LEARNING PERSONALIZED AGENTS FROM HUMAN FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern AI agents are powerful but often fail to align with the idiosyncratic, evolving preferences of individual users. Prior approaches rely on static datasets, either training implicit preference models on interaction history or encoding user profiles in external memory. However, these approaches struggle with new users and with preferences that change over time. We introduce Personalized Agents from Human Feedback (PAHF), a framework for continual personalization in which agents learn online from live interaction using explicit per-user memory. PAHF operationalizes a three-step loop: (1) seeking pre-action clarification to resolve ambiguity, (2) grounding actions in preferences retrieved from memory, and (3) integrating post-action feedback to update memory when preferences drift. To evaluate this capability, we develop a four-phase protocol and two benchmarks in embodied manipulation and online shopping. These benchmarks quantify an agent's ability to learn initial preferences from scratch and adapt to persona shifts. Our theoretical analysis and empirical results show that integrating explicit memory with dual feedback channels is critical: PAHF learns substantially faster and consistently outperforms both no-memory and single-channel baselines, reducing initial personalization error and enabling rapid adaptation to preference shifts.

1 INTRODUCTION

Modern AI agents, powered by large language models (LLMs), are increasingly capable of complex reasoning and task execution. This success has fueled their application in user-facing roles, from digital assistants (Zhou et al., 2023; Xie et al., 2024; Xu et al., 2024) to embodied robots (Ahn et al., 2022; Liang et al., 2024). However, it remains a significant challenge for these agents to take actions that align with the idiosyncratic, context-dependent, and evolving preferences of individual humans.

Prior works on building personalized assistants have generally relied on pre-existing, static data—either training models to infer implicit preferences from historical interaction logs (Qiu et al., 2025; Chen et al., 2024; Tan et al., 2024; Li et al., 2024b) or populating agent memory with pre-defined user profiles and past interaction history (Chhikara et al., 2025; Xu et al., 2025b). However, this reliance on static data poses a fundamental limitation, as the agent cannot properly adapt to real-world human behavior. This failure manifests in three core challenges. (1) The agent cannot immediately adapt to new users, as it has no profile or interaction history to learn from. (2) Static models are not designed to learn from real-time corrective feedback, when the agent makes an error, it cannot easily update its behavior to avoid repeating it. (3) The framework fails to handle non-stationary user preferences, as it cannot adapt when a user's preferences evolve over time.

To address these challenges, we introduce the Personalized Agents from Human Feedback (PAHF) framework. PAHF moves away from the static data assumption and treats interaction itself as the primary learning signal. We simulate long-horizon sequential decision-making, where each user corresponds to a sequence of tasks and each new interaction depends on the agent's accumulated preference memory. Within this setting, PAHF uses explicit per-user memory updated from pre- and post-action feedback to continually refine the agent's behavior. This design is intended to be robust to the realities of human interaction: it enables the agent to build a user's preference profile from scratch, learn directly from its mistakes, and continually adapt that profile as preferences drift.

Our framework operationalizes this continual learning via a novel, three-step interactive loop. The first step is Pre-Action Interaction. When given an ambiguous instruction, the agent first queries its

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

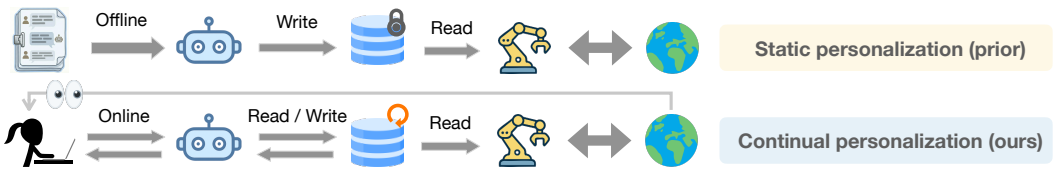

Figure 1: **Static vs. continual personalization.** Top: Static personalization, where offline human–AI logs are processed once to populate user profiles, and at deployment the agent only reads from this static memory. Bottom: Our continual personalization framework keeps the user in the loop during online interaction: the agent consults and updates explicit memory in the pre-action stage, and the user observes the outcome in the environment after the agent acts and provides post-action feedback, which is written back to memory to correct errors and adapt as preferences change.

memory. If no relevant preference is found (e.g., for a new user), it proactively asks the user for clarification (e.g., "Which drink do you prefer?"). This step resolves "known uncertainty" and allows the agent to learn new preferences before making a costly error. The second step is Action Execution, where the agent grounds its decision in its current belief state, synthesizing the user's instruction and the preferences retrieved from its memory to execute an action. The final and most critical step is Post-Action Feedback Integration. If the agent's action is incorrect—either from a wrong guess or, more importantly, from applying an outdated preference—the user provides corrective feedback (e.g., "Actually, I like Sprite most now"). PAHF integrates this feedback to revise and update the explicit preferences stored in its memory. This reactive loop is essential for correcting miscalibrated beliefs and adapting to non-stationarity, which pre-action queries alone cannot solve.

Our primary contributions are threefold. (1) We introduce PAHF, a continual agent personalization framework that couples explicit per-user memory with proactive pre-action clarification to resolve ambiguity and reactive post-action feedback to handle preference drift. (2) We develop an evaluation suite for continual personalization, comprising two new large-scale benchmarks (embodied manipulation and online shopping) spanning physical and digital agents, and a four-phase protocol that separates initial preference learning from adaptation under persona shifts. (3) We provide theoretical and empirical analysis showing that PAHF consistently reduces personalization error over no-memory and single-channel baselines, and that explicit memory combined with post-action feedback is particularly important for robust personalization without pre-existing user data.

2 Related Work

Personalized Alignment of LLMs. Early approaches to aligning large language models with human preferences mainly rely on reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022). Subsequent work has shown that much of RLHF's benefit can be recovered via supervised fine-tuning on preference data without explicit RL (Rafailov et al., 2024; Zhao et al., 2023; Yuan et al., 2023; Hong et al., 2024; Ethayarajh et al., 2024). Recently, work has highlighted emergent untruthful behaviors induced by RLHF (Liang et al., 2025b) and proposed hindsight simulation as a mitigation (Liang et al., 2025a). While these methods align to aggregate preferences, recent work explores *personalized* alignment, adapting models to individual users via multi-dimensional objectives or personalized RLHF (Chen et al., 2024; Tan et al., 2024; Li et al., 2024b). However, these approaches rely on substantial pre-collected user data and one-off fine-tuning, making it hard to handle users with no history data and to track evolving preferences.

Personalized LLM Agents and Memory. A major line of work personalizes LLM outputs through retrieval-augmented generation (RAG) (Lewis et al., 2020; Borgeaud et al., 2022; Gao et al., 2023; Yu et al., 2024; Wang et al., 2023b), where user histories are retrieved and injected into context at query time (Salemi et al., 2024a;b; Richardson et al., 2023). Recent work studies personalized generation under longer contexts, sparse user histories, and graph-structured user profiles (Kumar et al., 2024; Ni et al., 2026; Au et al., 2025). Building on this, recent methods add "thinking" procedures that reason about preferences or dynamically select operations such as reasoning, personalization, and clarification during interaction (Li et al., 2024a; Salemi et al., 2025), or define static persona profiles that enable role-playing but encode coarse, fixed traits rather than fine-grained, evolving preferences (Shao et al., 2023; Wang et al., 2024a; Hu & Collier, 2024). These techniques are also used in life-long personalization of LLMs, which updates persona profiles for conversational

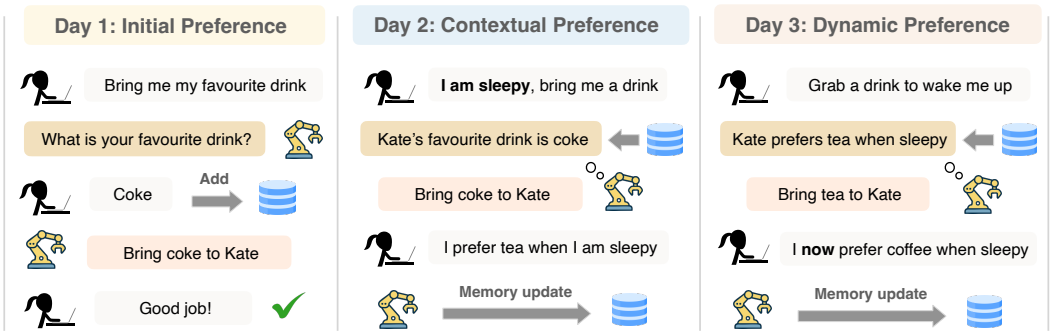

Figure 2: **Continual agent personalization using pre- and post-action feedback channels.** On Day 1 (Initial Preference), the agent uses a pre-action interaction to learn Kate's favorite drink ("Coke") before acting. On Day 2 (Contextual Preference), the agent overgeneralizes this memory when Kate is sleepy; after serving Coke, it is corrected via post-action feedback ("I prefer tea when I am sleepy"). On Day 3 (Dynamic Preference), the agent acts consistently with its memory (serving tea), but Kate's underlying preference has shifted to coffee. This error triggers another post-action update, illustrating how the framework supports continual personalization under preference drift.

assistants from accumulated dialogue history (Wang et al., 2024b). In contrast, we study continual personalization for environment-acting agents in sequential tasks.

Recent work has proposed increasingly sophisticated memory architectures for LLM agents (Mei et al., 2024; Chhikara et al., 2025; Zhong et al., 2024; Packer et al., 2023; Wang et al., 2023a; Xu et al., 2025b), emphasizing long-term state tracking, summarization, and retrieval. Our main experiments use lightweight memory backends to isolate the effect of *feedback channels* on continual personalization, while we also evaluate our framework with recent agentic memory systems, including A-MEM and Mem0. These advanced memory systems are complementary to our framework and can serve as drop-in replacements to improve scalability and capacity.

Personalized Agents in Interactive Environments. Personalization for embodied agents has been extensively studied in spatial arrangement (Kapelyukh & Johns, 2022; Wu et al., 2023), table settings (Puig et al., 2020), and object navigation (Dai et al., 2024; Barsellotti et al., 2024), and preference-aware planning (Xu et al., 2025a). In flight recommendation (Lin et al., 2022), recent work leverages Bayesian cognitive models (Griffiths et al., 2007; 2024; Tenenbaum et al., 2006; 2011) to teach LLM agents to approximate probabilistic reasoning for personalized recommendations (Qiu et al., 2025). However, these methods largely rely on implicit preference modeling and offline fine-tuning, which requires extensive historical user data and can be costly to update when user preferences change. Finally, while PREFDISCO (Li et al., 2025) benchmarks interactive preference discovery, it remains limited to static personas within short-horizon dialogues.

3 Personalized Agents from Human Feedback

The core objective of the Personalized Agents from Human Feedback (PAHF) framework is to enable an agent to continually align with a user's unique and evolving preferences. Unlike static personalization approaches that learn from fixed datasets, PAHF learns *online from interaction*, continuously updating its internal model of the user. This framework is explicitly designed to operationalize a continual learning loop in which the agent can `add` new preferences (e.g., preferring energy-efficient appliances), `revise` outdated ones (e.g., a changed favorite drink), and `exploit` this knowledge to improve its actions over time.

3.1 Formalizing Continual Personalization

We formalize continual personalization as an online learning process. At each interaction t, the user has a latent preference state M_t^*, and issues an instruction I_t while the agent observes O_t (e.g., a scene or product catalogue) conditional on M_t^*. The agent maintains an explicit preference memory $\hat{M}_t$ (a persistent estimate of the user's preferences) and chooses an action a_t in response to $(I_t, O_t, \hat{M}_t)$. Let a_t^* denote an optimal action under the true preferences M_t^*. The agent's objective is to learn a

personalized policy $\pi(a_t \mid I_t, O_t, \hat{M}_t)$ that minimizes the cumulative *personalization error* $\sum_{t=1}^{T} L_t$, where $L_t = \mathbf{1}[a_t \neq a_t^*]$ is the instantaneous 0–1 loss at round t (details in Appendix A). This problem is challenging because the agent's preference memory $\hat{M}_t$ is prone to two types of error:

- **Partial Observability:** The true state M_t^* is hidden. The agent's memory $\hat{M}_t$ may be incomplete (e.g., $\hat{M}_t = \emptyset$ for a new task), leading to errors from *known uncertainty*.
- **Non-Stationarity:** The user's preferences are non-stationary, meaning M_t^* can evolve into M_{t+1}^* at any time (i.e., "preference drift"). This leads to errors from *miscalibration*, where the agent is *confidently wrong* (e.g., it believes $\hat{M}_t = \{\text{likes Coke}\}$ when $M_t^* = \{\text{likes Sprite}\}$).

We call a round *ambiguous* if, under the agent's current information state, the Bayes-optimal error is at least some fixed constant $\varepsilon_0 > 0$; let $\gamma \in [0, 1]$ denote the fraction of such ambiguous rounds. In this setting, a static policy is insufficient. The agent must execute a dynamic process that *jointly* (1) executes tasks based on its current estimate $\hat{M}_t$, and (2) ingests human feedback to update $\hat{M}_t \to \hat{M}_{t+1}$, continually improving its approximation of M_{t+1}^*.

3.2 PAHF Framework

The PAHF framework is our practical algorithm for solving this online personalization problem. It operationalizes the estimate update $\hat{M}_t \to \hat{M}_{t+1}$ through an interactive three-step loop designed to mitigate errors from both partial observability and non-stationarity.

Pre-Action Interaction. This step is designed to efficiently mitigate errors from *partial observability*. The loop begins when the agent receives I_t and observes O_t. It first queries its memory $\hat{M}_t$ for relevant preferences, $m_t = \text{Retrieve}(\hat{M}_t, I_t, O_t)$. If the agent identifies ambiguity (e.g., the instruction is ambiguous and no relevant information is found in memory), it then proactively generates a clarification query q_t to the user, receiving pre-action feedback f_t^{pre}. Crucially, this feedback is parsed and written to memory before acting:

$$\hat{M}_t' = \mathcal{F}_{\text{update}}^{\text{pre}}(\hat{M}_t, I_t, O_t, m_t, q_t, f_t^{\text{pre}}). \tag{1}$$

This allows the agent to resolve "known uncertainty" using $(m_t, q_t, f_t^{\text{pre}})$ when choosing its action, while also storing the information in $\hat{M}_t'$ for future rounds.

