Ant International

Learning Preference Adaptation for Large Language Model Personalization via Verbal Reinforcement Learning

Yuting Liu[1,2] **Wei Wu**[2,*] **Jianzhe Zhao**[1] **Guibing Guo**[1,*]

[1]Software College, Northeastern University, China [2]Ant International

[*]Corresponding authors: Guibing Guo and Wei Wu.

liuyuting@stumail.neu.edu.cn, wuwei19850318@gmail.com, {guogb,zhaojz}@swc.neu.edu.cn

arXiv:2608.09507v2 [cs.CL] 12 Aug 2026

Abstract

Natural language user preferences provide an interpretable interface for LLM personalization. However, universal preference summaries often contain information irrelevant to a particular downstream task. Directly supplying the full preference summary therefore wastes context capacity and introduces cross-task distraction, while manually designing task-specific preference views is difficult to scale. In this work, we study *task-specific preference adaptation*: given a universal user preference summary and a downstream task, derive a task-conditioned representation that preserves sufficient decision-relevant evidence while removing redundant context. To this end, we propose AlignXada, a training-free meta-learning framework that induces reusable textual refinement policies for adapting universal preference summaries to task-specific ones. The refinement policy is iteratively optimized by a meta learner through verbal reinforcement learning. Across 13 tasks and three downstream models (39 task–model cells), AlignXada achieves an average gain of 3.82 points, improving 33 cells while retaining only 22.8% of the original profile tokens and outperforming RAG in 36 cells. An extended faithfulness analysis further shows that the refined profiles remain largely grounded in the source preferences while preserving task-relevant personalization signals, suggesting that profile-side adaptation serves as a practical complement to universal memory construction for lifelong personalized agents. Code is available at https://github.com/AntResearchNLP/AlignX-Family/tree/main/AlignXada.

Keywords: large language model personalization, task-specific preference adaptation, verbal reinforcement learning, meta-learning

1 Introduction

Personalization, the technique of aligning AI systems with human preferences, has played an important role in the training of large language models since their early development [Ouyang et al., 2022]. Recently, with the proliferation of personal AI agents [OpenClaw, 2026, Hermes, 2026] and the expanding use of LLMs in applications such as search [Baek et al., 2024], recommendation [Geng et al., 2022, Lyu et al., 2024, Peng et al., 2026], and dialogue systems [Otsuka et al., 2024], personalization techniques — particularly as a key component in building LLM memories [Wang et al., 2023, Xu et al., 2025, Chhikara et al., 2025, Yu et al., 2025, Zhong et al., 2024] — have become increasingly critical and continue to drive advances at the research frontier.

A central problem in personalization is user preference representation, which serves as the interface through which AI systems perceive user preferences and adapt their behaviors accordingly. Early studies modeled user preferences through user embeddings or by encoding user information directly into model parameters [Koren et al., 2009, Bao et al., 2023, Liu et al., 2025b, Zhang et al., 2025]. While effective, such black-box representations suffer from limited interpretability and are difficult to update in real time. With the emergence of LLMs, in-context learning has enabled a simple yet effective alternative by explicitly incorporating user behaviors into the model context as demonstrations [Du et al., 2026, Salemi et al., 2024]. However, such approaches are fundamentally constrained by the context capacity of LLMs, making it difficult to comprehensively capture user preferences while remaining highly sensitive to noise in the selected examples. More recently, the advancement of LLM reasoning capabilities has inspired efforts to infer universal user preferences from heterogeneous data sources and summarize them in natural language [Chen et al., 2025, Nam et al., 2025, Liu et al., 2026, Yang et al., 2026]. This paradigm offers significant improvements in interpretability and scalability. More importantly, it lays the foundation for lifelong personal agents, where preference representa-

