Adaptive Preference Arithmetic: Modeling Dynamic Preference Strengths for LLM Agent Personalization

Hongyi Nie
Northwestern Polytechnical University
hy_nie@mail.nwpu.edu.cn

Yaqing Wang*
Beijing Institute of Mathematical Sciences and Applications
wangyaqing@bimsa.cn

Mingyang Zhou
Tsinghua University
zhou-my24@mails.tsinghua.edu.cn

Feiyang Pan
Tsinghua University
pfy824@gmail.com

Quanming Yao
Tsinghua University
yaoaa@tsinghua.edu.cn

Zhen Wang*
Northwestern Polytechnical University
w-zhen@nwpu.edu.cn

Abstract

As large language models (LLMs) are increasingly used as personalized user assistants, effectively adapting to users' evolving preferences is critical for delivering high-quality personalized responses. While user preferences are often stable in content, their relative strengths shift over time due to changing goals and contexts. Therefore, *modeling these dynamic preference strengths can enable finer-grained personalization*. However, current methods face two major challenges: (i) limited user feedback makes it difficult to estimate preference strengths accurately, and (ii) natural language ambiguity limits the controllability of preference-guided generation. To address these issues, we propose **AdaPA-Agent**, a LLM-agent personalization framework that models dynamic preference strengths via *Adaptive Preference Arithmetic*. First, instead of requiring additional user feedback, AdaPA-Agent employs an alignment-based strength estimation module to estimate the strength of user preferences from the existing user-agent interaction. Then, it guides controllable personalized generation by linearly combining next-token distributions, weighted by the estimated strengths of individual preferences. Experiments on two personalization tasks-conversational recommendation and personalized web interaction-demonstrate that AdaPA-Agent better aligning with users' changing intents, and has achieved over 18.9% and 14.2% improvements compared to ReAct, the widely-used agent framework.

1 Introduction

Agents powered by large language models (LLMs) [1, 2] are increasingly being utilized as user assistants [3], helping individuals with tasks such as information retrieval and decision-making. Given that user needs are highly diverse and personalized, they often do not want assistants to provide generic responses. Instead, users expect these user assistants to understand their unique needs better and provide personalized services [4], such as travel planning [5], and shopping recommendations

*Corresponding authors

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

[6, 7]. Therefore, enhancing the ability of LLM agents to model users' personalized needs has become a key focus of current research [3].

Since user preferences fundamentally shape their personalized needs, recent approaches improve personalization of LLM agents by modeling and leveraging these preferences [8]. These methods can be divided into two categories: fine-tuning and training-free methods. Fine-tuning [9, 10] involves adjusting the model's parameters to fit personal preferences. However, this approach requires gathering large amounts of individual data and model training, making it both resource-intensive and costly. In contrast, training-free methods such as retrieval-augmented generation (RAG) and prompt engineering provide more flexible alternatives. RAG [8, 11] enables agents to retrieve relevant preference information in real-time, adapting responses without retraining the model. Prompt engineering [12, 13, 14] inserts the prompts of preference content into prompt templates, which can guide the model toward personalized outputs without modifying its underlying parameters.

In real-world scenarios, users often hold multiple coexisting preferences–such as favoring both healthy meals and junk food–but the influence of each preference on user's decision-making is not equal and can vary over time [15]. We term this influence as *preference strength*. As shown in Figure 1, without modeling this dynamic variation in preference strength, LLM agents may produce responses that fail to align with the user's current intent.

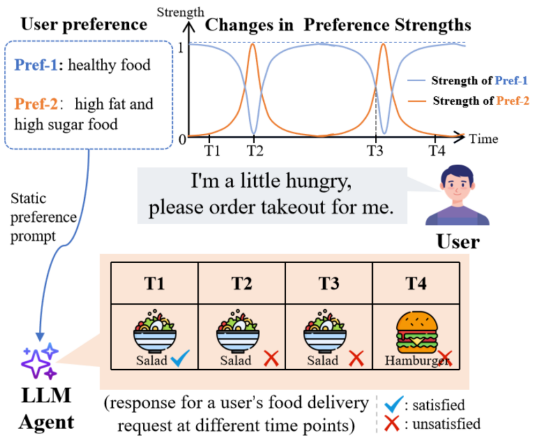

Figure 1: Illustration of an agent responding to a user's food delivery request over time. The user's needs are shaped by two preferences (Pref-1 and Pref-2), whose relative importance changes. To respond effectively, the agent must recognize and adapt to these shifts; otherwise, it may fail to meet expectations.

Although, existing methods make great progress in personalization, they still face two key challenges of modeling the dynamic preference strengths for LLM agents: (i) *Accurately estimating preference strengths without relying on explicit user feedback*. Existing methods such as ReAct [16] and Reflexion [17] often require user feedback to help adjust the strengths of preferences. Yet in real-world scenarios, user feedback is often limited in number, and frequent requests for feedback will disrupt normal interactions and significantly degrade user experience. (ii) *Effectively utilizing estimated strengths to guide content generation*. A common strategy is to input the preference strengths as text prompts to LLMs. However, this prompt-based method is fundamentally limited by the inherent ambiguity of natural language [18, 19], which may lead to misaligned personalized responses.

To address these challenges, we propose AdaPA-Agent, a framework that models and applies dynamic preference strengths through *Adaptive Preference Arithmetic*. AdaPA-Agent comprises two key components. First, to estimate preference strengths without explicit user feedback (Challenge (i)), we introduce **Alignment-Based Strength Estimation**, which combines *dual-side augmentation* with an *LLM-based alignment scorer* to measure how well each preference aligns with the current user-agent interaction. Second, to utilize these estimated strengths in generation (Challenge (ii)), we propose **Controllable Personalized Generation**. This component modulates the output distribution of an LLM by *linearly combining next-token distributions* conditioned on individual preferences and weighted by their inferred strengths. We validate AdaPA-Agent on two personalized agent tasks: *conversational recommendation* and *personalized web interaction*. Experimental results show that AdaPA-Agent consistently improves personalization quality over strong baselines, enabling LLM agents to generate responses that better reflect users' evolving intents and priorities.

Our key contributions can be summarized as follows:

- We identify two key challenges of modeling dynamic preference strengths for LLM-agent personalization: (i) accurately estimating preference strengths without relying on explicit user feedback, and (ii) effectively utilizing these estimated strengths to guide content generation.

- To address these challenges, we propose **AdaPA-Agent** that models dynamic preference strengths for LLM agents through *Adaptive Preference Arithmetic*. AdaPA-Agent introduces two novel components: *Alignment-Based Strength Estimation* for estimating preference strengths without explicit feedback, and *Controllable Personalized Generation*, which controls the personalized generation process by linearly combining next-token distributions of LLM based on the estimated strengths.
- We validate AdaPA-Agent on conversational recommendation and personalized web interaction tasks, achieving over 18.9% and 14.2% improvements compared to widely-used agent framework ReAct, respectively, demonstrating its effectiveness in adapting to users' dynamic preferences.

2 Related Work

2.1 LLM Agent

The emergence of LLMs has transformed autonomous agents, enabling them to perform tasks with human-like intelligence through powerful reasoning and knowledge capabilities. These agents typically feature modules for profiling [4], memory [20], planning [21], and action [22]. Profiling defines the agents role and personality [23, 24], while memory enables learning and adaptation [20]. Planning is enhanced by Chain-of-Thought (CoT) [25] and Tree of Thoughts (ToT) [26], which break down complex tasks using sequential or tree-based reasoning. Action modules execute plans by combining internal logic with external tools [27, 22]. Integrated reasoning and acting approaches, like ReAct [16], enable dynamic decision-making, while Reflexion [17] improves adaptability via self-reflection and feedback.

LLM agents are now widely used in domains like marketing [28], and software development [29]. As these agents become more capable, user demands have shifted from seeking general responses to expecting interactions that are more aligned with their personalized needs. Enhancing the ability of agents to provide tailored and personalized experiences has thus become a key focus of current research.

2.2 LLM Personalization

Real-world applications like personalized web interaction [27] and conversational recommendation [30] demand flexible models that adapt to evolving user needs. While traditional methods like collaborative filtering [31] are common, they lack semantic understanding and rich interactions. With LLMs, there's a shift toward language-based agents, offering more personalized and responsive experiences. LLM personalization mainly follows two approaches: fine-tuning and training-free methods. Fine-tuning [9, 10, 32] adjusts model parameters using user-specific data for improved alignment, though it requires significant resources. Training-free approaches like RAG and prompt engineering offer lightweight alternatives. RAG [8, 11] retrieves external data in real time, while prompt engineering [12, 13, 14, 33] adapts prompts to personalize outputs without modifying the model. Recent works further explore controllable text generation guided by user preferences. For instance, OPAD [34] and CoS [35] propose controllable generation methods that rely on explicit preference instructions to guide LLM outputs. AMPLe [36] infers preferences through multiple rounds of explicit user feedback.