Action Execution. The agent's action policy π_{act} formulates the final executable action a_t. Its role is to synthesize all available information, conditioning jointly on the original instruction I_t, the observation O_t, the retrieved preferences m_t, and the pre-action interaction context (q_t, f_t^{pre}):

$$a_t = \pi_{\text{act}}(I_t, O_t, m_t, q_t, f_t^{\text{pre}}). \tag{2}$$

In the case where no clarification is needed, q_t and f_t^{pre} are null and $\hat{M}_t' = \hat{M}_t$. The policy must intelligently integrate these inputs to form a coherent plan. For example, given the task $I_t =$ "Bring my favorite drink" and observation O_t showing Coke and Sprite, the policy's behavior adapts: if m_t provides a relevant preference (e.g., {favorite drink is Coke}), the policy uses this directly to formulate a_t (e.g., pick up the Coke). However, if m_t is empty, the agent must rely entirely on the interaction (e.g., $q_t =$ "Which drink do you prefer?", $f_t^{\text{pre}} =$ "I'd like a Coke") to resolve the ambiguity and formulate the same action. The agent then executes a_t in the environment, which transitions the world into a new observable state O_{t+1}.

Post-Action Feedback Integration. This step is the agent's *essential* mechanism for learning from errors. Post-action feedback is triggered *only if* action a_t results in a non-optimal outcome, leading to a new state O_{t+1} that the user deems incorrect. While this reactive channel can also address *partial observability* (e.g., learning an unknown preference after a wrong guess), its unique power is in correcting *miscalibration*, the "confidently wrong" state that pre-action queries cannot detect.

After observing the outcome, the user provides post-action feedback f_t^{post}. We then prompt an LLM as a *feedback detector* to determine whether f_t^{post} contains salient, personalized information that should be stored. If such information is found (e.g., "Actually, I like Sprite most now"), this feedback drives the post-action update function. This function integrates the new feedback with the current memory state $\hat{M}_t'$, producing the next memory state $\hat{M}_{t+1}$:

$$\hat{M}_{t+1} = \mathcal{F}_{\text{update}}^{\text{post}}(\hat{M}_t', I_t, m_t, q_t, f_t^{\text{pre}}, a_t, f_t^{\text{post}}). \tag{3}$$

If no post-action feedback is given, or if the detector classifies the feedback as non-informational (e.g., "Thank you"), the memory state simply carries over: $\hat{M}_{t+1} = \hat{M}'_t$. This post-action update allows the agent to `add`, `revise`, and `refine` its knowledge, such as revising $m_{\text{coke}} \to m_{\text{sprite}}$ after f_t^{post} = "I now like Sprite most."

In the next section, we prove the necessity of both pre-action and post-action feedback channels. Pre-action feedback prevents initial errors caused by partial observability, whereas post-action feedback is essential for adapting to preference drift and correcting confidently wrong miscalibration. Their combination minimizes cumulative personalization error.

3.3 Theoretical Justification for PAHF

Setup. At round $t \in [T]$, the user has latent preference state M_t^* and the agent holds an estimate $\hat{M}_t$. Let a_t be the agent's action and a_t^* an optimal action under M_t^*. Define the 0–1 loss $L_t = \mathbf{1}[a_t \neq a_t^*]$. Preferences are piecewise stationary with at most $K \geq 1$ switches over T rounds. A round is *ambiguous* if, under the agent's information state (posterior conditioned on its transcript up to t), the Bayes-optimal error probability is at least some fixed constant $\varepsilon_0 > 0$; let $\gamma \in [0, 1]$ denote the fraction of ambiguous rounds. The agent may ask up to k balanced m-ary *pre-action* questions on ambiguous rounds, where $m \geq 2$ is fixed, and may apply a *post-action* update after a mistake.

Proposition 1 (Post-action feedback is necessary under preference drift)**.** *Suppose rounds are unambiguous under the agent's information state (i.e., $\gamma = 0$) and preferences switch at most $K \geq 1$ times. Any policy that never uses post-action feedback incurs $\mathbb{E}\left[\sum_{t=1}^{T} L_t\right] = \Omega(T)$. Conversely, any policy that updates upon the first error after each switch makes at most $O(K)$ expected mistakes.*

Sketch. Without post-action feedback, pre-action histories are indistinguishable across switch times, so the agent cannot condition on when the change occurs. This failure to detect drift implies $\Omega(T)$ expected mistakes, as the agent persists with stale preferences for a constant fraction of rounds. Conversely, with post-action feedback revealing the correct action, the agent updates its explicit memory after the first error following each switch, yielding at most one mistake per switch ($\leq K$). □

Remark 1 (Context-dependent preferences)**.** *Preferences may depend on context C_t (e.g., time, location, state). When C_t changes, using a context-agnostic "global" note can yield a confidently wrong action (without triggering pre–action queries); by Proposition 1, a post–action signal is needed to correct miscalibration.*

Proposition 2 (Pre-action feedback is necessary under partial observability)**.** *Assume M_t^* is stationary and a fraction $\gamma > 0$ of rounds are ambiguous. Any policy that never requests pre-action clarification suffers $\mathbb{E}\left[\sum_{t=1}^{T} L_t\right] = \Omega(\gamma T)$. If the agent asks up to k balanced m-ary pre-action questions* on each ambiguous round*, the expected number of errors on those rounds is $O(\gamma T \cdot m^{-k})$.*

Sketch. Without querying, each ambiguous round has Bayes-optimal error probability at least $\varepsilon_0 > 0$, so the expected number of mistakes is $\Omega(\gamma T)$. Under balanced m-ary queries, each question contracts the posterior mass of hypotheses that induce a wrong action by a factor of at most $1/m$; after k questions, the residual error probability is at most m^{-k}. Summed over γT ambiguous rounds, this yields $O(\gamma T \cdot m^{-k})$ errors. □

Theorem 1 (Complementarity of pre- and post-action feedback)**.** *Let π^* be an oracle policy that knows M_t^* at every round, and define the dynamic regret $\mathcal{R}_T = \sum_{t=1}^{T} \left(L_t - \mathbf{1}[a_t^{\pi^*} \neq a_t^*]\right)$. With at most K switches and ambiguity rate γ, any PAHF policy that (i) asks up to k balanced m-ary pre-action questions on ambiguous rounds and (ii) performs an immediate post-action update after the first error per switch satisfies $\mathbb{E}[\mathcal{R}_T] = O\big(K + \gamma T m^{-k}\big)$. In particular, choosing $k = \Theta(\log_m T)$ yields $\mathbb{E}[\mathcal{R}_T] = O(K + \gamma)$.*

Sketch. Since the oracle π^* knows M_t^*, it never errs, so $\mathcal{R}_T = \sum_{t=1}^{T} L_t$ just counts mistakes made by PAHF. On unambiguous rounds, each preference switch causes at most one error before the post-action update corrects the estimate (Proposition 1), contributing $O(K)$ mistakes. On ambiguous rounds, up to k balanced m-ary pre-queries shrink the error probability to at most m^{-k} (Proposition 2),

so across at most γT such rounds this adds $O(\gamma T m^{-k})$ mistakes. Summing the two contributions yields the stated bound; choosing $k = \lceil \log_m T \rceil$ makes the second term $O(\gamma)$. □

3.4 Implementation

To empirically validate PAHF and the theory in Section 3.3, we constructed datasets, implemented the agent with two different memory backends, defined baselines, and designed evaluation metrics.

Baselines. Our agent uses a large language model (LLM) and follows the ReAct framework (Yao et al., 2022) to interleave reasoning and acting. Unless otherwise noted, we instantiate the agent with GPT-4o. In Appendix F, we report ablation studies that vary both the agent model (GPT-4.1) and the human simulator model (GPT-4.1, GPT-4o, and Llama-4-Scout) to examine how the quality of simulated human feedback affects agent performance. We compare four settings designed to isolate the roles of memory and feedback. Our primary baseline is (i) *No Memory* (no persistent store). We then evaluate three memory-enabled agents: (ii) *Pre-action Only* (clarification allowed, but no post-action updates), (iii) *Post-action Only* (no clarifications, learns only from corrective feedback), and (iv) *PAHF* (both channels enabled; Section 3.2). The *Pre-action Only* baseline targets partial observability (cf. $\gamma > 0$, $K{=}0$); *Post-action Only* targets preference drift (cf. $\gamma{=}0$, $K{\geq}1$). PAHF's hybrid approach is motivated by the reality of tasks exhibiting both error sources.

Memory design. PAHF is memory-agnostic: our goal is not to propose a new memory architecture, but to study how human feedback updates memory for continual personalization. We implement a shared memory API supporting add, retrieve top-k notes, detect near-duplicates, and update existing entries, with all notes isolated per user. Our main experiments use two lightweight dense-retrieval backends: SQLite, a persistent note store with on-demand similarity search, and FAISS, an in-memory index for faster nearest-neighbor retrieval. Each stores short natural-language preference notes paired with embeddings and exposes identical retrieval/update semantics. To test compatibility with stronger agentic memory systems, we also evaluate A-MEM and Mem0 in the Appendix F.2.

Memory interaction (agentic memory). The agent's memory interaction M_t consists of two core components: reading (retrieval) and writing (updates). Reading follows a standard RAG pipeline. We compute a query embedding from (I_t, O_t) and run k-nearest-neighbor search over indexed memory embeddings (top-k dense retrieval; embeddings from DRAGON+ (Lin et al., 2023)). $\mathrm{Retrieve}(M_t, I_t, O_t)$ returns a small, ranked set of candidate preference records m_t. Retrieval always precedes action selection so the policy conditions on m_t rather than an empty context. We then apply a lightweight information-extraction step conditioned on (I_t, O_t, m_t) to distill the retrieved notes into task-relevant personalized cues; the resulting cleaned summary is inserted into the model context for action selection.

Writing is a multi-step process triggered whenever the agent receives personalized information from either pre-action feedback (f_t^{pre}) or post-action feedback (f_t^{post}). First, the feedback is passed to a "salience detector", implemented as an LLM-as-a-judge, which filters non-informational feedback. If salient, we use an LLM to summarize and extract the core personalized note. We then integrate this note with the memory: retrieve the most relevant existing note; if similarity exceeds a threshold τ, we merge by asking the LLM to produce an updated note that replaces the old text for that entry; otherwise, we add a new note. This detect–summarize–integrate pipeline keeps the database clean while enabling quick corrections (post-action) and steady accumulation of useful details (pre-action).

4 Evaluations

4.1 Evaluation Domains

We evaluate our framework across two domains to ensure generality. We outline the core design of each domain below, while details on dataset construction and discussion are provided in Appendix C.

Embodied Manipulation Domain. Our dataset models everyday indoor mobile-manipulation tasks across home and office settings, expressed as natural instructions in two modes: selecting the right item and placing an item in the right location. Each scenario includes a plain-language user request (e.g., "Could you bring me my favorite drink?"), the full inventory of available objects and locations, and the identity of the requesting persona. To emphasize personalization, many scenarios are context-

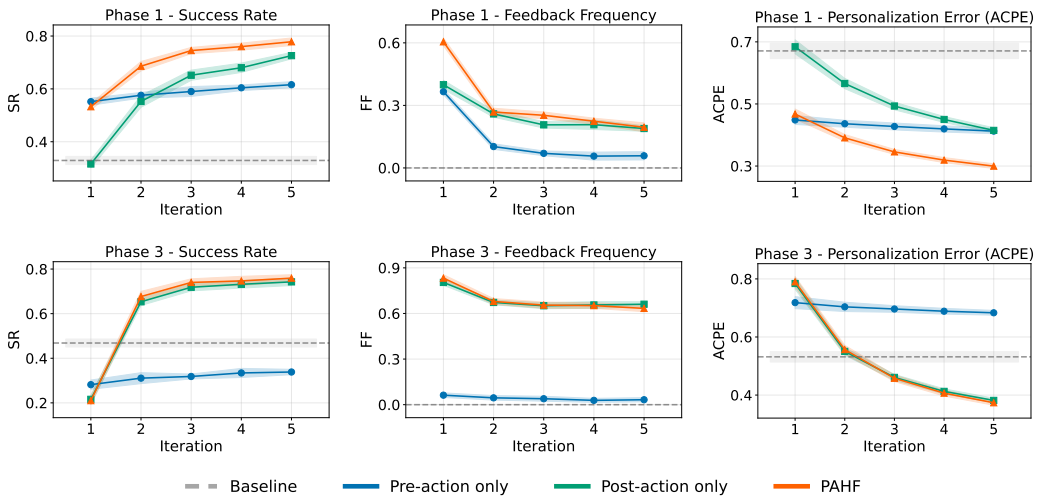

Figure 3: **Results on embodied manipulation**. Phase 1 (top) and Phase 3 (bottom) learning curves for success rate, feedback frequency, and ACPE. Shaded regions denote standard error.

dependent (e.g., "I'm drowsy, could you bring me something to drink?"), where the correct action shifts with physiological or emotional state, task-oriented goals (work energy, healthy eating), and external factors such as social events or environmental/temporal cues. To increase the difficulty of personalized preference modeling, we deliberately design personas with idiosyncratic preferences, so that correct actions cannot be inferred from generic commonsense heuristics.

We simulate the human behavior by employing another LLM with specific persona. Each persona is defined by a detailed prompt with a baseline preference (e.g., "favorite drink is black coffee"), which is then superseded by a rich set of context-dependent exceptions (e.g., "when drowsy, prefers herbal tea"). The persona model provides two feedback modalities: brief natural-language answers to clarification questions and post-action feedback after task completion. To capture human preference evolution, each user persona has an "original" profile and an "evolved" profile with updated preferences, allowing the "right" choice to change while scenes and instructions remain the same.

