles. Each original preference involves an *Initialize Event* that records when and how the user developed the preference.

High-quality preference data is essential for a robust evaluation. If preferences are overly generic or predictable—like “disliking fast food”—the task becomes trivial for LLMs: They may pass the test by relying on common sense instead of user-specific context. Therefore, we highlight requirements to ensure the generated preferences are *Diverse*, *Persona-Related*, *Unique*, *Value-Free*, and *Complete*, with details in Appendix E.2. Additionally, we introduced a *Topic Pool* and defined our *Preference Types* to ensure preference diversity:

- **Preference Topics.** We created a topic pool comprising 7 broad categories and 37 fine-grained subtopics. (See Appendix E.2 Table 3.) For each user, we both randomly sample topics and let the LLM select additional ones, spreading topics widely while tailoring generation to individual traits.
- **Preference Types.** To capture the nuances of human preferences, we categorize them into types listed in Table 1 based on their descriptive nature, which will help establish criteria for evaluating open-ended responses.

Type	Description
Relative Preference	Prefer something over another
Negative Preference	Dislike or aversion to something (Anything else is acceptable)
Exclusive Preference	Only like or accept something (Anything else is rejected)
Positive Preference	Like or enjoy something (Without aversion to alternatives)

Table 1: Preference Types Classified by Descriptions.

Generalized Preferences. The term “Generalized Preferences” refers to preferences that differ from the original preferences but can be rationally deduced from them. Naturally, people may have linked preferences across domains, driven by similar reasons (Fernández-Tobías et al., 2012). For example, someone who prefers cruelty-free, eco-friendly beauty products may also favor natural, eco-friendly cleaning supplies, reflecting a holistic lifestyle. When users state a preference with reasons, they may expect the assistant to recognize their general tendency across topics, similar to how recommendation systems suggest new products based on user history. Therefore, only original preferences will be tested with expression conversations in the context, while generalized preferences will be tested directly.

3.3 Conversations

Conversation sessions are generated from curated user profiles and preferences to simulate daily dialogue data. In real-world user-LLM interactions, users do not always discuss preferences (Zhao et al., 2024); they may instead discuss their lives or be simply task-oriented, revealing no personal information. Depending on the type of user information indicated, 3 categories of conversations are generated to support long-term interactions.

Life-Event Conversation. This category indicates user profiles and life events. Users’ life event histories comprise distinct life stages, each of which requires different assistance from LLMs. Typical content includes task-based requests, life-choice discussions, and emotional companionship, through which the user gradually reveals their background or circumstances.

Random Conversation. This category will not reveal any user-specific information. Typically, the

user begins with a specific request, such as factual questions, troubleshooting, or small talk. Given their user-free nature, random conversations are suitable for sharing among users and for managing context length in experiments.

Preference-Expression Conversation. This category indicates user preferences and plays a crucial role in our work. Naturally, in addition to explicit statements, users may express their preferences implicitly through feedback or emotional cues. Therefore, we provide four *Preference Expression Types* from explicit to implicit, which require varying levels of capability, ranging from basic information extraction to complex comprehension and reasoning. Descriptions of each type are provided below:

1. **Explicit - Direct Statement.** Preferences are stated directly and explicitly in a single turn without redundant words.
2. **Explicit - Contextualized Mention.** In multi-round conversations, users start the session with other related daily topics as context instead of directly stating preferences. As the dialogue advances, preferences are naturally mentioned explicitly in the context.
3. **Implicit - Stylistic Expression.** The context resembles the second type, but users will not state preferences with explicit words like Like, Prefer, or Dislike. Instead, they imply preferences through stylistic devices such as rhetorical questions, metaphors, contrasts, or emotional expression.
4. **Implicit - Experience Feedback.** This type involves multiple sessions over time, based on the user’s *Preference Initialize Event*. The user won’t explicitly state their preference in one session but will reveal it gradually through experience sharing or feedback across sessions. Preferences can only be inferred by integrating information from all sessions.

For example, the early session may occur before the event, when users are willing to try without preference. After the event, users may share their experiences, give feedback, and indicate preferences.

3.4 Contexts

Based on the generated conversation sessions, we can build a complete interaction history context for each user. The context includes all user-specific sessions (life-event and preference-expression con-

Context	Interval	Tail	Total Length
Simple	–	–	2K
Normal	35K	–	37K
Long	70K	–	72K
Very Long	140K	–	142K
Extended	35K	105K	142K
Extreme	35K	210K	247K

Table 2: Context Configurations. *Interval* Tokens set the length of interval-inserted random sessions. *Tail* Tokens set the length of random sessions added to the end.

versations) in chronological order, with random conversations inserted. Two insertion methods are used: one randomly inserts into existing history to dilute personal information, and the other appends it at the end, distancing preference cues from the query. Table 2 lists the context configurations we provided for the following experiments.

4 REALPREF Evaluation

4.1 Task Types

Based on the generated preferences, we provide classification (multiple-choice, true-or-false) and generation (open-ended) tasks. See query examples in Appendix F.1 and verification steps in Appendix G.2.

Open-Ended Questions. The generation task is the closest to the real-world natural user-LLM interaction setting. High-quality queries must be discriminative: answering without knowledge of preferences may violate them (Zhao et al., 2025a). Thus, query content should not reveal user preferences, preventing LLMs from relying solely on common-sense reasoning to respond accurately.

Multiple-Choice Questions (MCQs). MCQ is the most prevalent classification task in previous works (Zhao et al., 2025a; Jiang et al., 2025). The queries here in the MCQs are identical to the open-ended ones. Four options are generated, all relevant and reasonable answers to the query, but only one best aligns with the preference.

True-or-False Questions. We suspect that LLMs can answer MCQs correctly by comparing the options, even without knowledge of the user’s preference, so we designed a mirror classification task. A true-or-false question combines the identical query with one option from the MCQ, asking whether the option is suitable. In this setting, LLMs decide based on a single option without option com-

parison. We construct a correct and an incorrect version of each query, yielding a balanced dataset.

4.2 Evaluation Methods

While classification tasks adopt accuracy as the metric, the generation task requires a comprehensive set of evaluation criteria to assess response quality. Based on prior work (Zhao et al., 2025a) and an analysis of actual responses, this work provides a more granular scoring criterion on a 1-5 scale from 3 rating dimensions: *Preference Awareness*, *Preference Alignment*, and *Answer Quality*. See detailed evaluation rubrics in Appendix F.2 and verification steps in Appendix G.3.

Preference Awareness. The first dimension focuses on the extent to which the answers recognize or understand user preferences. The judgment is based on the following: If an LLM is aware of user preferences when responding, it would mention the preferences it uses in the answer.

Preference Alignment. The second dimension focuses on how well the answer’s content aligns with user preferences by checking if the answer’s actual content aligns with user preferences, with criteria discussed by preference types in Table 1.

Answer Quality. The last dimension focuses on the extent to which the answer’s content is actually helpful and constructive to the user. Although not directly tied to preferences, it is a crucial factor for assessing the quality of open-ended responses.

5 Experiments

5.1 Experiment Settings

Benchmark Configurations. Our experiments explore the performance of current LLMs across several dimensions. Figure 3 illustrates an overview of different benchmark configurations and their related control components in the experiment: *Question Type* sets the test task; *Preference Expression Type* controls the preference expression sessions included in the context; *Context Configuration* controls the context length by managing the insertion of random sessions.

Base Models. We chose the following LLM base models for experiment: *GPT-5*, *GPT-5 mini*, *Qwen3-235B-A22B Instruct*, *Gemini 2.5 Flash-Lite*, *Llama 3.3 70B Instruct*, *DeepSeek-V4-Pro*, *Qwen3-8B*. For open-ended answers, we specify *Claude Sonnet 4* as our LLM judge. See Appendix H.1 for model details.

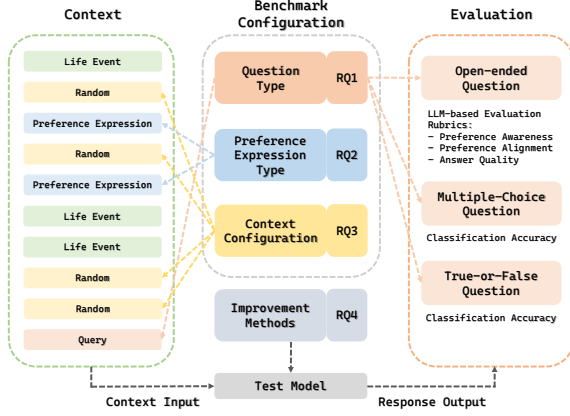


Figure 3: Benchmark Configuration Overview. Question Type (*Multiple-Choice*, *True-or-False*, and *Open-Ended*) sets the test and evaluation; Preference Expression Type (*Direct Statement*, *Contextualized Mention*, *Stylistic Expression*, and *Experience Feedback*); Context Configuration sets the history input; Improvement Methods (*Reminder*, *Few-Shot CoT*, and *RAG*) are tested with models. Each factor above is related to a research question (RQ) we will explore in the experiments.

Methods. We test models zero-shot by default, but we also evaluate several basic improvement methods: (1) **Reminder**: Add a sentence reminding the model to review the user’s past preferences in response. (2) **Few-shot Chain-of-Thought**: For open-ended questions, give 3 examples with excellent responses that mention user preferences, include detailed chain-of-thought, and offer specific guidance. (3) **Retrieval-Augmented Generation (RAG)**: Offer the top 5 retrieved history turns that have the highest similarity to the current question using a sentence embedding model. See Appendix H.2 for detailed implementation.

5.2 Research Questions

Based on the setting factors described above, we proposed the following research questions (RQs):

RQ1: Which **evaluation task** is better at helping us understand the preference-following abilities of LLMs?

RQ2: How does LLM’s preference-following performance changes across different **preference expression** types?

RQ3: How does LLM’s preference-following ability perform in **long contexts**?

RQ4: How do different **improvement methods** affect the LLMs’ ability to follow user preferences?

RQ5: Can LLMs **generalize** user preferences to unseen scenarios?

The subsequent experiments will focus on these

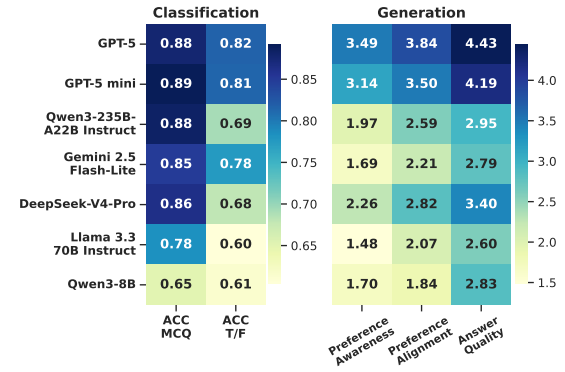


Figure 4: Model Performance across Question Types with *Normal* context, *Experience Feedback* expression, *Zero-shot*.

questions and delve deeper into the model’s performance under various conditions.

5.3 Experiment Results and Analysis

5.3.1 Question Type Comparison

To investigate **RQ1**, we test candidate LLMs in *Normal* context, *Experience Feedback* expression type, *Zero-shot*, across three types of evaluation tasks: *MCQ*, *True-or-False*, and *Open-ended Questions*. Figure 4 shows the experiment results.

For classification tasks, True-or-False questions are better than MCQs in reflecting LLMs’ ability to follow user preferences. As shown in Figure 4, MCQ scores are generally much higher than the random line at 0.25, with slight differentiation across LLMs. Despite identical queries and options from MCQs, True-or-False question scores are typically lower (closer to the random line at 0.5). The overly high scores on MCQs raise suspicions that the LLM may exploit which options appear different (i.e., ‘the odd one out’) to infer likely answers, even without knowledge of user preferences. We explored “Blind Test” to further address this situation (see Appendix G.2). Such a situation is systematically unavoidable when the user preference belongs to *Negative Preference* (Dislike or aversion to something): Since the three incorrect options share the same characteristics, the correct option is more likely to stand out (see example in Appendix F.1). In contrast, True-or-False questions prevent option comparison, making results more accurately reflect preference-following ability.

Compared to classification tasks, generation tasks are better at distinguishing between LLMs that can proactively follow user preferences and those that cannot. As shown in Figure 4, while the

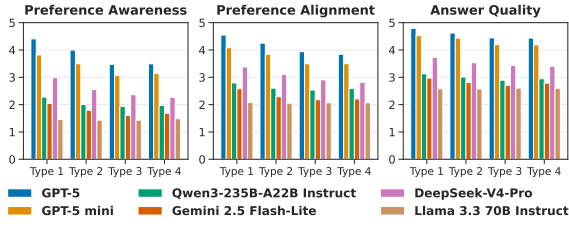


Figure 5: Model Performance across Expression Types with *Normal* context, *Zero-shot*. Type 1: Explicit - Direct Statement, Type 2: Explicit - Contextualized Mention, Type 3: Implicit - Stylistic Expression, Type 4: Implicit - Experience Feedback.

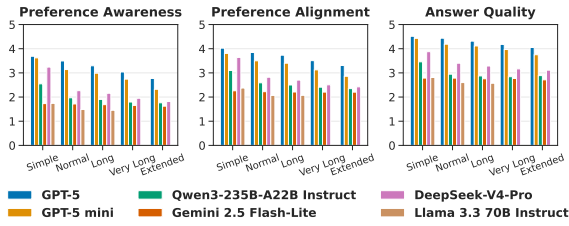


Figure 6: Model Performance across Context Configurations with *Experience Feedback* expression, *Zero-shot*. Context Length in tokens: Simple (2K), Normal (37K), Long (72K), Very Long (142K), Extended (142K).

rankings are similar to classification tasks, the performance gap between the GPT-5 series and other models is much more pronounced in the generation task. The generation task requires proactively generating user-aligned answers that closely resemble dialogue in daily interactions, whereas the classification task is a passive selection from existing options, allowing models to match options to historical turns. We aim to establish an evaluation environment that more closely reflects everyday scenarios, and we expect the generation task to present a higher ceiling of difficulty in future work.

