

PREFINFER: Preference Inference from User-LLM Interaction History

Yufei Chen*, Hao Peng*, Yunjia Qi, Lei Hou, Bin Xu†, Juanzi Li
Department of Computer Science and Technology, Tsinghua University
{chen-yf25}@mails.tsinghua.edu.cn

Abstract

Large language model (LLM) personalization aims to generate responses that align with individual user preferences, making accurate preference inference critical. In practice, user-LLM interaction histories naturally contain rich preference signals. However, extracting preferences from these histories remains underexplored and challenging: the history can be long and noisy, preferences may evolve over time, and preferences may vary across situations even within the same topic. To address these challenges, we develop PREFINFER, a model that accurately infers user preferences from the user’s interaction history with an LLM. Specifically, we introduce PREFMINE, an automated data construction method that synthesizes high-quality preference inference data, where each sample includes a synthetic interaction history, a current user request, and the user’s preference for the current request, and construct PREFMINE-15K for training PREFINFER. Experiments show that PREFINFER achieves advanced performance on preference inference, and the preferences inferred by PREFINFER effectively guide LLMs to generate responses better aligned with user preferences. We conduct thorough ablation studies on the key design choices of PREFMINE, validating their effectiveness. We have released the code, data, and models to facilitate further research¹.

1 Introduction

Large Language Models (LLMs) have become a core technology in modern conversational AI systems (OpenAI et al., 2024a). In these systems, an important capability is personalization (Jang et al., 2024; Chen et al., 2025; Guan et al., 2025), where LLMs generate responses that are not only helpful and accurate, but also aligned with users’ individual

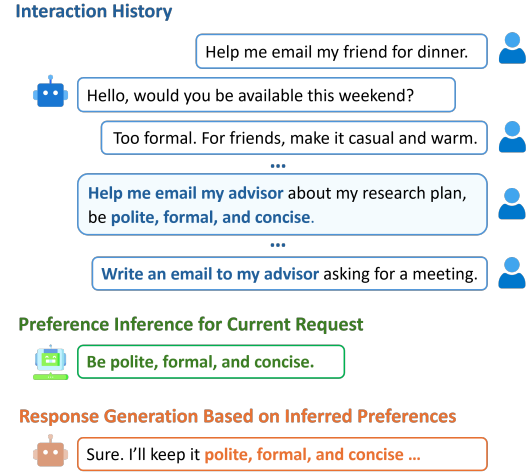


Figure 1: Overview of the preference inference task with a simplified example.

preferences, e.g., formatting style. This capability improves user satisfaction for each interaction and provides a better user experience.

A critical prerequisite for LLM personalization is correctly inferring user preferences. In practice, a user’s interaction history with an LLM contains rich preference signals and thus provides an important source for preference inference. However, identifying preferences relevant to a current request from such history remains underexplored and challenging (Kim et al., 2025; Zhao et al., 2025a; Nie et al., 2026). The history can be long and noisy, user preferences may evolve over time, and request-relevant preferences are often highly situation-dependent, varying across different situations even for the same topic. For example, the style of an email may vary depending on whether it is addressed to a friend or an academic advisor.

To address these challenges, we develop PREFINFER, a model that infers request-relevant user preferences from past user-LLM interactions. The inferred preferences are then used to guide downstream LLMs toward responses better aligned with

*Equal contribution.

†Corresponding author: xubin@tsinghua.edu.cn

¹<https://github.com/THU-KEG/PREFINFER>

the user, as illustrated in Figure 1. A straightforward way to train such a model is to construct supervised data by collecting real user-LLM interaction histories and manually annotating the request-relevant preferences. However, real interaction histories are often long and noisy, making this process costly and difficult to scale. We therefore design PREFMINE, a fully automated synthetic data construction method for generating training instances. The design of PREFMINE is guided by three key challenges of preference inference. First, to capture situation-dependent preferences, we synthesize relevant interaction sessions and a corresponding current request, together with same-topic sessions in different situations where the applicable preferences may differ. Second, to model preference evolution, we include earlier sessions with the same topic and situation but outdated preferences, requiring the model to identify the most recent applicable preference. Third, to simulate long and noisy histories, we sample unrelated real user-LLM sessions as noise in the interaction history. Each training instance pairs the interaction history with a current request and the corresponding request-relevant preference. This process yields 10,000 instances. We further add 4,000 instances to improve robustness to negatively expressed preferences and non-mainstream preferences. While these 14,000 instances are within 8,192 tokens, we additionally construct two long-context subsets, each containing 500 instances, with context-length budgets of 65,536 and 131,072 tokens, respectively. Together, these instances form PREFMINE-15K.

We apply supervised fine-tuning (SFT) (Ouyang et al., 2022) with LoRA (Hu et al., 2022) to Qwen3-4B-Instruct-2507 (Qwen Team, 2025) on PREFMINE-15K to obtain PREFINFER. Experiments show that PREFINFER achieves advanced performance on preference inference, and the preferences inferred by PREFINFER effectively guide LLMs to generate responses better aligned with user preferences. Moreover, our method generalizes well to implicit preference types not explicitly constructed in our training data, and remains effective on user-LLM interaction histories of up to approximately 103k tokens.

We conduct ablation studies on the key design choices of PREFMINE, validating their effectiveness. The same-topic sessions situated in different situations improve situation-dependent preference inference, while earlier sessions with outdated preferences mainly benefit changing preference cases.

Negation augmentation substantially improves performance across preference types, and long-context training data brings larger gains as test histories become longer, especially for implicit preferences.

In summary, our contributions are threefold: (1) We propose PREFMINE, an automated and annotation-free method for constructing preference inference data. (2) Using this method, we construct PREFMINE-15K to train PREFINFER, a model specialized for request-relevant preference inference from user-LLM interaction history. (3) We show that the preferences inferred by PREFINFER effectively guide LLMs to generate responses better aligned with user preferences.

2 Method

2.1 Preliminary

In this paper, we focus on inferring user preferences from user-LLM interaction histories. Let $H = \{s_1, \dots, s_n\}$ denote a user’s interaction history with an LLM, where each session s_i contains one or more turns of user-LLM dialogue, and let q denote the user’s current request. Given (H, q) , the task is to infer a preference description p that is both implied by the history and relevant to the current request. Here, p is a concise natural-language statement capturing the request-relevant user preference, e.g., “Prefer traditional methods and avoid external dependencies.” Accurately inferred preferences can guide the generation of user-preferred responses and further be stored as external memories for retrieval, advancing LLM personalization.

2.2 Overview of PREFMINE

We design a fully automated pipeline, namely PREFMINE, that uses an LLM to synthesize interaction histories, user queries, and the corresponding preferences. PREFMINE mainly proceeds in two stages. First, we describe the core pipeline, which generates preference-grounded (H, q, p) instances from seed topics (§ 2.3): given a seed topic, we first synthesize a user preference p , then generate an interaction history H conditioned on both the topic and p . This *preference-first* design ensures that the resulting histories faithfully reflect the target preference without requiring manual annotation. Then, we enrich the interaction histories to capture the realistic challenges of preference inference (§ 2.4), including *negation* and *long-context augmentation*.

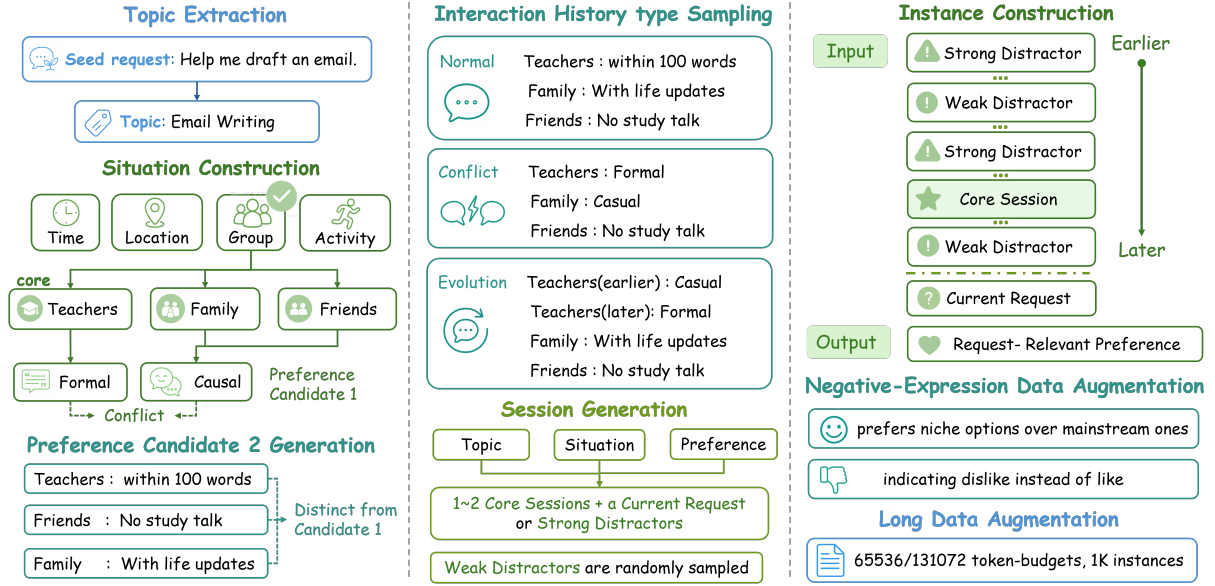


Figure 2: Overview of the training data construction method PREFMINE.