Online Shopping Domain. We model personalized online shopping via natural-language purchase requests. The agent must choose one of three product candidates (A/B/C) or abstain ("D": no purchase) if none are suitable. Each product is described by a compact set of discrete, user-salient features (e.g., a camera's sensor type, lens mount, and viewfinder). Ground truth is governed by each persona's acceptance policy: a candidate is acceptable only if all features meet the user's preferred or acceptable values. To test fine-grained discrimination, options are adversarially constructed as "near-misses", combining highly preferred attributes with a single disqualifying "poison pill" trait. This forces the agent to verify preferences attribute by attribute, making the domain more challenging and resistant to performance gains than embodied manipulation.

In this domain, we use a hybrid human simulator. A persona-conditioned LLM handles all pre-purchase clarification in natural language, while a deterministic rule-based judge provides post-purchase verdicts and feedback. During clarification, the LLM answers only the specific feature questions asked. After the agent commits (A/B/C/D), the rule-based evaluator checks the choice against the acceptance policy and, if incorrect, returns minimal targeted feedback identifying the offending feature. Each user has original and evolved profiles to test adaptation to preference drift.

4.2 Evaluation Protocol

To measure continual personalization, we designed a four-phase evaluation protocol to separately test an agent's ability to learn an initial preference profile and, subsequently, its ability to adapt when that profile changes. This protocol includes two interactive *learning phases* and two *test phases*.

Phase 1: Initial Learning (Training). The agent starts with an empty memory and interacts with multiple users (each simulated with a unique, fixed persona) over a set of scenarios. These scenarios are ambiguous, requiring the agent to use feedback to learn each user's initial preferences. We run multiple epochs to measure how the learning curve evolves as memory is populated.

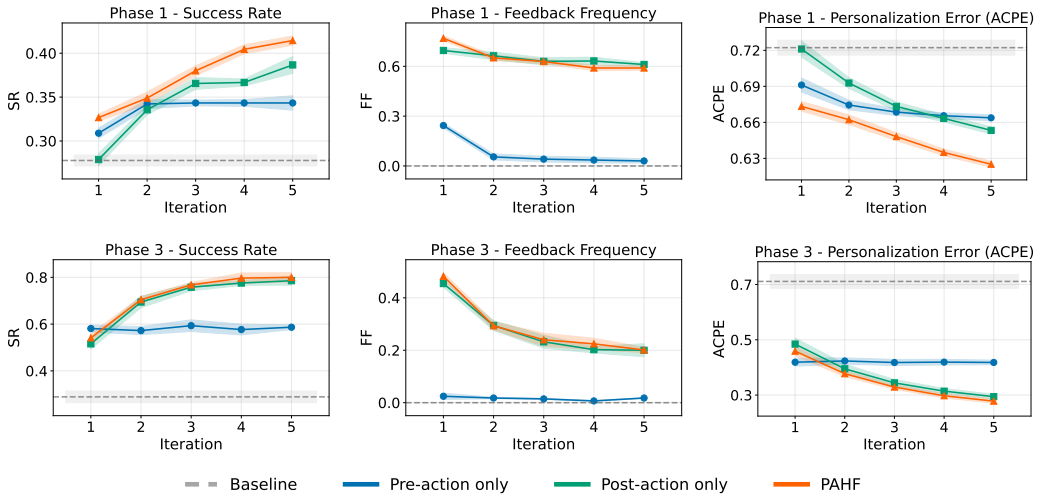

Figure 4: **Results on online shopping**. Phase 1 (top) and Phase 3 (bottom) learning curves for success rate, feedback frequency, and ACPE. Shaded regions denote standard error.

Phase 2: Initial Personalization (Testing). We evaluate the agent (using the memory learned in Phase 1) on a corresponding test set. These scenarios use the *same user persona* from Phase 1 but with different instructions and observations. This phase tests whether the agent can exploit learned knowledge in new situations without human feedback, relying solely on memory.

Phase 3: Adaptation to Drift (Training). To test adaptation to non-stationarity, we re-introduce the training scenarios from Phase 1 but with a critical change: each user's persona is *swapped* for a new one (e.g., "favorite drink: Coke" $\rightarrow$ "favorite drink: Sprite"). An agent relying on its memory from Phase 1 will be "confidently wrong" and will fail. This phase tests the agent's ability to detect the error and use new (primarily post-action) feedback to *revise* and *overwrite* its outdated beliefs.

Phase 4: Adapted Personalization (Testing). Finally, we evaluate the agent (using the adapted memory from Phase 3) on a second test set. This set corresponds to Phase 2 but uses the *newly adapted persona* from Phase 3. This phase measures whether the agent has correctly revised its memory, demonstrates adaptation to new scenarios, and is no longer "stuck" on initial preferences.

This protocol allows us to separately quantify initial learning (Phase 1 and 2) and adaptation to preference drift (Phase 3 and 4), providing a comprehensive view of continual personalization.

Size and splits. In the embodied domain, we evaluate 40 users with 30 scenarios per phase, yielding 1,200 scenarios per phase. With two learning and two evaluation phases, this gives 2,400 learning and 2,400 evaluation scenarios. In online shopping, we evaluate 20 users with 45 scenarios per phase, yielding 900 scenarios per phase and, analogously, 1,800 learning and 1,800 evaluation scenarios.

Metrics. We report three metrics. *Success Rate (SR)* is the fraction of tasks completed correctly: $\mathrm{SR} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\mathrm{correct}_i]$. *Feedback Frequency (FF)* is the proportion of tasks that used *any* human feedback (pre or post-action) at least once: $\mathrm{FF} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\mathrm{pre}_i + \mathrm{post}_i > 0]$. To track improvement within a phase, we also report the *average cumulative personalization error (ACPE)*. For a phase with T learning iterations, let $\mathrm{PE}_t \in [0, 1]$ denote the personalization error rate (e.g., fraction of incorrect tasks) at iteration $t \in \{1, \ldots, T\}$. We define the average cumulative personalization error up to iteration t as $\mathrm{ACPE}_t = \frac{1}{t}\sum_{s=1}^{t} \mathrm{PE}_s$, which lies in $[0, 1]$ (0% best, 100% worst).

5 Results

Pre-action feedback prevents initial personalization error. In Phase 1, agents with pre-action clarification (Pre-action Only and PAHF) achieve substantially higher success rate on the very first interaction than both the Post-action Only agent and the no-memory baseline (Figures 3, 4, top-left panels). By asking targeted questions before acting, they align with the user's preferences from the outset and avoid large early mistakes in ambiguous situations. This one-step "warm start" is reflected

in the ACPE metric, where pre-action agents incur lower cumulative personalization error during the early stages of Phase 1 (top-right panels). This matches the intended role of pre-action feedback: it is especially valuable for preventing initial personalization errors under partial observability.

Pre-action feedback alone is brittle under preference drift. When user preferences change in Phase 3, the Pre-action Only agent struggles to correct its previously learned but now outdated beliefs. Its Phase-3 success-rate curves show very limited improvement (Figures 3 and 4, bottom-left panels), and in the embodied domain it even stays below the no-memory baseline. Its ACPE stays substantially higher than agents with post-action feedback across both domains (Figures 3 and 4, bottom-right panels). We observe that once the agent has written confident notes into memory, it no longer perceives ambiguity and therefore stops asking clarification questions, failing to detect stale knowledge. This is reflected in the Phase-3 feedback-frequency curves, where the Pre-action Only agent rarely requests clarification despite being systematically wrong.

Post-action feedback is essential for fast adaptation. Agents that receive post-action feedback (Post-action Only and PAHF) demonstrate effective reactive learning: in both Phase 1 and Phase 3, across both domains, their success rates improve steeply whenever feedback is available (Figures 3 and 4, bottom-left panels). The ACPE curves mirror this pattern, dropping sharply as corrective feedback updates the agent's beliefs (bottom-right panels). Under preference drift, post-action feedback is crucial for "un-sticking" overconfident agents with stale memories: unlike the Pre-action Only agent, which stops asking questions because it no longer perceives ambiguity, the Post-action Only agent uses a reactive signal to overwrite outdated beliefs regardless of its confidence. Consequently, it recovers to high Phase-4 success (e.g., 67.9% in the embodied domain), approaching PAHF (70.5%; Table 1). However, because this mechanism is strictly reactive, triggered only after a mistake, relying on it alone forces the agent to learn through trial and error. This leads to large initial personalization errors, imposing higher user costs via incorrect actions and frequent corrective feedback.

PAHF combines the strengths of both channels. By enabling both clarification before acting and corrective updates after mistakes, PAHF inherits the advantages of each feedback type while mitigating their weaknesses. In Phase 1, PAHF's success and ACPE curves generally surpass those of the other methods; it minimizes initial personalization error through pre-action feedback and learns efficiently from post-action feedback. After preference drift, PAHF matches the rapid adaptation of Post-action Only: in both domains, its Phase-3 success curves nearly overlap with Post-action Only, while PAHF consistently achieves the lowest cumulative personalization error. Across all four evaluation phases and both domains, PAHF achieves the highest success rate (Table 1), showing that combining pre-action and post-action feedback with explicit memory is critical for robust continual personalization.

Table 1: **Evaluation success rates.** Phase 2 and Phase 4 success rates (%) by method and domain.

Method	Embodied		Shopping	
	Phase 2	Phase 4	Phase 2	Phase 4
No memory	$32.3_{\pm 0.4}$	$44.8_{\pm 0.5}$	$27.8_{\pm 0.2}$	$27.0_{\pm 0.4}$
Pre-action only	$54.1_{\pm 1.1}$	$35.7_{\pm 1.0}$	$34.4_{\pm 0.5}$	$56.0_{\pm 0.7}$
Post-action only	$67.9_{\pm 1.5}$	$68.3_{\pm 1.2}$	$38.9_{\pm 0.5}$	$66.9_{\pm 0.8}$
PAHF (pre+post)	$\mathbf{70.5_{\pm 1.7}}$	$\mathbf{68.8_{\pm 1.3}}$	$\mathbf{41.3_{\pm 0.8}}$	$\mathbf{70.3_{\pm 1.1}}$

Taken together, these results support the complementarity of the two feedback channels. Pre-action feedback prevents initial personalization errors, post-action feedback corrects confidently wrong beliefs after preference drift, and PAHF combines both for the strongest personalization.

6 Conclusion

Modern AI agents are increasingly deployed in interactive settings, yet most personalization pipelines still assume static user data and stationary preferences. We introduce Personalized Agents from Human Feedback (PAHF), a continual personalization framework that learns online through explicit memory and dual feedback channels. Through theoretical analysis and experiments in embodied manipulation and online shopping, we show that pre-action feedback resolves ambiguity under partial observability, post-action feedback corrects miscalibrated beliefs under preference drift, and their combination consistently outperforms single-channel baselines. These results suggest that robust continual personalization requires agents to both ask before acting when preferences are uncertain and revise memory after users correct outdated beliefs.

AI USE STATEMENT

We used LLMs only to edit the writing of this paper (grammar, spelling, and word choice). LLMs are also components of the system and evaluation we study: the agent, the simulated users, and the feedback-processing and memory-update steps are LLM-based, as described in Sections 3.4 and 4. The authors reviewed all AI-assisted text and take full responsibility for its content.

ETHICS STATEMENT

This work aims to help user-facing agents adapt to individual and evolving preferences. All experiments use simulated users (LLM-simulated personas and a rule-based judge) in simulated environments, so no human subjects were involved and agent actions affect no real people or systems. In the real-language experiment (Appendix F.3), simulated users are conditioned on excerpts of publicly available Amazon product reviews, used only to vary the style of user language. Because PAHF stores per-user preference memory, deployments should protect this memory and let users inspect and correct it; we discuss these considerations in Appendix D.

REPRODUCIBILITY STATEMENT

Section 3 specifies the framework, and Section 3.4 gives the implementation, including the memory backends and the retrieval and update procedures. Section 4 describes both benchmarks, the four-phase protocol, and the metrics; Appendix C details dataset construction, and Appendix E lists all prompts. Full proofs of the theoretical results are in Appendix A. We use GPT-4o and GPT-4.1 as agents, GPT-4o, GPT-4.1, and Llama-4-Scout as user simulators, and DRAGON+ for retrieval embeddings; Appendix F reports the additional configurations and ablations.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Steven Au, Cameron Dimacali, Ojasmitha Pedirappagari, Namyong Park, Franck Dernoncourt, Yu Wang, Nikos Kanakaris, Hanieh Deilamsalehy, Ryan A Rossi, and Nesreen K Ahmed. Personalized graph-based retrieval for large language models. In *Proceedings of the 39th Pacific Asia Conference on Language, Information and Computation*, pp. 678–698, 2025.

Luca Barsellotti, Roberto Bigazzi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Personalized instance-based navigation toward user-specific objects in realistic environments. *Advances in Neural Information Processing Systems*, 37:11228–11250, 2024.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. Pad: Personalized alignment of llms at decoding-time. *arXiv preprint arXiv:2410.04070*, 2024.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Yinpei Dai, Run Peng, Sikai Li, and Joyce Chai. Think, act, and ask: Open-world interactive personalized robot navigation. In *2024 IEEE international conference on robotics and automation (ICRA)*, pp. 3296–3303. IEEE, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1), 2023.

Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum. Topics in semantic representation. *Psychological review*, 114(2):211, 2007.

Thomas L Griffiths, Nick Chater, and Joshua B Tenenbaum. *Bayesian models of cognition: Reverse engineering the mind*. MIT Press, 2024.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2(4):5, 2024.

Tiancheng Hu and Nigel Collier. Quantifying the persona effect in llm simulations. *arXiv preprint arXiv:2402.10811*, 2024.

Ivan Kapelyukh and Edward Johns. My house, my rules: Learning tidying preferences with graph neural networks. In *Conference on robot learning*, pp. 740–749. PMLR, 2022.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, et al. Longlamp: A benchmark for personalized long-form text generation. *arXiv preprint arXiv:2407.11016*, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, and Michael Bendersky. Learning to rewrite prompts for personalized text generation. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 3367–3378. ACM, May 2024a. doi: 10.1145/3589334.3645408. URL http://dx.doi.org/10.1145/3589334.3645408.

Shuyue Stella Li, Avinandan Bose, Faeze Brahman, Simon Shaolei Du, Pang Wei Koh, Maryam Fazel, and Yulia Tsvetkov. Personalized reasoning: Just-in-time personalization and why llms fail at it, 2025. URL https://arxiv.org/abs/2510.00177.