However, current methods still struggle with modeling preference strengths and controlling generation when user feedback is sparse or inconsistent. Our work addresses these gaps by improving preference modeling and guiding LLM agents to produce more personalized outputs.

3 Target Formulation

This paper focuses on personalization tasks where LLM-based agents generate responses tailored to user preferences. This includes (i) *conversational recommendation*, where the agent interacts with users to refine intent and recommend suitable items, and (ii) *personalized web interaction*, where the agent performs context-aware actions such as search or comment generation based on inferred preferences. Here, we formulate the personalization task of LLM agent as follows:

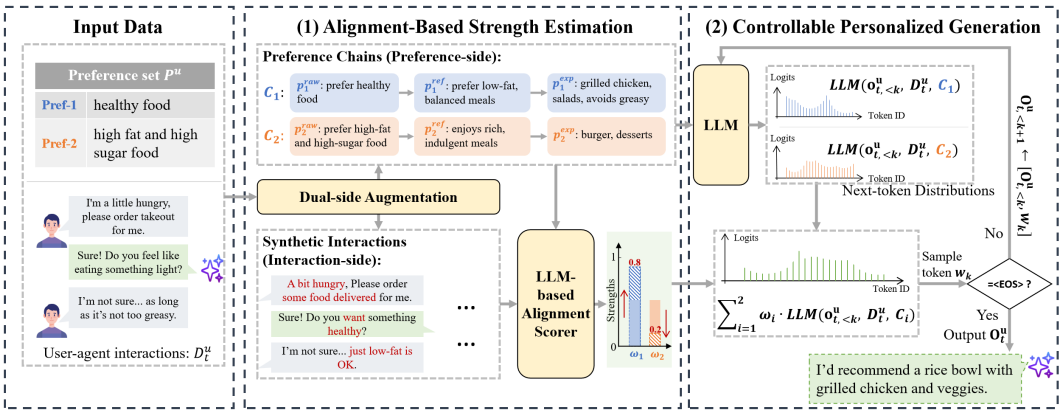

Figure 2: Overall framework of AdaPA-Agent, which includes two components: (1) **Alignment-Based Strength Estimation**, which computes alignment scores between structured preference chains and augmented user-agent interactions to infer the relative importance of preferences. (2) **Controllable Personalized Generation**, which combines next-token distributions from preference-conditioned LLM outputs, weighted by their estimated strengths. Our method enables the agent to generate responses that reflect the user's evolving priorities without requiring explicit feedback.

Definition 3.1 ***Personalization Task of LLM Agent [27, 6].*** *Given a user u, his user-agent interaction and intent at current time t is $\mathcal{D}_t^u$ and $\mathcal{I}_t^u$, and historical interactions is $\mathcal{H}^u = \{\mathcal{D}_i^u\}_{i=1}^{t-1}$. The intent $\mathcal{I}_t^u$ is influenced by the user's preference set $\mathcal{P}^u = \{p_i^u\}_{i=1}^M$, where M is the number of preferences. $\mathcal{P}^u$ can be extracted from $\mathcal{H}^u$, and each preference p_i^u has a corresponding strength ω_i. The goal of the LLM agent is to infer the preference information from user-agent interactions and use it to generate a response $\mathcal{O}_t^u$ that aligns with the user's intent $\mathcal{I}_t^u$.*

4 Methodology

Existing personalization methods often encode user preferences as static prompts or memories, overlooking how their *relative strength* evolves over time. However, modeling dynamic preference strength is challenging due to limited user feedback and the ambiguity of natural language. To address this, we propose **AdaPA-Agent**, a LLM-agent personalization framework that captures preference dynamics via *Adaptive Preference Arithmetic*. As shown in Figure 2, it includes: (1) **Alignment-Based Strength Estimation** (Section 4.1), and (2) **Controllable Personalized Generation** (Section 4.2). The first component estimates preference strengths adaptively without extra feedback through a dual-side augmentation module and an LLM-based alignment scorer. These strengths are then used in generation by combining next-token distributions conditioned on preferences in the second component. This allows the agent to adapt responses continuously based on updated preference weights. The following sections detail each component.

4.1 Alignment-Based Strength Estimation

Finding an observed signal that correlate the strength of user preferences is important for estimation without user feedback. Our key insight is that *the stronger a preference, the more frequently it manifests in user-agent interactions*. Based on the insight, we reformulate strength estimation as an *alignment* problem: for each preference, we measure how well it semantically aligns with the user-agent interaction and treat this alignment score as a proxy for strength. A higher alignment score indicates that the interaction provides stronger support for the preference, hence implying a larger strength for that preference.

However, user expressions are often vague and diverse in practice, which makes alignment scoring unreliable when using naïve methods such as keyword overlap or static embedding similarity. To address this challenge, we incorporate two key techniques: (i) dual-side augmentation, which enriches both the preference (via preference chains) and interaction sides (via conversation-level paraphrases) with more informative and semantically diverse representations; and (ii) an LLM-based

scoring module, which assesses the alignment between each augmented preference-interaction pair in a context-aware and interpretable manner.

4.1.1 Dual-Side Augmentation

Preference-side augmentation. We first extract the set of preferences $\mathcal{P}^u$ from historical interactions $\mathcal{H}^u$. However, user preferences often emerge at multiple semantic levels, i.e., from abstract self-descriptions to concrete behavioral cues. To capture this spectrum, we construct a three-stage *preference chain* for each preference p_i^u using a CoT prompt:

$$\mathcal{C}_i \;=\; \mathrm{CoT}(p_i^u) \;=\; \big[p_i^{\mathrm{raw}} \;\rightarrow\; p_i^{\mathrm{ref}} \;\rightarrow\; p_i^{\mathrm{exp}}\big], \tag{1}$$

where p_i^{raw} is the original description of the preference extracted from p_i^u, p_i^{ref} is a context-aware reformulation refining p_i^{raw}, and p_i^{exp} enumerates concrete items or behaviors exemplifying p_i^{raw} and p_i^{ref}. This three-level structure supplies coarse-to-fine preference information for the subsequent alignment scoring. The complete prompt is provided in Appendix A, Prompt 1.

Interaction-side augmentation. Since user intent is typically implicit and linguistically diverse, computing an alignment score from a single interaction can be highly biased. To reduce this variance, we enrich the interaction side by paraphrasing the original user-agent interaction $\mathcal{D}_t^u$ into multiple semantically equivalent variants:

$$\mathbf{g}(\mathcal{D}_t^u) \;=\; \big\{\mathcal{D}_{t_1}^u,\, \mathcal{D}_{t_2}^u,\, \ldots,\, \mathcal{D}_{t_K}^u\big\}, \tag{2}$$

where we employ an LLM as a generation function $\mathbf{g}(\cdot)$ (Prompt 2, Appendix A), each synthetic interaction $\mathcal{D}_{t_k}^u$ ($k = 1, 2, \ldots, K$) preserves the same number of turns as $\mathcal{D}_t^u$ but rewrites utterances with alternative lexical and syntactic choices to enrich the expression of user intent. The augmented set $\mathcal{A}^u = \mathbf{g}(\mathcal{D}_t^u) \cup \mathcal{D}_t^u$ forms a more complete pool of interaction variants for alignment scoring.

4.1.2 LLM-Based Alignment Scoring

Compared to static embeddings or keyword matching, LLMs offer a richer understanding of semantics, enabling context-aware and fine-grained alignment. Therefore, we employ an LLM to assess the alignment score of each preference-interaction pair. Specifically, given the structured preference chains $\mathcal{C}_i$ and the augmented interaction set $\mathcal{A}^u$, we compute a alignment score s_i for each preference p_i^u as:

$$s_i \;=\; \frac{1}{|\mathcal{A}^u|} \sum_{\mathcal{D} \in \mathcal{A}^u} \mathbf{f}\big(\mathcal{C}_i,\, \mathcal{D}\big), \tag{3}$$

where $\mathbf{f}(\cdot, \cdot)$ is an LLM-based scorer returning a fine-grained alignment score (1-10) for a preference-interaction pair (Prompt 3, Appendix A). We then normalize the scores to obtain relative strength of each preference:

$$\omega_i \;=\; \frac{s_i}{\sum_{j=1}^{M} s_j}. \tag{4}$$

Through this process, we can obtain and adaptively update relative strength of each preference without requiring additional user feedback.