2.3 Core Pipeline of PREFMINE

Data Design To provide request-relevant preference signals while reflecting the key challenges of preference inference, we design each interaction history H to consist of 8 sessions drawn from three types: (1) **core sessions**, which are directly relevant to q and reveal p ; (2) **strong distractors**, which are on the same topic as the core sessions but either correspond to different situations or express outdated preferences, thereby targeting situation-dependent preferences and preference evolution; and (3) **weak distractors**, which are sampled from unrelated real user-LLM interactions to simulate long and noisy histories. Figure 2 gives an overview of the automated construction method. We use GPT-OSS-120B (OpenAI et al., 2025) to automatically construct topic, situations and their corresponding preferences, and to generate **core sessions** and **strong distractors** conditioned on the topic, situations, and preferences. Specifically, q is generated together with the core sessions.

Topic and Situation Construction We sample initial user requests from the Gen subset of Infinity-Instruct (Li et al., 2025a), filter potentially toxic content using LLaMA-Guard-3-8B (Grattafiori et al., 2024), and map each request to a conversation topic. Given a topic, we randomly select one situation dimension from *time*, *location*, *group*, and *activity*. The LLM then generates 3–5 situations by varying the selected dimension. One situation is designated as the core situation, while the others

serve as distractor situations.

Preference Candidate Generation For each situation, we generate two preference candidates. Candidate 1 is constructed to support situation-dependent conflicts: the core situation receives a preference A , while distractor situations receive the opposite preference $\neg A$. Candidate 2 is an additional preference distinct from Candidate 1, generated conditioned on the situation. This gives each situation two possible preferences and allows us to instantiate different interaction history types. More details are provided in Appendix A.1.

Interaction History Type Sampling This step determines the **preference** assigned to each situation. To reflect the situation-dependent and evolving nature of user preferences, we sample one of three interaction history types with equal probability for each training instance: *Normal*, where each situation uses its own preference candidate 2; *Preference Conflict*, where the core situation uses its own candidate 1, one distractor situation is randomly selected to use its own candidate 1, and the remaining distractor situations randomly use either their own candidate 1 or their own candidate 2; and *Preference Evolution*, where the core situation is split into two stages: before evolution, it follows the opposite preference $\neg A$; after evolution, it follows A . In this type, all distractor situations use their own preference candidate 2.

Session Generation For the core situation, we generate 1–2 sessions as **core sessions**. For **strong**

distractors, we first sample the total number of strong distractors from 1 to 6, and then distribute these sessions across distractor situations using multinomial sampling. Some distractor situations may receive no sessions under this sampling process. Each generated session is a 2–8 turn feedback dialogue under the specified topic and situation, where the user iteratively provides feedback to refine the assistant’s response toward the preference. We require each session to cover all aspects of its assigned preference and terminate it once the user expresses satisfaction. For the core situation, we additionally generate a **current request** q consistent with the same topic and situation. **Weak distractors** are randomly sampled, as detailed in Appendix A.2.

Instance Construction We place the generated core sessions and strong distractors into 8 session slots, ensuring that all evolved sessions appear after the corresponding outdated sessions in evolution cases. The remaining slots are filled with randomly sampled weak distractors. We filter out samples longer than 8,192 tokens, resulting in 10,000 training instances that form **PREFMINE-10K**.

2.4 Scaling to Realistic Histories

The core pipeline (§ 2.3) produces high-quality preference-grounded (H, q, p) instances, yet a gap still remains between synthesized and real interaction histories. To characterize this gap, we sample and analyze 50 conversation histories from Wild-Chat (Zhao et al., 2024) and identify two salient properties of real-world histories that pose challenges for preference inference: (1) Long: real histories can span tens of thousands of tokens, substantially increasing the difficulty of preference inference; (2) Entangled preference signals: users express not only what they prefer but also what they dislike that may mislead inference. To close this gap, we design two enrichment modules: *negation augmentation*, which injects preference-confounding signals, and *long-context augmentation*, which scales histories to realistic lengths.

Negation Augmentation Models should account for users’ dislikes in order to avoid responses that conflict with their explicitly stated negative preferences. However, prior studies have shown that language models often exhibit limited sensitivity to negation (Kassner and Schütze, 2020; Rezaei and Blanco, 2025). To improve robustness to negation, we add a training subset PREFMINE-neg. Each

instance contains 1 core session and 7 weak distractors, with a total length at most 8,192 tokens.

We construct 3,000 feedback-based instances. Each instance is defined by a request and two contrasting preferences: a major preference that is more mainstream or commonly preferred, and a minor preference that is a non-mainstream alternative. The core session is a short feedback dialogue, where the user first issues a request, the assistant provides a response, and the user then gives feedback. The feedback either explicitly favors the minor preference while disfavoring the major preference, or only expresses dissatisfaction with the major preference. The target preference is directly derived from the user’s feedback, rather than inferred by comparing candidate options. We further construct 1,000 instances where the user directly expresses preferences in an utterance through a negative form. To vary the strength of preference signals, we include both strong lexical cues, such as *prefer* and *avoid*, and weaker or more implicit expressions, such as *find ... a bit uninteresting* and *will go for*. Detailed statistics are presented in Table 7.

Long-Context Augmentation In real-world applications, dialogue histories can be substantially long, yet prior studies have shown that LLMs often struggle to effectively process and utilize long-context inputs (Liu et al., 2024; Bai et al., 2024). To improve long-context performance, we add two 500-instance subsets with token budgets of 65,536 and 131,072. We denote these subsets as Long-64k and Long-128k. Details are provided in Appendix A.3. We then combine these instances with PREFMINE-neg and PREFMINE-10K to form the final PREFMINE-15K. We manually inspected approximately 50 samples and found little noise. Detailed data quality evaluations are provided in Section A.5. The statistics of the training set are shown in Table 8. We additionally conduct a data contamination check and find negligible overlap with the evaluation data. Details are provided in Appendix A.4.

2.5 Model Training

We train PREFINFER using SFT (Ouyang et al., 2022) with LoRA (Hu et al., 2022) on the Qwen3-4B-Instruct-2507 (Qwen Team, 2025) backbone. The model is trained for 2 epochs with a learning rate of 5×10^{-6} and a global batch size of 8 on NVIDIA A800 GPUs. Details are provided in

Method	P	R	F1
Qwen3-4B-Instruct-2507	41.47	54.15	46.97
DeepSeek-V3.2	<u>47.81</u>	<u>64.33</u>	<u>54.85</u>
Qwen3.5-35B-A3B	46.70	63.77	53.92
PREFINFER (ours)	68.12	67.52	67.82
+ Retrieval	37.67	47.67	41.28

Table 1: Direct Inference results on CUPID. We report precision (P), recall (R), and F1 on the full set. Retrieval uses Qwen3.5-35B-A3B as the inference model. The highest scores are in **bold** and the second highest are underlined.

Appendix B.3.

3 Experiments

3.1 Experimental Setup

Evaluation Settings We evaluate our method from two perspectives: whether models can accurately infer request-relevant preferences, and whether the inferred preferences can effectively guide LLMs to generate responses better aligned with user preferences. This leads to three evaluation settings: **(1) Direct Inference.** Given the interaction history H and the current request q , the goal is to infer the user preference p relevant to q . **(2) Classification.** Given (H, q) and four candidate responses, the goal is to select the response that best aligns with the user’s preference. **(3) Generation.** Given (H, q) , the goal is to provide an answer that aligns with the user’s preference.

Compared Methods We compare the following methods across three settings: **(1) Full-History.** We evaluate three strong LLMs: Qwen3-4B-Instruct-2507 (Qwen Team, 2025), Qwen3.5-35B-A3B (Qwen Team, 2026), and DeepSeek-V3.2 (DeepSeek-AI et al., 2025). For all settings, the model is given the full interaction history H together with the current request q as input. **(2) Retrieval-Based.** For each current request q , we retrieve the top-10 utterances in H that are most semantically similar to q . For all settings, we concatenate the retrieved utterances in chronological order with the current request q and provide the resulting input to Qwen3.5-35B-A3B, a strong model among the full-history baselines. **(3) Memory-Based Methods.** PersonaVLM (Nie et al., 2026) maintains a user-centric memory database and retrieves relevant memories to support Classification and Generation. SeCom (Pan et al., 2025) constructs segment-level conversational memories through

dialogue segmentation and compression, and retrieves request-relevant memories for downstream personalization. **(4) PPOpt.** PPOpt (Ma et al., 2026) leverages the interaction history to rewrite the current request into an optimized prompt for personalized response generation. For Classification and Generation, the optimized prompt is provided to the downstream LLM in place of the original interaction history. **(5) Ours.** For **Direct Inference**, we evaluate PREFINFER directly on (H, q) . For **Classification** and **Generation**, we first use PREFINFER to infer p from (H, q) , and then evaluate the three general LLMs using (p, q) as input. **(6) Two-Stage Prompting.** We replace PREFINFER in our two-stage pipeline with the downstream LLM itself. This verifies that the gains of our method do not merely come from decomposing the task into two stages, but from PREFINFER’s stronger preference inference ability. Detailed implementation settings are provided in Appendix B.2.