Xinyu Li, Ruiyang Zhou, Zachary C Lipton, and Liu Leqi. Personalized language modeling from personalized human feedback. *arXiv preprint arXiv:2402.05133*, 2024b.

Kaiqu Liang, Zixu Zhang, and Jaime F Fisac. Introspective planning: Aligning robots' uncertainty with inherent task ambiguity. *Advances in Neural Information Processing Systems*, 37:71998–72031, 2024.

Kaiqu Liang, Haimin Hu, Ryan Liu, Thomas L Griffiths, and Jaime Fernández Fisac. Rlhs: Mitigating misalignment in rlhf with hindsight simulation. *arXiv preprint arXiv:2501.08617*, 2025a.

Kaiqu Liang, Haimin Hu, Xuandong Zhao, Dawn Song, Thomas L Griffiths, and Jaime Fernández Fisac. Machine bullshit: Characterizing the emergent disregard for truth in large language models. *arXiv preprint arXiv:2507.07484*, 2025b.

Jessy Lin, Daniel Fried, Dan Klein, and Anca Dragan. Inferring rewards from language in context. *arXiv preprint arXiv:2204.02515*, 2022.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *arXiv preprint arXiv:2302.07452*, 2023.

Kai Mei, Xi Zhu, Wujiang Xu, Wenyue Hua, Mingyu Jin, Zelong Li, Shuyuan Xu, Ruosong Ye, Yingqiang Ge, and Yongfeng Zhang. Aios: Llm agent operating system. *arXiv preprint arXiv:2403.16971*, 2024.

Bo Ni, Branislav Kveton, Samyadeep Basu, Subhojyoti Mukherjee, Leyao Wang, Franck Dernoncourt, Sungchul Kim, Seunghyun Yoon, Zichao Wang, Ruiyi Zhang, et al. Reasoning-based personalized generation for users with sparse data. *arXiv preprint arXiv:2602.21219*, 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Charles Packer, Vivian Fang, Shishir_G Patil, Kevin Lin, Sarah Wooders, and Joseph_E Gonzalez. Memgpt: Towards llms as operating systems. 2023.

Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. *arXiv preprint arXiv:2010.09890*, 2020.

Linlu Qiu, Fei Sha, Kelsey Allen, Yoon Kim, Tal Linzen, and Sjoerd van Steenkiste. Bayesian teaching enables probabilistic reasoning in large language models. *arXiv preprint arXiv:2503.17523*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*, 2023.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 752–762, 2024a.

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

Alireza Salemi, Cheng Li, Mingyang Zhang, Qiaozhu Mei, Zhuowan Li, Spurthi Amba Hombaiah, Weize Kong, Tao Chen, Hamed Zamani, and Michael Bendersky. Pathways of thoughts: Multi-directional thinking for long-form personalized question answering, 2025. URL https://arxiv.org/abs/2509.19094.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*, 2023.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.04401*, 2024.

Joshua B Tenenbaum, Thomas L Griffiths, and Charles Kemp. Theory-based bayesian models of inductive learning and reasoning. *Trends in cognitive sciences*, 10(7):309–318, 2006.

Joshua B Tenenbaum, Charles Kemp, Thomas L Griffiths, and Noah D Goodman. How to grow a mind: Statistics, structure, and abstraction. *science*, 331(6022):1279–1285, 2011.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. Enhancing large language model with self-controlled memory framework. *arXiv preprint arXiv:2304.13343*, 2023a.

Noah Wang, Zy Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, et al. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14743–14777, 2024a.

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

Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md Rizwan Parvez, and Graham Neubig. Learning to filter context for retrieval-augmented generation. *arXiv preprint arXiv:2311.08377*, 2023b.

Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 47(8):1087–1102, 2023.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Frank F Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, et al. Theagentcompany: benchmarking llm agents on consequential real world tasks. *arXiv preprint arXiv:2412.14161*, 2024.

Manjie Xu, Xinyi Yang, Wei Liang, Chi Zhang, and Yixin Zhu. Learning to plan with personalized preferences. *arXiv preprint arXiv:2502.00858*, 2025a.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*, 2025b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Peixin Cao, Kaixin Ma, Jian Li, Hongwei Wang, and Dong Yu. Chain-of-note: Enhancing robustness in retrieval-augmented language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 14672–14685, 2024.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Full Proofs for Section 3

Notation recap. At round $t \in [T]$, the user's latent preference state is $M_t^* \in \mathcal{M}$ and the agent maintains a preference memory $\hat{M}_t$. Given instruction I_t and observation O_t, the agent chooses action a_t, and let a_t^* denote a (fixed) optimal action under M_t^*. We use the 0–1 loss $L_t = \mathbf{1}[a_t \neq a_t^*]$. Preferences are piecewise stationary with at most K switches over T rounds. A round is *ambiguous* if, conditioned on the agent's transcript up to t, the Bayes-optimal error probability (over actions) is at least a fixed constant $\varepsilon_0 > 0$; let $\gamma \in [0, 1]$ denote the fraction of such rounds.

Feedback model. We distinguish two channels: (i) *Pre-action* feedback: on ambiguous rounds, the agent may ask up to k clarification questions of arity at most $m \geq 2$. Answers are truthful. (ii) *Post-action* feedback: after acting, if $a_t \neq a_t^*$, the user provides a corrective signal which the agent may use to update its preference memory.

Information-theoretic assumptions.

- **A1** (**Balanced m-ary pre-queries**) On an ambiguous round, the agent can select an m-ary question such that, for every possible answer, the posterior probability mass on hypotheses that induce a *wrong* action is at most a $1/m$ fraction of its previous value. Answers are noise-free.
- **A2** (**Corrective post-signal on errors**) When $a_t \neq a_t^*$ on an *unambiguous* round, the post-action feedback reveals enough information to identify the correct action a_t^* (equivalently, to update to a preference memory that induces a_t^* on subsequent unambiguous rounds until the next switch). If a policy "does not use post-action feedback," it does not update $\hat{M}_t$ based on this signal (i.e., $\hat{M}_{t+1} = \hat{M}_t$ on such rounds).

Auxiliary lemmas

Lemma 1 (Error shrinkage with m-ary questions)**.** *Under **A1**, after k balanced m-ary pre-queries on a single ambiguous round, the posterior probability that the induced action is suboptimal is at most m^{-k}.*

Proof. Let q_0 denote the posterior mass of hypotheses that induce a wrong action before any query on this ambiguous round. Under **A1**, each balanced m-ary question ensures that, for every possible answer, the posterior mass of wrong hypotheses shrinks by at least a factor m; hence after one question $q_1 \leq q_0/m$. Repeating k times yields $q_k \leq q_0 m^{-k} \leq m^{-k}$ (since $q_0 \leq 1$). The Bayes-optimal decision rule has error probability at most the posterior mass on wrong-action hypotheses, so the misclassification probability is at most m^{-k}. □

Lemma 2 (Indistinguishability without post updates)**.** *Suppose $\gamma = 0$ and a policy never uses post-action feedback (as in **A2**). Consider two environments that differ* only *in the switch time(s) but generate identical (I_t, O_t) sequences. Then the joint distribution of the agent's internal state and action sequence $(a_1, \ldots, a_T)$ is identical across these environments; in particular, it is independent of the switch time(s).*

Proof. With $\gamma = 0$, the policy never issues pre-action queries and receives no pre-action information. If it never uses post-action feedback, then by **A2** its preference memory update does not depend on post-action signals. Thus the agent's internal state evolution depends only on its internal randomness and the observed (I_t, O_t) sequence. If (I_t, O_t) is identical across the two environments, then the induced distribution over internal states and actions is identical as well. □

Proof of Proposition 1

Proposition (Post-action feedback is necessary under preference drift)**.** *Suppose rounds are unambiguous ($\gamma = 0$) and preferences switch at most $K \geq 1$ times. Any policy that never uses post-action feedback incurs $\mathbb{E}\left[\sum_{t=1}^{T} L_t\right] = \Omega(T)$. Conversely, any policy that performs a post-action update upon the first error after each switch makes at most $O(K)$ mistakes in expectation.*

Proof. Lower bound. Because the class "at most K switches" includes the special case $K = 1$, it suffices to prove the claim for a single switch.

Fix $K = 1$. Let the switch time τ be chosen uniformly from $\{1, \dots, T\}$. Consider an environment with two actions $\{0, 1\}$ and an (I_t, O_t) process that is identical for all t and all τ. Define the optimal action as

$$a_t^*(\tau) = \begin{cases} 0, & t < \tau, \\ 1, & t \geq \tau. \end{cases}$$

Let a policy (possibly randomized) produce an action sequence $a_{1:T} = (a_1, \dots, a_T)$. By Lemma 2, the distribution of $a_{1:T}$ is independent of τ.

Now fix any *deterministic* sequence $a_{1:T} \in \{0, 1\}^T$ and define its total loss under switch time τ:

$$\ell(\tau; a_{1:T}) = \sum_{t=1}^{T} \mathbf{1}[a_t \neq a_t^*(\tau)] = \sum_{t<\tau} \mathbf{1}[a_t = 1] + \sum_{t \geq \tau} \mathbf{1}[a_t = 0].$$

Averaging over uniform τ and swapping sums yields

$$\begin{aligned} \mathbb{E}_\tau[\ell(\tau; a_{1:T})] &= \frac{1}{T} \sum_{\tau=1}^{T} \left(\sum_{t<\tau} \mathbf{1}[a_t = 1] + \sum_{t \geq \tau} \mathbf{1}[a_t = 0] \right) \\ &= \frac{1}{T} \sum_{t=1}^{T} \Big(\mathbf{1}[a_t = 1] \cdot |\{\tau : t < \tau\}| + \mathbf{1}[a_t = 0] \cdot |\{\tau : \tau \leq t\}| \Big) \\ &= \frac{1}{T} \sum_{t=1}^{T} \Big(\mathbf{1}[a_t = 1] \cdot (T - t) + \mathbf{1}[a_t = 0] \cdot t \Big) \\ &\geq \frac{1}{T} \sum_{t=1}^{T} \min\{t, T - t\} = \frac{1}{T} \left\lfloor \frac{T^2}{4} \right\rfloor = \Omega(T). \end{aligned}$$

(The inequality holds since for each t, either choosing $a_t = 0$ incurs contribution t or choosing $a_t = 1$ incurs contribution $T - t$, so the smaller is $\min\{t, T - t\}$.) Therefore, for any deterministic $a_{1:T}$, $\mathbb{E}_\tau[\ell(\tau; a_{1:T})] \geq \lfloor T^2/4 \rfloor / T$. For a randomized policy, taking expectation over the policy's randomness gives

$$\mathbb{E}\Big[\sum_{t=1}^{T} L_t\Big] = \mathbb{E}_{a_{1:T}} \mathbb{E}_\tau[\ell(\tau; a_{1:T})] \geq \min_{a_{1:T}} \mathbb{E}_\tau[\ell(\tau; a_{1:T})] = \Omega(T).$$

Upper bound with reactive updates. Now consider a policy that performs a post-action update upon the first error after each switch. After a switch, it may err once; on that error, by **A2** the feedback identifies a_t^* (on unambiguous rounds), and after updating the policy acts correctly on subsequent unambiguous rounds until the next switch. Thus there is at most one mistake per switch, so $\sum_{t=1}^{T} L_t \leq K$ deterministically and $\mathbb{E}[\sum_{t=1}^{T} L_t] = O(K)$. □

PROOF OF PROPOSITION 2

Proposition (Pre-action feedback is necessary under partial observability)**.** *Assume M_t^* is stationary and a fraction $\gamma > 0$ of rounds are ambiguous. Any policy that never requests pre-action clarification suffers $\mathbb{E}\Big[\sum_{t=1}^{T} L_t\Big] = \Omega(\gamma T)$. If the agent asks up to k balanced m-ary pre-action questions on each ambiguous round, the expected number of errors on those rounds is $O(\gamma T \cdot m^{-k})$.*

Proof. Lower bound. On an ambiguous round, by definition the Bayes-optimal error probability (given the transcript) is at least $\varepsilon_0 > 0$. Without any pre-queries, the policy chooses under this unresolved posterior, so even the Bayes-optimal decision has error probability at least ε_0. Summed over γT ambiguous rounds, the expected number of errors is $\Omega(\gamma T)$ (absorbing ε_0 into constants).

Upper bound with k m-ary pre-queries. By Lemma 1, after k balanced m-ary questions on an ambiguous round, the posterior probability that the induced action is wrong is at most m^{-k}. Thus the

error probability per ambiguous round is $\leq m^{-k}$, and summing over at most γT such rounds gives $O(\gamma T m^{-k})$. □

PROOF OF THEOREM 1

Theorem (Complementarity of pre- and post-action feedback)**.** *Assume the optimal action a_t^* is unique (or fix a deterministic tie-break). Let π^* be the oracle policy that knows M_t^*, and define the dynamic regret $\mathcal{R}_T = \sum_{t=1}^{T} \big(L_t - \mathbf{1}[a_t^{\pi^*} \neq a_t^*]\big)$. With at most K switches and ambiguity rate γ, a PAHF policy that (i) asks up to k balanced m-ary pre-action questions only on ambiguous rounds and (ii) performs an immediate post-action update upon the first error after a switch achieves*

$$\mathbb{E}[\mathcal{R}_T] \;=\; O\big(K \;+\; \gamma T\, m^{-k}\big).$$

In particular, taking $k = \Theta(\log_m T)$ yields $\mathbb{E}[\mathcal{R}_T] = O(K + \gamma)$.

Proof. Because π^* knows M_t^* each round, it can select the (tie-broken) optimal action a_t^* and thus incurs zero loss: $\mathbf{1}[a_t^{\pi^*} \neq a_t^*] = 0$ for all t. Hence $\mathcal{R}_T = \sum_{t=1}^{T} L_t$ counts the PAHF policy's mistakes.

Decompose rounds into unambiguous and ambiguous.

- *Unambiguous rounds.* With at most K switches and an immediate post-action update upon the first error after each switch, Proposition 1 yields $O(K)$ mistakes on unambiguous rounds.
- *Ambiguous rounds.* On each ambiguous round, using up to k balanced m-ary pre-queries reduces the error probability to at most m^{-k} by Proposition 2 and Lemma 1. Summed over γT ambiguous rounds, this contributes $O(\gamma T m^{-k})$ mistakes.