4.2 Controllable Personalized Generation

Once the set of preference strengths $\{\omega_i\}_{i=1}^M$ is obtained, the remaining question is how to use these weights to steer an LLM so that its response continuously reflects the user's current intent. Given the inherent ambiguity in natural language prompting, directly inserting numeric weights into a text prompt is unreliable [18]. Recent work [19] has demonstrated that complex textual style can be effectively generated by arithmetically combining the next-token distributions of language models conditioned on different basic styles. This motivates the proposal of preference arithmetic, a technique that combines multiple preference-conditioned distributions in a formulaic way, using the dynamically estimated strengths ω_i as weights. This allows for fine-grained control over the influence of each preference p_i^u on the generated response $\mathcal{O}_t^u$.

Specifically, at each step of generating the response $\mathcal{O}_t^u$, the LLM produces a next-token distribution. Let $\mathbf{P}_{\text{opt}}(\cdot)$ be the optimal personalized next-token distribution, which is conditioned on the

user's intent $\mathcal{I}_t^u$. We model $\mathbf{P}_{\text{opt}}(\cdot)$ as a weighted combination of M individual next-token distributions, where each individual distribution $\mathbf{P}_{p_i^u}(\cdot)$ is conditioned on a specific preference p_i^u from the user's preference set $\mathcal{P}^u$. The overall next-token distribution for generating the current token w_k of the response $\mathcal{O}_t^u$, given the partially generated response prefix $\mathcal{O}_{t,<k}^u$ and the current user-agent interaction $\mathcal{D}_t^u$, is defined as:

$$\mathbf{P}_{\text{opt}}(w_k|\mathcal{O}_{t,<k}^u, \mathcal{D}_t^u, \mathcal{I}_t^u) = \sum_{i=1}^{M} \omega_i \cdot \mathbf{P}_{p_i^u}(w_k|\mathcal{O}_{t,<k}^u, \mathcal{D}_t^u, p_i^u), \tag{5}$$

where each preference-conditioned distribution $\mathbf{P}_{p_i^u}(\cdot)$ is generated by the LLM when prompted with the specific preference p_i^u in the context of the current interaction $\mathcal{D}_t^u$ and the already generated part of the response $\mathcal{O}_{t,<k}^u$. For preference p_i^u, we can use its augmented representation, the preference chain $\mathcal{C}_i$ (from Section 4.1.1), to construct a comprehensive prompt for the LLM. Thus, the i-th preference-conditioned next-token distribution is:

$$\mathbf{P}_{p_i^u}(w_k|\mathcal{O}_{t,<k}^u, \mathcal{D}_t^u, p_i^u) = \text{LLM}((\mathcal{O}_{t,<k}^u, \mathcal{D}_t^u, \mathcal{C}_i)). \tag{6}$$

This formulation enables a continuously steerable decoding process, where each preference influences generation proportionally to its weight. Equivalently, the combined distribution can be viewed as the solution to a weighted KL minimization problem with closed-form optimum [19]:

$$P^\star = \arg\min_P \sum_{i=1}^{M} \omega_i \, D_{\text{KL}}\big(P \,\|\, Q_i\big), \tag{7}$$

$$P^\star(w_k) = \text{softmax}\Big(\sum_{i=1}^{M} \omega_i \log Q_i(w_k)\Big). \tag{8}$$

Here, $P^\star(w_k)$ represents the **KL-optimal compromise** among the preference-conditioned distributions Q_i, where stronger preferences (larger ω_i) dominate while weaker yet relevant ones still contribute to a balanced, intent-aligned generation.

To generate the full response $\mathcal{O}_t^u$, tokens are sampled autoregressively. As illustrated in Algorithm 1, at each generation step k, the corresponding token w_k is sampled from the combined distribution: $w_k \sim \sum_{i=1}^{M} \omega_i \cdot \text{LLM}((\mathcal{O}_{t,<k}^u, \mathcal{D}_t^u, \mathcal{C}_i))$, and extend the context as: $\mathcal{O}_{t,<k+1}^u = [\mathcal{O}_{t,<k}^u, w_k]$. This process continues iteratively until the sampled token w_k is the end-of-sequence token (`EOS`), ensuring the generation of coherent and personalized responses.

5 Experiments

In this section, we evaluate our AdaPA-Agent[2] on two typical LLM personalized agent tasks (conversational recommendation and personalized web interaction). Additionally, we validate the effectiveness of our method design through a series of ablation experiments. In the following, we first present the experimental setup and baselines, and then provide a detailed presentation and analysis of the experimental results.

5.1 Experimental Setup

5.1.1 Baseline

The proposed AdaPA-Agent is compared with the following baselines: **ReAct** [16] combines reasoning and action generation to improve decision-making in interactive tasks, allowing models to adapt dynamically to new information and user needs. **Reflexion** [17] uses verbal reinforcement learning for self-reflection and feedback, allowing language agents to continuously refine their actions and reasoning, making them more adaptable to personalized interactions. **SimToM** [37] introduces perspective-taking to improve theory-of-mind (ToM) capabilities, helping models better understand and predict human intentions, thus tailoring responses based on individual user preferences. **RecMind** [38] leverages an LLM-powered autonomous recommender agent with a self-inspiring algorithm, enabling zero-shot personalized recommendations by considering historical information and

[2]The code is available at: `https://github.com/Sirius11311/AdaPA`.

Maximum Steps	Method	Long-Term Tasks		Short-Term Tasks		All Tasks	
		RSR	AIR	RSR	AIR	RSR	AIR
3	ReAct [16]	$30.27_{\pm 9.78}$	$2.81_{\pm 0.09}$	$30.19_{\pm 7.73}$	$2.89_{\pm 0.04}$	$30.20_{\pm 5.20}$	$2.85_{\pm 0.05}$
	Reflexion [17]	$38.45_{\pm 9.62}$	$2.74_{\pm 0.09}$	$28.69_{\pm 3.79}$	$2.85_{\pm 0.03}$	$33.80_{\pm 6.80}$	$2.79_{\pm 0.05}$
	RecMind [38]	$39.77_{\pm 9.65}$	$\underline{2.73}_{\pm 0.13}$	$32.50_{\pm 4.44}$	$2.85_{\pm 0.05}$	$36.40_{\pm 4.73}$	$2.78_{\pm 0.08}$
	InteRec [39]	$\underline{45.67}_{\pm 11.55}$	$\mathbf{2.66}_{\pm 0.16}$	$\underline{32.87}_{\pm 4.48}$	$\underline{2.76}_{\pm 0.08}$	$\underline{39.60}_{\pm 6.73}$	$\underline{2.71}_{\pm 0.10}$
	SimTom [37]	$40.22_{\pm 10.68}$	$2.78_{\pm 0.09}$	$31.19_{\pm 5.58}$	$2.87_{\pm 0.05}$	$36.01_{\pm 4.67}$	$2.82_{\pm 0.06}$
	AdaPA-Agent	$\mathbf{46.10}_{\pm 9.65}$	$\mathbf{2.66}_{\pm 0.10}$	$\mathbf{37.05}_{\pm 3.94}$	$\mathbf{2.62}_{\pm 0.06}$	$\mathbf{41.45}_{\pm 5.82}$	$\mathbf{2.64}_{\pm 0.08}$
5	ReAct [16]	$57.47_{\pm 9.53}$	$4.19_{\pm 0.19}$	$40.99_{\pm 4.59}$	$4.46_{\pm 0.09}$	$49.60_{\pm 4.60}$	$4.32_{\pm 0.14}$
	Reflexion [17]	$67.88_{\pm 7.80}$	$3.81_{\pm 0.24}$	$51.40_{\pm 6.30}$	$4.24_{\pm 0.18}$	$60.20_{\pm 4.80}$	$4.01_{\pm 0.19}$
	RecMind [38]	$66.58_{\pm 9.29}$	$3.77_{\pm 0.30}$	$58.92_{\pm 4.72}$	$4.00_{\pm 0.18}$	$62.82_{\pm 5.88}$	$3.88_{\pm 0.19}$
	InteRec [39]	$\underline{70.28}_{\pm 10.35}$	$\underline{3.67}_{\pm 0.38}$	$\underline{61.83}_{\pm 6.44}$	$\underline{3.88}_{\pm 0.15}$	$\underline{66.36}_{\pm 5.91}$	$\underline{3.77}_{\pm 0.23}$
	SimTom [37]	$66.20_{\pm 9.71}$	$3.88_{\pm 0.30}$	$52.58_{\pm 7.46}$	$4.17_{\pm 0.17}$	$59.45_{\pm 4.37}$	$4.02_{\pm 0.17}$
	AdaPA-Agent	$\mathbf{71.27}_{\pm 9.37}$	$\mathbf{3.64}_{\pm 0.30}$	$\mathbf{63.04}_{\pm 4.84}$	$\mathbf{3.83}_{\pm 0.19}$	$\mathbf{67.24}_{\pm 4.41}$	$\mathbf{3.73}_{\pm 0.23}$
7	ReAct [16]	$73.50_{\pm 7.77}$	$5.06_{\pm 0.39}$	$51.92_{\pm 3.31}$	$5.59_{\pm 0.21}$	$63.40_{\pm 4.60}$	$5.31_{\pm 0.27}$
	Reflexion [17]	$79.70_{\pm 5.74}$	$4.51_{\pm 0.43}$	$62.08_{\pm 4.86}$	$5.20_{\pm 0.22}$	$71.40_{\pm 2.73}$	$4.83_{\pm 0.27}$
	RecMind [38]	$\mathbf{81.26}_{\pm 5.38}$	$4.40_{\pm 0.43}$	$65.24_{\pm 5.20}$	$4.82_{\pm 0.26}$	$73.80_{\pm 2.93}$	$4.59_{\pm 0.24}$
	InteRec [39]	$79.64_{\pm 5.81}$	$\mathbf{4.26}_{\pm 0.47}$	$\underline{69.60}_{\pm 7.84}$	$\mathbf{4.51}_{\pm 0.27}$	$\underline{75.00}_{\pm 3.33}$	$\mathbf{4.38}_{\pm 0.17}$
	SimTom [37]	$79.98_{\pm 4.94}$	$4.44_{\pm 0.42}$	$64.07_{\pm 7.01}$	$5.07_{\pm 0.32}$	$72.60_{\pm 3.27}$	$4.73_{\pm 0.30}$
	AdaPA-Agent	$\underline{80.36}_{\pm 5.64}$	$\underline{4.41}_{\pm 0.42}$	$\mathbf{70.45}_{\pm 6.21}$	$\mathbf{4.51}_{\pm 0.30}$	$\mathbf{75.41}_{\pm 3.07}$	$\underline{4.46}_{\pm 0.36}$