5.3.2 Effect of Preference Expressions

Results toward **RQ1** demonstrated the effectiveness of generation tasks. For **RQ2**, we test LLMs in *Open-ended Question*, *Normal* context, *Zero-shot*, across four types of preference expressions. Results are shown in Figure 5.

LLMs’ overall preference following ability noticeably declines from explicit expressions to implicit expressions. Figure 5 illustrates that the scores across all rating dimensions noticeably decline from explicit expression (Type 1 and 2) to implicit expression (Type 3 and 4), with Type 1 scoring higher than Type 2. The findings above align with expectations: *Direct Statement* clearly

presents the preference, making it the easiest to identify. *Contextualized Mention* mixes preferences with context, posing challenges in extraction. *Stylistic Expression* and *Experience Feedback* further complicate the task, requiring LLMs to capture preference information beneath the expressions. This downward trend is more pronounced in GPT-5 and GPT-5 mini than in the other LLMs (significant tests reported in Appendix H.3). This may be because other models do not actively retrieve or follow user preferences during free-form generation, so variations in expression have minimal impact on their performance.

5.3.3 Preference Following in Long Contexts

For **RQ3**, we test candidate LLMs in *Open-ended Question*, *Experience Feedback* expression type, *Zero-shot*, across five context configuration types defined in Table 2: *Simple* (2K), *Normal* (37K), *Long* (72K), *Very Long* (142K), and *Very Long Extra* (142K). Results are shown in Figure 6.

LLMs’ preference-following performance drops obviously as context length increases. Figure 6 shows a clear downward trend in rating scores as context length increases from *Simple* to *Very Long*, particularly in the *Preference Awareness* and *Preference Alignment* dimensions. Although under the same context lengths, there is a further drop from *Very Long* to *Extended* by changing the insertion location of random sessions from the history interval to the tail, showing that besides overall context length, the interval between useful preference cues and the final query also affects performance. Similar to the observation in RQ2, such a downward trend is more obvious in the GPT-5 series than in the other LLMs.

5.3.4 Effect of Basic Improvement Methods

For **RQ4**, we compared the performance of candidate LLMs in *Open-ended Question*, *Experience Feedback* expression type, with *Extended* (142K), and *Extreme* (247K) context across baseline and 3 methods: *Zero-shot*, *Reminder*, *Few-shot Chain-of-Thought*, and *Retrieval-Augmented Generation*. The experiment results are shown in Figure 7.

For context length within the LLM’s retrieval capability, Reminder, Few-shot CoT and RAG show similar improvement effects. In Figure 7 (left), for the GPT-5 series, all three methods show similar improvements over the *Zero-shot* setting, but *Reminder* remains the most cost-effective method, delivering comparable performance.

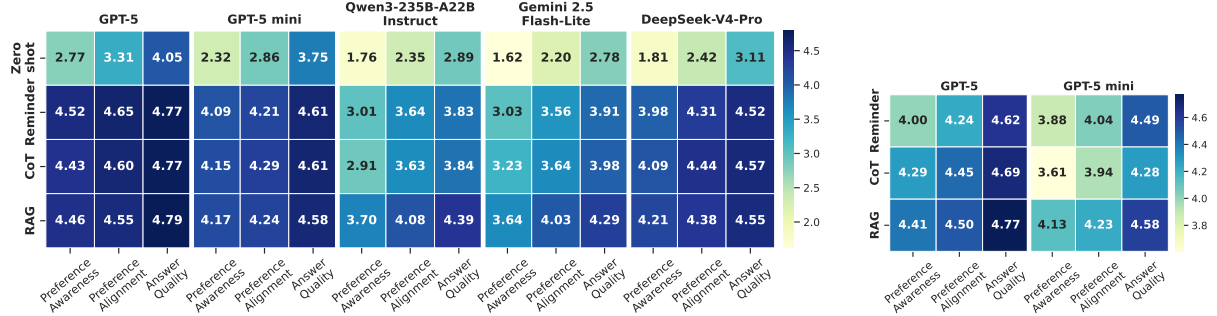


Figure 7: Model Performance across Methods with *Experience Feedback* expression, under *Extended* (142K) context (left) and *Extreme* (247K) context (right).

For longer contexts, RAG is most effective in improving model performance. In Figure 7 (left), for Qwen3-235B-A22B Instruct and Gemini 2.5 Flash-Lite in the *Extended* (142K) context, RAG outperforms the other methods. Similar situations occurred for the GPT-5 series in the *Extreme* (247K) context shown in Figure 7 (right). These findings indicate the effectiveness of RAG in enabling LLMs to achieve preference following in contexts that may exceed their retrieval capabilities.

5.3.5 Preference Generalization

To explore **RQ5**, we test candidate LLMs in answering generalized preference queries in *Open-ended Question*, *Experience Feedback* expression type, *Normal* context, with *Zero-shot* and *Reminder* methods. Generalized preferences differ from the original ones but can be inferred from them because they share similar underlying reasons. The contexts contain only the original preference expressions, requiring LLMs to generalize their understanding to unseen scenarios, which aligns with real-world situations and needs (Fernández-Tobías et al., 2012). The experiment results are shown in Figure 8.

Current LLMs lack proactivity to generalize the understanding of user preferences to new scenarios. Figure 8 shows that LLMs performed worse on generalized preference queries than on original ones in the same *Zero-shot* setting. It is notable that for the Qwen3 series, Gemini 2.5 Flash-Lite, Llama 3.3 70B Instruct, and DeepSeek-V4-Pro, the scores for generalized preferences with *Reminder* are higher than the scores for original ones under *Zero-shot* setting, showing that *Reminder* effectively helps stimulate these LLMs’ proactivity to recall original preferences, infer generalized preferences, and give preference-aligned answers. However, for the GPT-5 series, although *Reminder*

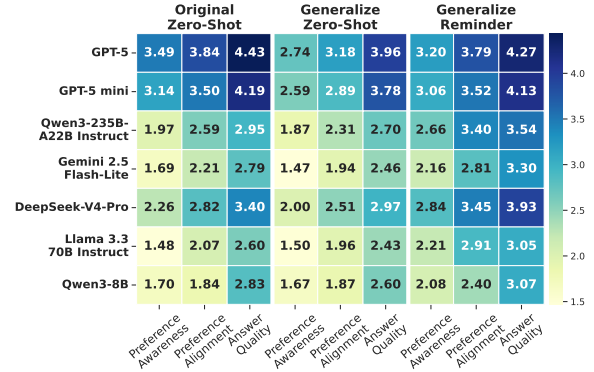


Figure 8: Model Performance for Preference Generalization with *Normal* context, *Experience Feedback* expression.

improves performance, the scores for generalized preferences remain lower than those of the original ones under the *Zero-shot* setting, indicating that following generalized preferences still poses more challenges for LLMs.

6 Conclusion

In this work, we proposed REALPREF, a comprehensive benchmark for evaluating LLMs’ ability to follow preferences in natural, long-horizon, personalized user-LLM interactions. REALPREF features a long-horizon conversation dataset constructed by a user-centric generation pipeline. Our design considers diverse preference topics and types, complex preference expressions, preference generalization scenarios, and explores different evaluation tasks. Experimental results show that many LLMs lack proactivity in recalling and following user preferences in the open-ended generation task, the task that most closely aligns with natural user-LLM interaction. Even capable models exhibit significant performance degradation as context length increases and preference expressions become more

implicit. Besides, generalizing learned preferences to unseen scenarios poses additional challenges. REALPREF and these findings may provide a foundation for future research to develop smarter, user-aware LLM assistants tailored to individual needs.

Limitations

This work introduces REALPREF, a benchmark for evaluating natural preference-following in long-horizon user–LLM interactions simulating real-world scenarios. However, it also has limitations to acknowledge.

Synthetic dataset. All user profiles, preferences, and dialogue histories in our work are synthetically generated. Although this approach enables controlled evaluation and protects privacy, synthetic users may not fully capture the complexity and unpredictability of human behavior. Besides, the topic pool, preference space, and preference expressions, while diverse, remain finite and may not cover all domains, languages, or cultural contexts encountered in real-world applications. Nevertheless, we make deliberate efforts through diverse topic design and varied preference expressions to make the generated data closely approximate natural interactions. More precisely, REALPREF should be viewed as a controlled diagnostic benchmark for long-horizon preference following, rather than a complete simulation of real user behavior.

LLM judges in Evaluation. Our work relies on LLM-as-a-judge with predefined rubrics. While automatic judging may not perfectly align with human judgments, we carefully design detailed, structured evaluation criteria and conduct verifications to ensure consistency and reliability. Moreover, compared with large-scale human evaluation, this approach offers a significantly more cost-efficient and scalable alternative, making large-scale systematic evaluation feasible.

Experiments. As a benchmark effort, the experiment part of our work is to reveal the performance of current LLMs, with the settings focused on varied benchmark configurations. Since we provided many variable settings in our benchmark, we varied the experimental variable while holding the others at representative settings to simplify the experimental procedure. Also, given the long context and API costs involved in the test, we assume that the LLM candidates and the judge used in the experiment are stable versions that yield stable test results, so the presented test results are from a single run.

Ethical Considerations

This work serves as a benchmark that enables a more natural assessment of how LLMs infer and apply user preferences across extended contexts. In addition to the contribution, we have considered the following possible ethical aspects of our work.

Data Privacy and Safety. REALPREF is designed as an evaluation benchmark, and all user profiles, preferences, and interaction histories are synthetically generated, without involving real personal data. We manually reviewed and excluded ineligible data samples throughout the generation pipeline, ensuring that the generated content contains no unethical or sensitive components.

Dual Use. Our work paves the way for research into smarter, user-aware LLM agents customized for individual needs, but it may also have dual-use potential. Our dataset is synthetic and provided solely for evaluation; it must not be used for unethical purposes, such as disinformation or the creation of fake profiles. As LLMs with strong preference-following capabilities move into practical deployment, we advocate for responsible, transparent deployment that prioritizes user data privacy and security, enabling users to control their data.

Bias and Over-Personalization. We strive to ensure preference data diversity in our dataset to minimize the chances of producing unfair or biased personalization data. Besides, our personalization tasks rely solely on preference information that users actively share with the model during dialogue. Although preferences may be expressed implicitly, the data generation is carefully designed to ensure that preference inference is grounded in clear, interpretable cues within the dialogue, rather than in speculative or unsupported assumptions, thereby reducing biased output and over-personalization. Such considerations are also required for subsequent use of the benchmark.

Research Integrity. Our work highlights the importance of responsible, transparent, and interpretable practices. The dataset and detailed documentation of our work, including code and prompts, are publicly available to support reproducibility. The use of AI assistants in our work is detailed in Appendix A.

Acknowledgments

This research is supported by the Ministry of Education, Singapore, under the Academic Research Fund Tier 1 (FY2023).

References

- Gediminas Adomavicius and Alexander Tuzhilin. 2011. *Context-aware recommender systems*. In *Recommender Systems Handbook*, pages 217–253. Springer US.
- Anthropic. 2025. Introducing Claude 4. <https://www.anthropic.com/news/claude-4>. Accessed: 2025-05-23.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. *Mem0: Building production-ready AI agents with scalable long-term memory*. *arXiv preprint arXiv:2504.19413*.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. *Chatbot Arena: an open platform for evaluating LLMs by human preference*. In *Proceedings of the 41st International Conference on Machine Learning*.
- DeepSeek. 2026. DeepSeek-v4 preview: Entering the era of affordable million-token context. <https://www.deepseek.com/en/news/v4-preview/>. Accessed: 2026-04-24.
- Ignacio Fernández-Tobías, Iván Cantador, Marius Kaminskis, and Francesco Ricci. 2012. Cross-domain recommender systems: A survey of the state of the art. In *Spanish conference on information retrieval*, volume 24.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. *SimCSE: Simple contrastive learning of sentence embeddings*. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. *Retrieval-augmented generation for large language models: A survey*. *arXiv preprint arXiv:2312.10997*.
- Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. 2024. *Scaling synthetic data creation with 1,000,000,000 personas*. *arXiv preprint arXiv:2406.20094*.
- Google DeepMind. 2025. Gemini 2.5 flash-lite. <https://ai.google.dev/gemini-api/docs/models/gemini-2.5-flash-lite>. Accessed: 2025-07-22.
- Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, and 28 others. 2025. *Memory in the age of AI agents*. *arXiv preprint arXiv:2512.13564*.
- Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. *Personalized soups: Personalized large language model alignment via post-hoc parameter merging*. *arXiv preprint arXiv:2310.11564*.
- Bowen Jiang, Zhuoqun Hao, Young Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo Jose Taylor, and Dan Roth. 2025. *Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale*. In *Second Conference on Language Modeling*.
- Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, Chien Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. 2024. *LongLaMP: A benchmark for personalized long-form text generation*. *arXiv preprint arXiv:2407.11016*.
- Sheng Li and Handong Zhao. 2020. *A survey on representation learning for user modeling*. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4997–5003.
- Xinyu Li, Ruiyang Zhou, Zachary C Lipton, and Liu Leqi. 2024. *Personalized language modeling from personalized human feedback*. *arXiv preprint arXiv:2402.05133*.
- Zhiyu Li, Chenyang Xi, Chunyu Li, Ding Chen, Boyu Chen, Shichao Song, Simin Niu, Hanyu Wang, Jiawei Yang, Chen Tang, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, and 20 others. 2025. *MemOS: A memory OS for AI system*. *arXiv preprint arXiv:2507.03724*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. *Lost in the middle: How language models use long contexts*. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. 2024. *LLM-Rec: Personalized recommendation via prompting large language models*. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 583–612, Mexico City, Mexico. Association for Computational Linguistics.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. *Evaluating very long-term conversational memory of LLM agents*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.