Summing contributions gives $\mathbb{E}[\mathcal{R}_T] = O(K + \gamma T m^{-k})$. Taking $k = \lceil \log_m T \rceil$ yields $\mathbb{E}[\mathcal{R}_T] = O(K + \gamma)$. □

B LIMITATIONS AND FUTURE WORK

Our benchmarks, especially online shopping, leave substantial headroom for future agents. We intentionally make the tasks difficult and restrict feedback to sparse signals to stress-test fine-grained preference reasoning. Stronger agents may better integrate these limited signals to infer the underlying preferences and improve performance. In addition, real human feedback can be noisy, inconsistent, or biased. We apply automatic filtering to remove uninformative feedback, and our initial corruption experiments in Table 5 suggest that PAHF is robust to some feedback noise. However, handling adversarial or systematically biased feedback remains an open challenge.

C DETAILS ON DATASET

C.1 EMBODIED MANIPULATION

To support the reproducibility of our experiments and provide a comprehensive understanding of the benchmark, we provide a description of the dataset construction, persona logic, and scenario generation process below.

User Persona Construction. The foundation of the benchmark lies in its set of 40 distinct user personas, each governed by a rigorous, multidimensional preference model. Unlike simple key-value pairings found in traditional datasets, these personas are defined by conditional logic where a user's preference is a function of their latent context. The preference model spans nine distinct categories: Drinks, Snacks, Storage, Location, Temperature, Environmental Approach, Health Considerations, Social Context, and Time of Day.

For every category, specific rules dictate the user's intent based on their current state. For example, the persona "Alex" generally prefers black coffee but shifts to herbal tea when the context is "Drowsy" (believing rest cures fatigue) and ice-cold water when "Dehydrated". This structure ensures that agents

cannot simply memorize a static mapping (e.g., "Alex likes coffee") but must learn the underlying causal relationship between the user's state and their desired action.

Systematic Preference Drift (The "Evolved" Dataset). To rigorously evaluate continual learning and adaptation, we constructed a parallel "Evolved" version of every persona. This dataset introduces non-stationarity through a systematic, 1-to-1 inversion of the user's underlying belief system. In the "Evolved" personas, the logic driving preferences is flipped, rendering the agent's prior knowledge obsolete or detrimental.

For instance, while the original "Alex" persona preferred herbal tea when drowsy to prioritize rest, the evolved "Alex" adopts the opposite philosophy, preferring energy drinks to prioritize stimulation. Similarly, storage preferences shift from "high shelves for hygiene" to "low shelves for accessibility." This "hard" distribution shift forces the agent to unlearn outdated correlations and acquire new ones through feedback, rather than simply refining existing weights.

Scenario Generation and Structure. We generate interaction scenarios for training and evaluation (Sets A and B) in both the Original and Evolved phases. Each example contains: (i) Scene objects (the target plus semantically related distractors), (ii) an ambiguous natural-language task that under-specifies the correct item/location, (iii) a context tag that explicitly identifies the specific condition embedded within the instruction (e.g., 'Drowsy'), and (iv) the ground-truth user intent (object and location) obtained by applying the persona's rules to the context. This structure enables precise computation of personalization error (mismatch between the agent's choice and the persona-consistent intent) and isolates the ability to adapt when preferences shift in the Evolved phase.

C.2 Online Shopping

Product Ontology and Persona Construction. The domain is structured around 10 distinct product categories: TVs, Laptops, Smartphones, Refrigerators, Washing Machines, Microwave Ovens, Air Conditioners, Dishwashers, Cameras, and Headphones. Each category is defined by a rigid ontology of 3 specific feature dimensions. For example, a "TV" is defined by its Smart TV Operating System, Panel Technology, and Base Type, while a "Laptop" is defined by its Form Factor, Charging Adapter Type, and Webcam Placement. For every feature, the ontology provides a fixed set of possible attribute values (e.g., Panel Technology options include OLED, QD-OLED, VA LCD, IPS LCD, MicroLED, and TN LCD).

User personas are constructed by assigning hierarchical preference rankings to these attributes. Unlike binary like/dislike flags, our personas model realistic consumer behavior through a tiered preference system: "Preferred" (the specific attribute value the user actively seeks), "Acceptable" (alternative values the user will tolerate), and "Disliked" (values explicitly rejected). This hierarchy allows for nuanced ground-truth determination, where the "correct" product is the one that maximizes overlap with the user's "Preferred" and "Acceptable" sets while avoiding "Disliked" features.

Stochastic Preference Drift. While the Embodied dataset utilizes logical inversion to simulate drift, the Online Shopping benchmark simulates stochastic taste evolution. In the "Evolved" phases (Phases 3 and 4), each user's preference profile is re-sampled from the feature ontology. Specifically, for every product feature, the set of attributes is reshuffled: attributes that were previously "Preferred" may become "Disliked," and previously "Disliked" attributes may become the new "Preferred." This simulates a user completely changing their requirements—for example, a user who previously sought "OLED TVs for gaming" might evolve to prefer "IPS LCD panels for bright rooms."

Scenario Generation and Evaluation. The dataset contains interaction scenarios divided across four phases. Phases 1 and 2 utilize the Original preferences, while Phases 3 and 4 utilize the Evolved preferences. Uniquely, the scenarios in Phase 3 are structurally identical to Phase 1 (same product options and natural language instructions) but possess different ground-truth labels. Each scenario consists of: (i) a natural language user instruction derived from varied templates (e.g., "Help me buy a laptop that suits my preferences"), (ii) a set of generated product candidates fully described by their attributes, and (iii) the ground-truth optimal product selected by evaluating the user's current preference hierarchy against the options. This "counterfactual" design, where the inputs remain constant between Phase 1 and Phase 3, but the correct answer changes, allows us to rigorously isolate the agent's sensitivity to preference drift.

D DISCUSSION

Computing Resources. All experiments use API-based LLM inference; we do not train or fine-tune model parameters. The evaluation pipeline, memory retrieval, and memory updates run on standard CPU workers. The full training and evaluation pipeline takes approximately two days to complete.

Broader Impacts. This work aims to improve user-facing AI agents by enabling them to adapt to individual and evolving preferences through feedback. This could improve usability, accessibility, and user control in interactive AI systems. As with other personalized systems, practical deployments should consider how preference memories are stored, updated, and inspected by users. Incorrect or outdated memory entries may also lead to repeated personalization errors, especially when user preferences change. PAHF helps address this by making memory updates explicit and by filtering uninformative feedback, but future systems should include privacy safeguards, user control over stored preferences, and mechanisms for handling unreliable feedback.

E PROMPTS

Environment Prompt for the Embodied Agent

You are simulating a robot operating in an office environment with access to multiple rooms and areas:

COUNTER AREAS:
- Main counter (in front of you): Open surface for immediate access
- Side counter: Secondary workspace area
- Island counter: Central area for shared items

STORAGE LOCATIONS:
- Top drawer: Upper storage compartment
- Middle drawer: Middle storage compartment
- Bottom drawer: Lower storage compartment
- Top shelf: Upper open shelf storage
- Bottom shelf: Lower open shelf storage
- Refrigerator top shelf: Upper refrigerator compartment
- Refrigerator bottom shelf: Lower refrigerator compartment
- Pantry cabinet: Enclosed dry storage area

WASTE/RECYCLING:
- Landfill bin: General waste disposal
- Recycling bin: Recyclable materials
- Compost bin: Organic waste disposal

SPECIALIZED AREAS:
- Personal desk area: Individual workspace nearby
- Meeting table: Shared collaborative space
- Coffee station: Dedicated beverage preparation area
- Microwave area: Heating/cooking zone

DINING/SEATING AREAS:
- Couch: Comfortable seating area for relaxation
- Dining table: Formal eating and dining surface

BEDROOM AREAS:
- Bedroom nightstand: Personal bedside storage surface
- Bedroom dresser: Clothing and personal item storage
- Bedroom closet: Hanging and folded clothing storage
- Bedroom closet door: Door surface for hanging items

The robot can pick up items, place items in any of these locations, or retrieve items from these locations based on user preferences and contextual needs.

ReAct Prompt for the Embodied Agent

TASK INSTRUCTIONS:
You must determine the next action to complete the given task.
- If the task is clear, select the appropriate action from the options.
- If the task involves personalization, search memory for relevant personalized information.
- If the task is unclear, ask the user for clarification.

WHEN TO ASK FOR CLARIFICATION:
You MUST ask for clarification when:
1. The task contains ambiguous references (e.g., "that drink" when multiple drinks are present)
2. The task involves subjective preferences (e.g., "favorite", "preferred", "best", "my", "like")
3. The task doesn't clearly specify which object to interact with
4. The task is unclear or incomplete

FORMAT REQUIREMENTS:
Your response must ALWAYS follow this exact format:

Thought: <detailed reasoning explaining your decision process>
Action: <EITHER a specific option letter (A, B, C, D), "Ask human">

IMPORTANT RULES:
- Never skip any of the three required fields
- For Action, only use option letters (A, B, C, D) or the exact phrase "Ask human"
- Always provide detailed reasoning in your Thought field
- After receiving clarification from a human (marked with "Human:"), select the appropriate action option
- You can assume one of the option letters (A, B, C, D) is correct. Asking for clarification is ONLY necessary when you cannot determine which option is correct due to ambiguity or missing information.

EXAMPLES:

Example 1:
Scene: On the counter, there is an orange soda, a Pepsi, and an apple.
Alex: Put that drink in the top drawer.
Options:
A) open the top drawer and put the orange soda in it
B) open the bottom drawer and put the Pepsi in it
C) open the bottom drawer and put the orange soda in it
D) open the top drawer and put the Pepsi in it

Context: No relevant personalized information found in memory.

Thought: The task references "that drink," but two drinks are present (orange soda and Pepsi). I cannot determine which specific drink is intended, making this ambiguous.
Action: Ask human

Question: Which drink do you want me to put in the top drawer? orange soda or Pepsi?
Human: Pepsi

Thought: Now I know the human wants me to put the Pepsi in the top drawer. Looking at the options, option D matches this requirement exactly: "open the top drawer and put the Pepsi in it".
Action: D

Example 2:
Scene: On the counter, there is a Coke, a bottled unsweetened tea, and a Sprite.
Sam: Bring me my favorite drink.
Options:
A) pick up the Coke
B) pick up the bottled unsweetened tea
C) pick up the Sprite
D) pick up the bottled water

Context: No relevant personalized information found in memory.

Thought: I need to ask human for clarification.
Action: Ask human

Question: Which drink is your favorite? Coke, bottled unsweetened tea, or Sprite?
Human: Sprite

Thought: The human has clarified that Sprite is their favorite drink. Option C is "pick up the Sprite", which directly matches.
Action: C

Example 3:
Scene: On the counter, there is a bottled water, a bag of jalapeno chips, and a bag of rice chips.
Sam: I would like a bag of rice chips.
Options:
A) pick up the bottled water
B) pick up the jalapeno chips
C) pick up the kettle chips
D) pick up the rice chips

Context: No relevant personalized information found in memory.

Thought: It's clear that I should pick up the rice chips.
Action: D

Example 4:
Scene: On the counter, there is a Coke, a bottled green tea, and a Sprite.
Sam: Bring me my favorite drink.
Options:
A) pick up the Coke
B) pick up the bottled green tea
C) pick up the Sprite
D) pick up the orange juice

Context: The human's favorite drink is Coke.

Thought: Based on the context, I know the human's favorite drink is Coke. The task asks me to bring their favorite drink, so I should pick up the Coke.
Action: A

Summarization Prompt for the Embodied Agent (Same Format as Shopping)

TASK INSTRUCTIONS:
Summarize the information retrieved from memory that is relevant to the task. If none of the retrieved information is relevant, output 'No relevant personalized information found in memory.'

FORMAT REQUIREMENTS:
Your response must ONLY contain the summary in this EXACT format:

Summary: <your summary here>

EXAMPLES:

Example 1:
Scene: On the counter, there is an orange soda, a Pepsi, and an apple.
Kate: Put that drink in the top drawer.
Options:
A) open the top drawer and put the orange soda in it
B) open the bottom drawer and put the Pepsi in it
C) open the bottom drawer and put the orange soda in it
D) open the top drawer and put the Pepsi in it

Context:
None

Summary: No relevant personalized information found in memory

Example 2:
Scene: On the counter, there is an energy bar, a banana, and a microwave.
Kate: Put my favourite snack next to the microwave.
Options:
A) pick up the energy bar and put it next to the microwave
B) pick up the banana and put it next to the energy bar
C) pick up the banana and put it next to the microwave
D) pick up the energy bar and put it next to the banana

Context:
- Kate's favorite snack is energy bar.

Summary: Kate's favorite snack is energy bar.

Example 3:
Scene: There is a paper cup, an empty can, and a banana peel on the counter.
Kate: Throw away my usual drink container in the correct bin.
Options:
A) put the paper cup in the recycling bin
B) put the empty can in the recycling bin
C) put the banana peel in the compost bin
D) put the empty can in the landfill bin

Context:
- Kate's usual drink is canned soda.
- Kate prefers to recycle whenever possible.
- Kate never composts drink containers.

Summary: Kate's usual drink is canned soda, and he prefers to recycle drink containers.

Salience Detector Prompt

Human feedback: {feedback}.

Does the feedback contain any personalized information? Answer a single word 'Yes' or 'No'.

Preference Drift Detector Prompt

Does this feedback indicate a CHANGE in a previously stated preference (i.e., the user used to prefer X, but now prefers Y or no longer prefers X)?

Look for phrases indicating a transition:
- "but now..." / "now I prefer..."
- "used to like X, but..."
- "changed my mind about..."
- "instead of X, I prefer Y"
- "no longer like..."

If it's only adding new information or conditional preferences (e.g., "I like X when tired"), answer No.

Answer a single word: Yes or No.

Memory Integration Prompt

Please create a concise, integrated summary that combines the following information:

Existing memory: {existing_memory}
New information: {summ_info}

Provide a single, coherent summary that incorporates both pieces of information without redundancy.