Table 1: Main results of the AdaPA-Agent method and baseline models on the conversational recommendation task. The table presents recommendation successful rate (RSR) and average interaction rounds (AIR) for long-term, short-term, and overall performance, evaluated under different maximum steps (3, 5, and 7).

previous states. **InteRec** [39] integrates long-term and short-term memory mechanisms to enhance personalized recommendations, improving the understanding of user preferences and context for more relevant interactions.

5.1.2 Implementation Details

For AdaPA-Agent, we use Llama-3.1-8B-Instruct [40] as the local LLM to generate the next-token distribution. In the interaction-augmentation phase, we set $K = 4$ for the conversational recommendation task and $K = 2$ for the personalized web interaction task. In the preference arithmetic phase, we set $M = 2$ for both two tasks, i.e., we choose top 2 preferences to generate the personalized next-token distribution. In the conversational recommendation task, GPT-4o [41] serves as the user simulator, while DeepSeek V2.5 [42] supports all baselines and AdaPA-Agent. For the personalized web interaction task, GPT-4o is used consistently across all baselines and AdaPA-Agent. To reduce generation randomness, we set the LLM temperature to 0 during all evaluations.

5.2 Conversational Recommendation Task

5.2.1 Task Introduction

(1) Task Description: Conversational recommendation systems personalize suggestions through real-time interactions. Users express needs via natural language, while the agent clarifies and recommends accordingly. To provide recommendations that better align with the user's intent, the agent needs to be able to model the changes in the user's preference strength. **(2) Task Construction**: To evaluate how well our method models preference strength, we divide user movie preferences into long-term and short-term types. Based on the Reddit-Movie dataset [6], we filter 984 unique users with sufficient historical data. Then, we extract their stable preferences from historical data as long-term and select unrelated movies to define short-term preferences. A dynamic simulation environment is built using an LLM-based user simulator [43, 44], where each task randomly assigns a dominant preference type to simulate the conversation. This setup allows controlled variation in preference strength, enabling direct and systematic evaluation of our method. We introduce the setup details of task settings and user simulator in Appendix C. **(3) Evaluation Metrics**: The task is limited to T rounds. If the agent recommends the target movie within k rounds ($k \neq T$), it succeeds; otherwise, it fails. Performance is measured by: *Recommendation Successful Rate (RSR)* $=$

Method	Search		Recommendation		Review		Overall	
	F. Acc	R. Acc	F. Acc	R. Acc	F. Acc	R. Acc	F. Acc	R. Acc
Random Memory	0.974	0.640	0.296	0.018	0.996	0.442	0.745	0.357
Last Memory	0.937	0.626	0.432	0.028	**1.000**	0.442	0.782	0.357
Relevant Memory	0.928	0.622	0.492	**0.030**	**1.000**	0.443	0.800	0.356
ReAct [16]	0.903	0.605	0.560	0.027	0.996	0.444	0.815	0.350
RecMind [38]	0.981	0.645	0.226	0.017	0.990	0.442	0.721	0.359
AdaPA-Agent	**0.987**	**0.654**	**0.592**	0.027	**1.000**	**0.457**	**0.851**	**0.367**

Table 2: Main results of the AdaPA-Agent method and baseline models on the single-turn track settings of personalized web interaction. The table shows the function accuracy (F. Acc.) and response accuracy (R. Acc.) for the three web services: Search, Recommendation and Review.

Method	Search			Recommendation			Review			Overall		
	F. Acc	R. Acc	Avg. Steps	F. Acc	R. Acc	Avg. Steps	F. Acc	R. Acc	Avg. Steps	F. Acc	R. Acc	Avg. Steps
Random Memory	0.999	0.680	4.193	0.703	0.042	4.474	1.000	0.448	2.007	0.896	0.380	3.564
Last Memory	0.996	0.676	4.229	0.708	**0.045**	4.252	**1.000**	0.449	2.007	0.897	0.381	3.597
Relevant Memory	0.996	0.686	4.233	0.715	0.042	4.564	0.999	0.448	2.008	0.899	0.383	3.609
ReAct [16]	0.996	0.674	4.657	0.218	0.013	5.468	0.974	0.448	2.129	0.718	0.369	4.198
Reflexion [17]	**1.000**	0.686	5.406	0.281	0.014	6.145	0.976	0.449	2.145	0.741	0.373	4.579
RecMind [38]	0.997	0.642	6.728	0.347	0.026	6.003	0.997	0.451	2.107	0.771	0.364	4.938
InteRec [39]	0.999	0.642	3.110	0.618	0.022	3.008	**1.000**	0.447	2.001	0.867	0.362	2.706
AdaPA-Agent	0.999	**0.698**	5.352	**0.768**	0.039	3.485	**1.000**	**0.455**	2.004	**0.917**	**0.386**	3.592

Table 3: Main results of the AdaPA-Agent method and baseline models on the multi-turn track settings of personalized web interaction. The table shows the function accuracy (F. Acc.), response accuracy (R. Acc.) and average steps (Avg. Steps) for the three web services: Search, Recommendation and Review.

$\frac{N_s}{N} \times 100\%$, *Average Interaction Rounds (AIR)* $= \sum_{i=1}^{N} \frac{k_i}{T}$, where N is the total tasks, N_s the successful recommendations, and k_i the interaction rounds in task i. Higher RSR and lower AIR indicate better efficiency.

5.2.2 Results Analysis

The results in Table 1 compare AdaPA-Agent with baseline models under varying maximum steps (3, 5, and 7) in the conversational recommendation task. AdaPA-Agent consistently outperforms baseline methods in terms of RSR while maintaining competitive step efficiency. AdaPA-Agent maintains a stable and robust performance across both long-term and short-term tasks, unlike some baseline methods that exhibit imbalances. In contrast, baseline methods often show trade-offs. For instance, RecMind performs well on short-term tasks but suffers on long-term preferences, while Reflexion achieves moderate long-term results but lags on short-term tasks. These inconsistencies suggest a lack of adaptive preference modeling. AdaPA-Agent addresses this gap by estimating dynamic preference strengths and adjusting generation accordingly, leading to stable performance across diverse user intents. Although InteRec has explicitly considered the long-term preference and short-term preference, it still suffers the problem of modeling the preference strength and has a lower performance than AdaPA-Agent. Furthermore, AdaPA-Agent strikes an optimal balance between RSR and AIR, making it more efficient in real-time applications. At just 3 steps, it outperforms all baselines on overall RSR (41.45% vs. 38.60% by RecMind and 36.01% by SimTom), and achieves the lowest AIR (2.64). This demonstrates that AdaPA-Agent can effectively estimate preference strengths with minimal feedback and provide accurate recommendations even within short interactions, confirming its effectiveness in real-world conversational recommendation systems.