Evaluation Datasets We evaluate our method on two public datasets: **CUPID** (Kim et al., 2025) and **PrefEval** (Zhao et al., 2025a). **CUPID** supports both **Direct Inference** and **Generation** settings, assessing whether models can identify the preference relevant to a current request by distinguishing which prior interaction contexts are applicable. Its prior interaction histories average 8,186.7 tokens per instance. It contains three subsets: *Consistent*, *Contrastive*, and *Changing*, which evaluate aligned evidence, conflicting cross-context evidence, and within-context preference shifts, respectively. **PrefEval** evaluates preference following under long dialogue histories through both **Classification** and **Generation**. It inserts varying amounts of irrelevant dialogue context between the preference-revealing conversation and the current request. We use its 25-, 150-, and 300-turn settings, corresponding to roughly 9k, 54k, and 103k tokens of interaction history, respectively. PrefEval includes *Explicit Preference*, where the user directly states a preference, and two implicit subsets: *Implicit Choice-Based*, where the model needs to infer the underlying preference by comparing a user-selected option with alternatives, and *Implicit Persona-Driven*, where the preference is subtly expressed in a longer persona-centered conversation. Notably, our training data does not explicitly construct these two implicit interaction patterns. More details are provided in Appendix B.1.

Method Turns (Tokens)	CUPID N/A	PrefEval Classification			PrefEval Generation			Overall N/A
		25 (9k)	150 (54k)	300 (103k)	25 (9k)	150 (54k)	300 (103k)	
Qwen3-4B-Instruct-2507	7.93	64.83	53.90	48.97	68.87	37.97	30.20	54.86
+ Qwen3-4B-Instruct-2507	7.63	61.63	48.80	42.63	50.50	31.63	21.23	47.53
+ PREFINFER	8.58	93.93	92.50	89.30	90.97	87.70	83.67	89.12
DeepSeek-V3.2	7.68	87.80	68.50	54.70	92.80	78.93	53.40	73.28
+ DeepSeek-V3.2	7.98	88.47	88.10	85.57	85.30	85.97	81.60	84.97
+ PREFINFER	<u>8.50</u>	95.17	93.77	90.63	92.73	<u>90.37</u>	<u>86.77</u>	<u>90.63</u>
Qwen3.5-35B-A3B	7.47	84.63	71.27	70.27	93.53	85.90	86.37	80.95
+ Qwen3.5-35B-A3B	7.91	78.43	68.30	67.87	73.03	59.80	59.23	69.39
+ Retrieval	7.50	85.20	75.43	73.03	87.50	73.60	69.80	77.08
+ PREFINFER	8.58	<u>95.07</u>	<u>93.17</u>	<u>89.70</u>	93.87	90.93	86.87	90.77
PersonaVLM	5.54	73.47	N/A	N/A	51.53	N/A	N/A	N/A
PPOpt	7.12	57.70	N/A	N/A	37.23	N/A	N/A	N/A
SeCom	7.03	71.03	N/A	N/A	56.70	N/A	N/A	N/A

Table 2: Results on several benchmarks under different interaction-history lengths. The highest scores are in **bold** and the second highest are underlined. Overall is averaged over all seven settings after scaling CUPID Generation by 10. $X + \text{PREFINFER}$ denotes using PREFINFER for preference inference and X as the downstream model. $X + X$ denotes Two-Stage Prompting, where X is used for both preference inference and downstream tasks.

3.2 Experimental Results

Results on Direct Inference Table 1 reports Direct Inference results on CUPID. We observe that: (1) our method consistently outperforms all baselines on the full test set. Compared with the strongest baseline, DeepSeek-V3.2, PREFINFER improves the overall F1 from 54.85 to 67.82, yielding a gain of 12.97. In contrast, the retrieval-based baseline performs poorly on this setting. (2) Our method brings a large gain in precision, improving it from the second-best 47.81 to 68.12, while also achieving the best recall (67.52), demonstrating a strong balance between precision and recall. To assess inference stability, we run PREFINFER on Direct Inference three times. Details are provided in Appendix B.5.

Results on Classification and Generation As shown in Table 2, we observe that: (1) Our method improves most settings, and the gains tend to be larger for longer histories. (2) Weaker LLMs benefit more from our method. Qwen3-4B-Instruct-2507 achieves the largest gain on PrefEval, and with PREFINFER, its performance becomes comparable to, and in some cases even surpasses, the baseline performance of DeepSeek-V3.2 and Qwen3.5-35B-A3B. (3) Across all three downstream models, replacing the prompted preference-inference stage with PREFINFER improves the overall score, suggesting that the gains cannot be explained solely by two-stage decomposition or context compression. (4) Our method remains effective across in-

Method	Explicit	Choice	Persona
Qwen3-4B-Instruct-2507	60.30	55.30	52.10
+ Qwen3-4B-Instruct-2507	55.77	51.00	46.30
+ PREFINFER	96.77	88.83	90.13
DeepSeek-V3.2	76.73	71.43	62.83
+ DeepSeek-V3.2	89.27	88.27	84.60
+ PREFINFER	97.80	89.63	92.13
Qwen3.5-35B-A3B	87.17	72.47	66.53
+ Retrieval	84.77	69.43	79.47
+ Qwen3.5-35B-A3B	81.93	73.67	59.00
+ PREFINFER	<u>97.30</u>	<u>89.13</u>	<u>91.50</u>

Table 3: Results on PrefEval Classification across three preference types, averaged over 25-, 150-, and 300-turn settings. We report results on Explicit preference (explicit), and two out-of-distribution preference types not covered in the training set: Implicit Choice-Based (choice) and Implicit Persona-Driven (persona).

teraction histories of different lengths, indicating strong scalability to long-context settings, including histories up to 103k tokens. (5) Finally, PersonaVLM, PPOpt and SeCom perform substantially worse than other methods in aggregate under the evaluated 25-turn setting. Meanwhile, the retrieval-based baseline consistently underperforms our method. These results suggest that the retrieval- and memory-based baselines considered here are insufficient. In summary, PREFINFER can effectively guide LLMs to generate responses better aligned with user preferences.

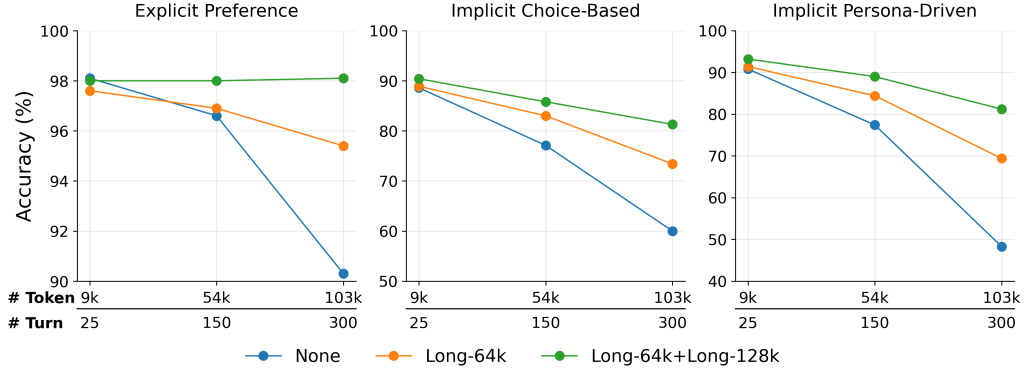


Figure 3: Ablation on long-context training data for the three preference types in PrefEval Generation under different interaction-history lengths. None, Long-64k, and Long-64k+Long-128k denote three inference models trained by removing both long-context subsets, keeping only Long-64k, and keeping both Long-64k and Long-128k, respectively. For each variant, the trained inference model first infers the request-relevant preference, and Qwen3.5-35B-A3B then generates the response conditioned on the inferred preference.

Training Set	All			Consistent		Contrastive		Changing	
	P	R	F1	P	R	P	R	P	R
PREFMINE-10K-weak	59.74	64.99	62.25	64.20	69.21	60.58	65.57	54.52	60.16
PREFMINE-10K-no-evolve	67.40	66.44	66.92	73.95	72.72	71.90	70.83	56.59	55.71
PREFMINE-10K	<u>67.44</u>	66.98	<u>67.21</u>	70.51	<u>69.85</u>	67.88	67.74	64.02	<u>63.33</u>
PREFMINE-10K + neg	68.04	66.74	67.39	<u>72.01</u>	68.51	66.98	66.48	65.24	65.23
PREFMINE-6K + neg	66.73	64.19	65.60	69.66	66.62	66.98	66.48	<u>65.00</u>	62.27

Table 4: Ablations on the core PREFMINE pipeline and negation augmentation using inference models trained on different training sets. Each row corresponds to an inference model trained with the specified training data, evaluated on CUPID Direct Inference and its three subsets. “neg” denotes PREFMINE-neg. We report Direct Inference results on CUPID and its three subsets, including precision (P), recall (R), and F1 on the full set, and precision/recall on each subset.

Training Set	Explicit	Choice	Persona
PREFMINE-10K	66.2	66.6	67.6
PREFMINE-10K + neg	96.9	90.7	93.1
PREFMINE-6K + neg	97.0	90.5	93.1

Table 5: Ablation results on PrefEval Classification with $\sim 9k$ tokens of interaction history. We report results on Explicit Preference (explicit), Implicit Choice-Based (choice), Implicit Persona-Driven (persona). “neg” denotes PREFMINE-neg. Adding PREFMINE-neg consistently improves performance across all three preference types.

4 Analysis on PREFMINE

This section provides a targeted analysis of the key design choices in PREFMINE. We first evaluate the generalization ability of our method in § 4.1, testing whether its gains extend to preference types not explicitly covered by the training data rather than merely overfitting to the training preference format. We then analyze the necessity of using

strong distractors in § 4.2, which is central to constructing challenging instances for preference inference. Next, §§ 4.3 and 4.4 examine the effectiveness of the two data augmentation strategies introduced in PREFMINE, validating their contributions to the overall method design. Finally, in § 4.5, as PREFMINE-15K always contains task- and situation-specific preferences relevant to the current request, we investigate how PREFINFER behaves when such preferences are absent.