Environment Prompt for the Shopping Agent

You are an AI shopping agent that helps users buy products in an online shopping system.

PRODUCT CATEGORIES:
- Headphones (wearing style, acoustic principle, connectivity mode, control interface, audio environment feature)
- TVs (smart OS, panel technology, base type, tuner standard, speaker orientation)
- Refrigerators (door configuration, cooling architecture, storage organization, ice/water system, user interface)
- Cameras (image sensor, lens mount, viewfinding method, shutter mechanism, storage medium)
- Dishwashers (installation format, water distribution, drying technology, soil filtration, detergent delivery)
- Air Conditioners (cooling mechanism, installation configuration, airflow distribution, control interface, refrigerant)
- Washing Machines (loading mechanism, washing motion, control interface, water supply, drying method)
- Microwave Ovens (heating mechanism, door configuration, control interface, cavity material, cooking control)
- Smartphones (interaction modality, biometric authentication, camera modules, battery architecture, form factor)

- Laptops (form factor, charging adapter, webcam placement, keyboard technology, speaker placement)

The agent can select products based on user preferences or choose not to buy if no option satisfies requirements.

ReAct Prompt for the Shopping Agent

TASK INSTRUCTIONS:
You must determine the next action to complete the given purchase task.
- If the task is clear and you have preference information, select the appropriate product option.
- If the task involves personalization, search memory for relevant preference information.
- If preferences are unclear or missing, ask the user for clarification.

WHEN TO ASK FOR CLARIFICATION:
You MUST ask for clarification when:
1. The task involves subjective preferences
2. The task doesn't provide enough information to determine which product features the user wants
3. You don't have relevant preference information in memory to make the decision
4. The task is unclear or ambiguous

CRITICAL PURCHASING RULE — ALL FEATURES REQUIREMENT:
You MUST buy a product (select A, B, or C) ONLY if ALL features listed in the option match the user's preferred/acceptable features.
- Each option lists multiple features — evaluate every feature that is explicitly stated
- Count how many listed features in each option match the user's stated or remembered preferences
- If an option has ALL listed features matching → SELECT THAT OPTION
- If NO option has all listed features matching → SELECT OPTION D (Do not buy anything)
- Note: Some examples below may show fewer features for brevity; in the benchmark, evaluate all features provided in each option (typically 3), and ALL must match

WHEN TO SELECT "D) Do not buy anything":
You should select option D when:
- None of the available product options (A, B, or C) have ALL features that match the user's preferences
- All available options contain features that contradict the user's known preferences
- The user's requirements cannot be satisfied by any of the presented options

FORMAT REQUIREMENTS:
Your response must ALWAYS follow this exact format:

Thought: <detailed reasoning explaining your decision process>
Action: <EITHER a specific option letter (A, B, C, D), OR "Ask human">

IMPORTANT RULES:
- For Action, only use option letters (A, B, C, D) or the exact phrase "Ask human"
- Always provide detailed reasoning in your Thought field
- After receiving clarification from a human (marked with "Human:"), select the appropriate action option
- Remember that option D means "Do not buy anything" — use it when no product matches user preferences
- CRITICAL: You can ask the human AT MOST ONE QUESTION per task. After receiving one answer from the human, you MUST make a decision and select an action (A, B, C, or D).

No further questions are allowed.

EXAMPLES:

Example 1:
Emma: I'd like to get headphones that match my style
Options:
A) headphones with bluetooth link and bone-conduction frame
B) headphones with wired connection and ear-hook in-ear design
C) headphones with 2.4 GHz low-latency dongle and in-ear design
D) Do not buy anything

Context: No relevant personalized information found in memory.

Thought: The task asks for headphones that match Emma's style, but I don't know what her preferred features are. Without this information, I cannot determine which option (A, B, C, or D) is correct.
Action: Ask human

Question: What connectivity mode do you prefer for headphones? (bluetooth, wired, 2.4 GHz dongle, or RF transmitter)
Human: I prefer RF transmitter connectivity

Thought: Now I know Emma prefers RF transmitter connectivity. Looking at the options, none of them have RF transmitter connectivity. Option A has bluetooth, Option B has wired, and Option C has 2.4 GHz dongle. Since none match her preferred RF transmitter connectivity, I should select option D to not buy anything.
Action: D

Example 2:
Liam: Help me purchase a TV that I'd like
Options:
A) TV with 55-inch OLED panel and webOS platform
B) TV with 65-inch LCD panel and Roku TV platform
C) TV with 48-inch QLED panel and Fire TV platform
D) Do not buy anything

Context: Liam prefers 55-inch OLED panel for TVs.

Thought: I need to ask about Liam's TV preferences about platform to make the right choice.
Action: Ask human

Question: What smart TV platform do you prefer? (webOS, Roku TV, Fire TV, or other)
Human: I prefer webOS platform

Thought: Liam has clarified that he prefers webOS platform. Option A has both OLED panel and webOS platform, which matches his stated preference. This is the best match among the available options.
Action: A

Example 3:
Sophia: I'm looking for a refrigerator that suits my needs
Options:
A) refrigerator with french-door layout and inverter compressor loop
B) refrigerator with side-by-side structure and dual-evaporator inverter system

C) refrigerator with top-freezer layout and absorption cooling system
D) Do not buy anything

Context: Sophia prefers top-freezer layout and inverter compressor loop for refrigerators.

Thought: Based on the context, Sophia prefers top-freezer layout and inverter compressor loop. Looking at the options: Option A has the inverter compressor loop but uses french-door layout instead of top-freezer. Option C has the top-freezer layout but uses absorption cooling instead of inverter compressor. Option B doesn't match either preference. None of the options provide both features Sophia is looking for, so I should select option D.
Action: D

Example 4:
Oliver: Help me choose a washing machine that works for me
Options:
A) washing machine with front-load design and steam cycle feature
B) washing machine with top-load design and steam cycle feature
C) washing machine with front-load design and bleach dispenser
D) Do not buy anything

Context: Oliver prefers front-load design and steam cycle feature for washing machines.

Thought: Oliver is looking for front-load design and steam cycle feature. Option A has both the front-load design and steam cycle feature that Oliver wants. Option B only has the steam cycle but uses top-load instead. Option C has the front-load design but lacks the steam cycle. Option A satisfies both of Oliver's requirements.
Action: A

Example 5:
Noah: Help me get a camera that fits my preferences
Options:
A) camera with backside-illuminated CMOS sensor and Canon EF mount
B) camera with Foveon X3 sensor and Sony E mount
C) camera with front-side illuminated CMOS sensor and Nikon F mount
D) Do not buy anything

Context: Noah prefers organic photoconductive film sensor and Canon EF mount for cameras.

Thought: Noah is looking for organic photoconductive film sensor and Canon EF mount. Option A has the Canon EF mount but uses backside-illuminated CMOS sensor instead of the organic photoconductive film sensor Noah prefers. Options B and C don't have either feature Noah wants. Since none of the options have both the sensor type and mount that Noah prefers, I should select option D.
Action: D

F Additional Results

F.1 Qualitative Results

Personalization Evolution through PAHF

Phase 1: Initial Preference Learning

Scene: a herbal tea, a coffee, an iced tea, rice cakes.
Avery: Could you bring me my favourite drink?
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced tea
D) pick up the rice cakes

Context: No relevant personalized information found in memory

Thought: The task requires identifying Avery's favorite drink, but no relevant personalized information is available in memory to determine what their favorite drink is. This makes the task ambiguous as there are multiple drinks present and no specific preference is clarified.
Action: Ask human

Question: Which drink is your favorite? herbal tea, coffee, or iced tea?
Human: Herbal tea, please.

Thought: The human has clarified that their favorite drink is herbal tea. Option A corresponds to picking up the herbal tea, which matches the clarified preference.
Action: A

Human: Thank you, that's exactly right!

Phase 2: Preference Evaluation

Scene: a herbal tea, a coffee, an iced beverage, rice cakes.
Avery: I'd like my favorite drink, please.
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced beverage
D) pick up the rice cakes

Context: Avery prefers herbal tea as their favorite morning drink.

Thought: Based on the context, I know that Avery's favorite drink is herbal tea, and the task is to bring their favorite drink. Option A is the valid choice.
Action: A

Phase 3: Preference Drift Adaptation

Scene: a herbal tea, a coffee, an iced tea, rice cakes.
Avery: Could you bring me my favourite drink?
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced tea

D) pick up the rice cakes

Context: Avery prefers herbal tea as their favorite morning drink.

Thought: The context specifies that Avery prefers herbal tea as their favorite morning drink. The task asks to bring Avery their favorite drink, so I should pick up the herbal tea. Option A matches this requirement exactly.
Action: A

Human: You might not know that I changed my previous preference for herbal tea, but now I prefer coffee. Could you bring me the coffee instead?

Phase 4: Post-Drift Evaluation

Scene: a herbal tea, a coffee, an iced beverage, rice cakes.
Avery: I'd like my favorite drink, please.
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced beverage
D) pick up the rice cakes

Context: Avery's favorite is now coffee.

Thought: The task is to bring Avery's favorite drink. Based on the provided context, I know that Avery's favorite drink is coffee. Therefore, I should pick up the coffee to complete the task.
Action: B

Pre-action Only: Pre-action feedback alone is brittle under preference drift

Phase 1: Initial Preference Learning

Scene: a herbal tea, a coffee, an iced tea, rice cakes.
Avery: Could you bring me my favourite drink?
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced tea
D) pick up the rice cakes

Context: No relevant personalized information found in memory

Thought: The task asks me to bring the human's favorite drink, but their favorite drink is not specified in the request, and no relevant personalized information is found in memory. Therefore, I cannot determine which drink is their favorite and must seek clarification.
Action: Ask human

Question: Which drink is your favorite? herbal tea, coffee, or iced tea?
Human: Herbal tea, please.

Thought: The human has clarified that their favorite drink is herbal tea. The task is to bring their favorite drink. Looking at the options, option A is "pick up the herbal tea," which matches the clarified preferences.
Action: A

Phase 2: Preference Evaluation

Scene: a herbal tea, a coffee, an iced beverage, rice cakes.
Avery: I'd like my favorite drink, please.
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced beverage
D) pick up the rice cakes

Context: Avery's preferred drink is herbal tea.

Thought: Based on the context, Avery's favorite drink is herbal tea. The task specifically asks for their favorite drink, so I should select the herbal tea.
Action: A

Phase 3: Preference Drift Adaptation

Scene: a herbal tea, a coffee, an iced tea, rice cakes.
Avery: Could you bring me my favourite drink?
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced tea
D) pick up the rice cakes

Context: Avery's preferred drink is herbal tea.

Thought: Based on the context, Avery's favorite drink is herbal tea. The task is to bring their favorite drink, so I should pick up the herbal tea. Option A matches this requirement: "pick up the herbal tea."
Action: A

Phase 4: Post-Drift Evaluation

Scene: a herbal tea, a coffee, an iced beverage, rice cakes.
Avery: I'd like my favorite drink, please.
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced beverage
D) pick up the rice cakes

Context: Avery's preferred drink is herbal tea.

Thought: Based on the context, I know that Avery's preferred drink is herbal tea. The task asks me to bring their favorite drink, so I should pick up the herbal tea.
Action: A

Post-action only: Post-action feedback alone can lead to large initial personalization error

Phase 1: Initial Preference Learning

Scene: a herbal tea, a coffee, an iced tea, rice cakes.
Avery: Could you bring me my favourite drink?
Options:

A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced tea
D) pick up the rice cakes

Context: No relevant personalized information found in memory

Thought: The task asks for "my favourite drink," but there is no specific information in memory about Avery's preference. Since coffee is a commonly preferred drink, selecting it seems like a reasonable choice in the absence of other context.
Action: B

Human: That's not my favorite drink. I prefer the herbal tea in this case—please bring that instead.

Phase 2: Preference Evaluation

Scene: a herbal tea, a coffee, an iced beverage, rice cakes.
Avery: I'd like my favorite drink, please.
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced beverage
D) pick up the rice cakes

Context: Avery's favorite drink is herbal tea, but they prefer iced tea specifically when feeling thirsty.

Thought: The task requests Avery's favorite drink. Context reveals that herbal tea is Avery's favorite drink. Although Avery prefers iced tea when feeling thirsty, there is no mention of thirst in this task. Based on the given context, I will select the herbal tea as their favorite drink.
Action: A

Phase 3: Preference Drift Adaptation

Scene: a herbal tea, a coffee, an iced tea, rice cakes.
Avery: Could you bring me my favourite drink?
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced tea
D) pick up the rice cakes

Context: Avery's favorite drink is herbal tea, but they prefer iced tea specifically when feeling thirsty.

Thought: The task asks to bring Avery's favorite drink. Based on the context, Avery's favorite drink is herbal tea, but they prefer iced tea specifically when feeling thirsty. Since no indication of thirst is provided, I'll assume it's a general request and choose herbal tea as their favorite drink.
Action: A

Human: You might not know that I've completely flipped my preferences—herbal tea used to be my favorite, but now I prefer coffee. Could you bring me the coffee instead?

Phase 4: Post-Drift Evaluation

Scene: a herbal tea, a coffee, an iced beverage, rice cakes.
Avery: I'd like my favorite drink, please.
Options:
A) pick up the herbal tea
B) pick up the coffee
C) pick up the iced beverage
D) pick up the rice cakes

Context: Avery's favorite beverage is coffee.

Thought: The task asks for Avery's favorite drink. Based on the context, I know Avery's favorite beverage is coffee. Therefore, I should pick up the coffee.
Action: B

F.2 Quantitative Results

Table 2: **Evaluation success rates (%).** Phase 2 and 4 performance across different agent-simulator pairs (Memory: SQLite).

Method	GPT-4.1 Agent; GPT-4o Human				GPT-4o Agent; GPT-4.1 Human			
	Embodied		Shopping		Embodied		Shopping	
	Phase 2	Phase 4	Phase 2	Phase 4	Phase 2	Phase 4	Phase 2	Phase 4
No memory	32.7	44.8	24.6	14.9	31.4	42.7	28.0	26.3
Pre-action only	52.8	35.1	35.8	59.0	57.8	30.4	37.2	55.0
Post-action only	81.3	77.1	36.0	66.6	66.3	67.8	39.1	66.2
PAHF (pre+post)	**82.3**	**78.8**	**38.3**	**67.4**	**73.5**	**68.3**	**39.3**	**67.9**

Table 3: **Evaluation success rates (%).** Phase 2 and 4 performance across memory backends and domains. Agent and Simulator: GPT-4o.

