HYPOTHESES-GUIDED SELF DISTILLATION FOR CONTINUAL PERSONALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

As people increasingly interact with LLM assistants in daily life, continually adapting to individual preferences has become essential for effective long-term interactions. However, user preferences are rarely stated in full, and instead emerge through heterogeneous, latent, and noisy signals, with existing methods relying on raw interaction histories or costly reward-based optimization to manage personalization. We introduce HYPREFLECT, a framework that enables continual personalization by inferring explicit, uncertainty-aware preference hypotheses from diverse user signals, reflectively refining them as new evidence accumulates, and incorporating the resulting user model through hypotheses-guided self-distillation. Experiments across three personalization settings: online personalization, multi-session interactions, and implicit behavioral signals, show that HYPREFLECT outperforms a range of baselines, including raw-history and incremental-update methods. We further demonstrate strong generalization to unseen users and cross-domain settings, along with stability across context budgets, reusable hypotheses, and more focused personalization. These results suggest a step towards reliable and interpretable continual personalization through explicit, revisable user preference hypotheses.[1]

1 INTRODUCTION

Conversational assistants that continuously adapt to individual users can provide increasingly effective and personalized support in diverse settings, including writing assistance (Mysore et al., 2024), conversational recommendation (Liang et al., 2024; Kim et al., 2025; Zhu et al., 2025), and education (Liu et al., 2025a). Personalization allows assistants to better address a user's needs (Zhang et al., 2018; Salemi et al., 2024; Zhang et al., 2025; Zhao et al., 2025b), while continual adaptation enables long-term interactions with users by improving relevance, as assistants develop a richer understanding of users over time (Xu et al., 2022; Zhong et al., 2024; Li et al., 2025).

Achieving such continual personalization requires an assistant to understand the user from diverse interaction signals, and maintain as well as revise that understanding throughout future interactions. However, users rarely articulate their preferences in full, and may not always be consciously aware of them. Figure 1 illustrates how evidence about a user's preferences emerges incrementally from explicit feedback, implicit behavior, and user-profile information, requiring it to be consolidated into an explicit representation that can be revised as new evidence arrives. These signals are often sparse, noisy, and context dependent, making it difficult to distinguish stable preferences from task-specific requirements. Moreover, evidence may be distributed across long multi-turn and multi-session histories (Maharana et al., 2024; Wu et al., 2025), which language models struggle to track consistently as conversations grow (Laban et al., 2025). Retrieval-based methods surface relevant past interactions (Salemi et al., 2024; Liu et al., 2025c), but incur repeated cost over growing histories and struggle with gradually emerging preferences. Summary-based methods compress user histories (Zhang, 2024; Ramos et al., 2024), but fail to capture latent and evolving preferences (Su et al., 2026; Shi et al., 2025). Approaches that infer personas (Balepur et al., 2025) provide a more explicit representation, but do not address how such representations should be continually revised. Therefore, the central problem is to transform an evolving stream of interactions into a user representation that is

[1] Code will be released upon acceptance.

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.

both reliable and revisable, while consolidating heterogeneous evidence and reconsidering earlier beliefs when they become uncertain or contradicted.

This representation of inferred preferences should further guide the model response. Inference-time methods retrieve and interpret user information during generation (Shen et al., 2026), but do not learn how preferences should shape future responses. Supervised fine-tuning needs personalized target responses, that are generally unavailable in continual interactions. RL-based methods, such as Group Relative Policy Optimization (GRPO; Shao et al., 2024), require reward models and repeated sampling, making personalization costly (Wan et al., 2025; Nam et al., 2026). Self-distillation offers a scalable alternative (Buening et al., 2026), but relies on raw histories, leaving unclear which preferences are learned and whether they generalize to future contexts.

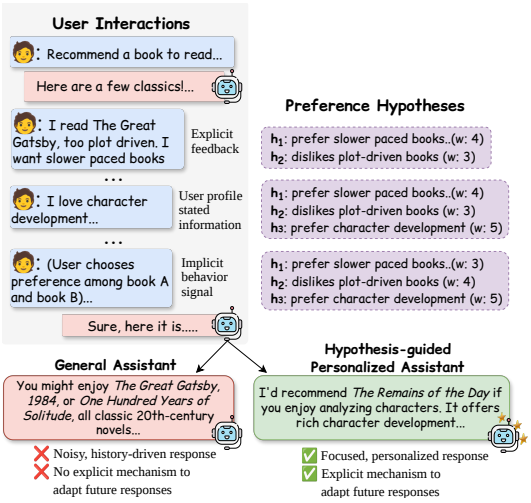


Figure 1: Preference hypotheses are generated across personalization settings from explicit feedback, implicit behavioral signals, and user profile information, along with confidence weights. They maintain explicit, revisable beliefs about the user and guide models to produce personalized responses.

In this work, we introduce HYPREFLECT, a framework for continual personalization that learns from heterogeneous user signals while maintaining explicit, uncertainty-aware, and revisable preference hypotheses. HYPREFLECT first distills local preference signals from each interaction into explicit hypotheses, for an interpretable representation of user preferences. It then reflectively refines the hypotheses across interactions so that earlier beliefs can be reconsidered as new evidence accumulates, enabling continual adaptation to evolving user preferences. Finally, the refined hypotheses guide self-distillation, teaching the model how to use the inferred preferences in future responses, without requiring ground-truth dialogue responses or an external reward model.

We evaluate HYPREFLECT across three personalization settings covering explicit feedback, multi-session interactions, and implicit behavioral choices, against a comprehensive suite of baselines, including GRPO, self-distillation over raw interaction histories, incrementally updated summaries and hypotheses, and reflective prompting. Across these settings, HYPREFLECT achieves relative improvements of up to 4.4%, 10.3%, and 5.0% respectively[2], while generalizing effectively to unseen users, shifted user characteristics, and cross-domain personalization. Further analyses demonstrate robustness across context budgets, hypothesis reusability, and more focused personalization. Overall, our findings show that reflective refinement enables reusable and revisable preference hypotheses over long interaction histories, and that models benefit substantially from these explicit user representations, enabling reliable and scalable continual personalization.

2 PROBLEM FORMULATION

Continual Personalization considers an assistant that repeatedly interacts with a user. At interaction t, the assistant observes the preceding conversation history $C_{<t}$ and the current user turn u_t, which may contain feedback f_{t-1} on the preceding assistant response, a new request q_t, or both. The assistant then generates a response: $y_t \sim \pi_\theta(\cdot \mid C_{<t}, u_t)$. As interactions accumulate, the history provides increasing signals about the user's preferences. The goal therefore is to continually infer, maintain, and use this understanding to personalize future responses.

Self-Distillation for Continual Personalization. Self-distillation uses the same model as both a student and a teacher, with the teacher conditioned on richer information. For continual personalization, self-distillation offers a simple and scalable alternative to costly reward optimization by

[2]Maximum relative improvement per dataset with HYPREFLECT+SUM in Table 1.

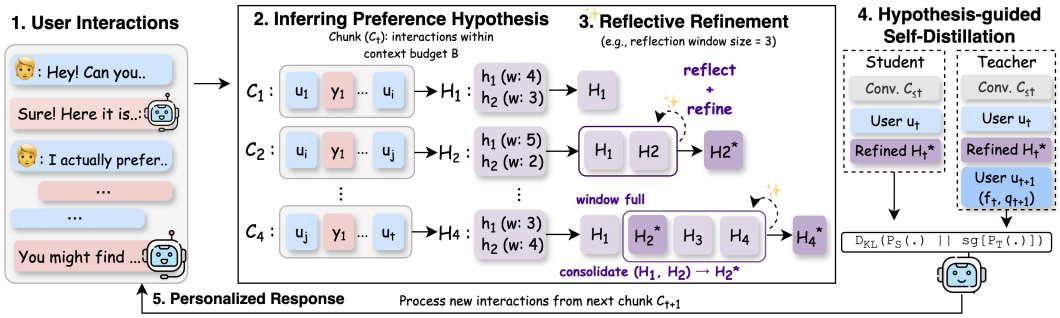


Figure 2: Overview of HYPREFLECT. Preference hypotheses are generated from context-bounded chunks (Steps 1–2) and reflectively refined over time (Step 3). The refined user state conditions both student and teacher during self-distillation, with the teacher seeing the next user turn as privileged information (Step 4). The resulting model produces continually personalized responses (Step 5).

treating the subsequent user turn $u_{t+1} = \{f_t, q_{t+1}\}$ as privileged information when ground-truth personalized responses are unavailable (Buening et al., 2026).

Specifically, the student distribution is $\pi_t^{\mathrm{S}} = \pi_\theta(\cdot \mid C_{<t}, u_t)$, while the privileged teacher distribution is $\pi_t^{\mathrm{T}} = \pi_\theta(\cdot \mid C_{<t}, u_t, u_{t+1})$. The teacher's feedback-informed predictions are distilled into the student by minimizing

$$\mathcal{L}_{\mathrm{SD}}(\theta) = \sum_t D_{\mathrm{KL}}\left(\pi_t^{\mathrm{S}} \middle\| \mathrm{stopgrad}\left[\pi_t^{\mathrm{T}}\right]\right).$$

This enables learning from preferences revealed by subsequent interactions without an external teacher or reward model. However, directly internalizing preferences from raw interaction histories may not generalize well beyond the current interaction.

Preference Hypotheses Representation. Hence, we introduce an explicit abstraction layer between raw interactions and response generation. We represent the model's understanding of the user as a set of preference hypotheses H_t and formulate personalized response generation as $y_t \sim \pi_\theta(\cdot \mid C_{<t}, q_t, H_t)$. This representation makes inferred preferences explicit, reusable, and revisable as new evidence arrives (Liu et al., 2026). The key challenge here is how to construct hypotheses from local interaction signals, refine and maintain them over long histories, and use them to guide self-distillation.