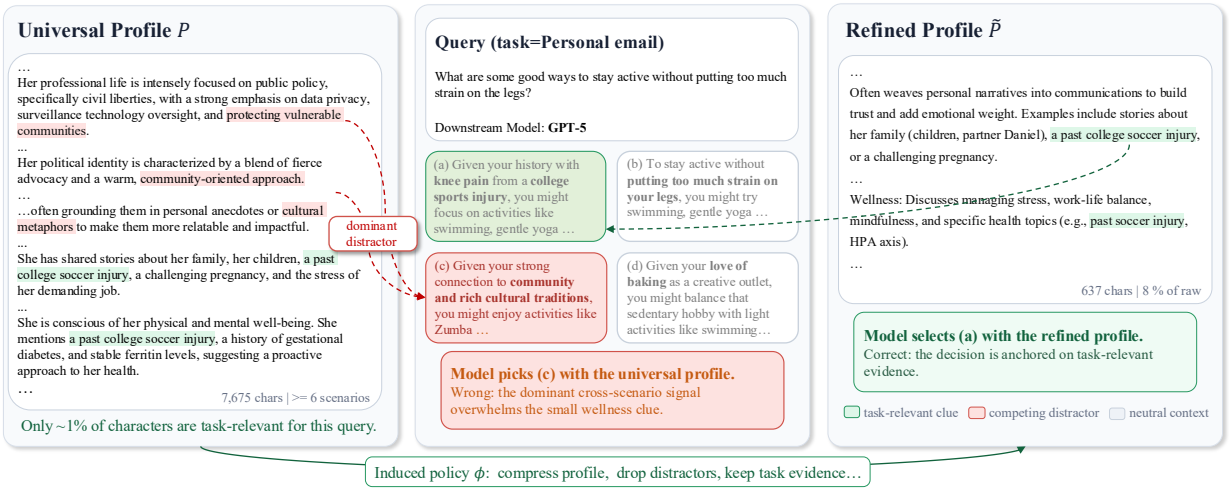

Figure 1: An example for task-specific preference adaptation. Given a query, only a small portion of the universal profile P is relevant to the task, while the remaining information may act as distractors and lead to suboptimal personalization. AlignXada refines P by retaining task-relevant evidence and removing distracting context, enabling the downstream model to ground its response in the appropriate user information.

tions can continuously evolve as user–agent interactions accumulate over time.

In this work, we study LLM personalization from the perspective of universal user preference interfaces. Rather than pursuing improved methods for constructing universal preference summaries, we assume a universal user profile is already available and investigate a complementary yet practically important problem: *how to effectively adapt the universal profile to specific tasks*. Our motivation stems from the observation that universal preference summaries, while comprehensive, inevitably contain redundant or task-irrelevant information that may act as noise in downstream applications. As illustrated in Figure 1, the user query is strongly associated with her past college injury, whereas information such as "community-oriented approach" is largely irrelevant. Such redundant information misleads the downstream model (i.e., GPT-5), resulting in a suboptimal response. In contrast, the result can be substantially improved by using an adapted profile that is shorter and more focused on task-relevant information.

Toward task-specific preference adaptation, we propose a framework for transforming universal preference summaries into task-aware representations. Ideally, the adapted representation should satisfy two desiderata: (1) *Sufficiency*, preserving all preference information necessary for downstream personalization; and (2) *Compactness*, removing redundant or task-irrelevant information to reduce noise and improve context efficiency. To this end, we propose AlignXada, a meta-learning framework for preference adaptation. Rather than directly training a model to refine user profiles, AlignXada employs a meta learner to learn structured refinement policies in natural language, which are subsequently used by a frozen refiner to adapt the universal user preference. This disentanglement between policy generation and preference refinement makes the adaptation process transparent and controllable, enabling human-in-the-loop diagnosis and refinement. The meta learner is optimized via verbal reinforcement learning, where refinement policies are iteratively improved using natural language feedback derived from task-specific demonstrations of user preferences. By avoiding parameter updates during policy learning, AlignXada naturally supports both open-source and proprietary models.

We evaluate AlignXada on a composite benchmark spanning nine conversational tasks and four ranking, rating, and generation tasks with three downstream models. Across 39 task–model cells, AlignXada improves 33 cells by an average of 3.82 points while reducing the profile token ratio to 22.8%. It strictly outperforms RAG in 36 cells, showing that task-oriented preference reorganization provides benefits beyond query-level retrieval. On PersonaMem-v2, a faithfulness audit further shows that 97.5% of refined-profile claims are supported by the source profiles, while 83.3% of available gold preference evidence is retained, suggesting that AlignXada mainly performs controlled task-specific compression and reorganization.

We summarize our contributions as follows:

- We formalize the *task-specific preference adaptation* problem, which aims to adapt universal user preferences to downstream tasks by removing redundant or task-irrelevant information, thereby paving the way for lifelong personalized agents equipped with memory.
- We introduce AlignXada, a meta-learning framework that induces natural language refinement policies from a small set of task-specific demonstrations. The policy is iteratively optimized via verbal reinforcement learning,

making AlignXada compatible with both open-source and proprietary LLMs.
- We conduct extensive evaluations across thirteen tasks and three downstream models. The results show that AlignXada consistently achieves a favorable trade-off between task performance and context-token usage, owing to the faithfulness and compactness of the refined preferences.