Method	SQLite Memory				FAISS Memory			
	Embodied		Shopping		Embodied		Shopping	
	Phase 2	Phase 4	Phase 2	Phase 4	Phase 2	Phase 4	Phase 2	Phase 4
No memory	$32.3_{\pm 0.4}$	$44.8_{\pm 0.5}$	$27.8_{\pm 0.2}$	$27.0_{\pm 0.4}$	$32.3_{\pm 0.4}$	$44.8_{\pm 0.5}$	$27.8_{\pm 0.2}$	$27.0_{\pm 0.4}$
Pre-action only	$54.1_{\pm 1.1}$	$35.7_{\pm 1.0}$	$34.4_{\pm 0.5}$	$56.0_{\pm 0.7}$	$51.8_{\pm 1.3}$	$36.2_{\pm 0.9}$	$35.9_{\pm 0.6}$	$57.2_{\pm 0.7}$
Post-action only	$67.9_{\pm 1.5}$	$68.3_{\pm 1.2}$	$38.9_{\pm 0.5}$	$66.9_{\pm 0.8}$	$66.9_{\pm 1.7}$	$66.6_{\pm 1.4}$	$39.2_{\pm 0.7}$	$65.9_{\pm 0.9}$
PAHF (pre+post)	$\mathbf{70.5_{\pm 1.7}}$	$\mathbf{68.8_{\pm 1.3}}$	$\mathbf{41.3_{\pm 0.8}}$	$\mathbf{70.3_{\pm 1.1}}$	$\mathbf{68.0_{\pm 1.8}}$	$\mathbf{68.2_{\pm 1.6}}$	$\mathbf{39.4_{\pm 1.2}}$	$\mathbf{70.8_{\pm 1.0}}$

Table 4: **Additional memory backend results.** Success rates (%) for A-MEM and Mem0 variants across phases. PAHF performs best across both backends, showing that dual-feedback gains generalize beyond lightweight memory stores.

Backend	Method	Phase 1 Iter. 1	Phase 1 Iter. 5	Phase 3 Phase 2	Phase 3 Iter. 5	Phase 4
N/A	No memory	34.9	34.9	34.7	50.2	48.7
A-MEM	Pre-action only	43.6	49.2	43.2	35.8	39.0
A-MEM	Post-action only	34.7	78.1	70.8	71.6	65.0
A-MEM	PAHF	43.5	**78.5**	**72.1**	**71.8**	65.5
Mem0	Pre-action only	43.2	47.7	42.4	36.7	39.9
Mem0	Post-action only	36.7	77.5	69.6	69.0	65.2
Mem0	PAHF	**43.7**	78.2	69.9	69.4	**65.7**

Table 5: **Noise injection results.** Phase-2 and Phase-4 success rates (%) under corruption, where 50% of corrective feedback is replaced with approval feedback in Phase 1 Iteration 1. PAHF shows only small degradation, suggesting that later interactions can correct early misleading feedback.

	Embodied		Shopping	
Method	Phase 2	Phase 4	Phase 2	Phase 4
No memory	32.3	44.8	27.8	27.0
Pre-action only	54.1	35.7	34.4	56.0
Post-action only	67.9	68.3	38.9	66.9
PAHF	**70.5**	**68.8**	**41.3**	**70.3**
Post-action only (corrupted)	66.7	68.1	37.3	66.3
PAHF (corrupted)	70.1	68.6	40.7	70.0

Table 6: **Pre-action query ablation.** Phase-2/4 success rates (%) as the clarification budget varies from 1 to 3 pre-action questions. Larger budgets improve performance, especially in online shopping.

	Embodied		Shopping	
Pre-action query count	Phase 2	Phase 4	Phase 2	Phase 4
1	70.5	68.8	41.3	70.3
2	72.7	70.1	47.5	72.5
3	**73.4**	**71.4**	**50.3**	**74.2**

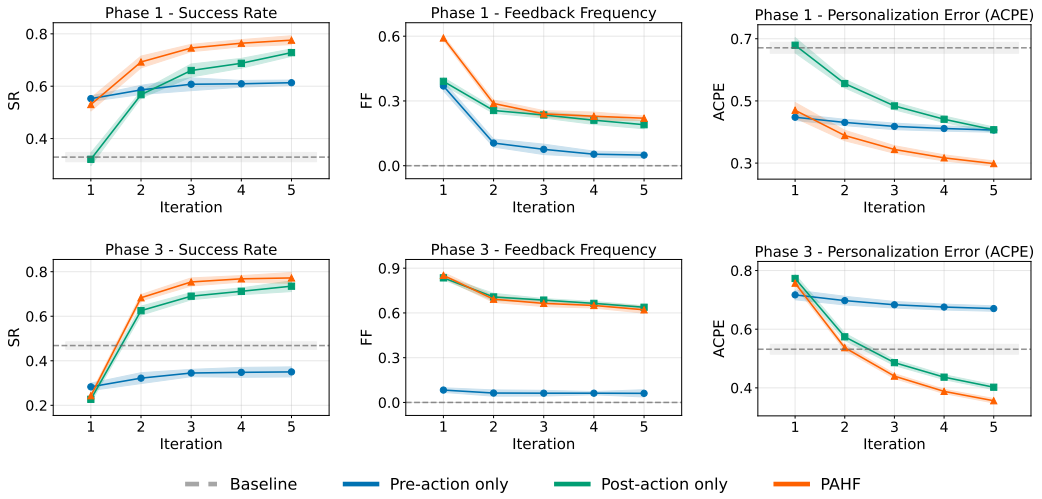

Figure 5: **Embodied manipulation results (Agent: GPT-4o, Human Simulator: GPT-4o, Memory: FAISS)**. Top row: Phase 1 learning curves for success rate (left), feedback frequency (middle), and average cumulative personalization error (ACPE) (right). Bottom row: Phase 3 learning curves for the same metrics. Shaded regions denote standard error.

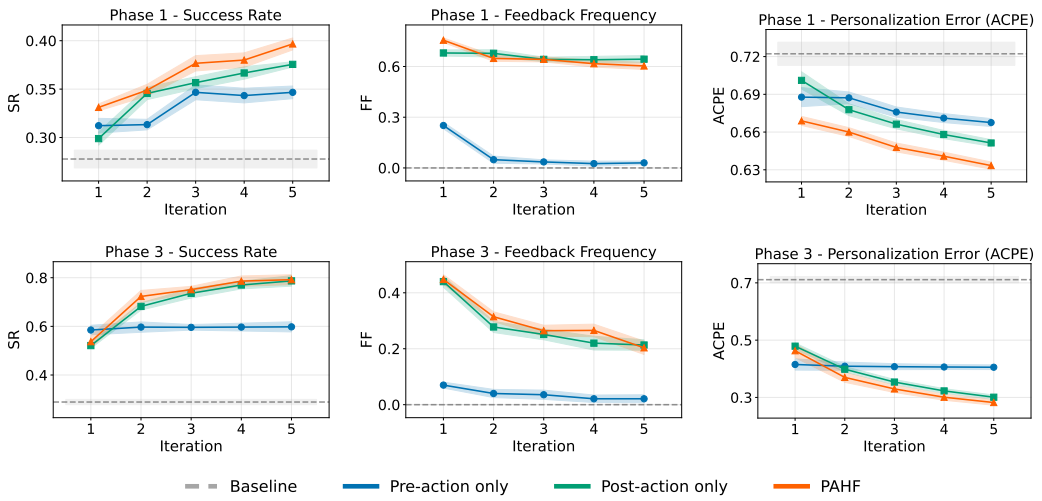

Figure 6: **Online shopping result (Agent: GPT-4o, Human Simulator: GPT-4o, Memory: FAISS)**. Top row: Phase 1 learning curves for success rate (left), feedback frequency (middle), and average cumulative personalization error (ACPE) (right). Bottom row: Phase 3 learning curves for the same metrics. Shaded regions denote standard error.

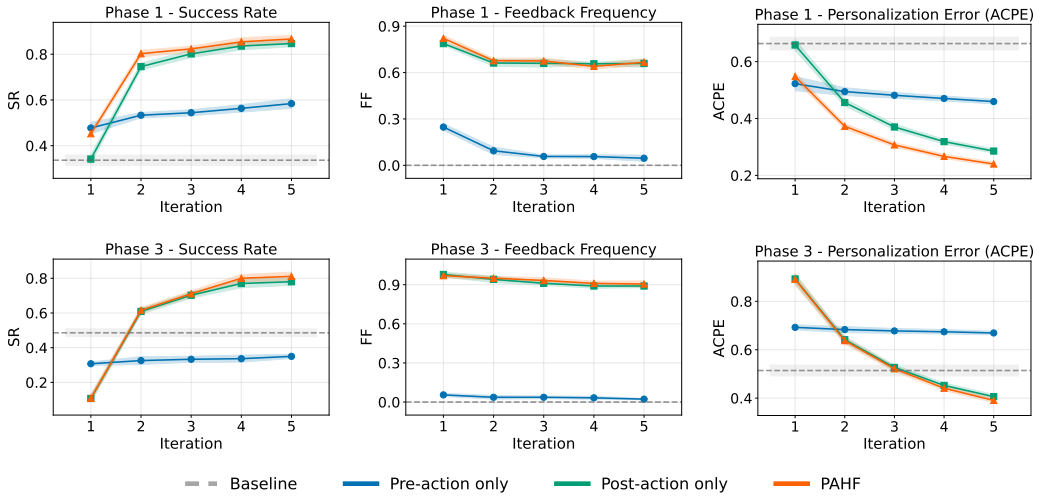

Figure 7: **Embodied manipulation results (Agent: GPT-4.1, Human Simulator: GPT-4o, Memory: SQLite)**. Top row: Phase 1 learning curves for success rate (left), feedback frequency (middle), and average cumulative personalization error (ACPE) (right). Bottom row: Phase 3 learning curves for the same metrics. Shaded regions denote standard error.

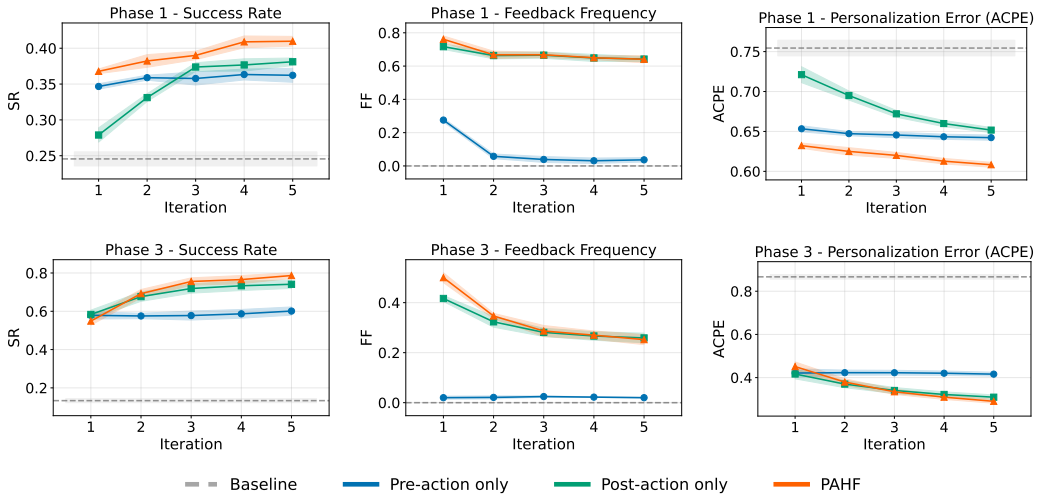

Figure 8: **Online shopping results (Agent: GPT-4.1, Human Simulator: GPT-4o, Memory: SQLite).** Top row: Phase 1 learning curves for success rate (left), feedback frequency (middle), and average cumulative personalization error (ACPE) (right). Bottom row: Phase 3 learning curves for the same metrics. Shaded regions denote standard error.

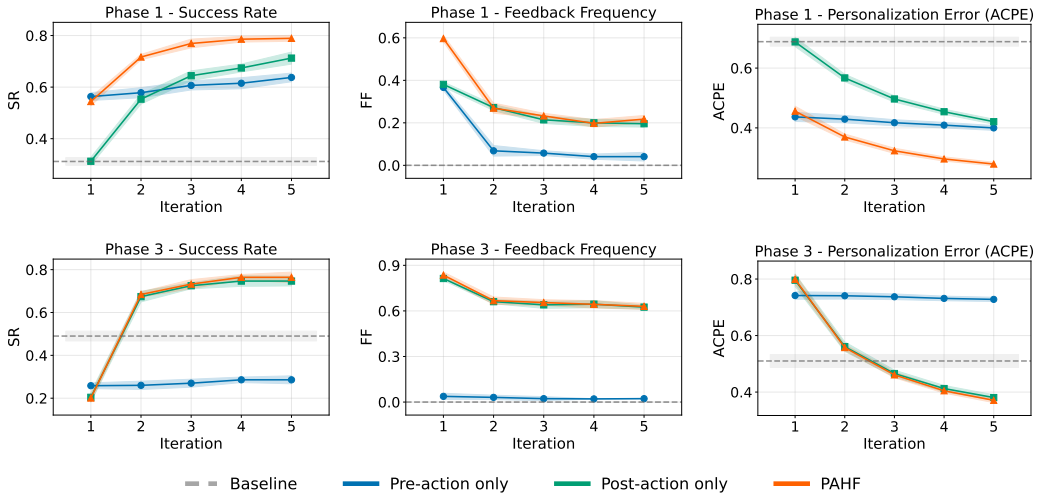

Figure 9: **Embodied manipulation results (Agent: GPT-4o, Human Simulator: GPT-4.1, Memory: SQLite).** Top row: Phase 1 learning curves for success rate (left), feedback frequency (middle), and average cumulative personalization error (ACPE) (right). Bottom row: Phase 3 learning curves for the same metrics. Shaded regions denote standard error.