3 HYPREFLECT: CONTINUAL USER MODELING THROUGH REFLECTIVE REFINEMENT

To support continual personalization, we propose HYPREFLECT, a framework that introduces an explicit preference hypothesis layer to better understand and respond to the user. Our framework consists of three stages. First, the model infers preference hypotheses from individual interaction chunks, capturing localized signals about the user (§3.1). Second, it recurrently refines these hypotheses by reflecting and consolidating user signals across chunks (§3.2). Finally, the refined hypotheses guide self-distillation, enabling the model to use subsequent user signals as privileged supervision for learning personalized response behavior (§3.3). We also consider a summary-augmented variant that jointly maintains summaries and hypotheses.

3.1 INFERRING PREFERENCE HYPOTHESES FROM INTERACTIONS

Hypotheses representation. To capture diverse interaction signals from the user (e.g., Fig. 1), we represent the model's understanding of the user as a revisable set of m preference hypotheses $H_t = \{h_{t1}, \ldots, h_{tm}\}$, where each candidate hypothesis $h_{tj} = (s_{tj}, w_{tj})$ pairs a natural-language statement s_{tj} describing a potential preference with a verbalized confidence score $w_{tj} \in \{1, \ldots, 5\}$. Maintaining multiple hypotheses preserves alternative beliefs that can be revised as new evidence arrives, while confidence distinguishes well-supported preferences from uncertain ones. This natural-

language representation makes the hypotheses interpretable, revisable, and directly usable for personalized response generation.

Inferring local hypotheses from interaction chunks. Inferring preferences from the full interaction history is often impractical due to context-length limitations and, in online settings, because future interactions are not yet available. We therefore partition the observed interaction history into consecutive chunks $\{C_1, \ldots, C_N\}$, each containing the longest sequence of complete interactions within a token budget B. For each chunk C_i, we infer a set of *local hypotheses* (Step 2 in Fig. 2) as

$$H_t = \textsc{GenerateHypotheses}(C_t) = \{h_{tj}\}_{j=1}^{m}.$$

The model is instructed to infer reusable preferences grounded in the user's instructions, corrections, feedback, and choices, while avoiding interaction-specific restatements and unsupported generalizations. We additionally define a summary-augmented variant, where we generate a local summary S_i for each interaction chunk. Summaries preserve relevant conversational context from C_i that grounds the hypotheses, whereas H_i captures reusable preferences that are explicitly stated or inferred from user signals.

3.2 Maintaining User Understanding through Reflective Refinement

As interactions accumulate, the model must maintain preference hypotheses that (1) *explain past interaction histories* and (2) *predict future user behavior*. Simply using the most recent hypotheses discards earlier preferences, whereas repeatedly reconsidering the full interaction history is computationally infeasible. A refinement mechanism should therefore preserve long-term preferences while remaining revisable and bounded in computational context size.

A simple approach is incremental updating, which iteratively refines a single running user state. However, since each update operates on an already compressed representation, over time it is either prone to error accumulation, or it may have valid earlier preferences overwritten by recent observations. We empirically show that this leads to unstable long-term personalization (§5.1).

Reflective refinement across chunks. To preserve evidence across multiple chunks, without relying exclusively on a recursively compressed state, we propose **reflective refinement** (Step 3 in Fig. 2). Rather than immediately merging each new local hypothesis set into a single running state, the model jointly reconsiders multiple independently generated local sets. By comparing hypotheses across interaction chunks, the model consolidates consistent evidence, resolves conflicts, merges redundant hypotheses, and revises beliefs that are no longer supported.

To keep refinement tractable, we maintain a bounded reflection window. Let H_i denote the local hypothesis set inferred from chunk C_i, H_i^* the refined hypothesis set after observing the first i chunks, and W the reflection-window size. We construct the refinement input as

$$\mathcal{I}_i = \begin{cases} [H_1, \ldots, H_i], & i \leq W, \\ [H^*_{i-W+1}, H_{i-W+2}, \ldots, H_i], & i > W, \end{cases}$$

and update the maintained hypothesis set as $H_i^* = \textsc{ReflectAndRefine}(\mathcal{I}_i)$.

When $i \leq W$, the model reflects over all available local hypothesis sets. Thereafter, it replaces older local sets with their refined representation while retaining the most recent $W - 1$ local sets. This keeps the refinement input bounded without discarding earlier preference information.

For our summary-augmented variant, we apply the same windowed recurrent refinement procedure to a jointly maintained summary and hypothesis set, $Z_i = (S_i, H_i)$, for each chunk, producing $Z_i^* = (S_i^*, H_i^*)$. The refined summary S_i^* consolidates the context used to ground and revise the hypotheses, while H_i^* maintains the refined preferences used for future personalization.

3.3 Hypothesis-Guided Self-Distillation

To effectively use the refined user state for personalized response generation, we incorporate it into self-distillation, training the model to translate maintained preferences into personalized responses. Building on the formulation in §2, we condition both the teacher and student on the same refined

state, providing a shared understanding of the user and making the teacher's targets learnable by the student. The teacher additionally observes the subsequent user turn $u_{t+1} = (f_t, q_{t+1})$ and uses it as privileged hindsight to generate a preference-grounded response.

At interaction t, given history $C_{<t}$, current turn u_t, refined hypotheses H_t^*, and subsequent turn u_{t+1}, the student distribution is $P_S(y_t) = \pi_\theta(y_t \mid C_{<t}, u_t, H_t^*)$ and the teacher distribution is $P_T(y_t) = \pi_\theta(y_t \mid C_{<t}, u_t, H_t^*, u_{t+1})$. The teacher uses u_{t+1} to provide feedback-informed supervision. We then minimize the reverse KL divergence from the student to the detached teacher:

$$\mathcal{L}_{\text{distill}} = \mathbb{E}_t \left[D_{\text{KL}} \left(P_S(\cdot) \,\|\, \text{stopgrad}[P_T(\cdot)] \right) \right].$$

The objective is computed token-wise over the generated response (Step 4 in Fig. 2).

4 Experimental Setup

4.1 Datasets and Evaluation

To test whether models can infer reusable user preferences from diverse interaction signals and use them to personalize future responses, we evaluate three complementary settings: continual adaptation from explicit user feedback, multi-session personalization using conversational user profiles, and preference inference from implicit behavioral signals. See Appendix B.1 and B.3 more details.

HelpSteer2 (Wang et al., 2024). To evaluate continual adaptation from **explicit user feedback**, we construct an online personalization setting from HelpSteer2. Each user is associated with a set of writing-style preferences, which are gradually revealed across multiple interactions via the user simulator providing feedback on one or two relevant preference dimensions after each model response.
`Evaluation` Models are trained over 250 interactions and evaluated every 50 interactions on held-out prompts using an LLM judge. We report pairwise win rates against the base model with respect to the target user's preferences.

HiCupid (Mok et al., 2025). To test personalization from **user profile information**, we use HiCupid, where user preferences and attributes are stated naturally across multi-session conversations. We train a model on QA interactions from 300 users and evaluate whether it can use previous user information to personalize future responses.
`Evaluation` We evaluate both seen-user and unseen-user generalization using held-out prompts and report pairwise win rates from an LLM judge[3] assessing personalization and response quality.

Flight Recommendation (Qiu et al., 2026). To evaluate preference inference from **implicit behavioral signals**, we use a Flight Recommendation task. We utilize multi-turn recommendation trajectories from 624 users, with up to 25 interactions per user, where user choices provide partial evidence about latent preferences. These preferences may include non-obvious trade-offs, requiring models to infer user-specific behavior rather than rely on fixed assumptions.
`Evaluation` We report final-round preference prediction accuracy on the original flight task, held-out feature settings, and cross-domain hotel transfer.

4.2 Model Variants and Training Setup

(1) ***Base.*** Uses the vanilla language model without fine-tuning, conditioned on the relevant raw interaction history, to generate a response. (2) ***GRPO.*** Samples four responses conditioned on the relevant raw interaction history. A reward model uses the user's hidden preferences to score each response's preference alignment (Appendix B.2 for details). (3) ***SD.*** Self-distillation is performed directly on the raw interaction history. When the history exceeds the context budget, only the most recent portion is retained; (4) ***SD-IncSum.*** The model is trained with an incrementally updated summary: a running summary S_i is maintained and refreshed after each chunk; (5) ***SD-IncHyp.*** Instead of a summary, the model maintains a running hypothesis set H_i, updated with the same incremental strategy; (6) **HypReflect.** Local hypotheses H_i are first generated from individual chunks and then consolidated into a refined set H_i^* through the reflective refinement process described in §3.2; (7) **HypReflect+Sum.** Reflective refinement is applied to the paired memory

[3]We use `deepseek-v4-flash` as the judge; Appendix C reports a cross-judge consistency analysis on a HelpSteer2 subset.

state $Z_i = (S_i, H_i)$, yielding $Z_i^* = (S_i^*, H_i^*)$, to examine whether summaries supply complementary information during hypothesis refinement.

Training setup. We use Qwen3.5-4B, Qwen3.5-9B (Yang et al., 2025), and Gemma4-4B (Team et al., 2026) as backbone LLMs. Additional hyperparameter and training details are provided in Appendix B.2.

5 RESULTS

Table 1: Performance on three continual-personalization benchmarks covering continual adaptation with explicit user feedback (HelpSteer2), user-profile information across multiple sessions (HiCupid), and implicit behavioral signals (Flight). We report win rate against Base for HelpSteer2 (averaged over 50–250 interactions) and HiCupid, and accuracy for Flight. (↑) and (↓) are relative to SD.