5.3 Personalized Web Interaction Task

5.3.1 Task Introduction

(1) Task Description: Personalized Web agents infer user preferences to enhance web services. We use the PWABench benchmark [27], which includes tasks like personalized search, recommenda-

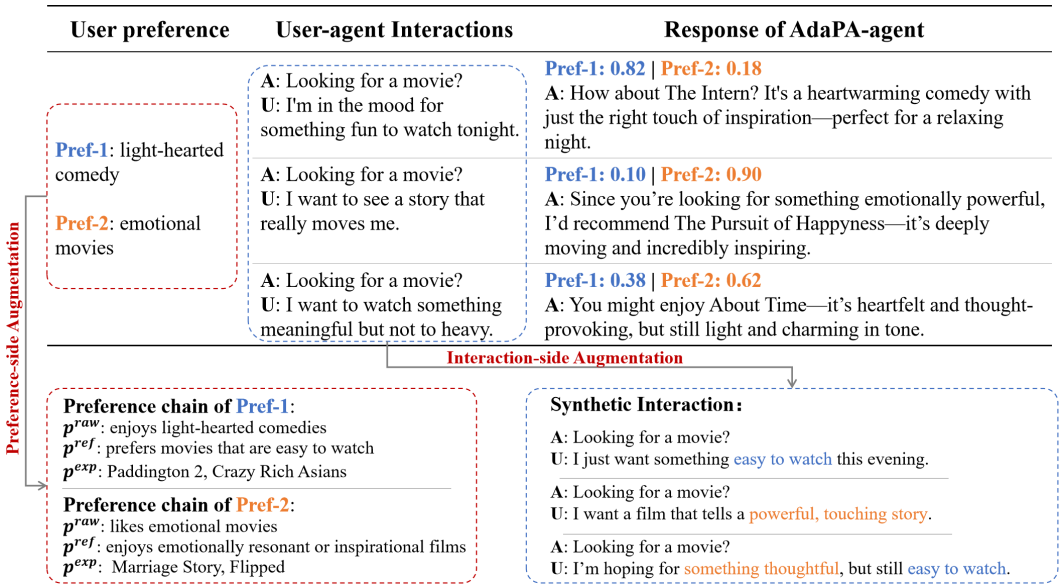

Figure 3: Case study of AdaPA-Agent in a movie recommendation scenario. The model dynamically adjusts preference strengths based on user-agent interactions and generates personalized responses.

tion, and review generation. These tasks require LLMs to select the right web function and parameters for personalized outputs. PWABench also provides three memory retrieval baselines–Random Memory, Last Memory, and Relevant Memory–that sample different historical data to assist agents. To better reflect real-world user behaviors, we do not restrict users to only two preferences as in the conversational recommendation task. Instead, each user may have multiple active preferences with diverse and dynamically shifting strengths, posing greater challenges for accurate modeling and adaptation. **(2) Evaluation Metrics**: the evaluation metrics for the system include three key components: (a) *Function Accuracy (F. Acc)*: evaluates the agent's ability to select the correct web function and parameters, scoring 1 for correct selection and 0 otherwise; (b) *Response Accuracy (R. Acc)*: for search and recommendation tasks, this metric uses the rank r of the target product within the returned product list as the performance indicator and is calculated as:

$$\text{R. Acc} = \begin{cases} 1 - \frac{r-1}{10}, & \text{if } r \leq 10 \\ 0, & \text{if } r > 10 \end{cases}; \tag{9}$$

and (c) *Average Steps (Avg. Steps)*: measures total actions needed to complete a task, where fewer steps indicate higher efficiency.

5.3.2 Results Analysis

In the personalized web interaction task, accurately understanding user intent requires distinguishing subtle differences in preference-driven behaviore.g., discerning whether a user prefers to search for a product or directly receive a recommendation. This challenge is particularly pronounced in single-turn interactions, where the agent lacks rich dialogue context. As shown in Table 2, AdaPA-Agent achieves the best overall performance across all services, reaching the highest overall function accuracy (F. Acc: 0.851) and response accuracy (R. Acc: 0.367). In particular, AdaPA-Agent significantly outperforms prior methods on recommendation-related interactions (F. Acc: 0.592), a setting where user intent is often implicit and preference strength is most influential. Compared to RecMind (0.226) and Random Memory (0.296), AdaPA-Agent improves recommendation accuracy by over 30 absolute points, showing its ability to infer the dominant preferences even with sparse interaction.

As shown in Table 3, in the multi-turn track, user feedback from multiple rounds is utilized to refine the model's personalized output, leading to performance improvements across all methods compared to the single-turn track. However, AdaPA-Agent stands out not only in achieving higher function accuracy and response accuracy than the baseline methods but also in maintaining relatively small average steps. This highlights the efficient preference modeling capabilities of AdaPA-Agent, as it

does not rely on excessive user feedback. While methods like Reflexion and ReAct, which depend on user feedback to update their understanding of user preferences, require more steps on average.

These results affirm the core advantage of AdaPA-Agent: by modeling preference strengths adaptively, it delivers both more accurate and more efficient personalized servicesespecially in settings where feedback is limited or user intent is ambiguous.

5.4 Case Study

We conduct a case study to examine how AdaPA-Agent dynamically adjusts to user preferences in a conversational movie recommendation scenario with two competing preferences: *light-hearted comedy* and *emotional movies*. As shown in Figure 3, across different user utterances, the preference weights inferred by AdaPA-Agent shift in a consistent and interpretable manner, accurately reflecting subtle changes in user intent. For instance, when the user mentions wanting a story that really moves me, the alignment score for the emotional preference sharply increases, prompting a matching recommendation. In contrast, when the user seeks something fun, the model shifts weight toward the light-hearted preference. This illustrates two key strengths of our approach: (1) the ability to capture nuanced preference signals through alignment-based estimation without explicit feedback, and (2) the controllability of generation via preference-weighted decoding, allowing the agent to produce responses that align precisely with the user's evolving priorities. This example highlights how AdaPA-Agent enables fine-grained and interpretable adaptation in real-time personalized interactions.

5.5 Additional Experiments

The appendix provides extensive ablation studies (Appendix D) that further validate the effectiveness of AdaPA-Agent in *modeling dynamic preference strengths*. These analyses demonstrate that (1) dual-side augmentation substantially enhances robustness and fine-grained preference estimation (Appendix D.1), (2) the proposed LLM-based alignment scorer more accurately captures semantic alignment between user preferences and interactions compared to embedding-based methods (Appendix D.2), and (3) the continuous preference arithmetic formulation enables more expressive and controllable generation than prompt-based methods (Appendix D.3).

6 Conclusion

We present **AdaPA-Agent**, a training-free framework for enhancing LLM-based agents through dynamic preference modeling. AdaPA-Agent addresses two key challenges in personalization: (i) estimating the relative strength of user preferences without relying on explicit user feedback, and (ii) effectively incorporating these strengths into the generation process. To solve the first challenge, we introduce *Alignment-Based Strength Estimation*, which leverages dual-side augmentation and an LLM-based alignment scorer to infer fine-grained preference weights from implicit user behaviors. To solve the second challenge, we propose *Controllable Personalized Generation*, which utilizes the estimated strengths via preference arithmetic to control the personalized generation process. Our experiments on two personalized agent tasks–conversational recommendation and personalized web agent–demonstrate that AdaPA-Agent consistently improves alignment with users' evolving intents and priorities by modeling dynamic preferences.

Acknowledgement

This work was supported in part by the National Science Fund for Distinguished Young Scholars under Grant 62025602; in part by the National Natural Science Foundation of China under Grant U22B2036, and Grant 11931015; in part by the Technological Innovation Team of Shaanxi Province (No. 2025RS-CXTD-009), the International Cooperation Project of Shaanxi Province (No. 2025GH-YBXM-017); in part by the Fok Ying-Tong Education Foundation, China, under Grant 171105; in part by the Fundamental Research Funds for the Central Universities under Grant D5000230366; in part by the Fundamental Research Funds for the Central Universities under Grant G2024WD0151, and in part by the Tencent Foundation and Xplorer Prize. Y. Wang is sponsored by Beijing Nova Program. Q. Yao is sponsored by CCF-Zhipu Large Model Innovation Fund (No. Zhipu202402).

References

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

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901, 2020.

[3] Xin Luna Dong, Seungwhan Moon, Yifan Ethan Xu, Kshitiz Malik, and Zhou Yu. Towards next-generation intelligent assistants leveraging LLM techniques. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 5792–5793, 2023.

[4] Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. Personal LLM agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459*, 2024.

[5] Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. TravelPlanner: A benchmark for real-world planning with language agents. In *International Conference on Machine Learning*, 2024.

[6] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. Large language models as zero-shot conversational recommenders. In *ACM International Conference on Information and Knowledge Management*, pages 720–730, 2023.

[7] Gangyi Zhang. User-centric conversational recommendation: Adapting the need of user with large language models. In *ACM Conference on Recommender Systems*, pages 1349–1354, 2023.