2 Related Work

2.1 LLM Personalization

As large language models (LLMs) evolve from generalized chatbots into personal AI agents, there has been growing interest in building personalized AI systems in which the behavior of a general-purpose LLM is aligned with individual preferences. Existing approaches can be broadly categorized into four groups. Retrieval-based methods identify relevant user records or profile elements from external memory or databases [Sun et al., 2025, Du et al., 2026, Liu et al., 2023, Shi et al., 2025], and leverage the retrieved content for downstream personalization. Parametric approaches encode persona information into trainable model parameters, such as LoRA modules, adapters, soft prompts, model-merging weights, and rerankers [Tan et al., 2024, Clarke et al., 2024, Li et al., 2023, Liu et al., 2025a, Jang et al., 2023, Zhuang et al., 2024], thereby adapting a general-purpose LLM toward user-specific behavior. In addition, motivated by the strong in-context learning capability of LLMs, prompt-based approaches append preference signals to the input context and steer response generation through personalized prompts [Dong et al., 2023, Yang et al., 2024b, Cheng et al., 2024, Li et al., 2025a]. More recently, with the emergence of lifelong personal agents and LLM memory systems, several studies have advocated learning universal user profiles as long-term and continuously evolving representations of user preferences [Li et al., 2025b, Liu et al., 2026, Yang et al., 2026]. AlignXada is motivated by this trend toward lifelong personalization. However, rather than constructing universal profiles themselves, AlignXada assumes such profiles are already available and studies how to adapt them to specific downstream tasks in real-world applications. In this sense, AlignXada complements existing efforts by bridging the gap between universal profile construction and task-oriented deployment.

2.2 Textual Optimization

With advances in LLM reasoning capabilities, recent work has used strong LLMs to optimize natural-language artifacts without gradient-based training. OPRO [Yang et al., 2024a] iteratively proposes task-level instructions based on reward trajectories, EvoPrompt [Guo et al., 2024] applies evolutionary operators to candidate prompts, TextGrad [Yuksekgonul et al., 2024] propagates verbal "gradients" through prompt computation graphs, and Reflexion [Shinn et al., 2023] generates reflective critiques for per-instance trajectories. In these methods, the optimized artifact is typically either a *task-level* instruction shared across users or a *per-instance* self-correction text, making it user-independent or instance-local. AlignXada extends this paradigm by formulating task-specific profile adaptation as the optimization of a reusable *user-conditional* rewrite policy that transforms each user's universal preference into a task-specific representation, rather than introducing a new general-purpose textual optimizer.

3 Methodology

Figure 2 presents an overview of AlignXada. In a nutshell, AlignXada consists of two stages: few-shot policy induction and policy deployment for task-specific personalization. During policy induction, the framework leverages a small support set of task-specific demonstrations, each consisting of a universal preference summary, a user query, and the corresponding user response, and iteratively refines the rewriting policy generated by a meta learner using natural language feedback. Once policy optimization converges, AlignXada enters the deployment stage, where the selected policy is consumed by a refiner to produce a refined profile for task adaptation. The refined profile is then provided to downstream models for personalized inference. Throughout the entire process, all models—including the meta learner, the refiner, and the downstream model—remain frozen. In the following, Section 3.1 formalizes the learning problem, and Section 3.2 presents the policy induction procedure based on verbal reinforcement learning.

3.1 Problem Formalization

Let u denote a user and P_u denote the corresponding universal preference summary, which may be obtained from an external LLM or memory system and is treated as prior knowledge in this work. Given a task τ, the objective is to derive a task-adapted profile $\tilde{P}_u$ from P_u such that the performance of a downstream model M for user u can be substantially improved on task τ.