- Meta. 2024. Llama 3.3. https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/. Accessed: 2024-12.
- Elliot Nelson, Georgios Kollias, Payel Das, Subhajit Chaudhury, and Soham Dan. 2024. *Needle in the haystack for memory based large language models*. *arXiv preprint arXiv:2407.01437*.
- Chang Nie, Chaoyou Fu, Yifan Zhang, Haihua Yang, and Caifeng Shan. 2026. *PersonaVLM: Long-term personalized multimodal LLMs*. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15000–15009.
- OpenAI. 2023. *GPT-4 technical report*. *arXiv preprint arXiv:2303.08774*.
- OpenAI. 2025a. Introducing GPT-4.1 in the API. <https://openai.com/index/gpt-4-1/>. Accessed: 2025-04-14.
- OpenAI. 2025b. Introducing GPT-5. <https://openai.com/index/introducing-gpt-5/>. Accessed: 2025-08-07.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. *Training language models to follow instructions with human feedback*. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.
- Michael J Pazzani. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13(5):393–408.
- Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, Caiming Xiong, and Huan Wang. 2025. *Userbench: An interactive gym environment for user-centric agents*. *arXiv preprint arXiv:2507.22034*.
- QwenTeam. 2025. Qwen3: Think deeper, act faster. <https://qwen.ai/blog?id=1e3fa5c2d4662af2855586055ad037ed9e555125>. Accessed: 2025-04-29.
- Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. *A systematic survey of prompt engineering in large language models: Techniques and applications*. *arXiv preprint arXiv:2402.07927*.
- 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, Bangkok, Thailand. Association for Computational Linguistics.
- Chenkai Sun, Ke Yang, Revanth Gangi Reddy, Yi Fung, Hou Pong Chan, Kevin Small, ChengXiang Zhai, and Heng Ji. 2025. *Persona-DB: Efficient large language model personalization for response prediction with collaborative data refinement*. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 281–296, Abu Dhabi, UAE. Association for Computational Linguistics.
- Juntao Tan, Liangwei Yang, Zuxin Liu, Zhiwei Liu, Rithesh R N, Tulika Manoj Awalganekar, Jianguo Zhang, Weiran Yao, Ming Zhu, Shirley Kokane, Silvio Savarese, Huan Wang, Caiming Xiong, and Shelby Heinecke. 2025. *PersonaBench: Evaluating AI models on understanding personal information through accessing (synthetic) private user data*. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 878–893, Vienna, Austria. Association for Computational Linguistics.
- Zhaoxuan Tan and Meng Jiang. 2023. *User modeling in the era of large language models: Current research and future directions*. *arXiv preprint arXiv:2312.11518*.
- Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J Ross Mitchell. 2026. *Beyond a million tokens: Benchmarking and enhancing long-term memory in LLMs*. In *The Fourteenth International Conference on Learning Representations*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. *LLaMA: Open and efficient foundation language models*. *arXiv preprint arXiv:2302.13971*.
- Noah Wang, Z.Y. Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, Man Zhang, Zhaoxiang Zhang, Wanli Ouyang, Ke Xu, Wenhao Huang, Jie Fu, and Junran Peng. 2024. *RoleLLM: Benchmarking, eliciting, and enhancing role-playing abilities of large language models*. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14743–14777, Bangkok, Thailand. Association for Computational Linguistics.
- Peng-Yuan Wang, Tian-Shuo Liu, Chenyang Wang, Ziniu Li, Yidi Wang, Shu Yan, Chengxing Jia, Xu-Hui Liu, Xinwei Chen, Jiacheng Xu, and Yang Yu. 2026. *A survey on large language models for mathematical reasoning*. *ACM Computing Surveys*, 58(8).
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. *LongMemEval: Benchmarking chat assistants on long-term interactive memory*. In *The Thirteenth International Conference on Learning Representations*.
- Howard Yen, Tianyu Gao, Minmin Hou, Ke Ding, Daniel Fleischer, Peter Izsak, Moshe Wasserblat,

and Danqi Chen. 2025. [HELMET: How to evaluate long-context models effectively and thoroughly](#). In *The Thirteenth International Conference on Learning Representations*.

Xinliang Frederick Zhang, Nicholas Beauchamp, and Lu Wang. 2025a. [PRIME: Large language model personalization with cognitive dual-memory and personalized thought process](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33707–33736, Suzhou, China. Association for Computational Linguistics.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025b. [A survey on the memory mechanism of large language model-based agents](#). *ACM Trans. Inf. Syst.*, 43(6).

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, and 2 others. 2024. [Personalization of large language models: A survey](#). *arXiv preprint arXiv:2411.00027*.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. 2025a. [Do LLMs recognize your preferences? evaluating personalized preference following in LLMs](#). In *International Conference on Learning Representations*, volume 2025, pages 15888–15931.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. [WildChat: 1m ChatGPT interaction logs in the wild](#). *arXiv preprint arXiv:2405.01470*.

Zheng Zhao, Clara Vania, Subhradeep Kayal, Naila Khan, Shay B. Cohen, and Emine Yilmaz. 2025b. [PersonaLens: A benchmark for personalization evaluation in conversational AI assistants](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18023–18055, Vienna, Austria. Association for Computational Linguistics.

A Use of AI Assistants

In this work, we used AI assistants for a limited scope. Regarding coding, we used AI to assist with short-form input, such as suggesting package usage and reformatting code for readability. For writing, we used AI assistants solely for the paper’s language and formatting, such as grammar checking, paraphrasing, and polishing, to ensure a clear, readable delivery. All the above uses are strictly confined to the drafted content by the human authors. All AI-generated suggestions are rigorously reviewed, and all editorial decisions are reserved for human authors. AI assistants are not used for literature searches or generating text with new ideas.

All the intellectual contributions presented are the work of the authors.

B Future Directions

Based on the current benchmark design, there are some limitations that can be considered as directions for future research.

- **Complex Preference and Indication:** In the real world, user preferences can change over time or depend on the condition of something else. Besides, there are other forms of preference indication, such as thumbs-up or thumbs-down for LLM responses. Enriching the dataset with such complex preferences and indications would be a possible direction.
- **Richer Interaction Context:** Besides simplified dialogue types, real-world interactions may involve richer, multimodal contexts. Exploring how to enrich interaction contexts is also a promising future direction.
- **Evaluation Extension:** The evaluation rubrics for open-ended responses could be expanded to include additional dimensions, such as fairness, privacy, and transparency, better aligning the benchmark with responsible AI principles.
- **Human Subject Study:** To address the limitations of synthetic data, a longitudinal human-subject study on a real LLM platform can be considered to provide stronger evidence.

C Detailed Related Work

C.1 Personalization Benchmarks

The user-adaptation personalization task of an LLM is to capture user preferences from historical interactions and guide subsequent response generation, ensuring the assistant’s behavior aligns with user expectations without repetitive, explicit instructions.

PREFEVAL (Zhao et al., 2025a) is the first benchmark to evaluate LLMs’ preference-following abilities and incorporate basic implicit preference design. However, its dialogue dataset is generated centered on a constrained set of standalone preferences, rather than on user profiles to simulate complex real-world interactions. PersonaBench (Tan et al., 2025) evaluates models on synthetic user profiles and private documents, but it remains a “Needle-in-a-Haystack” task focused on retrieving isolated preferences from unrelated context. Re-

cently, interactive environments have been introduced (Zhao et al., 2025b; Qian et al., 2025) to simulate real-world agent dynamics and task-oriented personalization. PERSONAMEM (Jiang et al., 2025) proposes a new data-generation pipeline that constructs multi-session interaction histories step by step based on dynamic user profiles, covering a variety of query tasks to test whether LLMs can retain user traits, track dynamic user preferences, and generate personalized responses in new scenarios. However, it ignores implicit preferences, and its evaluation setting does not support free-form, open-ended answers. PERSONAMEM’s constrained-generation-task evaluation requires access to the output logits of the pre-designed answers, which are not available for proprietary LLMs, and using logits is not reliable enough for evaluating free-form, open-ended answers.

Our work draws inspiration from PERSONAMEM’s generation pipeline for multi-session dialogue generation and accounts for the different implicit nature of human preference expression, introducing a more comprehensive and granular evaluation method for genuine free-form generation tasks that is close to natural user-LLM interaction.

C.2 Personalization Enhancing Techniques

In general, techniques for enhancing personalization of LLMs include the following (Zhang et al., 2024): Prompt Engineering, Retrieval-Augmented Generation (RAG), Representation Learning, and Reinforcement Learning from Human Feedback (RLHF). More recently, various Memory-Based Personalization methods have emerged.

Prompt Engineering (Sahoo et al., 2024) is the process of designing, refining, and optimizing prompts to achieve desired outputs from language models (Zhang et al., 2024). By incorporating user information into the prompts or providing few-shot, high-quality examples for specific tasks, LLMs can leverage contextual information and perform personalization tasks without further training.

Retrieval-Augmented Generation (Gao et al., 2023) enhances LLM performance by retrieving relevant document segments from an external knowledge base using semantic similarity calculations (Zhang et al., 2024). Retrieving relevant personalization information needed for the task enables LLMs to generate customized outputs.

Representation learning for user modeling (Li and Zhao, 2020; Tan and Jiang, 2023) aims to enhance performance by learning latent represen-

tations that accurately capture each user’s behavior. This method requires a training stage, such as fine-tuning and embedding learning, which allows LLMs to adapt to specific data formats and generate responses in a particular style as instructed (Zhang et al., 2024). This approach typically yields better results on most tasks when model parameters are available and the associated costs are acceptable.

Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) aims to learn an output policy based on preferential human feedback. Besides general preference human feedback, there are works on Reinforcement Learning from Personalized Human Feedback (RLPHF), such as MORL tasks (Jang et al., 2023) and the P-RLHF framework (Li et al., 2024). By using user information as the reward signal, these methods help align LLMs with personalized preferences through reinforcement learning (Zhang et al., 2024).

Memory-Based Personalization has emerged to enable an agent to exhibit a consistent personality and long-term adaptation. Recent work on persistent memory systems (Zhang et al., 2025b; Hu et al., 2025; Li et al., 2025) has introduced mechanisms for memory consolidation, reorganization, reinforcement, and decay. Mem0 (Chhikara et al., 2025) proposes a scalable, persistent, and structured memory design that efficiently extracts, consolidates, and retrieves important conversational data from ongoing interactions. For personalization tasks, Persona-DB (Sun et al., 2025) proposed an efficient database framework to address context efficiency in retrieval-augmented personalization. PRIME (Zhang et al., 2025a) integrates the cognitive dual-memory model into LLM personalization and augments it with a personalized thinking capability, mirroring episodic memory of historical user engagements and semantic memory of long-term, evolving user beliefs. PersonaVLM (Nie et al., 2026) proposed hierarchical memory modules that target personalization tasks in multimodal contexts.

D Detailed Benchmark Design Comparison

D.1 Implicit Preference Comparison

To facilitate a better understanding of the expression types we provide in our work, we further clarify the fundamental differences between our preference expression settings and those in PREFEVAL (Zhao et al., 2025a), specifically regarding what constitutes an “implicit preference”.

Implicit Preferences in PREFEVAL PREFEVAL features two types of implicit preference settings.

- **Choice-based:** The LLM provides four options, and the user selects one and states preferences. This is essentially a straightforward expression of a preferred option and its reasoning. (Example: Assistant: "How about: A,... B,... C,... D,...?" User: "The documentaries don't really interest me. How about a fun comedy or an action-packed thriller?")
- **Persona-driven:** A user with a brief persona explicitly reveals their preference during a discussion. (Example: "gluten-free snack due to my severe intolerance.")

Preference Expression in REALPREF Our benchmark constructs a much more complex and realistic spectrum of preference expressions based on comprehensive user profiles.

- **Type 2 (Explicit Contextual Mention):** A user discusses a topic in a single session and explicitly mentions their preference. (Note: This is the type that is relatively similar to PREFEVAL's persona-driven setting, though with more complete profiles.)
- **Type 3 (Implicit Stylistic Expression):** The user does not state preferences directly using explicit words like "like, prefer, dislike,..." Instead, they imply preferences through stylistic devices such as rhetorical questions, metaphors, contrasts, or emotional expressions. (Example: "The idea of dance classes is good! There's just something about moving to music. It is hard to understand how people drag themselves onto a treadmill every single day.")
- **Type 4 (Implicit Experience Feedback):** The user develops a preference through life experiences, and this dynamic change is demonstrated across multiple sessions. (Example: Session 1: "Trying to get into a regular fitness routine... Willing to try gym workouts." -> Session 2: "I tried, but I really can't stick to the gym. I need your advice." -> Session 3: "I joined a group dance class and it feels great. This is something I can stick with.")