4.1 Generalization to Implicit Preferences

As described in § 3.1, PrefEval covers both explicit and implicit preference types, while our training data does not explicitly construct the two implicit types, i.e., choice-based or persona-driven interaction patterns. Thus, these implicit subsets serve as an out-of-distribution evaluation of PREFINFER. Table 3 shows that the improvements are consistent across explicit and implicit subsets, demonstrating that PREFINFER learns a general ability to infer

user preferences, rather than only fitting the explicit preference patterns used in our data construction.

4.2 Analysis on Core Pipeline of PREFMINE

This section analyzes the impact of key design choices in the core pipeline on model performance. As described in §2.2, a main component of PREFMINE is strong distractors, which train the model to distinguish genuine user preferences from confounding signals. We analyze both the overall effect of strong distractors and the specific contribution of evolution-related distractors. For the former, we construct a weakened variant, PREFMINE-10K-weak, by replacing each strong distractor with a randomly sampled WildChat session matched within 200 tokens in length. For the latter, we construct PREFMINE-10K-no-evolve. In this variant, for *Preference Evolution* instances, we replace sessions that share the same topic and situation as the core sessions but appear earlier in the history with randomly sampled weak distractors. This removes outdated preferences from the same topic and situation, isolating the effect of evolution-related strong distractors.

We train models on the original, weak, and no-evolution variants, respectively. As shown in Table 4, removing strong distractors leads to a large performance drop across all subsets, especially on the *contrastive* and *changing* subsets. This indicates that same-topic but situation-inapplicable distractors are important for distinguishing request-relevant preferences from superficially similar ones. Comparing PREFMINE-10K-no-evolve with PREFMINE-10K, we find that removing evolution-related distractors performs well on the *consistent* and *contrastive* subsets, but much worse on the *changing* subset. In contrast, incorporating them substantially improves the *changing* subset and leads to more balanced performance across preference types.

4.3 Analysis on Negation Augmentation

As mentioned in § 2.4, negation augmentation is a key design for scaling to realistic histories, since real conversations often interleave preferences with dislikes. In this section, we analyze the effect of negation augmentation using PREFMINE-neg, a subset containing 4,000 instances specifically constructed for this purpose. The detailed construction of PREFMINE-neg is described in § 2.4. We conduct comparative experiments on PrefEval Classification with approximately 9k tokens of inter-

action history. The results show that PREFMINE-neg brings substantial and consistent improvements across all three preference types. Specifically, performance improves by more than 20 percentage points on all three types of preferences. The detailed results are shown in Table 5. We further evaluate the effect of PREFMINE-neg on the Direct Inference setting of CUPID. As shown in Table 4, we observe that the overall F1 score is slightly improved, with a marginal increase in precision and a slight decrease in recall. Notably, the improvement is not simply due to having more training samples. We construct PREFMINE-6K by randomly sampling 6K instances from PREFMINE-10K, such that PREFMINE-6K + neg and PREFMINE-10K both contain 10K instances. On PrefEval Classification, PREFMINE-6K + neg achieves substantially better performance, confirming the effectiveness of the negation augmentation module in PREFMINE. Moreover, this result reveals that LLMs may fail to correctly understand preference negation, deserving further attention.

4.4 Analysis on Long-Context Augmentation

To study the effect of long-context training data, we conduct an ablation on the Long-64k and Long-128k subsets in PREFMINE-15K. The length statistics of these subsets are reported in Table 8. As shown in Figure 3, we compare three settings: removing both subsets, keeping only Long-64k, and keeping both Long-64k and Long-128k. We observe two main findings. First, adding Long-64k and then Long-128k tends to yield larger overall gains as the test history becomes longer. This suggests that long-context training data is particularly important for handling long interaction histories. Second, the gains are not uniform across preference types. On *Explicit Preference*, the improvement is marginal, since the *None* setting already performs strongly. In contrast, *Implicit Choice-Based* and *Implicit Persona-Driven* show much larger improvements. This indicates that long-context data mainly benefits implicit preference understanding.

4.5 When no Request-Relevant Preference Exists

In PREFMINE-15K, relevance is defined relatively narrowly: the historical evidence should match both the task type and the situation of the current request. Under this definition, request-relevant preferences may often be absent from real interaction histories. PREFMINE-15K focuses on task- and

situation-specific preference inference rather than explicitly constructing such cases. We sample 100 CUPID test instances and remove the preference-revealing dialogues from each history. We then manually examine whether the preference inferred from the remaining history can be naturally applied to the current request without introducing obvious task-irrelevant constraints. All 100 inferred preferences pass this check. This preliminary analysis suggests that, when task- and situation-specific preference evidence is absent, PREFINFER tends to produce broadly applicable preferences rather than clearly incompatible ones, reducing the risk of degrading the quality or relevance of the downstream response.

5 Related Work

Personalization aims to tailor model responses to individual users by leveraging different sources of user information, including user-LLM interaction histories (Kim et al., 2025; Zhao et al., 2025a), explicit user profiles (Tao et al., 2025; Zhao et al., 2025b), and user behavior signals (Salemi et al., 2024; Kumar et al., 2024), where user-LLM interaction histories often provide the richest source of preference information.

Numerous methods have been proposed to leverage user-LLM interaction history for personalization. These methods can be broadly categorized into four groups: **(1) User-Specific Parameter Adaptation.** The core idea of this line of work is to perform additional training or parameter-efficient fine-tuning on a model using a particular user’s historical dialogues, thereby learning a personalized behavior pattern tailored to that user (Otsuka et al., 2024; Han et al., 2023). Such methods require extra training for each user, which can be time-consuming and difficult to scale in practice. **(2) Retrieval-Based Methods.** These methods retrieve historical dialogue segments that are most relevant to the current request and use them as context for downstream personalization (Zhao et al., 2025a; Zhang et al., 2026; Du et al., 2026). However, in long dialogue histories, relevant segments can be easily obscured by other contextual information, making retrieval less reliable. **(3) Memory-Based Methods.** This line of work compresses interaction history into memory representations, such as user profiles, which are then used to guide future responses (Nie et al., 2026; Li et al., 2025b; Wang et al., 2024; Pan et al., 2025). However, such

methods often need to decide what information to preserve before observing the future request, which may lead to the loss of details that later turn out to be crucial for request-relevant preference inference.

(4) Other Personalization Methods. For example, PPOpt (Ma et al., 2026) leverages interaction history to rewrite the current query into an optimized prompt for personalized response generation. Compared with prior approaches, our method requires no per-user retraining and avoids relying on retrieved segments or pre-compressed memory, while directly inferring request-relevant preferences from the interaction history, enabling more flexible preference inference.

6 Conclusion

In this work, we introduce PREFMINE, an automated training data construction method for request-relevant preference inference from user-LLM interaction history. Using PREFMINE, we construct PREFMINE-15K and train PREFINFER. Experiments show that PREFINFER achieves advanced performance on preference inference, and the preferences inferred by PREFINFER effectively guide LLMs to generate responses better aligned with user preferences. Moreover, our method generalizes well to implicit preference types that are not explicitly covered in our training data and remains effective on user-LLM interaction histories of up to approximately 103k tokens. Finally, we conduct thorough ablation studies on the key design choices of PREFMINE, validating their effectiveness. We hope our work and the released resources will contribute to future advances in LLM personalization.

Limitations

Our work has three primary limitations: (1) Model scale. Due to computational limitations, we conduct experiments only with a 4B preference inference model. Scaling to larger models may further improve performance. (2) Context length. Our model currently supports a maximum context length of 128k tokens. Extending PREFINFER to substantially longer interaction histories is left for future work. (3) Task scope. The established benchmarks used in our evaluation do not systematically cover histories with no relevant preference, weak evidence, or ambiguous evidence. Developing more comprehensive evaluation settings for these cases remains future work.

Ethical Considerations

We discuss the potential ethical concerns of this work. **(1) Intellectual property.** PREFMINE-15K is constructed from multiple sources, including Infinity-Instruct, which is released under CC BY-SA 4.0, and WildChat-4.8M, which is released under ODC-BY. PREFMINE-15K is released under CC BY-SA 4.0. The original upstream materials remain subject to their respective licenses and applicable terms of use, and we provide appropriate source attribution and license notices in the dataset documentation. **(2) Intended use.** This paper introduces PREFMINE, aiming to improve preference alignment by helping LLMs generate responses that better match request-relevant user preferences inferred from prior interactions. **(3) Potential risk control.** We apply LLaMA-Guard-3-8B to filter potentially toxic content from sampled initial user requests before using them for data construction. For weak distractors sampled from WildChat-4.8M, we use its non-toxic release and follow the dataset license and terms of use. However, automatic filtering may not remove all harmful or sensitive content. **(4) AI assistance.** We used ChatGPT to assist with paraphrasing and polishing some sentences during the writing of this paper.

Acknowledgements

We thank all the anonymous reviewers and meta-reviewers for their valuable comments. This work is supported by the Jing-Jin-Ji Regional Integrated Environmental Improvement–National Science and Technology Major Project (Grant No. 2026ZD1212206) and a grant from the Institute for Guo Qiang, Tsinghua University (Grant No. 2019QGB0003). We used ChatGPT for language polishing of text originally written by the authors. All AI-assisted edits were reviewed and verified by the authors.