A common approach to deriving $\tilde{P}_u$ from P_u is to learn a generative model R as a refiner and define $\tilde{P}_u = R(P_u, \mathcal{M}_\tau)$, where $\mathcal{M}_\tau$ specifies the task τ. In this work, we instantiate $\mathcal{M}_\tau$ as a support set $S^{(\tau)}$ defined as

$$S^{(\tau)} = \{(P_{u_i}, x_i, y_i)\}_{i=1}^{b}, \tag{1}$$

where P_{u_i} is the universal profile of user u_i, x_i denotes an input prompt, y_i denotes the corresponding response, and b is the size of the support set. Although the representational capacity of $S^{(\tau)}$ may be limited by the selection of (P_{u_i}, x_i, y_i) tuples and the budget b, such a formulation naturally aligns with real-world user-AI interactions: user prompts specify the task, while user responses provide signals of task-specific

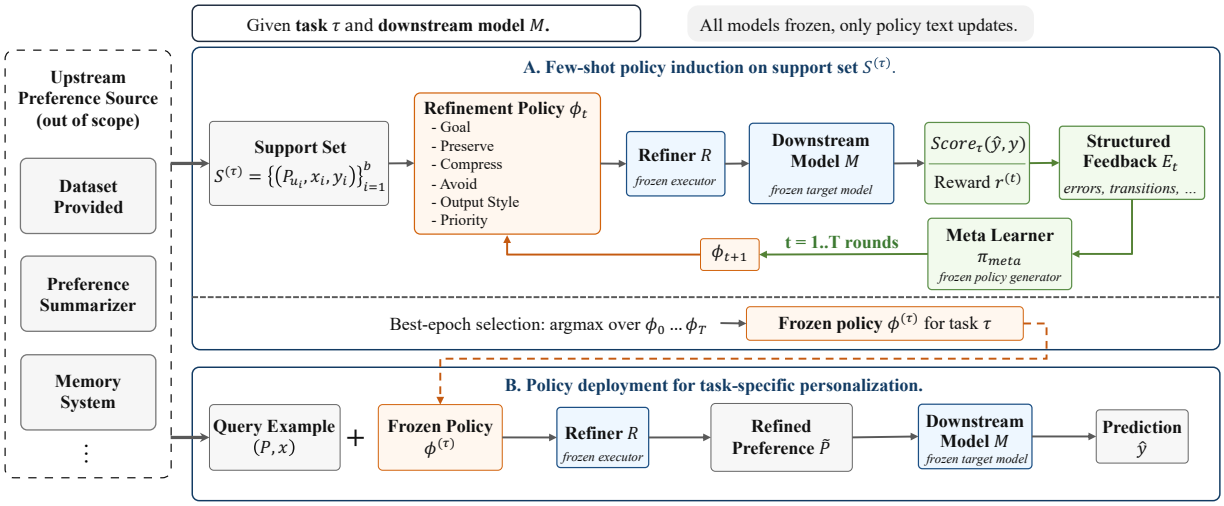

Figure 2: Overview of AlignXada. For each task, AlignXada induces a task-specific textual refinement policy from a small support set in several rounds and then freezes the selected policy for held-out deployment. All models remain frozen throughout, and only the policy text is updated during induction.

preferences and behavioral patterns, thereby alleviating the cold-start problem.

Instead of updating the parameters of R, we keep the model frozen and learn a textual refinement policy $\phi^{(\tau)}$ using a meta learner π_{meta}. The task-adapted profile is then defined as $\tilde{P}_u = R(P_u, \phi^{(\tau)})$. The learning objective is to maximize

$$\mathbb{E}_{(P_{u_i}, x_i, y_i) \sim S^{(\tau)}} \left[\text{Score}_\tau \left(M(x_i, \tilde{P}_{u_i}), y_i \right) \right], \tag{2}$$

where $\text{Score}_\tau(\cdot, \cdot)$ denotes the evaluation function for task τ.

We propose a verbal reinforcement learning approach to optimize Eq. (2), where the refinement policy $\phi^{(\tau)}$ is iteratively estimated from π_{meta} using feedback derived from $S^{(\tau)}$. Details are presented in next section.

3.2 Policy Induction

Overview. Algorithm 1 presents the task-specific policy induction procedure in AlignXada. Starting from a predefined task-agnostic initial policy ϕ_0 (detailed in Appendix F.1), AlignXada iteratively induces a refinement policy for task τ through two core flows: *rollout* and *update*. At each round t, the rollout flow evaluates the current policy ϕ_t on the support set $S^{(\tau)}$ by applying the refiner R, querying the downstream model M, and computing task-specific scores. The update flow then summarizes the rollout records into structured feedback E_t, which is used by the meta learner π_{meta} to revise the policy and produce the next policy ϕ_{t+1}. After all rounds, AlignXada returns the policy with the highest development-set performance.