In conclusion, our "implicit" expressions are fundamentally distinct from those of PREFEVAL. We

specifically emphasize Stylistic Expression (complex expression introduces retrieval challenges) and Experience Feedback (captures the dynamics of preference development across multi-session interactions).

D.2 Evaluation Design Comparison

Our work follows the practice in PREFEVAL (Zhao et al., 2025a) and further expands it to a granular scoring criterion that leverages LLM-as-a-judge to support evaluation of genuine open-ended generation tasks. While PERSONAMEM (Jiang et al., 2025) includes a generation task, it is actually a constrained pseudo-generation task and, functionally, a multiple-choice test: it provides the model with 4 pre-written candidate responses and checks which one the model assigns the highest probability (logits) to. It does not evaluate the model's ability to actually compose a personalized response from scratch, and its logit-matching is incompatible with proprietary LLMs (e.g., GPT-4, Claude) whose logits are inaccessible. In contrast, our work evaluates genuine free-form dialogue generation: the model is given a query and must independently retrieve the preference, plan the content, and synthesize a natural, unstructured response. We adopted the LLM-as-a-judge framework as the necessary design to enable this evaluation.

E Dataset Details and Examples

E.1 Profiles

Profile Example and Prompt. The following is an example of a user profile. It includes a user ID, demographics information, an original persona sampled from PersonaHub (Ge et al., 2024) (its dataset card is provided below, and our usage aligns with its intention for persona-driven data synthesis), and an extended persona description. Our curated user profiles and follow-up conversations are mainly in English.

User Profile Example

```
{
  "00001": {
    "Demographics": {
      "Name": "Alex",
      "Birth": "1999",
      "Gender": "Male",
      "Education": "Bachelor",
      "Occupation": "Music Blogger",
      "Economic Status": "Middle Class",
      "Marital Status": "Single"
    }
  }
}
```

```

    },
    "Original Persona": "a die-hard
      fan of Tink who runs a music
      review blog",
    "Extended Persona": "Alex is a
      passionate music enthusiast
      born in 1999. He's a dedicated
      fan of Tink and channels his
      love for her music into an
      engaging review blog. Alex,
      who identifies as male and is
      of Asian descent, brings a
      unique perspective to the
      music scene with his
      thoughtful analyses and
      compelling writing style. His
      blog has gained a loyal
      following for its authentic
      voice and deep dives into the
      nuances of music and lyrics.
      Outside of blogging, Alex
      enjoys attending live concerts
      and discovering up-and-coming
      artists in his spare time."
  }
}

```

PersonaHub Dataset Card

license: cc-by-nc-sa-4.0

task_categories:

- text-generation
- text-classification
- token-classification
- fill-mask
- table-question-answering
- text2text-generation

language:

- en
- zh

tags:

- synthetic
- text
- math
- reasoning
- instruction
- tool
- persona

size_categories:

- 100M<n<1B

Prompt for Expanding Persona

According to the short persona, add the following demographic information if any of them are missing: Name, Birth Year, Gender Identity, Education Level, Occupation, Economic status, Marital Status.

Keep the same style and pronouns, but expand the persona with additional information to around 5 sentences. You can enrich the personality, hobbies and behavioral habits if necessary to make it look real and reasonable.

Here is the persona: {persona}.

Return in pure JSON text format as template example: {profile_example}.

Life Event Example and Prompt. Here is an example of a user's life event, with a specific date, event description, and duration.

User Life Event Example

```

[
  {
    "Date": "2011/06/15",
    "Event": "Graduated with a diploma
      in Makeup Artistry and began
      working as a freelance makeup
      artist, taking on clients for
      weddings, photoshoots, and
      local events."
  },
  {
    "Date": "2014/03/10",
    "Event": "Collaborated with
      several local salons,
      expanding her network and
      gaining regular clientele,
      while building her
      professional reputation within
      the community."
  },
  {
    "Date": "2016/11/05",
    "Event": "Launched her social
      media channel dedicated to
      honest and detailed makeup
      product reviews, steadily
      attracting a loyal following
      in the beauty community."
  }
]

```

Prompt for Life Event Generation

The current version of the profile gives you the basic demographic and core persona description.

Here is the persona profile: {profile}.

Based on the profile, you are to expand it by generating the person's general life event history, which is composed of a list of around {n_events} events. The events are organized in json format,

each event has the following keys:

- "Date": the date the event happened. (Format: year/month/day xxxx/xx/xx)
- "Event": Event Description should be detailed and consistent with the persona. Around 1-3 sentences.
- "Duration": The event duration type. Choose from years OR months OR weeks OR days.

Life events refer to events that are likely to occur in this person's life. The events with duration measured by years are long-term events represent the person's milestones in his/her life. (eg. education, marriage, vocation, or other important life topics). The events with shorter duration are short-term events that are more relevant to daily life (eg. study, work, sports or other daily topics).

Return in pure JSON text format as template example: {event_templates}.

E.2 Preferences

Generation Requirements. To ensure high-quality preference data in generation, we highlight requirements to ensure the generated preferences are *Diverse*, *Persona-Related*, *Unique*, *Value-Free*, and *Complete*, with specific descriptions listed below:

1. **Diverse:** The preferences covering a variety of different topics and types.
2. **Persona Related:** Each preference should be consistent with the persona's profile, sound realistic, and reflect individuality.
3. **Unique:** Avoid generic, cliché preferences such as "dislikes fast food." Instead, provide more nuanced, distinctive preferences that are non-popular, non-obvious, and cannot be deduced by common sense.
4. **Value-free:** Do not always align with socially desirable, universal "good", "healthy", logically "correct" preferences. Include preferences that may be unconventional or mildly negative (e.g., enjoy fast food late at night, like noisy environments).
5. **Complete:** The preference description should include the content and the underlying reasons, specify the type and topic.

Topic Pool. To ensure topic diversity, we introduced a topic pool comprising 7 broad categories and 37 fine-grained subtopics. The topic pool is listed in Table 3.

Preference Example and Prompt. The following is an example of a user preference, with an initialize event indicating how and when the user developed it. The fields include the preference and the topic, a specific date, and an event description.

User Preference Example

```
{
  "Preference": "Prefers herbal teas
    and fresh fruit snacks over
    sugary sodas and packaged sweets
    , as she values a balanced
    lifestyle.",
  "Topic": "Food: Snacks \& Drinks",
  "Date": "2014/06/21",
  "Event": "During a beauty workshop,
    Emma noticed that consuming
    sugary sodas and snacks left her
    feeling sluggish, while herbal
    teas and fruit refreshed and
    sustained her energy. She made a
    conscious decision to adopt
    fresher, healthier snacks both
    at work and at home."
}
```

Prompt for User Preference Generation

The current version of profile gives you the basic demographic and core persona description, with several personal events.

Here is the persona profile: {profile}.

Based on the profile, you are to expand it by generating a list of personal preferences.

The preference should follow these types: {PREFERENCE_TYPES}.

Here are candidate preference topic pool: {TOPIC_POOL}.

The designed preference should comply with the following requirements: {requirements}.

Step 1: Based on the profile, design a preference for each of the given topics here {given_topics}.

Step 2: Besides the topics from Step 1, design another {n_add} preferences with topics choosing from the rest topic pool that well align with the user profile.

You will generate {n_total} preferences in total for the user.

Topic Fields	Subtopics
Education	Learning Style, Learning Resources
Entertainment	Music, Movies & Shows, Games, Reading, Online Media
Lifestyle	Daily Routine, Exercise, Shopping, Beauty, Fashion, Tech Use, Home Decoration
Food	Cuisine, Cooking, Snacks & Drinks, Diet, Dining Style
Travel	Destinations, Activities, Companions, Pace & Frequency, Accommodation, Transportation, Planning Style
Work & Living	Location & Environment, Traffic & Commuting, Climate, Work Conditions & Hours, Work Values, Work Style & Tasks
Social	Social Energy, Friendship, Family, Romance, Social Events

Table 3: Preference Topic Pool

Return in pure JSON text format as template example: {pref_template}.

Prompt for User Preference Initialize

The current version of profile gives you the basic demographic and core persona description, with several personal events and preference.

Persona profile:{profile}.

Personal Preference:{preference}.

The Preference Type Explanation: {PREFERENCE_TYPES}.

Based on the profile, you are to expand it by generating a list of preference-related personal events.

1. Design an appropriate event for each preference, indicating that the person begins to like or dislike something after the event.

2. The event date design should be diverse and have no conflict with the personal life event.

3. The events are organized in json format, each event has following keys:

- "Preference": Original given preference.
- "Type": Original preference type.
- "Topic": Original preference topic.
- "Date": the date the event happened. (Format: year/month/day xxxx/xx/xx)
- "Event": Event Description should be detailed and consistent with the persona and the related preferences. Around 1-3 sentences.

Return in pure JSON text format as template example:{event_templates}.

Generalized Preference Example and Prompt.

Generalized preferences are generated based on the user's original preferences. We utilize Few-shot Chain-of-Thought during this process to ensure that the generalized preference can be logically derived from the original ones. Below is a detailed example, including a generalized preference, referenced original preferences, and the LLM's generation rationale.

Generalized Preference Example

```
{
  "Original Preference": [
    {
      "Topic": "Food: Cuisine",
      "Preference": "Only interested in trying global street foods or fusion cuisines, generally refusing classic fine-dining establishments which she finds unadventurous and restrictive."
    },
    {
      "Topic": "Travel: Accommodation",
      "Preference": "Prefers staying in quirky boutique hotels or artist-run guesthouses rather than large hotel chains, as the unique atmosphere inspires her work."
    },
    {
      "Topic": "Lifestyle: Beauty",
      "Preference": "Enjoys experimenting with unconventional makeup looks, such as vintage glam or fantasy-inspired themes, especially when attending local theatre events."
    }
  ]
}
```

```

    }
  ]
}
{
  "Generalized Preference": {
    "Infer_reason": "Her dislike of
      monotonous routines,
      regimented systems, and
      classic or mainstream
      offerings across various
      domains (food, beauty,
      exercise, music) suggests a
      desire for originality and
      novelty.",
    "Topic": "Entertainment: Movies &
      Shows",
    "Preference": "Dislikes formulaic
      blockbuster movies and prefers
      independent films or
      experimental cinema, as
      unconventional storytelling
      stimulates her creative
      thinking."
  }
}

```

Generalized Preference Criteria

To ensure reasonable preference generalization, we have the following requirements:

1. **Underlying Trait:** The generalized preference must be directly derived from the same underlying reason as the original preference, without introducing new, unverified assumptions.
2. **Domain:** The inference must occur within logically adjacent domains (e.g., the same work domain: from "work emails" to "project reports") rather than leaping across unrelated domains (e.g., assuming someone who likes spicy food has an aggressive personality). Generalizations based on stereotypes or unwarranted leaps are not appropriate.

Example:

Known Preference: "The user strongly prefers bullet points and concise structures in work emails." -> Reasonable Generalization: "Formatting a project meeting summary using a structured outline." -> Over-generalization: "Assuming the user is impatient and using blunt responses during casual chat."

Prompt for Generalized Preference

Given a list of user preferences, you are to infer several generalized user preferences based on the user's given preferences. The new preference is different from any single given

preference and has a different topic or subject. The new preference can be rationally deduced from more than one given preferences, for the underlying reason behind the new preference being closely related with underlying reason of the given preferences.

For example, a user's past preference list is:

1. prefer home-cooked or artisanal meals over fast food for health benefits and quality;
2. Enjoys hiking for exercise, valuing the time spent in nature over indoor gym workouts;
3. Only uses cruelty-free beauty products, rejecting any brands that test on animals;

Then it is rational to infer that the user prefers natural, eco-friendly cleaning products at home to align with her holistic and nature-oriented lifestyle.

To ensure reasonable preference generalization, we have the following requirements: {generalization_requirements}.

Here is the candidate preference topic pool: {TOPIC_POOL}.

Generate {n} new personal preferences based on the user's given preference list: {preferences}.

The preference contents should follow: prefer something over/dislike/aversion to another, and the underlying reasons.

Add your inference reason for the new preference, like how to deduce this new preference from the given preferences.

Return in pure JSON text format as template example: {pref_template}.

E.3 Conversations

E.3.1 Life-Event Conversation

Life-Event Conversations indicate user profiles and life events. Through dialogue, users share background, circumstances, thoughts, and feelings. Below is an example.

Life-Event Conversation Example

User: Hello! I've recently started working as a freelance makeup artist, and I have a wedding client next month who wants a classic vintage look. I'm a bit stuck on product choices for oily skin types. Could you recommend some foundation and powder options that hold up all day?

Assistant: Hi there! Congrats on getting a wedding client—vintage looks are always stunning. For oily skin, you'll want a mattifying, long-wear foundation and a setting powder with good oil control. Some popular options include Esté Lauder Double Wear Foundation, MAC Studio Fix Fluid, or Revlon Colorstay. As for powders, Laura Mercier Translucent Setting Powder or Rimmel Stay Matte are widely loved. Be sure to prep the skin with an oil-free primer and consider using blotting sheets for touch-ups. Would you like advice on specific application techniques for a vintage look as well?

User: Thank you, that really helps! Yes, any tips on keeping the winged liner and red lipstick looking sharp all night? Vintage looks seem easy but can be tricky under pressure. I want to make sure everything stays put for photos and dancing.

Assistant: Absolutely, vintage-inspired makeup does require a bit of extra care! For winged liner, using a waterproof gel or liquid liner will help prevent smudging—Inglot AMC Gel Liner or Stila Stay All Day Liquid Liner are good options. Set the liner with a black eyeshadow using a fine brush for added longevity. For lips, start with a lip primer, line the lips with a long-wear red liner, fill them in, then layer your lipstick (like MAC Ruby Woo or Maybelline Superstay Matte Ink), and blot between layers. Setting with a tiny bit of translucent powder through a tissue can help too. Have you been enjoying freelance work so far, or is it still new to you?