References

- Anthropic. 2024. The Claude 3 model family: Opus, sonnet, haiku. <https://api.semanticscholar.org/CorpusID:270640496>. Model card.
- Anthropic. 2026. Claude Sonnet 4.6 system card. <https://www.anthropic.com/system-cards>. System card.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. **LongBench: A bilingual, multi-task benchmark for long context understanding**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.
- Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. 2025. PAD: Personalized alignment of llms at decoding-time. In *International Conference on Learning Representations*, volume 2025.
- DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenhao Xu, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, and 245 others. 2025. **DeepSeek-V3.2: Pushing the frontier of open large language models**. Preprint, arXiv:2512.02556.
- DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, and 300 others. 2026. **Deepseek-v4: Towards highly efficient million-token context intelligence**. Preprint, arXiv:2606.19348.
- Linfeng Du, Ye Yuan, Zichen Zhao, Fuyuan Lyu, Emiliano Penaloza, Xiuying Chen, Zipeng Sun, Jikun Kang, Laurent Charlin, Xue Liu, and Haolun Wu. 2026. **Optimizing user profiles via contextual bandits for retrieval-augmented LLM personalization**. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31803–31817, San Diego, California, United States. Association for Computational Linguistics.
- 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.
- Google DeepMind. 2026. **Gemini 3.7 flash**. Model Card. Published August 13, 2026.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. **The llama 3 herd of models**. Preprint, arXiv:2407.21783.
- Jian Guan, Junfei Wu, Jia-Nan Li, Chuanqi Cheng, and Wei Wu. 2025. A survey on personalized alignment—the missing piece for large language models

- in real-world applications. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5313–5333.
- Xu Han, Bin Guo, Yoon Jung, Benjamin Yao, Yu Zhang, Xiaohu Liu, and Chenlei Guo. 2023. Personapkt: Building personalized dialogue agents via parameter-efficient knowledge transfer. In *Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP)*, pages 264–273.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2024. [Personalized soups: Personalized large language model alignment via post-hoc parameter merging](#). In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*.
- Nora Kassner and Hinrich Schütze. 2020. [Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.
- Tae Soo Kim, Yoonjoo Lee, Yoonah Park, Jiho Kim, Young-Ho Kim, and Juho Kim. 2025. [CUPID: Evaluating personalized and contextualized alignment of LLMs from interactions](#). 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](#). *Preprint*, arXiv:2407.11016.
- J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.
- Jijie Li, Li Du, Hanyu Zhao, Bo-wen Zhang, Liangdong Wang, Boyan Gao, Guang Liu, and Yonghua Lin. 2025a. Infinity instruct: Scaling instruction selection and synthesis to enhance language models. *arXiv preprint arXiv:2506.11116*.
- Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, Jihao Zhao, Yezhaohui Wang, Peng Liu, Zehao Lin, Pengyuan Wang, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, and 20 others. 2025b. [Memos: A memory os for ai system](#). *arXiv preprint arXiv:2507.03724*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paran-jape, 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.
- Yuchen Ma, Yue Huang, Wenjie Wang, Xiaonan Luo, Xiangliang Zhang, and Stefan Feuerriegel. 2026. [Synthetic interaction data for scalable personalization in large language models](#). *Preprint*, arXiv:2602.12394.
- 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, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024a. [GPT-4 technical report](#). *Preprint*, arXiv:2303.08774.
- OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.
- OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024b. [GPT-4o system card](#). *Preprint*, arXiv:2410.21276.
- Atsushi Otsuka, Kazuya Matsuo, Ryo Ishii, Narichika Nomoto, and Hiroaki Sugiyama. 2024. [User-specific dialogue generation with user profile-aware pre-training model and parameter-efficient fine-tuning](#). *Preprint*, arXiv:2409.00887.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. 2025. [SeCom: On memory construction and retrieval for personalized conversational agents](#). In *The Thirteenth International Conference on Learning Representations*.

Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).

MohammadHossein Rezaei and Eduardo Blanco. 2025. [Making language models robust against negation](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8123–8142, Albuquerque, New Mexico. Association for Computational Linguistics.

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.

Meiling Tao, Chenghao Zhu, Dongyi Ding, Tiannan Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2025. Personafeedback: A large-scale human-annotated benchmark for personalization. *arXiv preprint arXiv:2506.12915*.

Tiannan Wang, Meiling Tao, Ruoyu Fang, Huilin Wang, Shuai Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2024. Ai persona: Towards life-long personalization of LLMs. *arXiv preprint arXiv:2412.13103*.

Yingyi Zhang, Junyi Li, Wenlin Zhang, Pengyue Jia, Xianneng Li, Yichao Wang, Derong Xu, Yi Wen, Huifeng Guo, Yong Liu, and Xiangyu Zhao. 2026. [Evoking user memory: Personalizing LLM via recollection-familiarity adaptive retrieval](#). In *The Fourteenth International Conference on Learning Representations*.

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

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. [WildChat: 1M chatGPT interaction logs in the wild](#). In *The Twelfth International Conference on Learning Representations*.

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.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyang Luo. 2024. [LlamaFactory: Unified efficient fine-tuning of 100+ language models](#). In

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

A Data Construction Details

As described in § 2.3, unsafe data are filtered using Llama-Guard-3-8B with temperature 0.0 and top-p 1.0. GPT-OSS-120B is used for training data generation, with temperature 0.7 and top-p 0.95.

A.1 Preference Candidate 1 Construction

To increase the diversity of generated preferences, we sample a preference dimension for each instance. With probability 0.5, we select content_preference_dimension, which is defined in Table 16; otherwise, we uniformly sample one dimension from preference_dimensions, as listed in Table 17. We then provide the selected dimension and description fields to the LLM and prompt it to generate a corresponding preference.

A.2 Instance Construction

We construct the weak distractor pool from the non-toxic release of WildChat-4.8M². Specifically, we retain only conversations satisfying two conditions: (1) the number of turns is no greater than 8, and (2) the serialized conversation length satisfies $\text{len}(\text{str}(\text{conversation})) \leq 8000$, where conversation is a user-assistant dialogue represented as a list of dict objects. After filtering, we group conversations into buckets according to their turn counts. For each weak distractor, we first sample a turn bucket uniformly at random, and then apply random.choice to select one conversation from that bucket.

A.3 Long-Context Augmentation

To construct long-context training instances, we first generate 500 base instances using the same procedure as PREFMINE-neg described in § 2.4, with the additional constraint that each base instance contains at most 8192 tokens. These instances follow the subset definitions in Table 7: we sample 100, 100, 100, and 200 base instances following the definitions of subsets 0-3 in this table. Starting from these base instances, we then expand each instance to a substantially longer context by repeatedly appending external conversational content.

²<https://huggingface.co/datasets/allenai/WildChat-4.8M>

Table 14 summarizes the full procedure. Following the procedure, we construct Long-64k. To construct Long-128k, we use the same procedure and simply replace `UniformInt(45000, 65000)` with `UniformInt(85000, 125000)`. We then discard instances whose lengths exceed the corresponding context-length budgets, i.e., 65,536 tokens for Long-64k and 131,072 tokens for Long-128k.

A.4 Data Contamination Check

To assess potential data contamination, we conduct a 5-gram overlap analysis between our constructed training data and the evaluation sets. For each training instance, we concatenate its interaction history, current request, and target preference into a single text sequence. We apply the same serialization procedure to the evaluation instances from CUPID and PrefEval. For PrefEval, we use the 25-turn setting, corresponding to approximately 9k tokens of interaction history. The 25-, 150-, and 300-turn settings share the same preference-bearing conversation and current request, and differ only in the length of inserted irrelevant dialogue context. Since the longer settings are formed by taking longer prefixes from the same 300-turn irrelevant dialogue, the 25-turn context is a strict prefix of the longer contexts. Thus, the 25-turn setting already covers the preference-relevant content and current request that are most relevant for contamination analysis. Running the same 5-gram comparison on the longer settings would substantially increase computational cost while mainly adding irrelevant-context prefixes. Specifically, for each evaluation instance, we compute its word-level 5-gram set and compare it against every training instance. We report the maximum containment score:

$$\text{Containment}(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)},$$

where A and B denote the 5-gram sets of an evaluation instance and a training instance, respectively. The maximum containment scores are 0.0113 for CUPID and 0.0077 for PrefEval, indicating negligible overlap between our training data and the evaluation data.

A.5 Data Quality Evaluations

To assess the quality of PREFMINE-15K, we randomly sampled 100 instances and independently evaluated them using three LLM judges: GPT-4o (gpt-4o-2024-11-20) (OpenAI et al., 2024b), DeepSeek-V4-Pro (DeepSeek-AI et al., 2026), and

Gemini-3.7-Flash (Google DeepMind, 2026), with the temperature set to 0.0 and top-p to 1.0 for all judge-model calls. The evaluation prompt is shown in Table 13. As shown in Table 12, the sampled instances from PREFMINE-15K are generally well grounded, relevant to the corresponding requests, and largely free from unsupported or overly specific information, suggesting high overall data quality.

We further evaluate the human-likeness of user turns in PREFMINE-15K. Specifically, we sample 1,000 user utterances from PREFMINE-15K and 1,000 from WildChat (Zhao et al., 2024), pair them one-to-one, and ask DeepSeek-V3.2 to identify which utterance in each pair is AI-generated. We focus on user turns because the assistant turns in both datasets are LLM-generated. DeepSeek-V3.2 achieves only 47.8% accuracy, which is close to random chance. This result suggests that the synthetic user utterances in PREFMINE-15K are difficult to distinguish from user utterances in real WildChat conversations, providing further evidence that PREFMINE-15K exhibits strong human-likeness at the user-turn level.