[8] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In *Annual Meeting of the Association for Computational Linguistics*, pages 7370–7392, 2024.

[9] Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. PERSONALIZED SOUPS: Personalized large language model alignment via post-hoc parameter merging. In *Advances in Neural Information Processing Systems Workshop Adaptive Foundation Models*, 2023.

[10] Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. LLMs+ Persona-Plug= Personalized LLMs. *arXiv preprint arXiv:2409.11901*, 2024.

[11] Alireza Salemi, Surya Kallumadi, and Hamed Zamani. Optimization methods for personalizing large language models through retrieval augmentation. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 752–762, 2024.

[12] Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. ONCE: Boosting content-based recommendation with both open-and closed-source large language models. In *ACM International Conference on Web Search and Data Mining*, pages 452–461, 2024.

[13] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. LLM-Rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL*, pages 583–612, 2024.

[14] Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. Towards open-world recommendation with knowledge augmentation from large language models. In *ACM Conference on Recommender Systems*, pages 12–22, 2024.

[15] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S Yu, Yading Yue, and Bin Wu. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *ACM International Conference on Information and Knowledge Management*, pages 453–462, 2015.

[16] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

[17] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 8634–8652, 2023.

[18] Simran Arora, Avanika Narayan, Mayee F Chen, Laurel Orr, Neel Guha, Kush Bhatia, Ines Chami, and Christopher Re. Ask Me Anything: A simple strategy for prompting language models. In *International Conference on Learning Representations*, 2023.

[19] Jasper Dekoninck, Marc Fischer, Luca Beurer-Kellner, and Martin Vechev. Controlled text generation via language model arithmetic. In *International Conference on Learning Representations*, 2024.

[20] Kostas Hatalis, Despina Christou, Joshua Myers, Steven Jones, Keith Lambert, Adam Amos-Binks, Zohreh Dannenhauer, and Dustin Dannenhauer. Memory matters: The need to improve long-term memory in LLM-agents. In *AAAI Symposium Series*, pages 277–280, 2023.

[21] Xixi Wu, Yifei Shen, Caihua Shan, Kaitao Song, Siwei Wang, Bohang Zhang, Jiarui Feng, Hong Cheng, Wei Chen, Yun Xiong, et al. Can graph learning improve planning in LLM-based agents? In *Advances in Neural Information Processing Systems*, 2024.

[22] Yanchu Guan, Dong Wang, Zhixuan Chu, Shiyu Wang, Feiyue Ni, Ruihua Song, and Chenyi Zhuang. Intelligent agents with LLM-based process automation. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 5018–5027, 2024.

[23] Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role play with large language models. *Nature*, 623(7987):493–498, 2023.

[24] Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-LLM: A trainable agent for role-playing. In *Conference on Empirical Methods in Natural Language Processing*, pages 13153–13187, 2023.

[25] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, pages 24824–24837, 2022.

[26] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of Thoughts: Eliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, pages 11809 –11822, 2023.

[27] Hongru Cai, Yongqi Li, Wenjie Wang, Fengbin Zhu, Xiaoyu Shen, Wenjie Li, and Tat-Seng Chua. Large language models empowered personalized web agents. In *The Web Conference*, 2024.

[28] Kausik Lakkaraju, Sara E Jones, Sai Krishna Revanth Vuruma, Vishal Pallagani, Bharath C Muppasani, and Biplav Srivastava. LLMs for financial advisement: A fairness and efficacy study in personal decision making. In *ACM International Conference on AI in Finance*, pages 100–107, 2023.

[29] Md. Ashraful Islam, Mohammed Eunus Ali, and Md Rizwan Parvez. MapCoder: Multi-agent code generation for competitive problem solving. In *Annual Meeting of the Association for Computational Linguistics*, pages 4912–4944, 2024.

[30] Jiabao Fang, Shen Gao, Pengjie Ren, Xiuying Chen, Suzan Verberne, and Zhaochun Ren. A multi-agent conversational recommender system. *arXiv preprint arXiv:2402.01135*, 2024.

[31] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *International Conference on World Wide Web*, page 173182, 2017.

[32] Xiaoyu Zhang, Ruobing Xie, Yougang Lyu, Xin Xin, Pengjie Ren, Mingfei Liang, Bo Zhang, Zhanhui Kang, Maarten de Rijke, and Zhaochun Ren. Towards empathetic conversational recommender systems. In *ACM Conference on Recommender Systems*, pages 84–93, 2024.

[33] Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning LLM agents by learning latent preference from user edits. In *Advances in Neural Information Processing Systems*, 2024.

[34] Mingye Zhu, Yi Liu, Lei Zhang, Junbo Guo, and Zhendong Mao. On-the-fly preference alignment via principle-guided decoding. In *International Conference on Learning Representations*, 2025.

[35] Jerry Zhi-Yang He, Sashrika Pandey, Mariah L. Schrum, and Anca D. Dragan. Context steering: Controllable personalization at inference time. In *International Conference on Learning Representations*, 2025.

[36] Minhyeon Oh, Seungjoon Lee, and Jungseul Ok. Comparison-based active preference learning for multi-dimensional personalization. In *Annual Meeting of the Association for Computational Linguistics*, pages 33145–33166, 2025.

[37] Alex Wilf, Sihyun Lee, Paul Pu Liang, and Louis-Philippe Morency. Think twice: Perspective-taking improves large language models' theory-of-mind capabilities. In *Annual Meeting of the Association for Computational Linguistics*, pages 8292–8308, 2024.

[38] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. RecMind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL*, pages 4351–4364, 2024.

[39] Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. Recommender AI agent: Integrating large language models for interactive recommendations. *arXiv preprint arXiv:2308.16505*, 2023.

[40] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[41] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[42] Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, et al. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024.

[43] Jafar Afzali, Aleksander Mark Drzewiecki, Krisztian Balog, and Shuo Zhang. Usersimcrs: A user simulation toolkit for evaluating conversational recommender systems. In *ACM International Conference on Web Search and Data Mining*, pages 1160–1163, 2023.

[44] Xiaolei Wang, Xinyu Tang, Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. Rethinking the evaluation for conversational recommendation in the era of large language models. In *Conference on Empirical Methods in Natural Language Processing*, pages 10052–10065, 2023.

[45] Haozhe Xu, Xiaohua Wang, Changze Lv, and Xiaoqing Zheng. Beyond single labels: Improving conversational recommendation through llm-powered data augmentation. In *Annual Meeting of the Association for Computational Linguistics*, pages 15573–15590, 2025.

[46] Gustavo Adolpho Lucas de Carvalho, Simon Ben Igeri, Jennifer Healey, Victor S. Bursztyn, David Demeter, and Lawrence Birnbaum. A flash in the pan: Better prompting strategies to deploy out-of-the-box llms as conversational recommendation systems. In *International Conference on Computational Linguistics*, pages 8385–8398, 2025.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have discussed the claims made in the abstract and introduction in the main paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitation of our work in the Appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided the implementation details about our method in the main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided an open link for our code and dataset.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We have provided the details of our experimental setting in the main paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We have provided an error bar for our results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided the details of experiments compute resources about GPU using and training time.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and our paper conforms to it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed the broader impacts of our work in the Appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper has no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original paper that produced the code package.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We have introduced the details of our new dataset and provided an open link for our code and dataset.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We use LLMs for editing (e.g., grammar, spelling, word choice), data processing/filtering, and writing.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Prompts for Alignment-Based Strength Estimation

The prompt 1 is used for *preference-side augmentation*, a key step in estimating dynamic preference strengths. It constructs structured preference chains that represent user preferences across three semantic levels: from abstract to concrete. The model is instructed to think step by step and output the structured result strictly in JSON format.

Prompt 1: Reasoning Augmentation

Your goal is to help an AI agent better understand user preferences for personalized response generation.
Given the following user preference description:
{USER_PREFERENCE_CONTEXT}

Your task is to construct a structured preference chain with three levels of semantic granularity.

Please think step by step and provide:
1. raw: the original or high-level form of the preference
2. refined: a context-aware, clearer reformulation of the raw preference
3. example: representative items, actions, or behaviors that reflect the preference

Output the result in the following JSON format:
{
"raw": "<fill in raw preference>",
"refined": "<fill in refined preference>",
"example": ["<item_1>", "<item_2>", "..."]
}

This prompt is used for *interaction-side augmentation*, a key step in estimating dynamic preference strengths. It instructs the LLM to generate K paraphrased versions of a given user-agent interaction $\mathcal{D}_t^u$, maintaining the same structure and intent but varying the surface-level expressions. This helps expand the range of observable user behaviors, reduces scoring bias from any single expression, and enables more robust alignment between user intent and preferences during strength estimation.