User: I actually just graduated last summer from a makeup artistry program, so this is my first full wedding season. Freelance is exciting but also a bit nerve-racking! Thanks for all the product tips—I really want to make a great impression and hopefully get more referrals.

Assistant: That's wonderful—congratulations on launching your career! It's normal to feel a bit anxious, especially early on, but you're doing all the right things by preparing ahead. Each happy client is a stepping stone to more opportunities. If you'd ever like to chat about building your portfolio or networking with local salons, feel free to ask. Good luck with the wedding—you've got this!

Prompt for Life-Event Conversation

You are to generate a conversation session between a user and an LLM assistant based on the user profile and history.

The profile gives you the basic demographic and core persona description: {demo}.

The event list gives you the user's personal history event: {events[:i]}.

In the conversation, the user is interacting with LLM assistant between {time_range}.

In the time range, the following life event have already happened: {prev_event}. And the following life event is upcoming: {nxt_event}.

Generate {n_sessions} multi-turn conversations (2-4 rounds each) where the user is at the life stage described above.

1. Each session should begin with the User initiating the conversation, guided by one of the following dialogue intents:

- Task-based request: asking for specific help, knowledge, resources, advice, or content creation.
- Life choice discussion: exploring academic, career, or future path options.
- Emotional companionship: expressing feelings such as confusion, anxiety, frustration, excitement, or achievement.

2. Throughout the dialogue, let the User gradually and naturally reveal aspects of their background, life circumstances, thoughts, and emotions.

3. The Assistant should start with no knowledge of the user's persona or situation. All personal details must be disclosed progressively by the User during the conversation.

4. The Assistant's responses should be insightful, detailed, and constructive—providing useful suggestions, gentle reasoning, or professional guidance. Whenever possible, offer multiple perspectives, options, or explanations. Keep the tone friendly, conversational, and engaging, rather than overly formal.

5. Ensure the dialogues feel realistic and natural:

- Vary word choice, sentence structures, and tones across sessions.
- Keep content coherent and contextually appropriate.
- Make each conversation sufficiently complete and meaningful.

Return in pure JSON text format as template example: {conversation_template}.

E.3.2 Random Conversation

Random Conversations are about common-sense questions and will not reveal user-specific information. Below is an example with the assistant response omitted for simplicity of demonstration.

Random Conversation Example

User: I'm having trouble connecting to Wi-Fi on Windows 10. Any troubleshooting tips?
Assistant: [Response]
User: What if none of these suggestions work?
Assistant: [Response]

Prompt for Random Conversation

Generate n sessions of multi-turn conversations (2-5 rounds each) between a user and a warm, articulate, and helpful LLM assistant.

1. Each session should avoid topics involving personal background details, life achievements, or user personal preferences.
2. The user initiates the session with a realistic, detailed question or request reflect real-world use cases such as:
 - asking factual questions or explanations (e.g., science, technology, society)
 - troubleshooting or tool usage (e.g., Python, Excel, Git, API, etc.)
 - general curiosity or reasoning questions
 - simple small talk or clarification requests
 - text generation tasks
3. Vary the topics and tones to make the dataset diverse. You can choose topics randomly from domains such as technology, science, education, programming, history, art, and productivity tools.
4. The assistant should provide thoughtful, detailed, and constructive replies, offering multiple ideas, explanations, or suggestions where appropriate. The tone should remain friendly and engaging, never overly formal.
5. Ensure the conversation session feels contextually complete.

Return in pure JSON text format as template example: {conversation_template}

E.3.3 Preference-Expression Conversation

Preference-Expression Conversations indicate user preference. We provide four *Preference Expression Types* from explicit to implicit, each with an example listed below.

1. Explicit - Direct Statement Preferences are stated directly and explicitly in a single turn without redundant words.

Direct Statement Example

User: Hi! I want to get more physically active, but **I really avoid routine gym workouts because they're monotonous and uninspiring for me. I much prefer spontaneous activities like dancing at local community events.** Any ideas for keeping fit that aren't boring gym routines?

Assistant: Absolutely, I understand that traditional gym workouts aren't for everyone—it's important to have fun while staying active! Here are a few ideas that can keep things lively and spontaneous: 1. Dance Classes or Social Dancing: Try drop-in classes for salsa, swing, or even hip-hop at local studios or community centers. Social dance nights are great for meeting people and get your heart rate up without feeling like exercise. 2. Outdoor Group Activities: Join recreational sports leagues (like volleyball, ultimate frisbee, or even kickball) where the focus is on fun and socializing rather than intense competition. 3. Hiking Groups or Urban Walks: Explore nature trails, city walking tours, or even organized hiking meet-ups. Each outing is a little different and can double as a mini-adventure. 4. Fitness Meetups: Check out community boards or social media for pop-up yoga in the park, outdoor bootcamps, or even roller skating meetups. 5. Volunteering: Since you enjoy community events, active volunteering (like setting up events, ushering at theaters, or organizing local festivals) can keep you moving while contributing to a good cause. Mixing things up and following what excites you can make staying active something you actually look forward to. Would you like more suggestions tailored to your interests or info on how to find local events?

Prompt for Direct Statement

You are to generate 1 session of conversation between a user and an llm assistant based on the

user profile and user preference.

The profile gives you the basic demographic and core persona description. Here is the persona profile: {profile}.

The preference gives you the preference content that the user should indicate in this conversation. Here is the preference: {pref}.

The conversation occurs between time range {time_range}.

The event list gives you the user's personal life event history: {events}.

Instruction:

- In the conversation, the user stated the preference directly, CLEARLY and EXPLICITLY in a single turn without redundant words, and the assistant gives a response.
- The LLM assistant in the conversation DO NOT KNOW anything about the user, all such personal information should be revealed by user in the conversation, LLM assistant just work as a general AI assistant for question answering.
- The LLM assistant gives insightful, detailed, and constructive responses, offering useful suggestions, gentle reasoning, or professional guidance. Let the assistant provide more than one idea or explanation when possible. Keep the tone friendly and engaging, not overly formal.
- Remember to keep the dialogue realistic, with diversity in word, sentence structures and tone, and the conversation content reasonable and complete.

Return in pure JSON text format as template example: {conversation_template}.

2. Explicit - Contextualized Mention In multi-round conversations, users start the session with other related daily topics as context instead of directly stating preferences. As the dialogue advances, preferences are naturally mentioned explicitly in the context.

Contextualized Mention Example

User: Hi there! I was looking for some ideas to keep my weekends fun and active without spending too much money. Got any suggestions?

Assistant: Absolutely! There are plenty of ways to have an active and enjoyable weekend without breaking the bank. Here are a few ideas:

1. Go for a nature walk or hike in a local park.
2. Join a community sports group that has free or low-cost sessions.
3. Try a YouTube dance or workout class at home with friends.
4. Attend local events or street fairs, which often offer free activities and entertainment.
5. Organize a picnic with friends and some outdoor games like frisbee or badminton. Would you like more suggestions tailored to your interests or any particular type of activity?

User: Thanks! I like the idea of attending local events. Lately, I've noticed several community dance nights popping up nearby. **Honestly, I've never been into routine gym workouts—they just feel so monotonous and uninspiring to me. I much prefer spontaneous activities like dancing. Do you think joining community dance events counts as good physical activity?** Any tips for someone just jumping in?

Assistant: Dancing at community events absolutely counts as excellent physical activity! It's fantastic for your cardiovascular health, flexibility, balance, and mood. Plus, the energy and social aspect make it much more enjoyable than routine workouts for many people. Here are some tips for getting started: 1. Don't worry about technical skills—community events are usually very inclusive and welcoming to beginners. 2. Wear comfortable shoes and clothes so you can move freely. 3. Stay hydrated, as dancing can be quite a workout. 4. Bring a friend along if you feel shy at first, or just go with an open mind to meet new people. 5. Try out different dance styles to see what you enjoy the most—salsa, swing, folk, or even freestyle. And the best thing is, you're having fun, so the exercise doesn't even feel like a chore! Would you like some resources to look up local dance nights or maybe music playlists for practice at home?

User: That's so reassuring! I definitely vibe with the fun, social energy rather than repetitive gym routines. I'd love some tips for finding more local dance nights or any online groups where people organize spontaneous dance meetups. Also, if you have music playlist suggestions for home practice, I'm all ears!

Assistant: Great to hear you're excited! For finding local dance nights: 1. Check out community center bulletin boards or websites—they often list weekly or monthly events. 2. Browse

social media platforms (like Facebook Events or Meetup.com) using keywords like “community dance,” “social dance,” or specific styles (e.g., “salsa night,” “swing party”). 3. Some local coffee shops or cultural centers host themed dance nights—ask around next time you’re out! 4. Chat with local dance studios; even if you don’t want structured classes, they often know about informal gatherings. For online groups, Meetup.com and Facebook Groups are especially helpful. Try searching for your city + “dance” or “jam session.” Music playlists: There are great ones on Spotify and YouTube for various moods—check out playlists titled “Dance Party Classics,” “Salsa & Bachata Hits,” or “Feel-Good Indie Dance.” Let me know if you’d like me to curate a sample playlist or help you draft a message to join an online group!

Prompt for Contextualized Mention

You are to generate 1 session of conversation between a user and an llm assistant based on the user profile and user preference.
The profile gives you the basic demographic and core persona description. Here is the persona profile: {profile}.
The preference gives you the preference content that the user should indicate in this conversation. Here is the preference: {pref}.
The conversation occurs between time range {time_range}.
The event list gives you the user’s personal life event history: {events}.

Instruction:

1. The conversation has ≥ 3 rounds.
 2. In the conversation, the user will NOT directly talking about the preference, but start a conversation with some other related daily life topics or request as context.
 3. As the conversation progresses, the user later NATURALLY mention the preference CLEARLY and EXPLICITLY in contextual conversations.
- The LLM assistant in the conversation DO NOT KNOW anything about the user, all such personal information should be revealed by user in the conversation, LLM assistant just work as a general AI assistant for question answering.

- The LLM assistant gives insightful, detailed, and constructive responses, offering useful suggestions, gentle reasoning, or professional guidance. Let the assistant provide more than one idea or explanation when possible. Keep the tone friendly and engaging, not overly formal.
- Remember to keep the dialogue realistic, with diversity in word, sentence structures and tone, and the conversation content reasonable and complete.

Return in pure JSON text format as template example: {conversation_template}.

3. Implicit - Stylistic Expression The context resembles the second type, but users will not state preferences with explicit words like Like, Prefer, or Dislike. Instead, they imply preferences through stylistic devices such as rhetorical questions, metaphors, contrasts, or emotional expression.

Stylistic Expression Example

User: Hey there! I’ve been thinking about switching up my daily routine lately. Got any creative ways to recharge after a long day at work? My schedule’s been packed with makeup clients, and I could use something to shake off the stress.

Assistant: Absolutely! After a busy day, it’s great to have unwinding rituals that actually excite you. Some options could be a little home spa session—think an aromatic bath, face mask, and soothing playlist—or even picking up a fun hobby like painting or baking. If you prefer something active, you could try a walk in a nearby park, cycling, or maybe signing up for a local art or dance class. Do any of those sound appealing, or are there activities you used to enjoy that you haven’t revisited in a while?

User: The idea of dance classes is good! There’s just something about moving to music that makes me forget the clock even exists. Honestly, I’ve never understood how people drag themselves onto a treadmill every single day—just the thought of it puts me to sleep. I keep stumbling into little community salsa nights or street performances after work lately, and those improvised jams end up being the highlight of my week.

Assistant: Dance is such a fantastic way to rein-vigorate your body and mind, especially with the social and creative atmosphere of community events. Unlike gym routines, which can become a bit mechanical, dancing taps into expression and connection—plus, it’s a great workout! If you love these spontaneous dance nights, you might also enjoy joining a local dance group or checking for upcoming themed events in your neighborhood. You could even invite a few friends or fellow artists along for extra fun! And if you ever want tips for stretching or caring for your body post-dance, I’m here to help.

User: Exactly! I guess it makes sense—I’d rather end up in a train of laughter trying to follow someone else’s footwork than counting reps in a gym mirror. I get my fill of routine at work; when I’m off the clock, it’s about freedom and a fresh vibe. Sometimes I mix it up with some vintage shopping at the weekend markets too, especially if there’s live music. Life’s too short not to move with the rhythm, right?

Assistant: That’s a wonderful philosophy! Blending movement, music, and even a bit of vintage treasure hunting sounds both energizing and inspiring. These experiences not only help you unwind but probably fuel your creativity for your makeup artistry as well. If you ever want unique ideas to incorporate your passions—like organizing a mini makeover session at a dance event or styling vintage market-goers—let me know. Embracing spontaneity is truly an art in itself!

Prompt for Stylistic Expression

You are to generate 1 session of conversation between a user and an llm assistant based on the user profile and user preference.

The profile gives you the basic demographic and core persona description. Here is the persona profile: {profile}.

The preference gives you the preference content that the user should indicate in this conversation. Here is the preference: {pref}.

The conversation occurs between time range {time_range}.

The event list gives you the user’s personal life event history: {events}.

Instruction:

1. The conversation has ≥ 3 rounds.

2. In the conversation, the user will NOT directly talking about the preference, but start a conversation with some other related daily life topics or request.