B Experimental Details

B.1 Evaluation Details

This subsection describes the evaluation protocols for CUPID and PrefEval. For all judge-model calls, we set the temperature to 0.0 and top-p to 1.0. We follow the maximum-generation-length settings of the original benchmarks: `max_tokens` is set to 8192 for CUPID and 300 for PrefEval.

For the main evaluations, we use the full evaluation sets without sampling: all 756 instances from CUPID and all 3,000 instances from PrefEval. For PrefEval Classification, each instance is a four-way selection task with a gold label, and a prediction is considered correct if the model selects the candidate response corresponding to the gold label. We next describe the evaluation protocols for CUPID and the Generation setting on PrefEval.

Following the evaluation protocol of CUPID, we evaluate preference inference by first using GPT-4o (gpt-4o-2024-11-20) (OpenAI et al., 2024b) to decompose each inferred preference into a set of atomic checklists. These checklists are then compared against the gold preference using PrefMatcher-7B (Kim et al., 2025), which computes precision and recall for the preference inference results. For Generation, CUPID evaluates preference alignment by prompting GPT-4o with

	Claude: Neg	Claude: Pos
DeepSeek: Neg	377	107
DeepSeek: Pos	56	460

Table 6: Confusion matrix between DeepSeek-V3.2 and Claude Sonnet 4.6 on the sampled subset used for judge agreement analysis.

the gold preference and the model-generated response, and asking it to assign a score from 0 to 10 based on how well the response satisfies the preference.

For the PrefEval Generation setting, the original benchmark uses Claude 3 Sonnet (Anthropic, 2024) as the judge model to determine whether each response is both helpful and aligned with the user’s preference. A response is considered correct only if it satisfies both criteria. Since this model is no longer available, the closest replacement would be Claude Sonnet 4.6 (Anthropic, 2026). However, due to its high evaluation cost, we instead use DeepSeek-V3.2 (DeepSeek-AI et al., 2025) as the judge model. To verify the reliability of this substitution, we measure the agreement between the two judges on a sampled subset from the setting Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Full-history, 25 turns (9k tokens) and Implicit Persona-Driven. We choose this subset because its DeepSeek-V3.2 accuracy is close to 50%, making the judged labels relatively balanced and reducing the risk that agreement is inflated by an overly easy or skewed subset. Note that the judge only observes the target preference and the generated response, and does not observe the interaction history or the preference type. Therefore, the selected subset affects the balance of judged labels, but not the form in which the preference is expressed in the history. As shown in Table 6, the two judges achieve a Cohen’s κ score of 0.673 on this subset, indicating substantial agreement (Lan-dis and Koch, 1977). Based on this result, we use DeepSeek-V3.2 as the judge model for Generation on PrefEval.

B.2 Implementation Details of Compared Methods

For **Ours** and **Full-history**, we evaluate three backbone models: Qwen3-4B-Instruct-2507, DeepSeek-V3.2, and Qwen3.5-35B-A3B. For the **Retrieval** method, we use Qwen3.5-35B-A3B. Across these methods, we set the temperature to 0.0 and top- p to

1.0, and disable thinking mode for all model runs.

We disable thinking mode for both efficiency and fairness. Since the **Full-history** method takes the entire user-LLM interaction history as input, enabling thinking mode would introduce substantial additional test-time token consumption, increasing both cost and latency. It may also make the comparison unfair by giving some models extra test-time computation, as similarly noted in PrefEval for GPT-o1-preview.

Ours For Direct Inference, we adopt the full prompt template from CUPID for a fair comparison when evaluating³. For PrefEval, which does not provide a direct preference-inference setting, we use the same inference template of CUPID to infer the request-relevant preference, but remove the analysis part from the prompt during the preference inference stage. For Classification and Generation, we construct templates conditioned on the inferred preference, following the corresponding templates in PrefEval (Zhao et al., 2025a). For 5% of the training data, the inference template includes the analysis part. The resulting templates are shown in Table 18 and Table 19, respectively.

Full-history For the full-history baseline, we provide the LLM with the complete user-LLM interaction history together with the current request. For CUPID, we strictly follow the original prompt used in its paper. For PrefEval, since the original work evaluates five methods and shows that Reminder performs the best, we adopt Reminder in our experiments. The following reminder sentence is added immediately after the user’s question and before the LLM’s response: "In your response, please ensure that you take into account our earlier discussion, and provide an answer that is consistent with my preference."

Retrieval We implement retrieval using princeton-nlp/sup-simcse-roberta-large (Gao et al., 2021). Each utterance in the dialogue history is encoded independently, where an utterance is defined as a single message regardless of whether it is produced by the user or the assistant. We then compute cosine similarity between the embedding of the current request and each historical utterance embedding. Based on these similarity scores, we retrieve the top-10 most relevant utterances. To preserve the original

³https://github.com/kixlab/CUPID/blob/main/prompts/evaluation/preference_inferer.yaml

conversational flow, the retrieved utterances are reordered chronologically before being inserted into the prompt. Finally, the reordered top-10 utterances, together with the current request, are provided to the LLM as input. The inference templates are shown in Table 20 and the template of downstream preference-following tasks is shown in Table 21, respectively.

PersonaVLM For the PersonaVLM baseline, we follow the implementation setting of the original work. PersonaVLM is a long-term personalized multimodal agent framework that maintains a user-centric memory database and retrieves relevant memories for personalization (Nie et al., 2026). In our implementation, we adapt it to the text-only setting of our benchmark. Specifically, for each test instance, we sequentially add the user-LLM interaction history into the memory system turn by turn, following the chronological order of the dialogue history. After all historical turns are stored, we query PersonaVLM with the current user request and use its retrieved memories to support Classification or Generation. We adopt the same reminder prompting method as used in PrefEval (Zhao et al., 2025a).

PPOpt We use the publicly released PPOpt (Ma et al., 2026) checkpoint⁴ with the inference configuration provided in its model card. Given the interaction history and current request, PPOpt rewrites the current request into an optimized prompt. For downstream evaluation, we use Qwen3.5-35B-A3B for both Classification and Generation, with the optimized prompt serving as the personalized request. For Classification, the corresponding multiple-choice options are additionally appended to the downstream prompt.

SeCom We use the official SeCom (Pan et al., 2025) implementation and adopt the configuration in SeCom/secom/configs/mpnet.yaml⁵ for memory retrieval. We construct memory at the segment level and retrieve the top-1 most relevant segment for each current request. Before retrieval, memory segments are compressed using LLMingua-2 with a compression rate of 0.75. The retrieved segment is then provided to Qwen3.5-35B-A3B together with the current request for downstream Classification or Generation. For Classification, the corresponding multiple-choice options are additionally

appended to the downstream prompt.

B.3 Training Details

As shown in Table 8, our training data contains 1,000 ultra-long instances from the Long-64k and Long-128k subsets. To improve training efficiency, we group instances with similar sequence lengths into the same mini-batches. Therefore, we arrange the data by subset, placing instances from Long-64k, Long-128k, and the shorter subsets into separate groups whenever possible. We disable data shuffling, so that instances within each batch have more similar lengths and unnecessary padding is reduced. We implement SFT with LoRA using LLaMA-Factory (Zheng et al., 2024). Training PREFINFER used 4 NVIDIA A800 GPUs for approximately 16 GPU-hours. The full training configuration is shown in Table 15.

B.4 Result Details

The complete experimental results are reported in Tables 22, 23, 24, and 25. Except for the stability analysis in § B.5, all reported main results are from a single run over the full evaluation set.

B.5 Inference Stability

To minimize run-to-run variation, we set the decoding temperature to 0.0 for all inference runs. To assess the stability of PREFINFER, we rerun Direct Inference three times. Since the inferred preferences are free-form natural-language descriptions, semantically equivalent predictions may differ lexically across runs. We therefore assess inference stability using CUPID’s semantic preference-matching evaluation rather than exact string matching. As shown in Table 10, the aggregate Precision, Recall, and F1 scores are highly consistent across runs. We further measure per-instance stability using the F1 range, i.e., the difference between the maximum and minimum F1 across runs: 51.7% of instances have an F1 range of 0, 75.9% have a range within 0.10, 89.0% within 0.20, and only 1.9% exceed 0.50. Since CUPID evaluates each instance with only a few discrete checklist items scored as 0, 0.5, or 1, a single judgment change can cause a noticeable F1 shift. Under this coarse scoring scheme, these results indicate strong stability.

B.6 Efficiency Analysis

We report end-to-end model calls and token usage, as heterogeneous deployment settings make direct latency comparisons unreliable.

⁴<https://huggingface.co/HowieHwang/ppopt>

⁵<https://github.com/microsoft/SeCom>

Subset	Tone	Preference Type	Expression Form
0	Strong	Pos & Neg	Feedback
1	Weak	Pos & Neg	Feedback
2	Weak	Neg Only	Feedback
3	Strong & Weak	Neg Only	Direct

Table 7: Overview of the four subsets in PREFMINE-neg with different combinations of tone strength, preference type, and expression form. Here, Pos & Neg denotes the expression of both like and dislike, while Neg Only denotes the expression of dislike only. Each subset contains 1,000 instances.

The statistics are averaged over the full Generation evaluation set, consisting of all 756 CUPID instances and all 9,000 PrefEval instances across the 25-, 150-, and 300-turn settings, for a total of 9,756 instances. All token counts are computed using the Qwen3-4B-Instruct-2507 tokenizer. Let H denote the interaction history, q the current request, p the inferred preference, and y the final response. As described in Section 3.1, we compare three methods: **Full-History**, **Ours**, and **Two-Stage Prompting**.