	HelpSteer2 (Win Rate ↑)			HiCupid (Win Rate ↑)			Flight Recommendation (Acc. ↑)		
Method	Gemma4-4B	Qwen3.5-4B	Qwen3.5-9B	Gemma4-4B	Qwen3.5-4B	Qwen3.5-9B	Gemma4-4B	Qwen3.5-4B	Qwen3.5-9B
Base	–	–	–	–	–	–	40.9	40.3	41.7
GRPO	74.2	74.5	78.5	52.5	54.7	52.0	37.3	55.9	60.1
SD	87.0	84.5	89.0	57.3	69.7	63.3	61.8	73.8	74.4
SD-IncSum	85.8 (−1.2 ↓)	81.0 (−3.5 ↓)	86.6 (−2.4 ↓)	58.4 (+1.1 ↑)	67.9 (−1.8 ↓)	67.7 (+4.4 ↑)	50.5 (−11.3 ↓)	74.4 (+0.6 ↑)	67.0 (−7.4 ↓)
SD-IncHyp	89.6 (+2.6 ↑)	81.9 (−2.6 ↓)	88.2 (−0.8 ↓)	60.7 (+3.4 ↑)	71.6 (+1.9 ↑)	66.2 (+2.9 ↑)	57.3 (−4.5 ↓)	**74.8** (+1.0 ↑)	75.4 (+1.0 ↑)
HYPREFLECT	86.3 (−0.7 ↓)	**88.9** (+4.4 ↑)	90.3 (+1.3 ↑)	61.1 (+3.8 ↑)	71.1 (+1.4 ↑)	67.7 (+4.4 ↑)	62.3 (+0.5 ↑)	74.0 (+0.2 ↑)	**76.0** (+1.6 ↑)
HYPREFLECT+SUM	**90.0** (+3.0 ↑)	88.2 (+3.7 ↑)	**92.1** (+3.1 ↑)	**61.9** (+4.6 ↑)	**71.8** (+2.1 ↑)	**69.8** (+6.5 ↑)	**64.9** (+3.1 ↑)	74.2 (+0.4 ↑)	75.3 (+0.9 ↑)

5.1 MAIN RESULTS

Preference hypotheses provide a stronger representation than summaries. Table 1 compares performance across three personalization benchmarks. Under the incremental update strategy, SD-INCHYP outperforms SD-INCSUM in eight of nine settings, showing that abstracting interaction signals into reusable preference beliefs is generally more effective than retaining them as summaries. HYPREFLECT further improves over SD-INCHYP and outperforms SD in eight of nine settings, indicating that the effectiveness of preference hypotheses also depends on maintaining them accurately and consistently over time.

Reflective refinement makes personalization more reliable. Furthermore, HYPREFLECT outperforms SD in eight of nine settings and HYPREFLECT+SUM improves over it in all nine. HYPREFLECT+SUM also outperforms HYPREFLECT in seven settings, suggesting that summaries provide complementary evidence omitted during hypothesis abstraction and ground refinement in both observed interactions and inferred preferences. In contrast, SD-INCSUM and SD-INCHYP underperform SD in six and three settings, respectively. We find that incremental updates can propagate earlier errors and allow recent observations to overwrite valid prior information, resulting in unstable supervision over time. In contrast, reflective refinement maintains more reliable preference hypotheses as interaction histories grow.

Self-distillation teaches more effectively than GRPO. In addition, self-distillation substantially outperforms GRPO in Table 1 while training approximately $15\times$ faster; see the efficiency comparison in Appendix Table 9. These results suggest that distilling token-level teacher distributions provides more effective supervision for multi-turn personalization. At inference time, our method generates an average of 21.4 more tokens than GRPO and 24.6 more tokens than the base model, with a latency of 0.40 seconds per query compared with 0.14 for the base model. Appendix B.2 provides hyperparameter details and further training and evaluation efficiency comparisons.

Table 2: Comparison of HYPREFLECT against Base model and Base+RefHyp (base model reflective prompting approach).

	HiCupid (Win% ↑)		**Flight Rec.** (Acc. ↑)		
Model	vs. Base	vs. +RefHyp	Base	+RefHyp	HYPREFLECT
Gemma4-4B	61.1	59.7	40.9	47.9	**62.3**
Qwen3.5-4B	71.1	62.2	40.3	42.0	**74.0**
Qwen3.5-9B	67.7	57.6	41.7	46.8	**76.0**

Reflective hypotheses prompting yields consistent gains, which self-distillation further amplifies. Table 2 shows the effectiveness of reflective refinement with and without self-distillation. We implement BASE+REFHYP similarly to HYPREFLECT, refining preference hypotheses to generate responses without self-distillation. Even when used only through prompting, reflectively refining preference hypotheses improves the base model, with BASE+REFHYP gaining an average of 4.6 accuracy points on Flight Recommendation and

Table 3: Generalization results. Flight→Features: robustness across 2–8 features (with models trained on 4 features). Flight→Hotel: cross-domain transfer. HiCupid Unseen: performance on unseen users. (↑) and (↓) are relative to SD.

	Flight → Features (Acc. ↑)			Flight → Hotel (Acc. ↑)			HiCupid Unseen (Win Rate ↑)		
Method	Gemma4-4B	Qwen3.5-4B	Qwen3.5-9B	Gemma4-4B	Qwen3.5-4B	Qwen3.5-9B	Gemma4-4B	Qwen3.5-4B	Qwen3.5-9B
GRPO	38.8	51.8	54.4	40.0	53.1	55.9	51.5	53.1	51.0
SD	55.4	64.9	63.7	54.8	61.8	62.0	57.0	71.7	66.9
SD-IncSum	47.5 (−7.9 ↓)	65.6 (+0.7 ↑)	57.1 (−6.6 ↓)	48.0 (−6.8 ↓)	63.7 (+1.9 ↑)	57.1 (−4.9 ↓)	58.7 (+1.7 ↑)	69.9 (−1.8 ↓)	69.4 (+2.5 ↑)
SD-IncHyp	53.4 (−2.0 ↓)	64.8 (−0.1 ↓)	65.5 (+1.8 ↑)	52.8 (−2.0 ↓)	62.3 (+0.5 ↑)	62.9 (+0.9 ↑)	62.1 (+5.1 ↑)	71.8 (+0.1 ↑)	68.0 (+1.1 ↑)
HypReflect	55.6 (+0.2 ↑)	**65.8** (+0.9 ↑)	**66.0** (+2.3 ↑)	55.7 (+0.9 ↑)	63.6 (+1.8 ↑)	**63.8** (+1.8 ↑)	**62.9** (+5.9 ↑)	71.6 (−0.1 ↓)	68.1 (+1.2 ↑)
HypReflect+Sum	**57.7** (+2.3 ↑)	65.0 (+0.1 ↑)	65.9 (+2.2 ↑)	**56.5** (+1.7 ↑)	**64.1** (+2.3 ↑)	63.7 (+1.7 ↑)	62.6 (+5.6 ↑)	**73.3** (+1.6 ↑)	**70.3** (+3.4 ↑)

narrowing the gap to HypReflect on HiCupid, reducing the average win rate of HypReflect from 66.6% against Base to 59.8%. Combining reflective refinement with hypothesis-guided self-distillation yields substantially larger gains, as HypReflect outperforms Base+RefHyp by an average of 25.2 accuracy points on Flight Recommendation and achieves an average win rate of 59.8% against it on HiCupid. These results show that reflective refinement is effective even as inference-time prompting and combining it with self-distillation more effectively incorporates user preferences into personalized response behavior.

5.2 Generalization Results

Reflective refinement improves robustness under distribution shift. Table 3 reports results on unseen HiCupid users and two Flight Recommendation transfer settings, varying feature counts (2–8) after training on 4 features and transfer to the Hotel domain. Reflective methods remain the most robust. HypReflect+Sum improves over SD in all nine settings, often by substantial margins, and HypReflect outperforms SD in eight out of nine settings. In contrast, SD-IncSum produces large drops on the Flight transfer tasks. SD-IncHyp is more stable and improves performance in most cases, yet still underperforms the reflective approaches on several Flight settings. These results show that reflective refinement produces higher-quality hypotheses, providing more reliable conditioning signals during self-distillation, allowing the model to learn a cleaner mapping from user behavior to underlying preferences.

Self-distillation and GRPO offer complementary strengths. Although GRPO underperforms self-distillation overall, it shows smaller performance drops from seen to unseen tasks. While self-distillation shows a larger decline on unseen tasks, consistent with the findings of Kim et al. (2026), it still performs substantially better than GRPO. These results suggest complementary strengths of the two approaches, where GRPO better preserves performance on unseen tasks and self-distillation more effectively teaches responses aligned with user preferences.

5.3 Further Analysis

Reflective refinement produces reusable user hypotheses. Table 4 presents our analysis on the quality of the user hypotheses learned by HypReflect through the hypothesis-transfer experiment. For each user, we take the final hypothesis generated by the HypReflect model and provide it to the corresponding non-finetuned base model for answer generation. During evaluation, each base model receives the hypothesis generated for the same user. The transferred hypotheses substantially improve performance across all models on both datasets, yielding average gains of 11.5 accuracy points on Flight Recommendation and 10.0 winrate points on HiCupid. These results indicate that our reflective refinement approach produces informative and reusable user hypotheses, suggesting that the resulting personalization benefits may extend beyond the model used to generate them.

Table 4: Evaluation of the quality of the final user hypotheses generated by HypReflect, by transferring them to non-finetuned base models.

Model	Flight (Acc.)		HiCupid (Win Rate)	
	Base	w/ Refined Hyp.	Base	w/ Refined Hyp.
Gemma4-4B	40.9	**55.1**$_{+14.2}$	50.0	**53.7**$_{+3.7}$
Qwen3.5-4B	40.3	**48.8**$_{+8.5}$	50.0	**63.4**$_{+13.4}$
Qwen3.5-9B	41.7	**53.6**$_{+11.9}$	50.0	**62.8**$_{+12.8}$

Hypothesis conditioning enables more focused personalization. We further present qualitative analyses between SD and HypReflect in Appendix D.3. On HiCupid, explicit hypotheses help HypReflect emphasize the most relevant preference (e.g., narrative-driven explanations), while

SD remains more generic. On Flight Recommendation, they help prioritize the strongest discriminative preference (e.g., shorter duration) leading HYPREFLECT to select the correct option. These examples suggest that shared hypothesis conditioning enables the student to better capture the user's style and preferences, leading to more targeted and personalized responses. This observation further suggests the shared abstraction layer may facilitate the distillation of personalized behaviors from the teacher to the student.