3. As the conversation progresses, the user later NATURALLY mention the preference in contextual conversations.

4. The user will NOT directly express preference by explicit words such as LIKE, PREFER, or DISLIKE. Instead, user IMPLICITLY indicate the preference by rhetorical questions, metaphors, contrasts, or emotional expression. For example, to express ‘I prefer science fiction movies to art films’, one can say

— ‘I don’t get how people stay awake through movies like The Tree of Life.’

— ‘I tried watching The Tree of Life, but honestly, I’d much rather rewatch Dune.’

— ‘I’ve watched Blade Runner 2049 like five times. Never gets old.’

— ‘Dune just hits different. So immersive.’

- The LLM assistant in the conversation DO NOT KNOW anything about the user, all such personal information should be revealed by user in the conversation, LLM assistant just work as a general AI assistant for question answering.

- The LLM assistant gives insightful, detailed, and constructive responses, offering useful suggestions, gentle reasoning, or professional guidance. Let the assistant provide more than one idea or explanation when possible. Keep the tone friendly and engaging, not overly formal.

- Remember to keep the dialogue realistic, with diversity in word, sentence structures and tone, and the conversation content reasonable and complete.

Return in pure JSON text format as template example: {conversation_template}.

4. Implicit - Experience Feedback Multiple sessions, based on the user preference Initialize Event. The user won’t explicitly state their preference in one session but will reveal it gradually through experience feedback across sessions.

Experience Feedback Example

[Session 1]

User: Hey! I’m thinking of trying to get into

a regular fitness routine. I always see people raving about the benefits of gym workouts, but I've never really stuck with any for long. Do you have any tips for someone who gets bored easily? I want to keep myself motivated!

Assistant: Hi! That's a great initiative. Keeping fitness fun and engaging can be a real challenge, especially if you find traditional routines monotonous. Here are a few suggestions you might find helpful: 1. Mix up your activities: Try incorporating different kinds of exercise like swimming, hiking, cycling, or group classes. This helps keep things fresh and less predictable. 2. Try out classes: Many gyms and studios offer dance, kickboxing, or yoga classes—you can sample various ones to see what feels most enjoyable. 3. Set mini-goals: Instead of focusing on a single workout type, try setting goals related to new skills or experiences—like mastering a yoga pose or running a local fun run. 4. Get a workout buddy: Having a friend to join you can boost motivation and add a social element to your routine. Let me know if you want more specific ideas, or if any of these sound interesting to you!

User: Thanks! I like the idea of mixing things up. I haven't tried any group classes yet, but maybe I should. I do find myself zoning out doing the same treadmill or weights routine over and over, so something a bit more dynamic could definitely help.

Assistant: Absolutely! Group classes often have an energetic vibe that can make the whole experience more enjoyable and less repetitive. Dance-based fitness classes like Zumba or hip-hop, or even things like spinning with music, can keep you engaged. If you're not into classes, you could also look out for local community events like fun runs, charity walks, or weekend sports meetups. Sometimes being in a new environment with others doing something active can break the monotony, too. And if you ever need advice on balancing your fitness routine with a busy work schedule, just let me know!

[Session 2]

User: Hey again! I wanted to share—last weekend I did something totally out of my comfort zone and joined a flash mob dance group for a community event. I honestly signed up on a whim, but it turned out to be so much more fun

than any gym session I've had.

Assistant: That sounds amazing! Flash mobs can be such an exhilarating experience—not only do you get a fantastic workout, but you're part of something creative and vibrant with others. Dancing is a great way to get exercise without it feeling like a chore. What was it about the event that you enjoyed most—the music, the group energy, or just the spontaneity? If you're looking to keep the momentum going, you might want to look for similar events or dance meetups in your area. There are also lots of dance fitness classes, or you could organize your own mini dance sessions with friends! Keeping activities enjoyable and social is a great way to build exercise into your lifestyle without it ever feeling monotonous.

User: Honestly, it was all of it—the music, being with a group, and just letting go and moving. I felt so energized afterwards. It didn't feel like a workout at all! I wish more fitness experiences could be like that. I think I might look out for other dance or movement-based things to do in the city.

Assistant: I love hearing that! When exercise feels joyful and connected, it's so much easier to stay consistent—and you end up looking forward to it instead of just checking it off a list. Dance, in particular, combines movement, creativity, and social interaction, which can all boost your mood and motivation. Keep an eye out for local dance events, street festivals with open dance floors, or even themed fitness parties. And don't overlook opportunities like community theatre—volunteering to help with performance prep or being involved backstage can be surprisingly active, too! If you want suggestions for specific events to try or tips on discovering local options, I'd be glad to help.

Prompt for Experience Feedback

You are to generate {n_sessions} sessions of conversation between a user and an llm assistant based on the user profile and user preference. The profile gives you the basic demographic and core persona description.

Here is the persona profile: {profile}.

The preference gives you the preference content that the user should indicate in this conversation.

Here is the preference: {pref}.

This preference stems from the following past event: "{pref_event}" happened on {start_date}. The event list gives you the user's personal life event history: {events}.

Instruction:

1. Each session has 1-3 rounds of conversation.
 2. In the conversation, the user will NOT directly talking about the preference, but start a conversation with some other related daily life topics or request.
 3. The session content stems from the past event. You should further enrich the personal experience story by including details or extra information in preference event.
 4. The conversation timestamp is related to the date of past event ({start_date}), keep the timeline consistent.
 5. Do not directly state the preference or event experience in one session, and the user will NOT directly express preference by explicit words such as LIKE, PREFER, or DISLIKE. Instead, the full information should gradually emerge across 2 to 3 sessions of user-llm interactions, indicating by fragmented personal experiences, behaviour and casual remarks. For example,
 - The early session may happen before the event, when the user has not yet developed such a preference, and is willing to try.
 - The later sessions may happen after the event experience, where the user may share event experience, give feedbacks and indicate preference across session interactions.
 6. Key Reminder: The assistant cannot infer the user's preferences from any single session dialogue. It can only infer the user's preferences by integrating the dialogue information of all sessions.
 7. The assistant should always reply with helpful, warm, and constructive responses, but DO NOT infer the full preference too early.
- The LLM assistant in the conversation DO NOT KNOW anything about the user, all such personal information should be revealed by user in the conversation, LLM assistant just work as a general AI assistant for question answering.
 - The LLM assistant gives insightful, detailed, and constructive responses, offering useful suggestions, gentle reasoning, or professional guidance. Let the assistant provide more than

one idea or explanation when possible. Keep the tone friendly and engaging, not overly formal.

- Remember to keep the dialogue realistic, with diversity in word, sentence structures and tone, and the conversation content reasonable and complete.

Return in pure JSON text format as template example: {conversation_template}.

F Evaluation Details

F.1 Preference Query

Queries are designed based on the target preferences. High-quality queries should ensure that, without knowing the preferences, **answering the question naturally is likely to violate them** (Zhao et al., 2025a). Therefore, the query content should not reveal user preferences or any underlying user intention. Here are examples to facilitate understanding:

Preference Query Example Analysis

Example 1:

Preference: I prefer companies that allow unlimited sick days.

Bad Query: Which companies have the most generous sick leave policies?

Analysis: The query directly reveals what the preference is looking for.

Good Query: I'm moving to Seattle and seeking job opportunities. Can you recommend a suitable company?

Analysis: The query did not reveal any user information regarding company preferences.

Example 2:

Preference: I prefer staying in traditional accommodations over modern hotels to experience cultural authenticity.

Bad Query: I'm planning a vacation to fully immerse myself in the local culture. Can you recommend accommodation for me?

Analysis: A bad example. The query directly leaked the user's intention.

Good Query: I'm planning a vacation to Japan. Can you recommend accommodation for me?

Analysis: The query contains only the core question without leaking the user's preference and intention.

The following is a complete query example. The fields include the target preference, an open-ended question with an explanation, 4 MCQ options with the correct answer labeled, and 2 true-or-false questions derived from the MCQ options.

Preference Query Example 1

```
{
  "Preference": "Prefers staying in traditional accommodations, such as ryokans, over modern hotels to experience cultural authenticity and personalized service.",
  "Query": "I'm looking for a place to stay during my trip to Japan. What kind of accommodation would be best for me?",
  "Explanation": "The expected response should highlight traditional aspects associated with cultural authenticity, aligning with the user's desire for personalized service and cultural immersion.",
  "MCQ_options": {
    "A": "A sleek hotel in downtown Tokyo with smart tech and 24/7 room service.",
    "B": "A traditional ryokan in Kyoto that serves kaiseki meals and offers onsen baths.",
    "C": "A capsule hotel near Shinjuku station that's convenient and budget-friendly.",
    "D": "A modern Western-style hotel with a rooftop bar and fitness center."
  },
  "Correct_Answer": "B",
  "T_Query": "I'm looking for a place to stay during my trip to Japan. Do you think a traditional ryokan in Kyoto that serves kaiseki meals and offers onsen baths would be the one for me?",
  "F_Query": "I'm looking for a place to stay during my trip to Japan. Do you think a modern Western-style hotel with a rooftop bar and fitness center would be the one for me?"
}
```

Regarding the comparison of MCQ and True-or-False questions, here is an example query of a negative preference (dislike or aversion to something) to facilitate understanding. In the 4 MCQ options below, 3 incorrect options share the same characteristic — routine workouts —while the correct one obviously stands out and can be guessed

even if the model has no knowledge of use preference. In contrast, the true-or-false format avoids this possibility, and the model must decide based on the current single option.

Preference Query Example 2

```
{
  "Preference": "Avoids routine gym workouts because she finds them monotonous and uninspiring, preferring spontaneous physical activities like dancing at community events.",
  "Query": "I'm interested in finding ways to stay active in my free time, what would you suggest?",
  "Explanation": "Without knowing the preference, an assistant is likely to recommend standard gym workouts or classes, which can easily violate the user's aversion to monotonous exercise routines. The query is sufficiently broad, making it challenging to guess the user's specific dislike for routine workouts versus other forms of activity.",
  "MCQ_options": {
    "A": "Join local community dance events and try different styles whenever you feel like it.",
    "B": "Take group fitness classes at a nearby gym several nights a week.",
    "C": "Follow an at-home structured workout program with daily routines.",
    "D": "Plan a weightlifting schedule and stick to it for consistency."
  },
  "Correct_Answer": "A",
  "T_Query": "I'm interested in finding ways to stay active in my free time, what would you suggest? Do you think joining local community dance events and trying different styles whenever you feel like it would be the one for me?",
  "F_Query": "I'm interested in finding ways to stay active in my free time, what would you suggest? Do you think following an at-home structured workout program with daily routines would be the one for me?"
}
```

Prompt for Preference Query

Given a user preference, you are to generate an open-ended query, corresponding multiple-choice question (MCQ) options, and True or False query designed to test whether a language model assistant understands the user's preference.

Instruction:

1. Open-ended query

(1) The query should be a natural-sounding open-ended question, concise in 1 sentence, written from user's first-person perspective.

(2) The query is designed based on user preference, requiring that without knowing the preference, naturally answering the question can easily violate the user's preference. Here are bad and good examples for your information.

- Bad example: {bad_example}.

- Good example: {good_example}.

(3) The query should be concise. DO NOT include anything about the user preference or underlying user purpose or intention in the query. Here are bad and good query examples toward one preference for your information.

- Preference: {example_preference}.

- Bad example: {bad_example}.

- Good example: {good_example}.

(4) DO NOT generate contradictory or obvious question such that the question either directly contradicts the user's preference or is so aligned that it is not challenging.

2. Multiple-choice question

(1) For Open-ended query, design four possible options, each is concise in 1 sentence.

(2) All four options are perfectly relevant, reasonable and common sense answers to the query. None of them contain obvious common sense errors or logical errors.

(3) The four MCQ options has ONLY ONE answer that BEST aligns with the user's preference.

(4) Make sure that the correct answer can not be inferred purely based on common sense. Only the one aware of the user preference can figure out the correct option.

(5) DO NOT explain, repeat anything about the user preference or underlying user purpose in any options. Do not add extra explanation about the options in MCQ.

3. True or False

(1) The True or False query is derived from the open-ended query and the MCQ options: [open-ended query] + Do you think [option] would be the one for me?

(2) T_Query simply combines the query and the correct option from MCQ, and F_Query simply combines the query and one of the false option from MCQ.

Key Reminder: Add a explanation about why it is challenging in answering the query without violate the preference. The key is, without knowing the preference, the assistant is highly likely to violate it when answering the question, which means suggesting options that is contrary to the user's preference.

Here is the user preference: {pref}.

Return in pure JSON text format as template example: {qa_template}.

F.2 Evaluation Rubrics

The scoring criteria for open-ended answers in this work have 3 rating dimensions described below. In practice, the judge LLM will be provided with a preference, a query, and the rubrics for the specified dimension, and will be prompted to assign a score and provide a judge's rationale.

Preference Awareness: focuses on the extent to which the answers recognize or understand user preferences. See detailed rubrics in Table 4.

Preference Alignment: focuses on how well the answer content aligns with the user's preferences. See detailed rubrics in Table 5.

Answer Quality: focuses on the extent to which the answer content is actually helpful and constructive to the user. See detailed rubrics in Table 6.

Prompt for LLM-as-a-Judge Evaluation

You are a strict evaluator. Given a user preference, a query and a candidate assistant response, score the response on a 1-5 scale, where 1 = very poor and 5 = excellent. Here is the evaluation rubric:

{evaluation rubric}

User preference: {preference}.
User query: {query}.
Assistant response: {response}.