We use Qwen3-4B-Instruct-2507 as the downstream LLM for all three methods to ensure a controlled comparison. The average token lengths of H and q are 53,692 and 21, respectively. Full-history uses a 41-token prompt, while Two-stage and Ours use 310 tokens for preference inference and 61 tokens for response generation.

Table 11 shows that PREFINFER’s accuracy gains do not require substantially more processed tokens: its total token footprint is comparable to those of the full-history and two-stage pipelines.

Subset	Count	Max	Avg	P95
PREFMINE-10K	10000	8190	4877	7283
Long-64k	500	65484	55572	64785
Long-128k	500	130710	105700	124296
PREFMINE-neg	4000	8192	6739	8056

Table 8: Statistics of the four subsets in PREFMINE-15K dataset: number of instances, maximum length, average length, and 95th-percentile length, measured in total tokens. The length statistics of current requests and target preferences are reported in Table 9.

Text Field	Avg.	P95	Max
Current request	31.21	47	194
Preference	19.46	32	53

Table 9: Length statistics of current requests and preferences in PREFMINE-15K, measured in total tokens.

Table 10: Stability of PREFINFER across three independent inference runs on the full CUPID evaluation set.

Run	Precision	Recall	F1
Run 1	68.12	67.52	67.82
Run 2	67.35	66.98	67.16
Run 3	68.12	67.68	67.90
Mean	67.86	67.39	67.63
SD	0.44	0.37	0.40
Range	0.77	0.70	0.74

Method	# Calls	Inference		Generation		Total
		Input	Output	Input	Output	
Full-history	1	N/A	N/A	53,753	382	54,135
Two-stage	2	54,022	23	105	390	54,541
Ours	2	54,022	14	96	384	54,517

Table 11: End-to-end model calls and token usage for full-history, two-stage prompting, and our method. Token counts are averaged over the evaluated instances. Because the component-wise token counts and the total token counts are averaged and rounded independently, the sum of the displayed components may differ from the reported total by one token.

Criterion	Judge 1	Judge 2	Judge 3	Avg.	Una.	Maj.
Grounded in History	96.0%	86.0%	99.0%	93.7%	86.0%	97.0%
Relevant to Request	99.0%	91.0%	96.0%	95.3%	89.0%	97.0%
No Unsupported/Overly Specific Details	91.0%	86.0%	94.0%	90.3%	82.0%	93.0%
Overall Pass	91.0%	84.0%	94.0%	89.7%	81.0%	92.0%

Table 12: Data quality evaluation on 100 samples randomly sampled from PREFMINE-15K. Judge 1, Judge 2, and Judge 3 denote DeepSeek-V4-Pro, GPT-4o and Gemini-3.7-Flash, respectively. Avg. denotes the average pass rate across three models, Una. denotes the rate at which all three models give the same label, and Maj. denotes the pass rate under majority voting.

```

JUDGE_SYSTEM_PROMPT = """You are a strict data quality evaluator for preference-inference training
↔ data.

Each sample contains:
- a system prompt,
- a user prompt containing an interaction history and a current request,
- an assistant answer that should infer the user's preference.

Your task is to judge whether the assistant answer is a high-quality preference inference.

Return ONLY a valid JSON object with boolean fields:
{
  "grounded_in_history": True/False,
  "relevant_to_request": True/False,
  "not_over_specific_or_hallucinated": True/False,
  "overall_pass": True/False,
  "reason": "one short sentence"
}

Criteria:
1. grounded_in_history:
  True if the inferred preference is supported by the interaction history.
  False if it is mostly invented or unsupported.

2. relevant_to_request:
  True if the inferred preference is related to the current request.
  False only if it is unrelated to the current request.
  Do not require the preference to be highly useful, necessary, or directly actionable.

3. not_over_specific_or_hallucinated:
  True if the answer avoids unsupported details, sensitive claims, or overly specific assumptions.
  False if it invents specific facts not present in the history.

4. overall_pass:
  True only if the sample is suitable for training a preference inference model.
  Usually this should be true only when all three previous criteria are true.

Be strict but fair.
"""

def build_judge_prompt(system_prompt, user_prompt, assistant_answer):
    return f"""Please evaluate the following preference-inference training sample.

[System Prompt]

{system_prompt}

[User Prompt]

{user_prompt}

[Assistant Answer]

{assistant_answer}

Return only the JSON judgment.
"""

judge_prompt = build_judge_prompt(
    system_prompt,
    user_prompt,
    assistant_answer,
)

messages = [
    {"role": "system", "content": JUDGE_SYSTEM_PROMPT},
    {"role": "user", "content": judge_prompt},
]

```

Table 13: Prompt used by the three LLM judges for data quality evaluation.

Require: interaction history set $\mathcal{H}$, WildChat-4.8M dialogue set $\mathcal{D}$

Ensure: long-context instance set $\mathcal{X}$ with 500 instances

```

1:  $\mathcal{X} \leftarrow \emptyset$ 
2: Construct 500 base instances according to PREFMINE-neg (§ 2.4), with each instance containing at
   most 8192 tokens
3: for each base instance  $x$  do
4:   Sample token_len  $\leftarrow$  UniformInt(45000, 65000)
5:   while TokenLen( $x$ ) < token_len do
6:     Randomly sample a session  $s$  from the interaction history in  $x$ 
7:     Randomly sample a dialogue  $d$  from  $\mathcal{D}$ 
8:     for each turn  $t$  in  $d$  do
9:       if TokenLen( $x \oplus (s \leftarrow s \oplus t)$ )  $\geq$  token_len then
10:        Append turn  $t$  to session  $s$  in  $x$ 
11:        break
12:       else
13:        Append turn  $t$  to session  $s$  in  $x$ 
14:       end if
15:     end for
16:   end while
17:    $\mathcal{X} \leftarrow \mathcal{X} \cup \{x\}$ 
18: end for
19: return  $\mathcal{X}$ 

```

Table 14: Construction of long-context instances

Hyperparameter	Value	Hyperparameter	Value
Number of epochs	2	LoRA alpha	16
disable shuffling	True	LoRA dropout	0.05
Per-device train batch size	1	LoRA target modules	All
Gradient accumulation steps	2	Precision	BF16
Learning rate	5.0×10^{-6}	Gradient checkpointing	True
LR scheduler type	Cosine	Flash attention	FlashAttention-2
Warmup ratio	0.03	DeepSpeed	ZeRO-2
Max gradient norm	1.0	Liger kernel	True
Maximum sequence length	131072	Use v1 kernels	True
LoRA rank (r)	8		

Table 15: Training hyperparameters used for SFT LoRA fine-tuning Qwen3-4B-Instruct-2507 on 4 GPUs.

Field	Value
Dimension	Content Specification
Description	The user’s personalized requirements regarding what specific content the response should contain, rather than how that content is presented. This includes preferred topics, types of examples, information emphasis, exclusions, and other substantive choices. When this dimension is used, the generated preference should specify: (1) the topics or types of information the user wants included; (2) the concrete characteristics of examples or cases the user prefers; (3) what content should be avoided or excluded; and (4) constraints on <i>what is said</i> rather than <i>how it is said</i> . Example preferences include: “The answer must include cases from modern Chinese history; avoid examples from Western history.” and “Code examples must be in Python; do not use Java or C++.”

Table 16: Content-specific preference dimension used in preference generation.

Dimension	Description
Detail Level	Level of detail in the response
Depth	How fundamental or principle-based the explanation is
Breadth	Range of perspectives or coverage scope
Abstraction	Ratio of theoretical concepts to concrete examples
Timeliness	Inclusion of historical context or future trends
Structure	Organization format (e.g., list, outline, paragraph)
Visualization	Use of tables, code blocks, formulas, etc.
Sequencing	Logical order (e.g., chronological, by importance)
Modularity	Clear segmentation (e.g., definition–principle–case)
Rigor	Provision of evidence or reasoning process
Stance	Neutral objectivity vs. explicit inclination
Criticality	Examination of premises or assumptions
Solution Orientation	Idealized vs. practical solutions
Formality	Level of technical terminology used
Tone	Formal, casual, humorous, etc.
Cultural Adaptation	Use of localized expressions or context
Perspective	First-, second-, or third-person narrative
Guidance Style	Direct answer vs. Socratic questioning
Cognitive Scaffolding	Gradual progression from simple to complex
Redundancy Tolerance	Acceptance of reasonable repetition
Open-endedness	Active invitation for further questions
Safety Boundary	Caution on sensitive topics
Value Orientation	Reflection of specific cultural or ethical positions
Ethical Explicitness	Explicit labeling of ethical issues
Political Correctness	Caution in identity-related expressions
Actionability	Provision of concrete action steps
Tool Integration	Recommendation of specific tools or commands
Context Fit	Tailoring to the user’s specific situation
Cost Awareness	Analysis of time or economic costs
Role Perception	The LLM’s self-positioning (e.g., expert, assistant)
Error Tolerance	Tolerance for uncertainty or mistakes
Source Expectation	Requirement for information sources

Table 17: Other preference dimensions used in preference generation.