Effect of reflection before consolidation. Table 5 presents an ablation on HiCupid and Flight Recommendation to isolate the effect of the reflection step. For HiCupid, we evaluate users with long interaction histories. Our refinement step first reflects on the inferred hypothesis sets $\mathcal{I}$ and then consolidates them into an updated hypothesis set H^* based on reflection. We compare direct consolidation of $\mathcal{I}$ without reflection. Adding reflection before consolidation yields modest but consistent average gains of 1.0 and 2.8 points on HiCupid and Flight Recommendation, respectively, over direct consolidation of locally inferred hypothesis sets. Our qualitative analyses on both datasets in Appendix D.2 show that direct consolidation can retain broad, overlapping, or redundant preferences, whereas reflection before consolidation reorganizes the accumulated evidence into more specific and informative hypotheses. On Flight Recommendation, reflection also reevaluates recent behavioral evidence before consolidation, correcting misleading local inferences.

Table 5: Performance of HYPREFLECT with and without reflection during refinement.

Dataset	Method	Gemma4-4B	Qwen3.5-4B
HiCupid	HYPREFLECT w/o refl.	72.8	83.8
	HYPREFLECT	**74.2**	**84.5**
Flight	HYPREFLECT w/o refl.	59.0	74.2
	HYPREFLECT	**62.9**	**76.0**

5.4 EFFECT OF CONTEXT SIZE AND REFLECTION WINDOW

Reflective refinement stabilizes online personalization across context budgets. Figure 3 presents the effect of the context budget B on HelpSteer2 in an online setting, where performance is evaluated as the model incrementally receives new evidence about the user. Since responses can contain up to 512 tokens, we evaluate context budgets of 2048 and 4096 tokens. HYPREFLECT adapts quickly and remains stable under both budgets. With 2048 tokens, SD and SD-INCHYP show unstable early performance, especially for Qwen3.5-4B; for Gemma4-4B, both initially perform well but degrade as interaction history grows. Increasing the budget to 4096 improves SD-INCHYP, but it still underperforms HYPREFLECT, while SD again degrades during later interactions. These results suggest that reflective refinement improves robustness by reconsidering evidence across interaction chunks, whereas retaining more tokens alone does not ensure better online personalization.

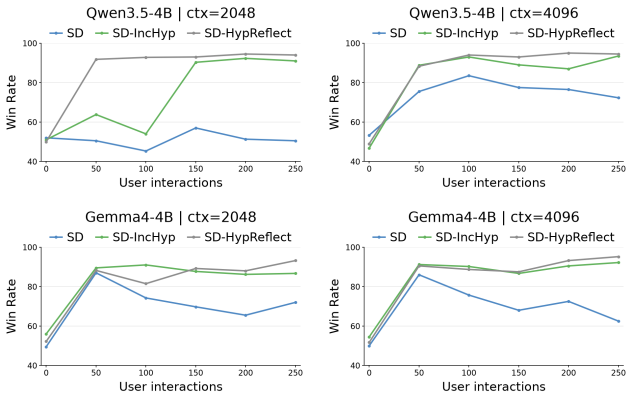


Figure 3: Win rate across user interactions under different context window sizes (2048, 4096) with Qwen3.5-4B and Gemma4-4B on HelpSteer2.

Preference inference from implicit behavior benefits as context budgets grow. Figure 4 examines the effect of context budget on Flight Recommendation, where user preferences are inferred from implicit behavioral signals. Flight uses fixed histories of short, structured choices over recurring attributes, so we evaluate smaller context budgets of 512, 1024, and 2048 tokens. Performance improves with larger context budgets, suggesting that additional context provides more evidence for inferring preferences from implicit behavioral signals. Across models and budgets, reflective refinement performs best in most settings, with clearer gains for Gemma4-4B and Qwen3.5-9B. Qwen3.5-4B, however, performs best with SD-INCHYP at larger budgets, suggesting that more context allows each user state to capture sufficient evidence, reducing the effects of noisy compression. These results demonstrate the robustness of reflective refinement across context budgets, supporting efficient continual personalization by reducing reliance on long, noisy interaction histories.

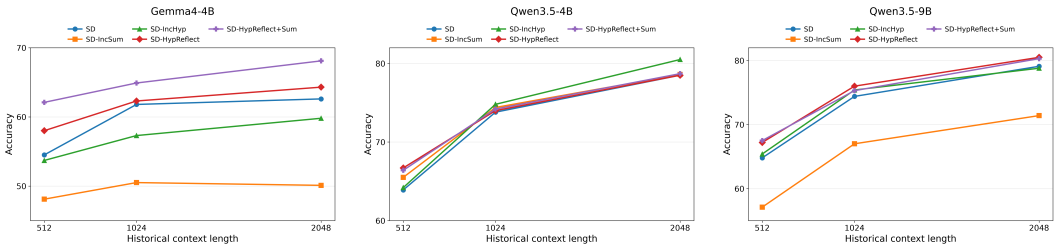


Figure 4: Accuracy across different context lengths for difference models on Flight.

Effect of reflection window size. Appendix Figure 10 compares different reflection window sizes on a subset of HelpSteer2. The window controls how many previous hypothesis sets are considered during refinement. Smaller windows are unstable early on but improve as evidence accumulates. A window of 5 enables quick adaptation and maintains stability, while increasing it to 10 yields little additional gain. We use a window of 5 across datasets and find it performs consistently well, indicating a reasonable default across datasets.

6 Related Work

Personalization in Conversational LLMs Personalized LLMs adapt responses as users' preferences, goals, and behaviors evolve. Prior work explores prompting, parameter adaptation, and alignment-based techniques (Zhang et al., 2025; Liu et al., 2025b; Hwang et al., 2023), while maintaining user representations through personas (Zhang et al., 2026a; Hwang et al., 2024), textual summaries (Nam et al., 2026), or RL-based profile updates (Zhao et al., 2025a). However, these methods typically target specific personalization setting rather than continually integrating diverse user signals. HypReflect studies continual personalization across diverse user signals, including feedback-driven, multi-session, and behavioral settings.

User Modeling User modeling aims to infer and represent user preferences, goals, and behaviors for personalized interactions. Prior work models users through profiles, behavioral patterns, and predefined attributes (Purificato et al., 2024). More recent approaches treat user understanding as an evolving inference process by maintaining beliefs over user goals (Deng et al., 2026), refining hypotheses through exploration-exploitation objectives (Zhou et al., 2024), inferring communication styles (Garbacea & Tan, 2025), explicitly modeling mental-state hypotheses (Hwang et al., 2026), and updating them through Bayesian inverse planning (Zhang et al., 2026b). However, these approaches often model specific aspects of user state rather than a unified, revisable representation integrating heterogeneous evidence over time.

Feedback-Driven Learning and Adaptation Recent work uses hindsight feedback to improve model behavior by distilling privileged information such as verified solutions (Zhao et al., 2026), combining self-distillation with policy optimization and external feedback (Hübotter et al., 2026), or leveraging subsequent user responses for alignment (Buening et al., 2026). Other methods adapt through interaction by refining user representations (Wan et al., 2025; Mehri et al., 2026) or modeling uncertainty over user hypotheses (Puri et al., 2026). However, these approaches often rely on task-specific feedback or costly optimization, limiting scalable long-horizon personalization. HypReflect instead uses user signals as privileged supervision while maintaining an evolving preference representation across interactions.

7 Conclusion

We introduce HypReflect, a reliable, interpretable framework for continual personalization that integrates heterogeneous signals from user conversations and feedback into explicit, uncertainty-aware, and revisable preference hypotheses. Our framework reflectively refines these hypotheses as evidence accumulates and incorporates inferred preferences into the model through hypothesis-guided self-distillation, maintaining a reliable and adaptable user model to guide future responses. Across three personalization settings spanning from implicit behavioral signals to multi-session interactions, HypReflect consistently outperforms approaches based on GRPO, raw histories, or incremental updates and remains stable across context budgets. Our work takes a step toward more effective continual personalization by turning heterogeneous user signals across diverse conversation settings into explicit, revisable hypotheses that support reliable adaptation over time.

AI USE STATEMENT

We used ChatGPT solely for language editing and proofreading during the preparation of this manuscript. Specifically, it was used to identify grammatical errors, improve sentence structure, enhance clarity and readability, and refine wording and phrasing. ChatGPT was not used to generate, develop, or substantively modify the research ideas, methodology, experimental design, analysis, results, or conclusions. All scientific content, interpretations, claims, technical decisions, code, and experiments were developed, implemented, and verified by the authors. The authors reviewed and approved all AI-assisted edits and remain fully responsible for the final manuscript.

ETHICS STATEMENT

Continual personalization systems rely on accumulating information from user interactions, which raises important privacy considerations. In real-world deployments, user data used to infer preferences should be collected with appropriate consent, handled securely, and subject to user control over storage, modification, and deletion. Although HYPREFLECT maintains explicit preference hypotheses rather than directly exposing raw interaction histories, these representations may still encode sensitive information about users and should be treated as private user data.

Personalization systems also introduce risks when inferred preferences are inaccurate, outdated, or overly generalized from limited evidence. Incorrect user models may lead to responses that reinforce assumptions about users or reduce their ability to explore alternative behaviors. Practical deployments of systems like HYPREFLECT should incorporate mechanisms for preference correction and transparency to ensure that inferred preferences remain aligned with users' intended goals.

Personalization may further amplify sycophancy, as models can interpret user preferences or prior feedback as signals to agree with the user's beliefs rather than merely adapt their responses. Over repeated interactions, this behavior may create a feedback loop in which the model increasingly reinforces the user's existing views, including inaccurate, biased, or harmful beliefs. This risk is particularly relevant when inferred preferences concern opinions or values rather than benign stylistic or task-related choices. Personalized systems should therefore distinguish between adapting to user preferences and endorsing user claims, preserve uncertainty when evidence is limited, and avoid treating agreement or positive feedback as evidence of factual correctness. They should also retain the ability to provide corrective information or alternative perspectives when appropriate.