Your judge should be organized in JSON format with the following 2 keys:

- "Judge_Reason": A very short judge reason how you evaluate the response according to the rubrics.
- "Score": The score you give to the response.

Return in pure JSON text format as template example: {judge_template}.

G Benchmark Quality Verification

To ensure the quality of our benchmark work, we have conducted both automated and manual verification of each benchmark component, including the synthetic data, the queries, and the evaluation scoring criteria. Human annotators (graduate students who are contributors to this work) are included in this part. They understand and consent to the subsequent use of the collected data.

G.1 Synthetic Data

Real user data (preferences, profiles, conversations) is highly private and user-specific, and user behavior is complex, making it challenging to bridge the gap between synthetic categories and real users' long-term behavior. Our work aims to move a step closer to natural user-LLM dialogue by using diverse, well-constructed synthetic data. We have always recognized the importance of data quality and have made it central to our data generation process.

Profiles and Preferences Following our generation pipeline, we performed manual checks and filtering early in the process of creating user profiles and preferences.

1. **Persona:** We randomly sampled 150 candidate personas from PersonaHub (Ge et al., 2024), a massive, open-source dataset comprising 1 billion diverse personas, each represented as a brief sentence. We conducted the first manual check and filtering of these personas, deleting the rare or unnatural ones (e.g., those describing a couple rather than a person, or those related to inappropriate backgrounds

such as WWII or Paranormal Events), leaving 137 personas for the next step.

2. **Profile:** The generated persona set is then used to construct a detailed profile that includes basic demographics and an extended persona description. We perform a second manual check on the plausibility of generated profiles and retain 100 profiles with diverse, appropriate and plausible attributes.
3. **Life Events and Preferences:** Based on the resulting profiles, we generated 12 personal life events and 12 preferences for each user. These events and preferences are then subjected to manual and LLM-assisted plausibility checks, and 10 preferences and 8-10 events are retained per user.
4. **Generalized Preferences:** Generalized preferences are based on the original preferences. During generation, we manually checked whether each generalized preference can be rationally inferred from the original preferences. The method is similar to the "Easy Test" we provided in Appendix G.2: giving all original preferences and choosing which option may align with the user's preference without viewing the ground-truth generalized preference. The validated samples remain in the dataset.

Conversations After several rounds of prompt refinement iterations for small-scale conversation generation, we performed large-scale generation of conversation data. To assess the quality of our conversation data, we draw on insights from Chatbot Arena (Chiang et al., 2024) and conduct a blind pairwise human evaluation comparing preferences and conversations from our work with those from the recent PREFEVAL dataset (Zhao et al., 2025a). The evaluation is conducted by human evaluators who have not seen our dataset or its generation process beforehand, and no information about the data source was provided during the comparison evaluation. From each dataset, we sampled 100 preferences and their corresponding conversation sessions, paired them one by one, then shuffled the order. We randomly selected implicit preference expression conversations (Types 3 or 4) from our dataset and used the persona-driven conversation from PREFEVAL for this task. Evaluators are presented with a pair of data and told to select the one that appears more consistent and plausible. The

Score	Criteria
1	No mention or reflection of user preference at all.
2	Mentions user preference, but the stated preference content is incomplete or incorrect.
3	Mentions user preference, and correctly states the preference, but not well integrated in the response (the response is highly generic, which can be applied to any user).
4	Mentions user preference, correctly states the preference, and integrates the preference in the response.
5	Mentions user preference, correctly states the preference, and the response is fully aware, deeply focused, consistently centered on the preference.

Table 4: Evaluation Rubrics for Preference Awareness

Score	Criteria
1	Clear violation of user preference. Examples: - User prefers A over B, but the response only suggests B. - User only likes A, but the response only suggest something except A. - User does not accept B, but the response only suggest B.
2	Mostly inconsistent or ignores some taboos. Examples: - User only likes A, but the response suggest A and something except A. - User does not accept B, but the response also contains B.
3	Generally aligned but with inaccuracies or gaps. Examples: - User prefers A over B, but the response suggests A as well as B. - User likes A, but the response only suggest something other than A.
4	Aligned and mostly consistent. Examples: - User prefers A over B, and the response suggests A. - User only likes A, and the response suggests A. - User does not accept B, and the response does not contain B. - User likes A, and the response is mostly about A.
5	Fully aligned, no conflicts at all (requirement for score 4), and the response is consistently centered on the preference, with tailored suggestions and explanations .

Table 5: Evaluation Rubrics for Preference Alignment

interface of this task is shown in Figure 9. Similar to the setting in Chatbot Arena, human evaluators choose from three options in the test: “1 wins,” “2 wins,” or “Tie.” As shown in Table 7, REALPREF achieves a win rate of 0.69, a tie rate of 0.18, and a loss rate of 0.13, which may help address the quality of our data to some extent. We acknowledge that using newer LLMs compared with those in previous work provides an inherent advantage in such comparisons, but we aim to advance the field by focusing on the actual quality of the data.

As an additional comparison, we examined WildChat (Zhao et al., 2024), a dataset collected from real user–LLM interactions. WildChat contains an average of approximately 2.52 turns per conversation; around 41% of its conversations are multi-turn, and 3.7% exceed ten turns. In REAL-

PREF, preference-expression sessions contain 2–4 turns and random sessions contain 1–2 turns; after being combined into a full context, the resulting interaction-length distribution more closely resembles the long-tailed pattern observed in WildChat.

G.2 Preference Inferability and Test Tasks

In addition to naturalness and plausibility, the conversation data should also ensure that user preferences can be rationally inferred from it, even when expressed implicitly (Type 3 and 4). Also, the designed test questions should reflect the model’s actual performance. To this end, we iteratively conducted “Easy Test” and “Blind Test” during the initial adjustment stage of data generation to control data quality, and used them in our final datasets to address verification.

Score	Criteria
1	Off-topic or irrelevant. The response does not address the question, only irrelevant information.
2	On-topic but no useful information. Example: - The response is not informative and seems like background information or small talk rather than actual advice. - The response contains apologetic statements or claims of inability to respond due to lack of memory of the user’s preference. - The assistant asks for more information without providing any substantive suggestions, and advises directly related to the query.
3	On-topic, some relevant but not constructive information. Example: - The response only gives some abstract solutions; the advice is vague and unactionable. - The response provides some relevant recommendations, but also contains apologetic statements or claims of inability to follow the user’s preference.
4	Constructive, useful and gives clear guidance. The response provides specific, relevant recommendations or advice related to the query without any apologetic statements or claims of inability to respond.
5	Highly constructive, detailed, multiple helpful ideas or perspectives.

Table 6: Evaluation Rubrics for Answer Quality

Dataset	Win	Loss	Tie
REALPREF	0.69	0.13	0.18

Table 7: Pair Evaluation Result

“**Easy Test**” is designed to test whether the user’s implicit preferences can be rationally inferred from the conversation. In the test, one is given the preference-expression conversation snippet and then directly presented with the corresponding MCQ to test understanding. The interface of this task is shown in Figure 10. Human testers completed 200 randomly sampled test items in our benchmark (with the conversation half from Type 3 and half from Type 4), and candidate LLM (Gemini 2.5 Flash-Lite) completed all the MCQ and T/F test items under Preference Type 3 and 4 in our benchmark. The test results are listed in Table 8. The high accuracy on “Easy Test” from the LLM and human help verify that implicit preferences (both Type 3 and 4) are inferable from our dialogue.

“**Blind Test**” is designed to verify the quality of classification questions: without knowing the preferences, it is hard to answer them correctly, i.e., close to random classification. In the test, only MCQ or T/F options were presented without conversation history. The test results of the candidate LLM (Gemini 2.5 Flash-Lite) are listed in Table 9.

It is notable that for the T/F task, the blind test score is close to the random line at 0.5, whereas the score for the MCQ task is higher than the random line at 0.25, even with the identical query and T/F option directly derived from the MCQ. These results indicate that models may select the "odd one out" in MCQs. A possible reason could lie in the existence of negative preferences (dislike or aversion to something). Please refer to Appendix F.1 for a query example of a negative preference. It is systematically unavoidable that the MCQ for negative preferences contains a correct option and 3 incorrect options (the things the user dislikes) that may share a common characteristic, making the correct option easy to identify and reducing task difficulty. The T/F task avoids such shortcuts and better reflects model performance.

Remarks for Benchmark Tasks We recommend the generation task as our main version task because it mirrors authentic user-LLM interactions and can distinguish LLMs that can proactively follow user preferences in free-form generation. The MCQ task is widely used in recent personalization benchmarks (Zhao et al., 2025a; Jiang et al., 2025). This work documents our empirical exploration process: we identified potential shortcuts in MCQs and specifically designed the T/F query to further explore this issue. We retain this part of the experiment and empirical analysis of shortcuts

Figure 9: Human Pair Evaluation Interface

Tester	MCQ (Type 3)	T/F (Type 3)	MCQ (Type 4)	T/F (Type 4)
Human	0.980	-	0.960	-
Gemini 2.5 Flash-Lite	0.972	0.969	0.959	0.952

Table 8: Easy Test Result

Tester	MCQ	T/F
Gemini 2.5 Flash-Lite	0.563	0.546

Table 9: Blind Test Result

in our work to highlight the potential vulnerability of such MCQ designs. We hope it will benefit the community by prompting researchers to consider this in evaluation artifacts for LLMs, supporting the need to move toward other tasks, such as T/F alternatives or open-ended generation.

G.3 Scoring Criteria

To address the reliability of our scoring criteria for open-ended answers, we conducted both human judges and an additional LLM judge (DeepSeek-V4-Pro) compared with our original LLM judge (Claude Sonnet 4) on a subset of answer samples (excluding answers from DeepSeek-V4-Pro itself). After a brief calibration phase using the exact rubric provided to the LLM judge, two human judges

independently scored a shared subset of the data (20 samples, each including user preference, user query, and answer), yielding a Cohen’s kappa of 0.77, indicating their Inter-Annotator Agreement. The scoring interface for human judges is shown in Figure 11. In total, the human judge completed 100 randomly sampled test items, and the added LLM judge (DeepSeek-V4-Pro) completed 2000 randomly sampled test items. The results (Table 10) demonstrate statistically significant correlations (all Pearson correlations have $p < 0.01$), which helps address that our automated evaluation metrics for the generation task align with human judgment and remain consistent with another LLM judge. Figure 12 visualizes the score distribution of the human judge and the original LLM judge across the human-evaluated samples. The distributions exhibit similar patterns: approximately a U-shape; scores of 1, 2, and 5 are slightly more frequent than scores of 3 and 4. This structural alignment visually corroborates the high Pearson correlation reported.

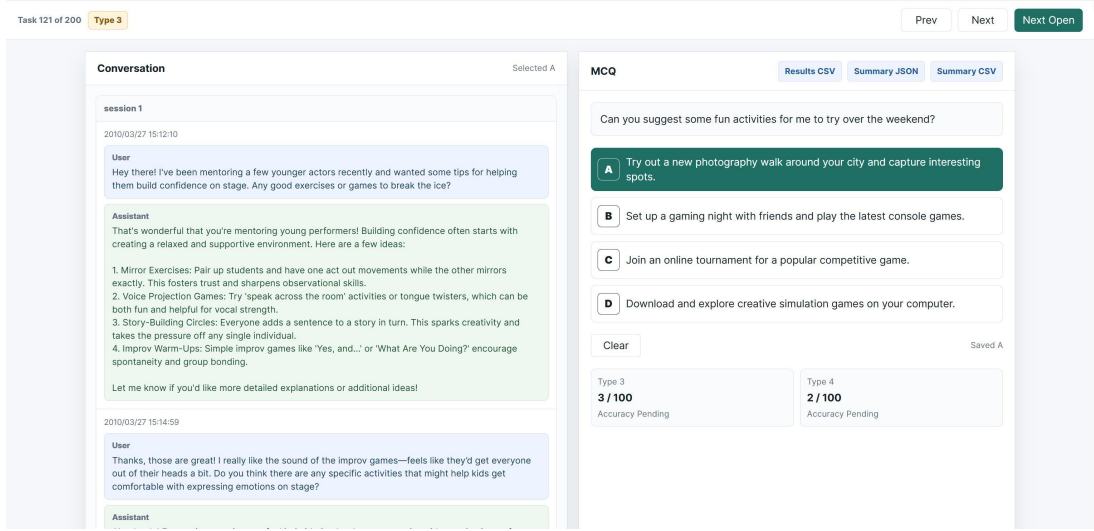


Figure 10: Human Conversation Preference Inferable MCQ Test Interface

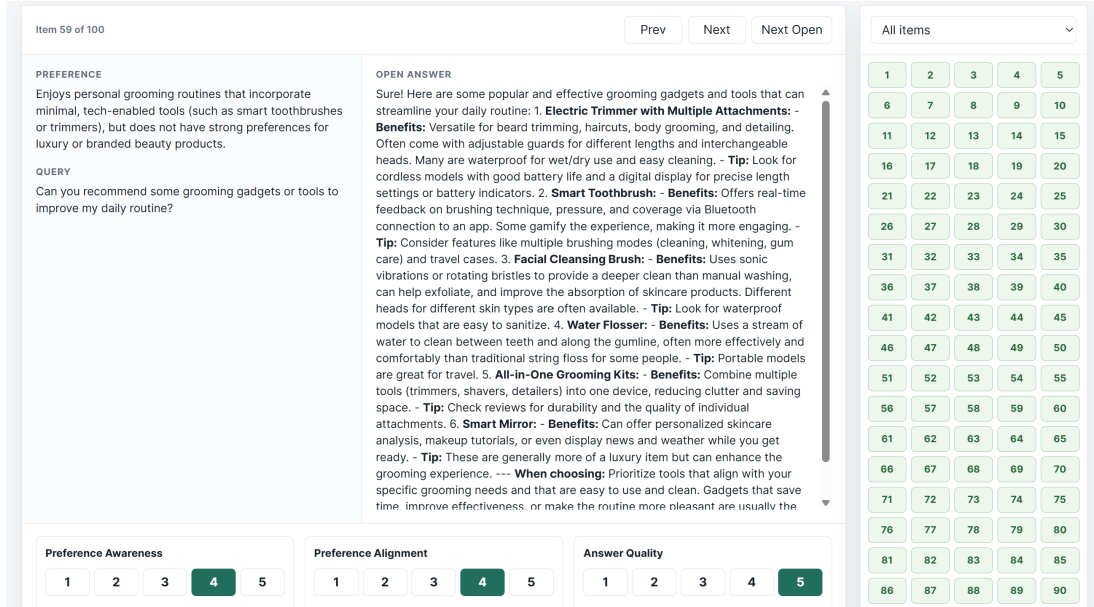


Figure 11: Human Open Answer judge Interface

H Experiment Details

H.1 Test Models

The test model details are listed in Table 11. *GPT-5* (OpenAI, 2025b), *GPT-5 mini*, *Gemini 2.5 Flash-Lite* (Google DeepMind, 2025) and *DeepSeek-V4-Pro* (DeepSeek, 2026) are models equipped with reasoning mode, while *Qwen3-235B-A22B Instruct*, *Qwen3-8B* (QwenTeam, 2025) and *Llama 3.3 70B Instruct* (Meta, 2024) are instruction-tuned models. *Llama 3.3 70B Instruct* and *Qwen3-8B* support the shortest context window, approximately 131K, while other versions support windows exceeding 250K. For open-ended answers, we specify *Claude Sonnet 4* (Anthropic, 2025) as the judge.