Prompt 2: Intuiton Augmentation

Please generate {K} semantically equivalent but lexically diverse conversations based on the following user-agent interaction:
User-Agent Interaction:
$\{\mathcal{D}_t^u\}$

Instructions:
1. Return only the {K} generated interactions, each preserving the number of turns and speaker roles.
2. Keep the user's underlying intent consistent with the original conversation.
3. Use varied wording and sentence structures to enhance linguistic diversity.

This prompt is designed for use in the *alignment scoring stage* of our framework, where an LLM is asked to assess how well a given user-agent interaction reflects a specific user preference. The input includes a structured preference chain (comprising raw, refined, and example-level descriptions) and the current interaction between user and agent. The LLM is instructed to reason step-by-step, interpret the semantics of the preference, and judge the degree of alignment exhibited in the dialogue. It outputs a single numerical alignment score ranging from 0 to 10, where higher values indicate stronger semantic alignment. This score serves as the foundation for computing the relative strength of each preference in the final generation stage.

Algorithm 1 Adaptive Preference Arithmetic (AdaPA-Agent)

Require: Accumulated user interaction data $\mathcal{D}_t^u$ up to time t,
User preference set $\mathcal{P}^u = \{p_i^u\}_{i=1}^M$,
Chain-of-thought generator $\mathrm{CoT}(\cdot)$,
Interaction paraphraser $\mathbf{g}(\cdot)$,
Alignment scorer $\mathbf{f}(\cdot,\cdot)$
Ensure: Personalized agent response $\mathcal{O}_t^u$

1: **## 1: Alignment-Based Strength Estimation**
2: $\mathcal{A}^u \leftarrow \mathbf{g}(\mathcal{D}_t^u) \cup \{\mathcal{D}_t^u\}$ # Interaction-side augmentation
3: **for** $i = 1$ **to** M **do**
4: $\quad \mathcal{C}_i \leftarrow \mathrm{CoT}(p_i^u)$ # Preference-side augmentation
5: $\quad s_i \leftarrow \frac{1}{|\mathcal{A}^u|} \sum_{\mathcal{D} \in \mathcal{A}^u} \mathbf{f}(\mathcal{C}_i, \mathcal{D})$ # Alignment scoring
6: **end for**
7: Normalize: $\omega_i \leftarrow \frac{s_i}{\sum_{j=1}^M s_j}$ for $i = 1{:}M$ # Strength weighting
8: **## 2: Controllable Personalized Generation**
9: Initialize: $\mathcal{O}_t^u \leftarrow [\,]$ # Empty response buffer
10: **while** $w \neq \texttt{EOS}$ **do**
11: $\quad \mathbf{P}_{\text{mix}} \leftarrow \sum_{i=1}^M \omega_i \cdot \mathrm{LLM}\big((\mathcal{O}_t^u, \mathcal{D}_t^u, \mathcal{C}_i)\big)$
12: $\quad w \sim \mathbf{P}_{\text{mix}}$ ## Sample next token
13: $\quad \mathcal{O}_t^u \leftarrow [\mathcal{O}_t^u, w]$ ## Update response prefix
14: **end while**
15: **return** $\mathcal{O}_t^u$

Prompt 3: Measuring Correlation Score

You are an alignment evaluator tasked with assessing how well a user-agent interaction aligns with a given preference chain.

Given:
- User-Agent Interaction: $\{\mathcal{D}_t^u\}$
- Preference Chain: $\{\mathcal{C}_i\}$, which includes a raw description, a refined version, and representative examples of a specific user preference.

Instructions:
1. Think step by step.
2. Judge how strongly the interaction reflects the intent or semantics of the given preference.
3. Provide a single alignment score between 0 and 10:
- 10 = perfectly aligned (preference is clearly implied or reflected)
- 0 = completely unrelated
4. Only return the numerical score.

B Algorithm for AdaPA-Agent

Algorithm 1 presents the core procedure of **Adaptive Preference Arithmetic (AdaPA-Agent)**, which progressively adjusts the influence of user preferences based on the accumulated interaction context to generate personalized responses. The algorithm operates in two stages that jointly enable adaptive, context-aware preference control.

- **1. Alignment-Based Strength Estimation (Lines 1-7):** At each interaction step t, the agent updates its internal record of user interactions $\mathcal{D}_t^u$. To enrich contextual diversity, it performs *interaction-side augmentation* by paraphrasing the interaction history via $\mathbf{g}(\cdot)$, forming an expanded set $\mathcal{A}^u$ (Line 2). For each user preference $p_i^u \in \mathcal{P}^u$, the agent generates reasoning traces using a CoT generator (Line 4), then evaluates the semantic alignment between each preference trace and the augmented interaction set using the alignment scorer $\mathbf{f}(\cdot,\cdot)$ (Line 5). The resulting alignment scores $\{s_i\}$ are normalized into preference

weights $\{\omega_i\}$ (Line 7), representing each preferences contextual relevance within the ongoing dialogue. This ensures that the model adaptively reflects preference shifts as user intent evolves over multiple turns.

- **2. Controllable Personalized Generation (Lines 8-15):** Initialized with an empty output prefix (Line 9), the agent generates a response token by token. At each step, it combines next-token distributions weighted by the computed ω_i across all preferences (Line 11), samples the next token (Line 12), and appends it to the output (Line 13), repeating until the end-of-sequence (EOS) token is reached.

By continually re-estimating preference strengths and integrating them into generation, AdaPA-Agent achieves *context-aware, progressive adaptation* across interaction turns. This allows the agent to maintain alignment with evolving user intent without requiring retraining or explicit feedback, resulting in more personalized and consistent dialogue behavior.

Prompt 4: LLM-Based User Simulator for Conversational Recommendation

You will play the role of a user interacting with a conversational movie recommendation system. Your task is to find a movie that matches your current taste, which is influenced by your preferences.
Role & Behavior Guidelines:

- Engage naturally with the agent by gradually revealing your preferences.
- Focus only on requesting or evaluating movie suggestions based on your preferences.
- **Never** mention the name of your target movie.

Task Information:

- In this task, your preference is: `{prefer_info}`.
- In this task, your target movie is: `{target_item}`.

Simulation Rules:

1. Start with vague intent (e.g., "I want to watch something meaningful").
2. Reveal preference cues as the agent asks follow-up questions.
3. Accept recommendations if they match the target movie.
4. Politely reject unrelated ones and give vague but helpful feedback (e.g., "That's not quite what I'm looking for").
5. Maintain a natural and preference-driven tone throughout.

IMPORTANT: Your role is to simulate a movie enthusiast who is exploring potential movie recommendations, not to reveal the exact title of the target movie–`{target_item}`. Keep the conversation natural and engaging, and always focus on requesting recommendations or giving feedback based on the suggestions you receive.

C Details of Conversation Recommendation Task Construction

To carefully evaluate our method's ability to model preference strength, we categorize user movie preferences into two types: long-term and short-term. This distinction allows for controlled evaluation of how well the method adapts to shifts in preference strength. The dataset of the task is based on Reddit-Movie [6], and we extract stable preferences from user's historical data as long-term preferences and generate a corresponding movie list. We select several movies unrelated to long-term preferences and regard their features as the user's short-term preferences.

Furthermore, we create a dynamic simulation environment where the agent interacts with a LLM-based user simulator. For each recommendation task, the user simulator randomly selects which type of preference (long-term or short-term) will dominate the interaction, allowing us to control the variation of preference strength. This enables us to directly manipulate and evaluate the model's response to dynamic shifts in user preferences, providing a clear validation framework for our method. To ensure the reliability of this simulation setup, our user simulator design follows common practices in recent literature [43, 44, 45, 46]. Specifically, we inject user attributes and interaction rules into prompt templates, and the LLM dynamically generates responses based on the accumulated dialogue context, enabling evolving user behavior over turns. We construct the user simulator for

the conversation recommendation task through the following prompt template (Prompt 4). To avoid exposing the target movie, we insert the rule that the LLM should not mention the target movie in the conversation. Besides, we use regular expressions in the code to detect and mask any accidental exposure (replacing it with "*") during the interaction between the LLM agent and the user simulator. Although our simulator may not perfectly replicate real users, it provides a controlled, consistent, and adaptive framework for evaluating the effectiveness of preference modeling strategies. To reduce generation randomness, we set the LLM temperature to 0 during all evaluations. For each reported result in Table 1, we ran 10 trials with different random seeds and computed the standard deviation across these runs.