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of our results. Detailed descriptions of the experimental setup, including training, evaluation, and LLM-based judging procedures, as well as prompts, are provided in Appendix B and in Appendix E. We plan to release our code upon acceptance.

REFERENCES

Nishant Balepur, Vishakh Padmakumar, Fumeng Yang, Shi Feng, Rachel Rudinger, and Jordan Boyd-Graber. Whose boat does it float? improving personalization in preference tuning via inferred user personas. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3371–3393, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.168. URL https://aclanthology.org/2025.acl-long.168/.

Thomas Kleine Buening, Jonas Hübotter, Barna Pásztor, Idan Shenfeld, Giorgia Ramponi, and Andreas Krause. Aligning language models from user interactions. In *Continual Adaptation at Scale: Towards Sustainable AI Workshop*, 2026. URL https://openreview.net/forum?id=HL3iloEczS.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel Weld. TLDR: Extreme summarization of scientific documents. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4766–4777, Online, November 2020.

Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.428. URL https://aclanthology.org/2020.findings-emnlp.428/.

Mengyi Deng, Zhiwei Li, Xin Li, Tingyu Zhu, Ying Zhao, Zhijiang Guo, and Wei Wang. Uncertainty-aware clarification in llm agents with information gain, 2026. URL https://arxiv.org/abs/2606.03135.

Cristina Garbacea and Chenhao Tan. Hyperalign: Interpretable personalized llm alignment via hypothesis generation, 2025. URL https://arxiv.org/abs/2505.00038.

EunJeong Hwang, Bodhisattwa Majumder, and Niket Tandon. Aligning language models to user opinions. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5906–5919, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.393. URL https://aclanthology.org/2023.findings-emnlp.393/.

EunJeong Hwang, Vered Shwartz, Dan Gutfreund, and Veronika Thost. A graph per persona: Reasoning about subjective natural language descriptions. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1928–1942, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.115. URL https://aclanthology.org/2024.findings-acl.115/.

EunJeong Hwang, Yuwei Yin, Giuseppe Carenini, Peter West, and Vered Shwartz. Infusing theory of mind into socially intelligent LLM agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11327–11360, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.551. URL https://aclanthology.org/2026.findings-acl.551/.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026. URL https://arxiv.org/abs/2601.20802.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of LLMs? In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=Az7guis46K.

Tongyoung Kim, Jeongeun Lee, SooJin Yoon, SungHwan Kim, and Dongha Lee. Towards personalized conversational sales agents: Contextual user profiling for strategic action. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5131–5154, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.275. URL https://aclanthology.org/2025.findings-emnlp.275/.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation, 2025. URL https://arxiv.org/abs/2505.06120.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. Hello again! LLM-powered personalized agent for long-term dialogue. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5259–5276, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.272. URL https://aclanthology.org/2025.naacl-long.272/.

Tingting Liang, Chenxin Jin, Lingzhi Wang, Wenqi Fan, Congying Xia, Kai Chen, and Yuyu Yin. LLM-REDIAL: A large-scale dataset for conversational recommender systems created from user behaviors with LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of*

the Association for Computational Linguistics: ACL 2024, pp. 8926–8939, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.529. URL https://aclanthology.org/2024.findings-acl.529/.

Ben Liu, Jihai Zhang, Fangquan Lin, Xu Jia, and Min Peng. One size doesn't fit all: A personalized conversational tutoring agent for mathematics instruction. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 2401–2410, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN 9798400713316. doi: 10.1145/3701716.3717527. URL https://doi.org/10.1145/3701716.3717527.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. A survey of personalized large language models: Progress and future directions, 2025b. URL https://arxiv.org/abs/2502.11528.

Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. LLMs + persona-plug = personalized LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9373–9385, Vienna, Austria, July 2025c. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.461. URL https://aclanthology.org/2025.acl-long.461/.

Yuting Liu, Jian Guan, Jia-Nan Li, Wei Wu, Jiang-Ming Yang, Jianzhe Zhao, and Guibing Guo. Text as a universal interface for transferable personalization, 2026. URL https://arxiv.org/abs/2601.04963.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.747. URL https://aclanthology.org/2024.acl-long.747/.

Shuhaib Mehri, Priyanka Kargupta, Tal August, and Dilek Hakkani-Tür. Multisessioncollab: Learning user preferences with memory to improve long-term collaboration, 2026. URL https://arxiv.org/abs/2601.02702.

Jisoo Mok, Ik-hwan Kim, Sangkwon Park, and Sungroh Yoon. Exploring the potential of LLMs as personalized assistants: Dataset, evaluation, and analysis. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10212–10239, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.504. URL https://aclanthology.org/2025.acl-long.504/.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. In Sachin Kumar, Vidhisha Balachandran, Chan Young Park, Weijia Shi, Shirley Anugrah Hayati, Yulia Tsvetkov, Noah Smith, Hannaneh Hajishirzi, Dongyeop Kang, and David Jurgens (eds.), *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pp. 198–219, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.customnlp4u-1.16. URL https://aclanthology.org/2024.customnlp4u-1.16/.

Hyunji Nam, Yanming Wan, Mickel Liu, Peter Ahnn, Jianxun Lian, and Natasha Jaques. Learning to summarize user information for personalized reinforcement learning from human feedback, 2026. URL https://arxiv.org/abs/2507.13579.

Isha Puri, Mehul Damani, Idan Shenfeld, Marzyeh Ghassemi, Jacob Andreas, and Yoon Kim. Reaching beyond the mode: Rl for distributional reasoning in language models, 2026. URL https://arxiv.org/abs/2603.24844.

Erasmo Purificato, Ludovico Boratto, and Ernesto William De Luca. User modeling and user profiling: A comprehensive survey, 2024. URL https://arxiv.org/abs/2402.09660.

Linlu Qiu, Fei Sha, Kelsey Allen, Yoon Kim, Tal Linzen, and Sjoerd van Steenkiste. Bayesian teaching enables probabilistic reasoning in large language models. *Nature Communications*, 2026. URL https://arxiv.org/abs/2503.17523.

Jerome Ramos, Hossein A. Rahmani, Xi Wang, Xiao Fu, and Aldo Lipani. Transparent and scrutable recommendations using natural language user profiles. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13971–13984, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.753. URL https://aclanthology.org/2024.acl-long.753/.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.399. URL https://aclanthology.org/2024.acl-long.399/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Jianzhi Shen, Keyu Mao, Minghao Shao, Chuanyang Jin, Yusong Wang, Ailiang Lin, Kotaro Funakoshi, Manabu Okumura, Tianmin Shu, and Muhammad Shafique. Hypertrace: Hypothesis-based preference tracing for online llm personalization, 2026. URL https://arxiv.org/abs/2609.09835.

Yunxiao Shi, Wujiang Xu, Zhang Zeqi, Xing Zi, Qiang Wu, and Min Xu. PersonaX: A recommendation agent-oriented user modeling framework for long behavior sequence. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5764–5787, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.300. URL https://aclanthology.org/2025.findings-acl.300/.

Jin Su, Runnan Fang, Yeqiu Li, Xiaobin Wang, Shihao Cai, Pengjun Xie, Ningyu Zhang, and Fajie Yuan. U-fold: Dynamic intent-aware context folding for user-centric agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 18043–18057, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.897. URL https://aclanthology.org/2026.findings-acl.897/.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. URL https://arxiv.org/abs/2607.02770.

Yanming Wan, Jiaxing Wu, Marwa Abdulhai, Lior Shani, and Natasha Jaques. Enhancing personalized multi-turn dialogue with curiosity reward, 2025. URL https://arxiv.org/abs/2504.03206.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J. Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer 2: Open-source dataset for training top-performing reward models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=PvVKUFhaNy.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory, 2025. URL https://arxiv.org/abs/2410.10813.

Jing Xu, Arthur Szlam, and Jason Weston. Beyond goldfish memory: Long-term open-domain conversation. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5180–5197, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.356. URL https://aclanthology.org/2022.acl-long.356/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Jiarui Zhang. Guided profile generation improves personalization with large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4005–4016, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.231. URL https://aclanthology.org/2024.findings-emnlp.231/.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1205. URL https://aclanthology.org/P18-1205/.

Weizhi Zhang, Xinyang Zhang, Chenwei Zhang, Liangwei Yang, Jingbo Shang, Zhepei Wei, Henry Peng Zou, Zijie Huang, Zhengyang Wang, Yifan Gao, Xiaoman Pan, Lian Xiong, Jingguo Liu, Philip S. Yu, and Xian Li. PersonaAgent: Bridging memory and action for personalized LLM agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 26421–26439, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1315. URL https://aclanthology.org/2026.findings-acl.1315/.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, Nesreen Ahmed, and Yu Wang. Personalization of large language models: A survey, 2025. URL https://arxiv.org/abs/2411.00027.

Zhining Zhang, Chuanyang Jin, Mung Yao Jia, Shunchi Zhang, and Tianmin Shu. Autotom: Scaling model-based mental inference via automated agent modeling, 2026b. URL https://arxiv.org/abs/2502.15676.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026. URL https://arxiv.org/abs/2601.18734.

Weixiang Zhao, Xingyu Sui, Yulin Hu, Jiahe Guo, Haixiao Liu, Biye Li, Yanyan Zhao, Bing Qin, and Ting Liu. Teaching language models to evolve with users: Dynamic profile modeling for personalized alignment, 2025a. URL https://arxiv.org/abs/2505.15456.

Zheng Zhao, Clara Vania, Subhradeep Kayal, Naila Khan, Shay B. Cohen, and Emine Yilmaz. PersonaLens: A benchmark for personalization evaluation in conversational AI assistants. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18023–18055, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.927. URL https://aclanthology.org/2025.findings-acl.927/.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: enhancing large language models with long-term memory. AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i17.29946. URL https://doi.org/10.1609/aaai.v38i17.29946.