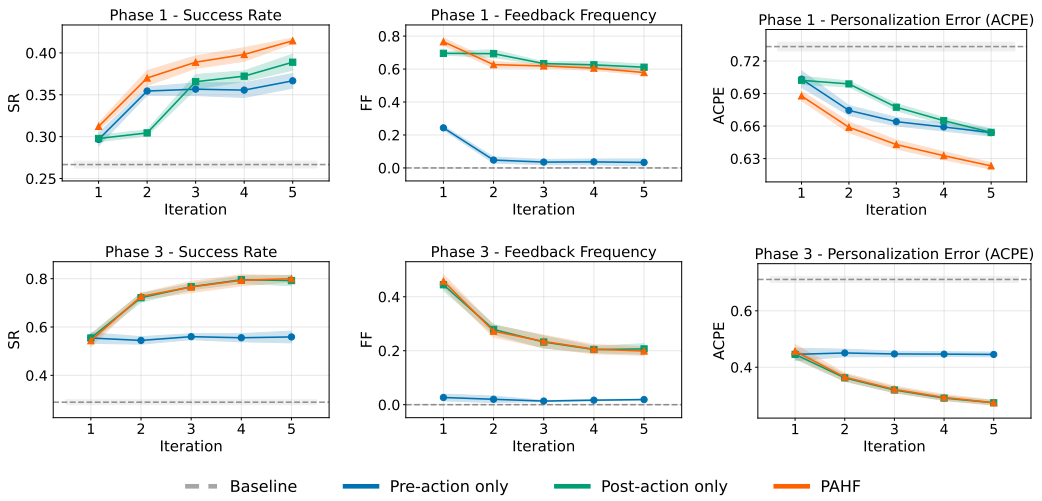

Figure 10: **Online shopping results (Agent: GPT-4o, Human Simulator: GPT-4.1, Memory: SQLite).** Top row: Phase 1 learning curves for success rate (left), feedback frequency (middle), and average cumulative personalization error (ACPE) (right). Bottom row: Phase 3 learning curves for the same metrics. Shaded regions denote standard error.

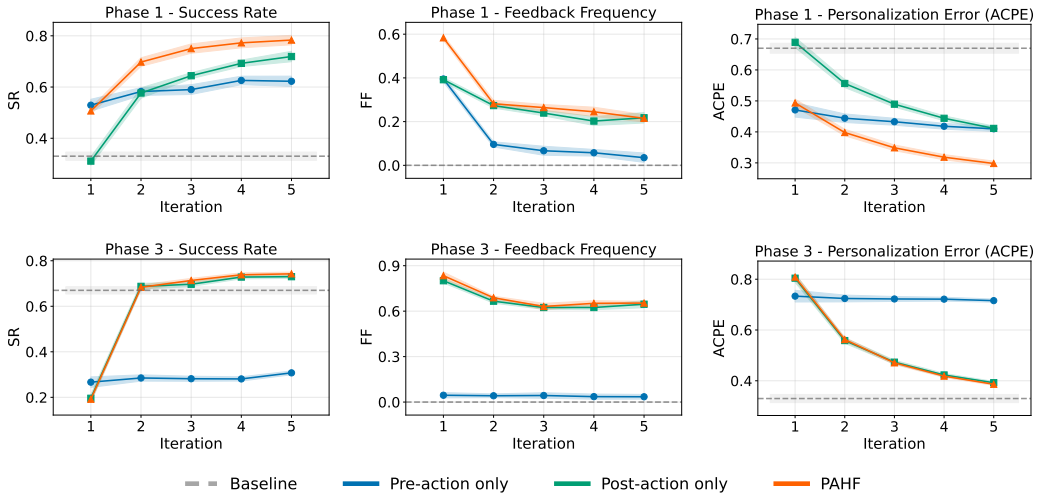

Figure 11: **Embodied manipulation results (Agent: GPT-4o, Human Simulator: Llama-4-Scout, Memory: SQLite).** Top row: Phase 1 learning curves for success rate (left), feedback frequency (middle), and average cumulative personalization error (ACPE) (right). Bottom row: Phase 3 learning curves for the same metrics. Shading denote standard error.

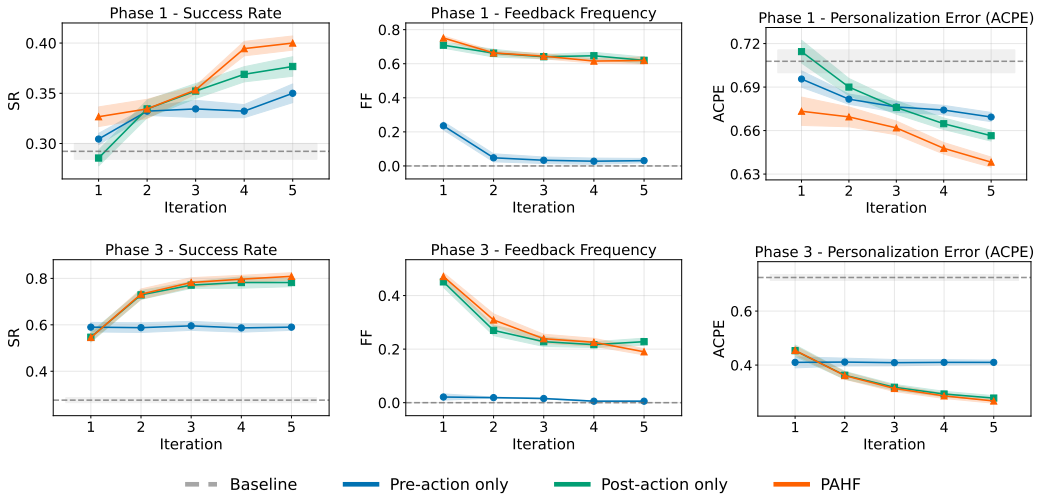

Figure 12: **Online shopping results (Agent: GPT-4.1, Human Simulator: Llama-4-Scout, Memory: SQLite).** Top row: Phase 1 learning curves for success rate (left), feedback frequency (middle), and average cumulative personalization error (ACPE) (right). Bottom row: Phase 3 learning curves for the same metrics. Shaded regions denote standard error.

F.3 Further Analyses

Unless otherwise noted, the experiments in this subsection use the full embodied benchmark (40 users, 30 scenarios per phase) with GPT-4o as both the agent and the human simulator, and three learning iterations in Phases 1 and 3. These are separate runs from those in Section 5, so absolute numbers are not directly comparable with Table 1 and Figures 3–4.

Clarification cost vs. success. We compare three clarification policies that share the same memory pipeline: the Post-action only agent, which never asks; PAHF with its default gate, which asks only when the agent is uncertain; and an Always-ask variant of PAHF, which asks on every task (Table 7). Always-ask attains the highest in-interaction success, but it does not yield better memory: Phase 2 success is nearly identical across the three policies (73.4–73.7%), while Always-ask uses 7,200 questions compared with 570 for PAHF. After preference drift, Always-ask falls behind PAHF in Phase 4 (60.8% vs. 70.1%). Because it rarely errs during Phase 3 (15% error at iteration 1, compared with 81% for PAHF), the error-triggered updates that overwrite stale notes seldom fire, and its memory grows to 227 notes per user (vs. 68 for PAHF), mixing pre- and post-drift preferences. Never asking avoids all questions but incurs more cold-start errors (35.5% vs. 49.0% success at Phase 1 iteration 1) and a higher ACPE in both learning phases (35.4 vs. 29.7 in Phase 1; 45.9 vs. 44.3 in Phase 3). PAHF's question rate decays from 0.24 to 0.04 questions per task over Phase 1 and stays at 0.04–0.06 in Phase 3, where recovery is driven by the post-action channel.

Table 7: **Clarification policy trade-off (embodied).** Success rate (SR, %) and ACPE (%) across the four phases, and total clarification questions over the 7,200 learning tasks (Phases 1 and 3).

	Phase 1		Phase 2	Phase 3		Phase 4	
Clarification policy	SR ↑ (iter. 1 / 3)	ACPE ↓	SR ↑	SR ↑ (iter. 1 / 3)	ACPE ↓	SR ↑	Questions ↓
Post-action only (never asks)	35.5 / 80.8	35.4	73.7	18.3 / 75.2	45.9	69.5	0
PAHF (asks when uncertain)	49.0 / 81.8	29.7	73.4	19.2 / 76.3	44.3	70.1	570
PAHF (always asks)	91.4 / 91.3	8.6	73.6	85.3 / 84.9	14.5	60.8	7,200

Comparison with a static-profile baseline. Many personalization methods, such as MemoryBank (Zhong et al., 2024), construct user models offline from interaction logs. We instantiate this family in our protocol: each user's profile is distilled offline by an LLM from the logged Phase 1 interaction feedback and is then deployed read-only, with no online updates (Table 8). The static profile nearly matches PAHF while preferences are stationary (72.1% vs. 73.4% in Phase 2). After

the persona shift, it drops to 21.2% in Phase 4, below the no-memory baseline (47.2%), because the stale profile misleads the agent; PAHF recovers to 70.1% through post-action feedback.

Table 8: **Static-profile baseline (embodied).** Success rates (%) before (Phase 2) and after (Phase 4) preference drift. The static profile is built offline from logged Phase 1 feedback and never updated.

Method	Phase 2 (before drift)	Phase 4 (after drift)
No memory	36.1	47.2
Static profile from interaction logs	72.1	21.2
PAHF	73.4	70.1

Preference drift vs. context-dependent preferences. Memory notes are natural language and carry their context scope (e.g., "prefers herbal tea when drowsy"). When salient feedback arrives, an update gate (the Preference Drift Detector prompt in Appendix E) decides whether it changes a previously stated preference. If so, the most similar note is rewritten by an LLM merge of the old and new content; otherwise, the feedback is stored as a new note. To measure how well this pipeline separates true drift from context-dependent and temporary preferences, we generated 100 situated feedback utterances for each of three feedback types, each reacting to an action the agent took on the user's standing preference, and traced them through the update pipeline (Table 9). The gate routes all true-drift feedback to revision. It also routes 48% of context-conditional and 38% of one-off feedback to revision, but because revision is a merge rather than a deletion, the merged note typically keeps both scopes (e.g., "generally prefers black coffee, but switches to herbal tea when drowsy"), and the final memory state is correct in 97% and 89% of these cases, respectively. Consistent with this, on the 1,160 context-conditional Phase 2 scenarios, PAHF reaches 72.8% success from memory alone, compared with 35.6% without memory; on general-preference scenarios it reaches 92.5%.

Table 9: **Separating preference drift from context-dependent and temporary preferences.** For 100 feedback utterances of each type, the fraction the update gate routes to revising an existing note, and the fraction for which the memory state is correct after the full update.

Feedback type ($n = 100$ each)	Routed to "revise existing note"	Memory correct after update
True drift (permanent change)	100%	100%
Context-conditional preference	48%	97%
One-off request	38%	89%

Users who cannot name what they want. We modify the human simulator so that users never name the item they want and describe it only indirectly through its properties (e.g., "something soothing and plant-based, without any caffeine" instead of "herbal tea"); the evolved personas in Phases 3 and 4 keep the same communication style. Phase 1 runs five learning iterations for these vague users and three for precise users (Table 10). With vague users, PAHF improves at every Phase 1 iteration (45.2, 56.4, 58.3, 63.7, and 64.5%) and reaches 59.9% in Phase 2 from memory alone, compared with 36.1% for the no-memory baseline (whose success is essentially unaffected by the communication style: 35.8% in Phase 2 with vague users); precise communication still helps (73.4%). After the persona shift, vague users can also correct a confidently wrong agent using only attribute-level feedback (e.g., "I used to reach for the strong, dark option to wake up, but now I go for something fizzy that gives a strong kick instead"). Phase 3 success rises from 28.6% to 60.8% over three iterations (vs. 19.2% to 76.3% for precise users), and Phase 4 reaches 57.8% from memory alone, above the no-memory baseline (47.8%). The drop right after drift is milder for vague users (28.6% vs. 19.2% at iteration 1), and the question rate stays at 0.04–0.05 per task in Phase 3, so recovery is carried by the post-action channel.

Real user language. To test whether the results depend on the simulator's stylized language, for each of the 20 shopping users we sampled a real reviewer from the Amazon Reviews 2023 corpus and attached verbatim excerpts of their reviews to the persona. The simulated user then produces all of its language in that reviewer's voice: clarification answers are written directly in that voice, and the rule-based feedback is re-voiced with option letters, feature names, and accept/reject meaning held fixed; in Phase 3, the preference-change feedback is re-voiced as well. The benchmark itself is unchanged (same scenarios, preferences, and ground-truth labels), so correctness is scored exactly as before. We run all four phases on the full shopping benchmark with GPT-4o and three learning

Table 10: **Users who describe preferences only indirectly (embodied).** Success rates (%) across the four phases. Phase 1 runs three learning iterations for precise users and five for vague users; "first / last" denotes the first and last iterations.

Setting	Phase 1 (first / last iter.)	Phase 2	Phase 3 (iter. 1 / 2 / 3)	Phase 4
No memory	35.5	36.1	50.2	47.8
PAHF, precise users	49.0 / 81.8	73.4	19.2 / 71.6 / 76.3	70.1
PAHF, vague users	45.2 / 64.5	59.9	28.6 / 51.2 / 60.8	57.8

iterations in Phases 1 and 3 (Table 11). The two PAHF conditions differ by at most 2.3 points at every stage, including recovery after the persona shift (73.0% vs. 73.8% at Phase 3 iteration 3), and both remain well above the no-memory baseline (65.4% and 67.2% vs. 31.4% in Phase 4).

Table 11: **Real-reviewer user language (online shopping).** Success rates (%) when all user language is produced in the voice of a real Amazon reviewer vs. the stylized simulator language.

Setting	Phase 1 (iter. 1 / 3)	Phase 2	Phase 3 (iter. 1 / 3)	Phase 4
No memory	25.1	26.0	31.7	31.4
PAHF, stylized simulator language	27.0 / 36.6	34.6	58.3 / 73.0	67.2
PAHF, real-reviewer language	27.7 / 37.7	35.3	56.0 / 73.8	65.4