D Ablation Studies of AdaPA-Agent

D.1 Effectiveness of Dual-side Augmentation

To assess the contribution of dual-side augmentation, we compare three variants of AdaPA-Agent in preference strength estimation: using both preference- and interaction-side augmentation (w/ D-side), using only preference-side augmentation (w/ P-side), and without any augmentation (w/o D-side). To validate dual-side augmentation, we simplified the conversational recommendation and personalized web interaction tasks. In the former, we use binary classification to predict whether a user's desired movie aligns with long-term or short-term preferences. In the latter, we use three-class classification to determine if the required service relates to search, recommendation, or review. As shown in Figure 4, the dual-side setup achieves the highest win rate in both tasks77.3% in conversational recommendation and 63.8% in personalized web interactionsignificantly outperforming the other variants. These results highlight that both augmentation dimensions are critical: while preference-side augmentation improves semantic richness, interaction-side augmentation enhances robustness against linguistic variability.

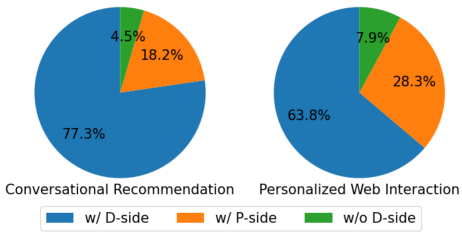

Figure 4: Win rates of AdaPA-Agent on using dual-side data augmentation (w/ D-side), only preference-side augmentation (w/ P-side) and without using any data augmentation (w/o D-side) in preference strengths estimation.

Relying solely on one side introduces bias or sparsity, whereas combining both provides more comprehensive signals for modeling fine-grained preference strengths. By contrast, dual-side augmentation enables AdaPA-Agent to capture both semantic intent and linguistic variability, significantly improving the accuracy of preference strength estimation.

K	Recall	Precision	F1-Score
0	25.32	70.21	37.22
1	43.73	66.70	52.83
2	56.27	72.43	63.34
3	68.79	75.86	72.15
4	**87.50**	**90.32**	**88.89**
5	78.13	80.64	79.35
6	71.88	71.88	71.88

Table 4: Effect of varying the number of generated user-agent interactions (K) on the binary classification results of the conversational recommendation task.

Building upon the experimental setup in Figure 4, we further investigate the effect of interaction-side augmentation on dynamic preferences modeling. As shown in the Table 4, K represents the amount of data augmentation, while recall, precision, and F1-score indicate the model's performance on binary classification for different levels of data augmentation. We observe that as K increases, the recall significantly improves, showing that more augmentation helps identify relevant samples. Similarly, precision and F1-score also increase, reaching their peak values at $K = 4$, after which they start to decrease as K continues to rise. These results show that, the initial improvement stems from the augmented interaction data providing useful preference information, but excessive augmentation introduces noise which results in poor performance. The same phenomenon also occurs in the personalized web interaction task.

D.2 Analysis of LLM-based Alignment Scorer

To evaluate the effectiveness of estimating preference-interaction alignment scores, we compare our proposed LLM-based method with a traditional embedding-based baseline using Sentence-BERT.

Maximum Steps	Embedding-based	LLM-based (ours)
3	35.39	41.45
5	58.56	70.45
7	67.58	75.41

Table 5: Recommendation success rate (RSR) of conversational recommendation using either an embedding-based alignment scorer (Sentence-BERT) or our proposed LLM-based alignment scorer. The LLM-based method consistently outperforms the embedding baseline across different maximum step limits, indicating its superior ability to estimate alignment between user preferences and user-agent interactions.

For the baseline, we encode both the preference description and the user-agent interaction with Sentence-BERT and treat the cosine similarity of the two embeddings as the alignment score. As shown in Table 5, the LLM-based alignment scorer consistently outperforms the embedding-based method across varying maximum step limits in the conversational recommendation task. This performance gap suggests that LLMs are better at capturing the nuanced alignment between user preferences and interaction trajectories. Unlike static embedding models, LLMs can perform contextual reasoning and infer implicit preference signals, leading to more reliable estimation of preference strength and, ultimately, more accurate and intention-aligned recommendations.

Prompt 5: Prompt Engineering

Here are the user's long-term and short-term preferences and their weights (0 to 1, where 0 means no relevance and 1 means the highest relevance):
User's long-term preference: {}
Long-term preference weight: {}
User's short-term preference: {}
Short-term preference weight: {}
Please respond to the user's query based on the long-term and short-term preferences and their weights.
User query: {}

D.3 Effectiveness of Preference Arithmetic

Figure 5 demonstrates the effectiveness of the preference arithmetic method in generating personalized responses based on varying strength combinations (ω_S, ω_L) of short-term and long-term preferences , compared to the traditional prompting engineering approach. Both preference arithmetic and prompt engineering consider two type of preferences and their weights, while the former uses weights as coefficients in arithmetic, the latter regards them as prompts. We use Llama-2-7b-chat as the backbone for these two method, while use GPT-4o to generate the ground truth. The semantic similarity between the generated content and the ground truth is calculated by Sentence-BERT. From the figure, it is evident that preference arithmetic consistently outperforms prompting engineering in terms of alignment with the ground truth. This indicates that adjusting preference-conditioned next-token distributions using preference strengths more effectively controls the influence of long- and short-term preferences on the generated content. Moreover, the preference arithmetic method is more accurate in distinguishing between strength differences, particularly when the strengths of long- and short-term preferences are close, such as in the (0.4, 0.6) strength pair. While the traditional prompting engineering method struggles with this, likely due to the limitations of understanding of numerical information in text. This ability allows preference arithmetic to generate more consistent personal-

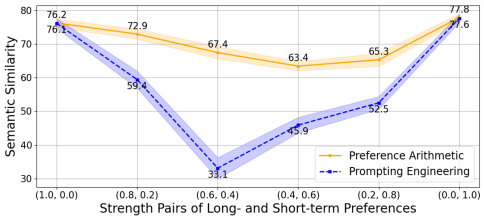

Figure 5: Performance comparison of the preference arithmetic and prompting engineering methods in generating personalized responses based on varying pairs of (ω_S, ω_L). The figure shows the alignment between generated content and ground truth generated by GPT-4o.

ized responses compared to prompting engineering. The prompt engineering template is designed as follows (Prompt 5):

We further compare AdaPA-Agent with a baseline method, termed **Semantic Strength Prompting (SSP)**, where the strengths of preferences are represented as five discrete textual descriptors (*weakest*, *weak*, *neutral*, *strong*, *strongest*). We evaluate both methods on the **Conversational Recommendation Task**, using Recommendation Success Rate (RSR) and Average Interaction Rounds (AIR) as metrics. As shown in Table D.3, AdaPA-Agent consistently outperforms the SSP baseline across all interaction settings. Although such discrete verbal schemes may appear expressive, they are inherently limited to a finite number of strength categories, leading to coarse control granularity. In contrast, AdaPA-Agent models preference strengths as continuous values within $[0, 1]$, enabling infinitely many combinations of strength settings. This formulation allows more nuanced and personalized control over how preferences influence generation. Moreover, since verbal descriptors like *weak* or *strong* can be semantically ambiguous and inconsistently interpreted by LLMs, token-wise distribution mixing provides explicit and interpretable control by directly manipulating strengths over next-token distributions, thereby reducing ambiguity and semantic drift. Therefore, these results demonstrate that AdaPA-Agent achieves superior expressivity and controllability compared to discrete strength prompting, enabling more robust and interpretable preference-conditioned generation.

Max Steps	Method	RSR	AIR
3	**AdaPA-Agent**	**41.45**	**2.64**
	SSP	34.20	2.80
5	**AdaPA-Agent**	**67.24**	**3.73**
	SSP	59.60	4.13
7	**AdaPA-Agent**	**75.41**	**4.46**
	SSP	70.82	4.79

Table 6: Performance comparison between AdaPA-Agent and Semantic Strength Prompting (SSP) on the conversational recommendation task.

E Experiments Compute Resources

In this work, all experiments are conducted on a machine with NVIDIA A6000*2 GPUs, each GPU has 48G memory.

F Limitations

While our framework shows promising results, it has several limitations that open directions for future work. First, our evaluation relies on simulated user interactions, which may not fully capture the nuance of real-world behaviorfuture work could include human-in-the-loop or live deployment studies to validate robustness. Second, the use of multiple LLM calls for alignment scoring increases computational costs, suggesting the need for more efficient approximation techniques. However, given the ongoing decrease in API call costs, we believe the trade-off between API usage and performance improvement in our method is worthwhile. Third, we focus on two domainsmovie recommendation and web interactionwhile extending the approach to broader tasks (e.g., education, health advice) remains unexplored.

G Broader Impacts

This work explores dynamic preference modeling for personalized LLM agents, which can enhance user experience in recommendation, web assistance, and interactive applications. By adaptively aligning with user intent without requiring explicit feedback, the proposed method supports more natural and efficient human-AI interactions. However, personalization systems must be carefully designed to avoid reinforcing user biases, exposing sensitive preferences, or leading to over-reliance on AI decisions. Future work should consider fairness, privacy, and user control in the deployment of adaptive preference-aware agents.