Yangqiaoyu Zhou, Haokun Liu, Tejes Srivastava, Hongyuan Mei, and Chenhao Tan. Hypothesis generation with large language models. In Lotem Peled-Cohen, Nitay Calderon, Shir Lissak, and Roi Reichart (eds.), *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pp. 117–139, Miami, FL, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.nlp4science-1.10. URL https://aclanthology.org/2024.nlp4science-1.10/.

Yaochen Zhu, Harald Steck, Dawen Liang, Yinhan He, Nathan Kallus, and Jundong Li. LLM-based conversational recommendation agents with collaborative verbalized experience. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 2207–2220, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.119. URL https://aclanthology.org/2025.findings-emnlp.119/.

A Methodology

A.1 Chunking

In the offline setting, we partition the complete history into consecutive, non-overlapping chunks $C_1, \ldots, C_N$, each containing the longest sequence of complete interactions that fits within a token budget B. We then infer a set of local hypotheses from each chunk. In the online setting, interactions are accumulated sequentially until adding the next complete interaction would exceed B. At that point, local hypotheses are inferred from the completed chunk, and accumulation begins for the next chunk. Before the first chunk is completed, the model uses an empty hypothesis set; afterward, it uses the latest maintained hypotheses while accumulating interactions for the next set of local hypotheses.

B Experimental Setting Details

B.1 Dataset Details

This section provides additional details about the construction of each evaluation benchmark. Figure 5 presents an example of each dataset's user question, preferences, feedback (user's follow up question) of HelpSteer2, HiCupid, and Flight recommendation datasets.

HelpSteer2 (Wang et al., 2024). We use eight synthetic user profiles, where each profile is assigned a combination of three writing-style preferences adapted from TL;DR (Cachola et al., 2020). These preferences define the target response behavior for each user. At each interaction, the user simulator (`deepseek-v4-flash`) receives the generated response and provides feedback on one or two relevant preference dimensions. This creates a sparse feedback setting where the model must identify relevant preference signals over time. We train a separate model for each user profile.

HiCupid (Mok et al., 2025). HiCupid contains personalized question-answering examples associated with individual users and their historical interactions. We train a single model jointly on 9K QA examples from 300 users. We use user's profile data as a privileged information. For evaluation, we construct two settings: seen-user generalization, where the model receives previous interactions from users observed during training, and unseen-user generalization, where the model must personalize responses for users not encountered during training. The evaluation sets contain 1.8K prompts from seen users and 7.5K prompts from 250 held-out users .

Flight Recommendation (Qiu et al., 2026). In Flight data, each user trajectory contains a sequence of recommendation interactions where user selections provide indirect evidence about underlying preferences. The benchmark contains 624 users and we use up to 25 interactions per user. User preferences may include non-obvious trade-offs, such as preferring higher prices or longer durations, requiring models to infer preferences from behavior rather than relying on standard assumptions. We evaluate across the original four-feature Flight setting, held-out feature settings with two to eight features, and transfer to the hotel recommendation domain.

HelpSteer2	HiCupid	Flight Recommendation
User Question: What is C#? **User preference examples (for user simulator)**: You are playing the role of a user who has the following preferences: - You specifically prefer CONCISE, CASUAL, and BEGINNER-FRIENDLY responses. You want short and clear answers, avoiding long, formal, or technically dense answers. - You prefer when the assistant uses plain language that focuses on clarity. You prefer short, straightforward answers. You dislike when the assistant responses with poetic, lyrical language. **Feedback example:** - The first sentence is unnecessary fluff. Just give me the list without the setup. - The response is clear and to the point, but it could be even shorter—just the question alone works fine for me.	**User Question**: What are the benefits of investing in consumer goods stocks? **User profile examples**: [persona]: "category": "Finance", "entity": "consumer goods stocks", "relation": "invests in", "sentiment": "POS", [profile]: "age": 32, "gender": "male", "income_range": "$40,000 - $60,000", "occupation": "sports journalist", "personality": "curious, analytical, engaging" **Feedback example:** - I agree. As someone who values market stability, should I prioritize companies with global operations?	**User Question**: Help me pick the best flight for my trip. Which flight is the best option? - Flight 1: departure time: 09:12 AM, duration: 12 hr 12 min, number of stops: 1, price: $640 - Flight 2: departure time: 05:12 PM, duration: 2 hr 27 min, number of stops: 1, price: $550 - Flight 3: departure time: 10:00 PM, duration: 14 hr 9 min, number of stops: 2, price: $640 **User profile examples**: departure_time=strong preference for later departures; duration=strong preference for longer durations; number_of_stops=slight preference for more stops; price=slight dislike for higher prices **Feedback example:** Flight 3

Figure 5: Example user question, preferences, and feedback (i.e., user's follow up message) of HelpSteer2, HiCupid, and Flight recommendation datasets.

B.2 Training Hyperparameters

We initialize our configurations based on the implementation of Buening et al. (2026) and the corresponding dataset-specific codebases, and tune the learning rate and batch size for each dataset. We explore learning rates of $\{1 \times 10^{-6}, 5 \times 10^{-6}, 1 \times 10^{-5}\}$ and observe relatively consistent performance on HelpSteer and HiCupid, whereas Flight performs best with 1×10^{-5}. We also evaluate batch sizes of 2, 4, and 8 on Flight and 16 and 32 on HiCupid. HiCupid is relatively insensitive to batch size, while Flight benefits from smaller batches; we therefore use batch sizes of 32 and 4, respectively. Performance is otherwise stable across the tested hyperparameters, except for the maximum history/chunk and generation lengths, which are limited by GPU memory. When memory permits, larger history/chunk sizes generally improve performance by preserving more information from previous interactions. The final configurations are reported in Table 6. We used NVIDIA A800 and RTX6000 Ada GPUs, and each experiment can be run on a single GPU node. Training and evaluation take approximately 15 hours on HelpSteer2, up to 17 hours on Flight, and up to 10 hours on HiCupid.

GRPO Setup Details. We follow GRPO implementation proposed by Shao et al. (2024) and design our GRPO setup to align with our self-distillation framework. Specifically, we instantiate the reward model from the same base model as the policy, supporting continual self-improvement without relying on a separate model for supervision. The reward model receives privileged information, either the user's future turn or hidden preferences, and evaluates how well each sampled response aligns with or incorporates this information (see Appendix E, Fig. 14). For the open-ended HiCupid and HelpSteer tasks, the reward model assigns scores from 0 (no relevance) to 4 (high relevance). For the flight task, where a ground-truth answer is available for each user, we instead use a binary correctness reward: 0 for incorrect answers and 1 for correct answers.

We evaluate rollout counts of $\{3, 4, 6\}$ and sampling temperatures of $\{0.7, 1.0, 1.3\}$, fixing top-p at 1.0 and the learning rate at 1×10^{-6}. Qwen3.5-4B generates sufficiently diverse responses at a temperature of 1.0 and achieves its best trade-off between training cost and performance with four rollouts. Qwen3.5-9B exhibits lower response diversity, but the same configuration provides the best balance between performance and training cost, as additional rollouts substantially increase cost. Gemma4-4B exhibits the lowest response diversity and performs best at a temperature of 1.3 with four rollouts; increasing to six rollouts incurs substantially higher training costs.

B.3 Evaluation Details

We follow the original dataset papers for evaluation, including an LLM-based user simulator and pairwise LLM judging where applicable. We use DeepSeek-V4-Flash as the judge for HelpSteer

Table 6: Training hyperparameters used for each dataset.

Hyperparameter	HelpSteer	Flight	HiCupid
Number of epochs	Online	1	1
Batch size	1	4	32
Learning rate	1×10^{-6}	1×10^{-5}	1×10^{-6}
LoRA rank	32	32	32
LoRA alpha	64	64	64
LoRA dropout	0	0	0
History/chunk size	4,096	1,024	1,024
Max generation length	512	400	512
Temperature	0.7	0.7	0.7
Top-p	1.0	1.0	1.0
Top-k	20	64	64
Reflection window size	5	5	5

Table 7: Evaluation settings used for each dataset.

Evaluation detail	HelpSteer	Flight	HiCupid
Evaluation data	256 validation prompts	624 users / 20 questions per user	300 Seen users, 250 Unseen users
Evaluation frequency	Every 50 interactions	After training	After training
Evaluator	deepseek-v4-flash	Exact answer matching	deepseek-v4-flash
Primary metric	Pairwise win rate	Accuracy	Pairwise win rate
Reference	Frozen base model	Ground-truth Flight	Frozen base model

and HiCupid, while Flight is evaluated through exact matching against the ground-truth choices. We additionally compare the judgments of DeepSeek-V4-Flash with those of DeepSeek-V4-Pro and Qwen3.7-Plus (see Table 8 and Appendix C) and observe consistent evaluation outcomes across the models. Given its comparable evaluation performance and lower inference cost, we select DeepSeek-V4-Flash as a practical balance between evaluation quality and cost. The complete evaluation settings are reported in Table 7. See Buening et al. (2026) (repo) for the LLM judge and user simulator prompts used for HelpSteer2, and Mok et al. (2025) (repo) for details of the LLM judge prompt used for HiCupid. Our code is also built based on these two repos. Running DeepSeek-V4-Flash as both the LLM judge and user simulator on HelpSteer2 costs approximately 10 USD and using it as the LLM judge on HiCupid costs approximately 10 USD.

C LLM JUDGE VALIDITY

We validate the use of `deepseek-v4-flash` as our primary LLM judge by comparing its evaluations with those of `deepseek-v4-pro` and `qwen3.7-plus`. Although the absolute scores vary across judges, all three produce the same ranking: HYPREFLECT consistently outperforms SD-INCHYP, which in turn outperforms SD. This agreement indicates that our main conclusions are robust to the choice of LLM judge.

D ADDITIONAL RESULTS

D.1 ANALYSES ON TRAINING AND INFERENCE COSTS

Table 9 reports efficiency on HiCuPID for Qwen3.5-4B and Qwen3.5-9B on a single RTX 6000 Ada GPU. Training time includes all method-specific generation and optimization, covering three rollouts per prompt and reward judging for GRPO, and hypothesis construction, reflection, refine-

Table 8: Evaluation results using different LLM judges on the subset of HelpSteer2. All judges produce the same relative ranking across methods.