Ours Classification Prompt

my query : {question}
my preference : {predict_preference}
I’m trying to decide on this and here are 4 options for my query:
{formatted_options}
Now, I’d like you to pick one of them as your top recommendation for me.
Important instructions for your response:

1. Choose only one option (A, B, C, or D) that best matches my preferences.
2. Your answer must be one of these options.
3. Don’t say things like “I can’t choose” or suggest alternatives not listed.
4. Answer example: <choice>B</choice>. Give me your answer in this exact format, without any additional explanation:
<choice>[A/B/C/D]</choice>

Table 18: The prompt template of our method for Classification.

```

REMINDER = "\nIn your response, please ensure that you take into account our earlier discussion, and
↪ provide an answer that is consistent with my preference."

messages = [
    {"role": "user", "content": f"My preference : {preference}"},
    {"role": "assistant", "content": "Ok, I got it."},
    {"role": "user", "content": f"{question} {REMINDER}"}
]

```

Table 19: The message-based prompt template of our method for Generation.

```

system_prompt: |-
    ## Your Objective

    You will be provided with the {k} most relevant exchanges retrieved from previous interaction sessions between a user and a
    conversational AI assistant. These exchanges are presented in chronological order, with the most recent exchange at the end.

    Your task is to infer the implicit preference that the user will consider in their current interaction with the AI assistant.
    Preferences are guidelines, requirements, constraints, or principles that the AI assistant should align with in order to
    satisfy the user in a specific context. You should infer implicit preferences that are not mentioned in the current request
    by analyzing these retrieved exchanges.

    Examples of Preferences:
    - "Responsibilities should be communicated with a stern and direct tone."
    - "Completely thorough and simplified resources must be prepared for even the most basic topics."
    - "A presentation should be built around an example, which should be universally understandable and inclusive."
    - "Inspection protocols for equipment must be prominently featured in any related instructions or manuals."

user_prompt: |-
    ## Top-{k} Most Relevant Exchanges from Previous Interactions

    {top_k_utterances}

    ---

    ## Current Interaction

    ### User

    {curr_request}

    ---

    ## Your Task

    Describe the user's most likely preference for the current interaction in a single sentence with at most 30 words. Focus on the
    most likely preference based on the retrieved exchanges, as you will be penalized for any incorrect details.

    Return your response in the format given below.

    ### Analysis

    <Describe your analysis of the top-{k} most relevant exchanges>

    ### Most Likely Preference

    <Describe the user's most likely preference for the current interaction as a single sentence with at most 30 words>

```

Table 20: Prompt used for Direct Inference with the retrieval method. It follows the CUPID inference template, with the full interaction history replaced by the top- k retrieved utterances.

Retrieval method prompt

Before answering my question, please consider the following context from our previous conversations. These are the {k} most relevant exchanges that we had previously, which may contain information about my preferences or prior discussions related to my query:

#Start of Context#

exchange 1. {retrieved_exchange_1}

exchange 2. {retrieved_exchange_2}

...

exchange {k}. {retrieved_exchange_k}

#End of Context#

Please use this context to inform your answer and adhere to any preferences I’ve expressed that are relevant to the current query. Note that not all contexts are useful for answering my question and there may be no context that is useful. Now, please address my question:

{question}

Table 21: Retrieval method prompt prefix used by the retrieval method for both Classification and Generation, adapted from the RAG prompt in PrefEval. For Classification, {question} includes current user request and option candidates; for Generation, {question} is the current user request.

Model	All			Consistent		Contrastive		Changing	
	P	R	F1	P	R	P	R	P	R
Qwen3-4B-Instruct-2507	41.47	54.15	46.97	40.28	52.31	37.67	50.07	46.39	60.08
DeepSeek-V3.2	<u>47.81</u>	<u>64.33</u>	<u>54.85</u>	<u>50.21</u>	<u>68.09</u>	46.28	61.85	<u>46.92</u>	<u>63.05</u>
Qwen3.5-35B-A3B	46.70	63.77	53.92	48.40	65.43	<u>48.69</u>	<u>65.85</u>	43.09	60.01
PREFINFER (ours)	68.12	67.52	67.82	71.56	70.55	67.14	67.46	65.84	64.53
Retrieval	37.67	47.67	41.28	37.98	47.05	34.76	42.85	40.25	47.11

Table 22: Direct Inference results on CUPID and its three subsets. We report precision (P), recall (R), and F1 on the full set. The highest scores are in **bold** and the second highest are underlined.

Model	All	Consistent	Contrastive	Changing
Qwen3	7.93	7.82	7.36	8.61
+ Qwen3	7.63	7.40	7.05	8.44
+ Ours	8.58	8.88	8.25	8.60
DeepSeek	7.68	7.53	7.19	8.32
+ DeepSeek	7.98	8.00	7.45	8.47
+ Ours	<u>8.50</u>	8.64	8.13	<u>8.73</u>
Qwen3.5	7.47	7.35	6.90	8.15
+ Qwen3.5	7.91	8.05	7.66	8.01
+ Ours	8.58	<u>8.81</u>	<u>8.20</u>	8.74
Retrieval	7.50	7.39	6.80	8.30
PersonaVLM	5.54	5.37	5.10	6.14
PPOpt	7.12	6.52	6.77	8.07
SeCom	7.03	6.92	6.65	7.51

Table 23: Generation results on CUPID and its three subsets. Ours, Qwen3, DeepSeek, and Qwen3.5 denote PREFINFER, Qwen3-4B-Instruct-2507, DeepSeek-V3.2, and Qwen3.5-35B-A3B. Best results are shown in **bold** and second-best results are underlined.

Model	25 (9k)				150 (54k)				300 (103k)			
	Explicit	Choice	Persona	All	Explicit	Choice	Persona	All	Explicit	Choice	Persona	All
Qwen3	74.1	63.9	56.5	64.83	57.8	52.8	51.1	53.90	49.0	49.2	48.7	48.97
+ Qwen3	73.5	59.6	51.8	61.63	51.3	50.6	44.5	48.80	42.5	42.8	42.6	42.63
+ Ours	96.8	91.3	93.7	93.93	96.9	89.3	91.3	92.50	96.6	85.9	85.4	89.30
DeepSeek	95.0	85.0	83.4	87.80	76.7	71.8	57.0	68.50	58.5	57.5	48.1	54.70
+ DeepSeek	90.8	87.7	86.9	88.47	89.2	89.8	85.3	88.10	87.8	87.3	81.6	85.57
+ Ours	97.7	<u>91.9</u>	95.9	95.17	98.0	90.1	93.2	93.77	97.7	86.9	87.3	90.63
Qwen3.5	95.1	82.5	76.3	84.63	86.4	67.3	60.1	71.27	80.0	67.6	63.2	70.27
+ Qwen3.5	88.9	78.6	67.8	78.43	79.8	70.3	54.8	68.30	77.1	72.1	54.4	67.87
+ Ours	<u>97.3</u>	92.3	<u>95.6</u>	<u>95.07</u>	<u>97.4</u>	<u>89.2</u>	<u>92.9</u>	<u>93.17</u>	<u>97.2</u>	<u>85.9</u>	<u>86.0</u>	<u>89.70</u>
Retrieval	92.5	77.7	85.4	85.20	82.0	66.2	78.1	75.43	79.8	64.4	74.9	73.03
PersonaVLM	78.6	68.9	72.9	73.47	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PPOpt	74.2	52.6	46.3	57.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SeCom	82.4	74.4	56.3	71.03	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 24: Results on PrefEval Classification under different lengths of interaction history (in distracting turns and tokens). For each setting, we report results on Explicit Preference (explicit), Implicit Choice-Based (choice), Implicit Persona-Driven (persona), and All. Ours, Qwen3, DeepSeek, and Qwen3.5 denote PREFINFER, Qwen3-4B-Instruct-2507, DeepSeek-V3.2, and Qwen3.5-35B-A3B.

Model	25 (9k)				150 (54k)				300 (103k)			
	Explicit	Choice	Persona	All	Explicit	Choice	Persona	All	Explicit	Choice	Persona	All
Qwen3	84.3	69.5	52.8	68.87	47.2	38.2	28.5	37.97	30.8	31.2	28.6	30.20
+ Qwen3	63.7	49.3	38.5	50.50	36.0	32.5	26.4	31.63	21.0	24.0	18.7	21.23
+ Ours	95.5	87.3	90.1	90.97	95.0	82.7	85.4	87.70	94.5	79.0	77.5	83.67
DeepSeek	<u>98.1</u>	<u>91.4</u>	88.9	92.80	93.2	81.9	61.7	78.93	70.4	52.9	36.9	53.40
+ DeepSeek	89.2	84.7	82.0	85.30	87.9	87.8	82.2	85.97	85.4	82.5	76.9	81.60
+ Ours	97.7	88.1	<u>92.4</u>	92.73	98.3	84.3	<u>88.5</u>	<u>90.37</u>	<u>97.9</u>	<u>81.5</u>	80.9	86.77
Qwen3.5	98.3	93.4	88.9	<u>93.53</u>	96.5	86.8	74.4	85.90	94.2	86.5	78.4	86.37
+ Qwen3.5	87.1	73.1	58.9	73.03	72.8	63.3	43.3	59.80	70.5	65.8	41.4	59.23
+ Ours	98.0	90.4	93.2	93.87	<u>98.0</u>	<u>85.8</u>	89.0	90.93	98.1	81.3	81.2	86.87
Retrieval	93.2	80.8	88.5	87.50	80.6	62.3	77.9	73.60	75.8	58.4	75.2	69.80
PersonaVLM	61.4	46.7	46.5	51.53	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
PPOpt	60.4	29.2	22.1	37.23	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SeCom	76.5	61.3	32.3	56.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 25: Results on PrefEval Generation under different lengths of interaction history (in distracting turns and tokens). For each setting, we report results on Explicit Preference (explicit), Implicit Choice-Based (choice), Implicit Persona-Driven (persona), and All. Ours, Qwen3, DeepSeek, and Qwen3.5 denote PREFINFER, Qwen3-4B-Instruct-2507, DeepSeek-V3.2, and Qwen3.5-35B-A3B.