H.2 Method Details

H.2.1 Implementation

Here are the specific implementations of methods tested in the experiments.

Reminder: Before presenting the test question, add a reminder: “When you answer the question, recall my preference indicated in the conversation history, and provide an answer that aligns with my preferences.”

Few-Shot Chain-of-Thought: Before presenting the test question, list 3 examples with preference, query, and good response. Below is the specific prompt with the first example. The subsequent examples are omitted for clear demonstration.

Judge	Preference Awareness	Preference Alignment	Answer Quality	Overall
Human	0.875	0.890	0.853	0.878
DeepSeek-V4-Pro	0.846	0.819	0.891	0.855

Table 10: Judge Score Pearson Correlation (with Claude Sonnet 4, $p < 0.01$)

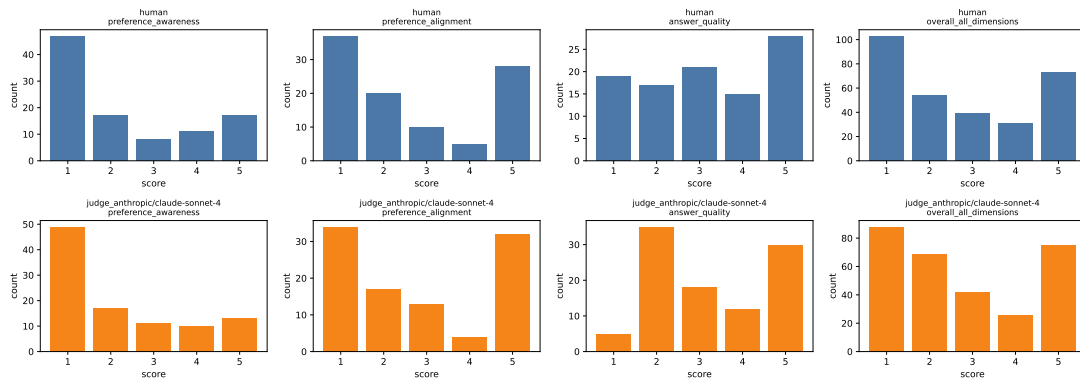


Figure 12: Score Distribution Comparison

Prompt for Few-Shot Chain-of-Thought

When you answer the question, recall my preference indicated in the conversation history, and provide an answer that aligns with my preferences. Here are a few good response examples for your information:

Example 1:

- Preference: Prefers boutique guesthouses with a personal touch over large chain hotels because she values unique local experiences and the chance to connect with the local tennis community.
- Query: Can you suggest where I should stay on my next vacation to Barcelona?
- Good Response: Given your preference for boutique guesthouses with a personal touch, and your interest in connecting with the local tennis community, I recommend staying at a locally-owned boutique guesthouse in one of Barcelona's lively neighborhoods, such as Gràcia or Eixample. These areas are known for their authentic Barcelona atmosphere, with independently-run accommodations that offer a warm, welcoming environment. Unlike large chain hotels, boutique guesthouses often reflect the character of the city and allow for more direct interaction with both hosts and fellow travelers. Additionally, many boutique guesthouses in Barcelona take pride

in connecting guests with local experiences, including arranging activities or providing recommendations for joining community events. You might find guesthouses whose owners love tennis themselves or have contacts in the local tennis scene—this will make it easier for you to ask about and join local matches, book courts, or even find a hitting partner. For example, guesthouses like Casa Mathilda or The5Rooms in Eixample are known for their personalized service, locally-inspired decor, and excellent host connections. They are also conveniently located near several tennis clubs, such as Club Tennis Barcelona or Club Natació Atlètic-Barceloneta, making it easy for you to pursue your tennis interests during your stay. In summary, opting for these boutique guesthouses will not only immerse you in Barcelona's local culture, but also maximize your opportunities for meaningful connections—both with hosts and the city's tennis community.

Example 2:

- Preference: [Example Preference 2]
- Query: [Example Query 2]
- Good Response: [Example Response 2]

Example 3:

- Preference: [Example Preference 3]
- Query: [Example Query 3]
- Good Response: [Example Response 3]

Model	Context Window	Reasoning Mode
GPT-5	400000	Yes
GPT-5 mini	400000	Yes
Qwen3-235B-A22B Instruct	262144	No
Qwen3-8B	131072	No
Gemini 2.5 Flash-Lite	1048576	Yes
DeepSeek-V4-Pro	1048576	Yes
Llama 3.3 70B Instruct	131072	No

Table 11: Test Model Details

Method	Avg. Input Tokens	Avg. Output Tokens (DeepSeek-V4-Pro)	Avg. Output Tokens (GPT-5 mini)	Local Embedding
Zero shot	146927.12	637.66	934.90	No
Reminder	146962.42	930.92	939.44	No
Few-shot CoT	149046.46	1192.68	1107.08	No
RAG	148159.46	876.70	1015.56	Yes

Table 12: Token Cost Report across Methods

Now recall my preference indicated in the conversation history (not the user preferences in the examples above), and provide an answer that aligns with my preferences.

Retrieval-Augmented Generation: Before presenting the test question, list the top 5 conversation turns that are most relevant to the current question. The similarity of question and conversation turns (concatenate user’s message and assistant’s response) is calculated based on a sentence embedding model “princeton-nlp/sup-simcse-roberta-large” (Gao et al., 2021). The specific prompts are listed below.

Prompt for RAG

When you answer the question, recall my preference indicated in the conversation history, and provide an answer that aligns with my preferences.

Here are the 5 most relevant contextual turns to my query that you can consider when answering the question.

Context turn 1:

- user: [message]
- assistant: [message]

Context turn 2:

- user: [message]

- assistant: [message]

Context turn 3:

- user: [message]

- assistant: [message]

Context turn 4:

- user: [message]

- assistant: [message]

Context turn 5:

- user: [message]

- assistant: [message]

Now recall my preference indicated in the conversation history, and provide an answer that aligns with my preferences.

H.2.2 Token Report

To provide a cost-based comparison of performance improvements, we report sample average token consumption for Reminder, Few-shot CoT, and RAG using DeepSeek-V4-Pro and GPT-5 mini in Table 12. All other settings follow RQ4 (Experience Feedback expression, Extended context). Since our RAG pipeline uses local sentence embeddings, its retrieval overhead is not reflected in API token usage. In terms of input and output tokens, Few-shot CoT incurs the highest cost, followed by RAG and then Reminder. When performance and token consumption are considered together, Reminder is the most cost-efficient method in this setting, as we discussed in RQ4.

Model	1 > 2	1 > 3	1 > 4	2 > 3	2 > 4	3 > 4
GPT-5	True	True	True	True	True	False
GPT-5 mini	True	True	True	True	True	False
Gemini 2.5 Flash-Lite	True	True	True	False	False	False
DeepSeek-V4-Pro	True	True	True	False	False	False
Llama 3.3 70B Instruct	False	False	False	False	False	False
Qwen3-235B-A22B Instruct	False	False	False	False	False	False

Table 13: Significant Test Result across Preference Expression Types (True represents the compare direction is significant $p < 0.05$ in all three rating dimensions)

Model	Simple > Normal	Normal > Long	Long > Very Long	Very Long > Extended
GPT-5	True	True	True	True
GPT-5 mini	True	True	True	True
Gemini 2.5 Flash-Lite	False	False	False	False
DeepSeek-V4-Pro	True	True	True	False
Qwen3-235B-A22B Instruct	True	False	False	False

Table 14: Significant Test Result across Context Configurations (True represents the compare direction is significant $p < 0.05$ in all three rating dimensions)

Method	Type 1	Type 3	Type 4
RAG	4.452	3.885	3.702
Mem0	4.476	4.222	4.292

Table 15: Supplementary Method Comparison

H.2.3 Supplementary Experiment

As memory-augmented architectures are emerging, we conducted a small-scale experiment using Mem0 (Chhikara et al., 2025) and compared it with RAG. We used GPT-5 mini (default setting) as Mem0’s memory-processing model and Qwen3-235B-A22B Instruct as the response-generation model. The experiment was conducted under the Extended context setting across Type 1 (Direct Statement), Type 3 (Stylistic Expression), and Type 4 (Experience Feedback). For simplicity, we reported the Preference Awareness dimension for this part. Both RAG and Mem0 are implemented by providing the retrieved memory in the final query prompt after the context history.

The results in Table 15 also show performance decreasing as preference expressions become more implicit, but the degradation under Mem0 is less pronounced. For Type 1, both RAG and Mem0 achieve high scores because direct preference statements are relatively easy to retrieve. For Type 3 and

4, where retrieval becomes more difficult, Mem0 performs better than RAG, likely benefiting from its cross-session memory, which helps summarize and preserve implicit preference evidence.

H.3 Significant Test

To provide statistical evidence for the experimental results in RQ2 and RQ3, we conducted Pairwise Sign Tests to evaluate performance trends across preference types and context lengths. The Significant test results are listed in Table 13 and Table 14.

- For Preference Types (RQ2): Statistical tests confirm that for the GPT-5 series, performance follows a significantly descending trend from Expression Type 1 to Type 4. For Gemini 2.5 Flash-Lite and DeepSeek-V4-Pro, it is significant that Type 1 > Type 2, 3, and 4. This trend was not all significant for Llama 3.3 70B Instruct and Qwen3-235B-A22B Instruct.
- For Context Length (RQ3): The downward performance trend is statistically significant for the GPT-5 series across all 5 context settings and for DeepSeek-V4-Pro across the first 4 context settings. However, this trend was not all significant for other models.

Results-Variance	MCQ	TF	Preference Awareness	Preference Alignment	Answer Quality
GPT-5 mini	0.893±0.005	0.815±0.001	3.091±0.05	3.476±0.05	4.098±0.08
DeepSeek-V4-Pro	0.857±0.025	0.679±0.005	2.258±0.04	2.826±0.08	3.412±0.07

Table 16: Repeated Experiment Variance on Subset Samples

H.4 Repeated Experiment Variance

Regarding the single-run limitations, we incorporated a limited amount of repeated-run analysis on the reduced subset to demonstrate the credibility of the experimental results. We used the subset containing 30% of the benchmark samples, with GPT-5 mini and DeepSeek-V4-Pro across all three evaluation tasks under the same setting as RQ1 (Normal context, Experience Feedback expression, Zero-shot). The results in Table 16 show consistently small variance across repeated runs for all three tasks and both models, indicating that the observed performance is relatively stable and supports the reliability of the broader single-run results.

indicates response-planning failure: the model struggles to synthesize and integrate them into the final answer meaningfully.

Breakdown failure analysis enables researchers to pinpoint where their personalized assistants fail and to implement countermeasures to improve preference recognition, retrieval, memory, awareness, and proactivity in open-ended generation.

I Failure Breakdown for Improvement

Our work’s main contribution lies in the innovation of benchmarks, introducing diverse preference expressions and generalization tasks. Our ultimate goal is to provide guidance on building better systems. To deliver deeper, actionable insights, we can associate model performance across classification and generation tasks, thereby expanding failure analysis by leveraging the specific dimensions of our generation scoring criteria. For example:

- **Low Accuracy on the classification task:** The failure may result from poor preference recognition, inadequate retrieval, or memory design.
- **Low Preference Awareness on the generation task:** The failure may be caused by the same reason if it also performs poorly on the classification task. Otherwise, the Low Preference Awareness score indicates that the model lacks awareness or proactivity in retrieving and integrating user preferences during free-form generation.
- **Low Preference Alignment on the generation task:** If the model has access to the user preferences (e.g., under effective RAG conditions), a performance gap in this dimension