	SD	SD-IncHyp	HYPREFLECT
Deepseek-v4-flash	77.1	90.3	93.0
Deepseek-v4-pro	74.1	86.3	87.1
Qwen3.7-plus	71.8	83.4	86.4

Table 9: Training and inference efficiency on HiCuPID with Qwen3.5 models. s/row is training time per data row; Extra tok. and Final tok. are average intermediate reasoning and final response token counts per query, respectively; s/query is inference time per query.

		Train	Inference		
Model	**Method**	**s/row**	**Extra tok.**	**Final tok.**	**s/query**
Qwen3.5-4B	Base	–	0.0	72.8	0.14
	GRPO	14.5	0.0	78.8	0.13
	HYPREFLECT	0.8	182.2	102.3	0.40
Qwen3.5-9B	Base	–	0.00	90.4	0.20
	GRPO	19.5	0.00	90.7	0.21
	HYPREFLECT	1.4	154.4	109.9	0.49

ment, and teacher-answer generation for HYPREFLECT. HYPREFLECT trains on average of 15.8× faster than GRPO. At inference, its final responses average 21.4 more tokens than GRPO and 24.6 more than the base model, with a latency of 0.40 seconds per query compared with 0.14 for the base model.

D.2 QUALITATIVE EXAMPLES FOR REFLECTION

Figures 6 and 7 presents some qualitative examples that compares between hypotheses consolidation without reflection and with reflection. We find that direct consolidation tends to transform local hypotheses into broad and partially overlapping meta-preferences, such as favoring actionable, multifaceted, or tailored advice. These abstractions omit concrete concepts from the original hypothesis sets and provide less distinctive guidance for personalization. Reflection instead explicitly identifies which hypotheses should be merged, retained, or removed, preserving specific preference anchors such as crowdfunding, local outreach, marketing, and the balance between project management and relationship building. This example suggests that reflection reduces vague generalization and redundancy while producing more specific and informative preference hypotheses.

D.3 QUALITATIVE EXAMPLES OF FINAL RESPONSES: SD VS. HYPREFLECT

Figures 8 and 9 present qualitative comparisons between the final responses generated by SD and HYPREFLECT, together with the preference hypotheses used by HYPREFLECT. In the HiCupid example, both methods provide generally relevant advice, but HYPREFLECT more explicitly emphasizes narrative structure and personification, reflecting the highest-weight hypothesis inferred from the user's interest in engaging, science-oriented content. In the Flight example, the hypotheses help HYPREFLECT resolve competing preferences by prioritizing the user's strong dislike of long travel durations. Because all three flights have the same price, HYPREFLECT correctly selects Flight 2, which has the shortest duration, whereas SD selects Flight 3. These examples illustrate how explicit, weighted hypotheses can help the model identify the most relevant user preferences and prioritize them when generating personalized responses.

D.4 REFLECTION WINDOW SIZE VARIATION

Figure 10 compares performance under different reflection window sizes on a subset of the HelpSteer2 setting at every 50 interactions.

E PROMPTS

HYPREFLECT follows the same four-stage procedure across datasets. First, the model generates local preference hypotheses from a recent chunk of interactions (Figure 11). Second, it reflects on the generated sets of local preference hypotheses to identify supported, conflicting, and less important hypotheses (Figure 12). Third, it produces a refined hypothesis set using a prompt nearly identical to that in Figure 11, augmented with the reflection traces and multiple sets of local hypotheses. Finally, the refined hypothesis set is provided to the model to guide the generation of the final personalized response (Figure 13).

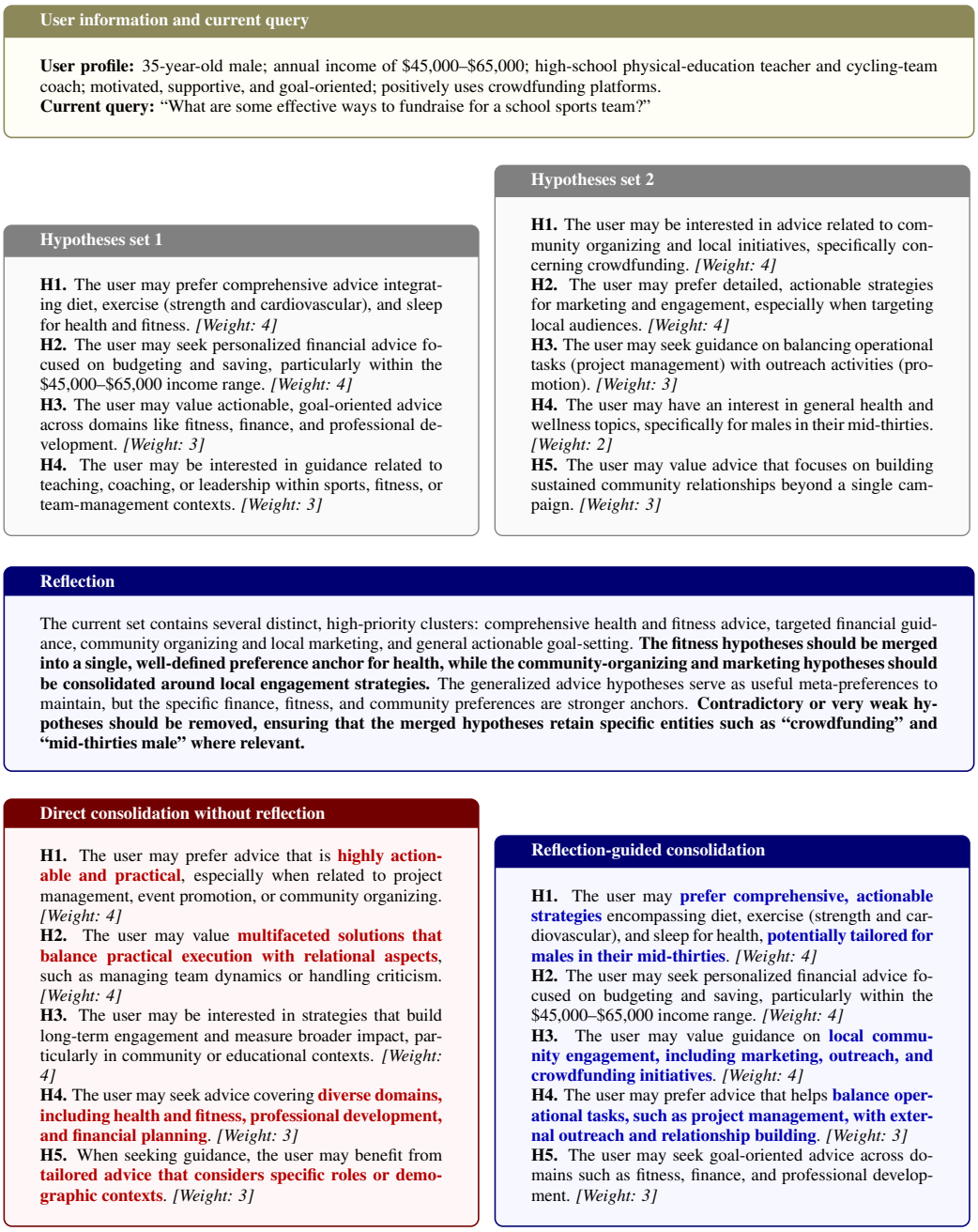


Figure 6: An example of hypothesis consolidation with and without explicit reflection.

Dataset-specific instantiations. Although all datasets follow the same overall prompt structure, we slightly adapt the preference-extraction instructions and output format to reflect their distinct evidence types and personalization targets. For Flight, the history contains a flight-recommendation question, the user's selected flight, and feedback. The hypotheses capture preferences over departure time, duration, number of stops, and price using a fixed set of directional labels. For HelpSteer2, the history contains a request, response, and user feedback. The hypotheses capture general response-style preferences, such as verbosity, tone, structure, and formatting, while excluding task-specific details. For HiCupid, the history consists of prior dialogues, and the hypotheses preserve relevant interests, constraints, roles, relationships, and other concrete personalization information. We retain at most three hypotheses for Flight and HelpSteer2 and five for HiCupid, reflecting

User preference profile and current choice

Hidden preference profile: Strong preference for later departures, longer durations, and higher prices; slight preference for more stops.
Current query: "Which flight is the best option?"
Flight 1: 06:00 AM; 30 min; 2 stops; $370. **Flight 2:** 06:00 AM; 8 hr 18 min; 0 stops; $100. **Flight 3:** 06:48 PM; 10 hr 15 min; 0 stops; $730.
Observed choice: Flight 3.

Hypotheses set 1

H1. Departure time = strong preference for later departures; duration = slight dislike for longer durations; stops = slight preference for more stops; price = slight dislike for higher prices. *[Weight: 4]*
H2. Departure time = strong preference for later departures; duration = slight preference for longer durations; stops = slight dislike for more stops; price = slight preference for higher prices. *[Weight: 2]*
H3. Departure time = strong preference for later departures. *[Weight: 3]*

Hypotheses set 2

H1. Departure time = slight preference for later departures; duration = slight preference for longer durations; stops = neutral; price = strong dislike for higher prices. *[Weight: 2]*
H2. Departure time = neutral; duration = neutral; stops = neutral; price = strong dislike for higher prices. *[Weight: 4]*

Reflection

The recent interaction window provides repeated evidence that the user selects later, longer, and more expensive flights over earlier, shorter, and cheaper alternatives. **This evidence directly conflicts with the hypotheses that infer a dislike for longer durations and higher prices.** Evidence regarding stops is less consistent and should therefore be retained with lower confidence. **Before consolidation, the conflicting hypotheses should be revised so that repeated behavioral evidence is prioritized over misleading local inferences.**

Direct consolidation without reflection

H1. Departure time = strong preference for later departures; duration = slight preference for longer durations; stops = slight dislike for more stops; price = slight dislike for higher prices. *[Weight: 3]*
H2. Departure time = strong preference for later departures; duration = slight dislike for longer durations; stops = strong dislike for more stops; price = strong dislike for higher prices. *[Weight: 3]*
H3. Departure time = strong preference for later departures; duration = slight preference for longer durations; stops = strong preference for more stops; price = slight dislike for higher prices. *[Weight: 3]*

Reflection-guided consolidation

H1. Departure time = **strong preference for later departures**; duration = **strong preference for longer durations**; stops = **slight preference for more stops**; price = **strong preference for higher prices**. *[Weight: 5]*
H2. Departure time = strong preference for later departures; duration = strong preference for longer durations; stops = slight dislike for more stops; price = strong preference for higher prices. *[Weight: 4]*
H3. Departure time = strong preference for later departures; duration = slight preference for longer durations; stops = slight preference for more stops; price = slight preference for higher prices. *[Weight: 3]*

Figure 7: A qualitative Flight Recommendation example comparing direct and reflection-guided hypothesis consolidation. Direct consolidation propagates misleading inferences about duration and price, whereas reflection re-evaluates conflicting evidence across the recent interaction window and produces a top hypothesis that exactly matches the hidden preference profile.

the richer, multi-session nature of the latter. Summary-enhanced variants additionally produce an `Evidence/summary note:` containing the supporting observations.

Dataset: HiCupid
Hidden user profile. 32-year-old man; sports journalist; income $40,000–$60,000; curious, analytical, and engaging; follows Mark Rober.

User query. What are some engaging ways to explain complex topics to a general audience?

Hypotheses supplied to HYPREFLECT.
H1: The user may prefer narrative-driven educational content that blends complex scientific concepts with relatable storytelling and personification (weight 5).
H2: The user may value high production quality while remaining open to cost-effective or free content-creation tools (weight 4).
H3: The user may prefer career guidance that balances existing skills with advanced opportunities in sports journalism (weight 4).
H4: The user may prefer learning methods that combine theoretical knowledge with hands-on application (weight 4).
H5: The user may prefer actionable health or financial guidance tailored to men in their early thirties (weight 3).

Final answer with HYPREFLECT. To explain complex topics to a general audience, lean heavily into analogies and metaphors, as these create instant relatable connections. Frame the topic as a narrative, personifying concepts or structuring the explanation like a mystery to be solved. Furthermore, focus on the "why it matters"—connecting the science directly to the audience's daily life or major societal issues makes the information immediately relevant and engaging.

Final answer with SD. To explain complex topics to a general audience, employ strong analogies and metaphors that connect the abstract concept to everyday experiences. Visual demonstrations and simplified models are crucial for making the invisible visible. Additionally, framing the topic as a compelling story—highlighting the "why" and the real-world implications—keeps the audience hooked.

Figure 8: Final response comparison between SD and HYPREFLECT on HiCupid

Dataset: Flight Recommendation
Hidden user profile. Slight preference for later departures; strong dislike for longer durations; slight preference for more stops; strong dislike for higher prices.

User query. Which flight is the best option?
- Flight 1: 10:00 PM; 20 hr; 0 stops; $910.
- Flight 2: 07:36 AM; 8 hr 18 min; 2 stops; $910.
- Flight 3: 09:12 AM; 14 hr 9 min; 2 stops; $910.

Hypotheses supplied to HYPREFLECT.
H1: Slight preference for later departures; strong dislike for longer durations; slight preference for more stops; strong dislike for higher prices (weight 4).
H2: Strong dislike for later departures, longer durations, more stops, and higher prices (weight 3).
H3: Slight preference for later departures; strong dislike for longer durations, more stops, and higher prices (weight 3).

User's choice: Flight 2.
HYPREFLECT answer: Flight 2 (correct).
SD answer: Flight 3 (incorrect).

Figure 9: Final response comparison between SD and HYPREFLECT on Flight Recommendation

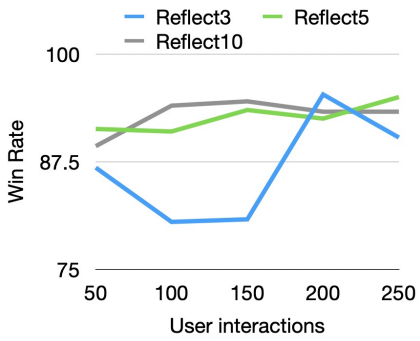


Figure 10: Performance with HYPREFLECT across different reflection window sizes on HelpSteer2

Local hypothesis generation

```
[SYSTEM]

You are given a bounded portion of a user's interaction history.
    Infer a compact set of persistent preference hypotheses that
    may help personalize future responses. The interaction history
     is evidence for preference inference only. Do not answer or
    continue any task contained in the history.

[PREFERENCE SPACE]

Infer preferences over <dataset-specific preference dimensions>.

[GUIDELINES]

- Use only evidence contained in the interaction history.
- Infer reusable preferences rather than merely describing the
    latest interaction.
- Express preferences as soft, conditional hypotheses.
- Avoid broad conclusions from weak or ambiguous evidence.
- Assign higher weights only when evidence is clear or repeated.
- Preserve or remove concrete details according to the dataset-
    specific personalization objective.
- Do not include unsupported preferences.

[CONVERSATION HISTORY]

<recent interaction chunk>

[TASK]
For each hypothesis, include a weight from 1 to 5 indicating how
    strongly the current interaction history supports it:
1 = weak inference from limited contextual evidence
2 = plausible inference with a clear but limited behavioral basis
3 = repeated behavioral evidence or a moderately clear stated
    preference
4 = directly stated preference or strong repeated behavioral
    evidence
5 = directly stated and repeatedly confirmed preference

Generate up to <maximum number> distinct preference hypotheses.

Return only:
hypothesis1: <dataset-specific preference statement> | weight:
    <1-5>
...
hypothesisN: <dataset-specific preference statement> | weight:
    <1-5>
```

Figure 11: Local hypothesis-generation prompt structure.

Hypothesis reflection

```
[SYSTEM]

You are given several preference hypothesis sets inferred from
    different portions of a user's interaction history, together
    with a previously refined hypothesis set when available.
    Reflect on how these hypotheses should be consolidated before
    a separate refinement step produces the updated preference
    memory. The hypotheses and summaries are evidence for
    preference analysis only. Do not answer or continue any task
    contained in them.

[GUIDELINES]

- Identify hypotheses that agree or describe the same preference.
- Identify conflicting, weak, ambiguous, unsupported, or overly
    broad hypotheses.
- Distinguish repeated evidence from isolated observations.
- Determine which hypotheses should be preserved, merged, narrowed
    , reweighted, or removed.
- Preserve dataset-specific information that remains useful for
    future personalization.
- Do not treat the reflection itself as additional evidence.
- Do not produce the final hypothesis set.

For each hypothesis, a weight from 1 to 5 indicates how strongly
    the current interaction history supports it:
1 = weak inference from limited contextual evidence
2 = plausible inference with a clear but limited behavioral basis
3 = repeated behavioral evidence or a moderately clear stated
    preference
4 = directly stated preference or strong repeated behavioral
    evidence
5 = directly stated and repeatedly confirmed preference

[LOCAL HYPOTHESIS SETS]

### Local hypothesis set max(N-W, 1)
<local hypotheses max(N-W, 1)>
...
### Local hypothesis set N
<local hypotheses N>

[TASK]

Write only a concise reflection that will guide the subsequent
    refinement step.
```

Figure 12: Hypotheses-reflection prompt across datasets.

Answer generation with preference hypotheses

```
[SYSTEM]

You are given hidden preference hypotheses inferred from the user's previous interactions. You may also receive recent conversation history and an evidence summary. Use this information silently to personalize the response. The current request remains the primary task and must be answered directly and completely. Generate the assistant's response to the current user request.

[GUIDELINES]

- Apply only hypotheses relevant to the current request.
- Give greater consideration to more strongly supported hypotheses.
- Treat every hypothesis as tentative rather than as a known fact.
- Prefer relevant and strongly supported hypotheses when hypotheses conflict.
- Ignore weak, irrelevant, or contradicted hypotheses.
- Do not force personalization when no hypothesis is relevant.
- Do not allow personalization to make the answer incomplete, incorrect, or unhelpful.
- Do not mention the hypotheses, evidence summary, interaction history, user profile, or personalization process.
- Follow the output format specified in system prompt.

For each hypothesis, include a weight from 1 to 5 indicating how strongly the current interaction history supports it:
1 = weak inference from limited contextual evidence
2 = plausible inference with a clear but limited behavioral basis
3 = repeated behavioral evidence or a moderately clear stated preference
4 = directly stated preference or strong repeated behavioral evidence
5 = directly stated and repeatedly confirmed preference

[RECENT CONVERSATION HISTORY]

<recent interaction history, if provided>

[HIDDEN USER PREFERENCE MEMORY]

<evidence summary, if provided>
<refined preference hypotheses>

[CURRENT USER REQUEST]

<current question or request>

[TASK]

Produce only the personalized response to the current request.
```

Figure 13: Prompt for answer generation conditioned on preference hypotheses.

System prompt: You are a reward judge. Your entire response must be exactly one character: 0, 1, 2, 3, or 4. Do not output words, explanations, punctuation, whitespace, or any other text.
User prompt: Output exactly one digit from this set: 0, 1, 2, 3, or 4.
Evaluate how well the assistant response satisfies the reference information.
The reference information may contain a correct answer, a preferred answer, user preferences or metadata, or feedback criticizing the response. Interpret it according to its meaning. Judge whether the response is correct, useful, and aligned with the preferences or corrections expressed in the reference information.

User request:
{ prompt }

Assistant response:
{ response }

Reference information:
{ feedback }

0 = completely incorrect or inconsistent
1 = mostly incorrect or poorly aligned
2 = partially correct or aligned
3 = mostly correct and aligned
4 = fully correct and aligned

Return only the single digit. Do not include an explanation or any other text.

Figure 14: Prompt used for reward judging in GRPO.