Rollout. At round t, AlignXada evaluates the current refinement policy ϕ_t on each support example $(P_{u_i}, x_i, y_i) \in S^{(\tau)}$. The refiner R applies ϕ_t to rewrite the universal preference summary into a task-adapted profile:

$$\tilde{P}_{u_i,t} = R(P_{u_i}, \phi_t). \tag{3}$$

The detailed prompt template is provided in Appendix E.1. The downstream model then conditions on the input x_i and the refined profile $\tilde{P}_{u_i,t}$ to generate a prediction:

$$\hat{y}_{i,t} = M(x_i, \tilde{P}_{u_i,t}). \tag{4}$$

This prediction is evaluated against the reference response y_i using the task-specific evaluation function:

$$s_{i,t} = \text{Score}_\tau(\hat{y}_{i,t}, y_i). \tag{5}$$

The rollout record at round t is then defined as

$$\mathcal{R}_t = \{(P_{u_i}, x_i, y_i, \tilde{P}_{u_i,t}, \hat{y}_{i,t}, s_{i,t})\}_{i=1}^{|S^{(\tau)}|}. \tag{6}$$

To reduce overfitting, AlignXada evaluates each policy on a development set $D^{(\tau)}$ disjoint from $S^{(\tau)}$. Applying Eqs. 3–5 to its examples yields development scores $s^D_{i,t}$, whose average defines the policy utility:

$$J_{D^{(\tau)}}(\phi_t) = \frac{1}{|D^{(\tau)}|} \sum_{i=1}^{|D^{(\tau)}|} s^D_{i,t}. \tag{7}$$

The evaluated policy and its development-set utility are stored in the history:

$$\mathcal{H} \leftarrow \mathcal{H} \cup \{(\phi_t, J_{D^{(\tau)}}(\phi_t))\}. \tag{8}$$

Maintaining this history allows AlignXada to retain all evaluated policies, as verbal policy updates are not guaranteed to improve monotonically.

Algorithm 1 Task-specific policy induction in AlignXada.

Require: Task τ; support set $S^{(\tau)}$; development set $D^{(\tau)}$; frozen meta learner π_{meta}; frozen refiner R; frozen downstream model M; evaluation function Score_τ; initial policy ϕ_0; number of rounds T
Ensure: Task-specific refinement policy $\phi^{(\tau)}$

$\mathcal{H} \leftarrow \emptyset$ ▷ history of evaluated policies
for $t = 0, 1, \ldots, T$ **do**
  $\mathcal{R}_t \leftarrow \emptyset$ ▷ support-set rollout records
  for all $(P_{u_i}, x_i, y_i) \in S^{(\tau)}$ **do**
    $\tilde{P}_{u_i,t} \leftarrow R(P_{u_i}, \phi_t)$
    $\hat{y}_{i,t} \leftarrow M(x_i, \tilde{P}_{u_i,t})$
    $s_{i,t} \leftarrow \text{Score}_\tau(\hat{y}_{i,t}, y_i)$
    $\mathcal{R}_t \leftarrow \mathcal{R}_t \cup \{(P_{u_i}, x_i, y_i, \tilde{P}_{u_i,t}, \hat{y}_{i,t}, s_{i,t})\}$
  end for
  $J_{D^{(\tau)}}(\phi_t) \leftarrow \frac{1}{|D^{(\tau)}|} \sum_{i=1}^{|D^{(\tau)}|} s^D_{i,t}$
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\phi_t, J_{D^{(\tau)}}(\phi_t))\}$
  if $t < T$ **then**
    $E_t \leftarrow \text{AGG}(\mathcal{R}_t)$
    $\phi_{t+1} \leftarrow \pi_{\text{meta}}(\phi_t, E_t)$
  end if
end for
$\phi^{(\tau)} \leftarrow \arg\max_{(\phi_t, J_{D^{(\tau)}}(\phi_t)) \in \mathcal{H}} J_{D^{(\tau)}}(\phi_t)$
return $\phi^{(\tau)}$

Policy update. After the rollout, AlignXada constructs structured feedback:

$$E_t = \text{AGG}(\mathcal{R}_t), \tag{9}$$

where $\text{AGG}(\cdot)$ converts the rollout records into a textual diagnostic summary. The feedback includes scalar signals, such as the average support-set score; instance-level signals, such as predictions and scores; and representative failure patterns derived from low-scoring examples. The complete feedback format is provided in Appendix E.3.

Given the current policy and structured feedback, the frozen meta learner produces a revised policy:

$$\phi_{t+1} = \pi_{\text{meta}}(\phi_t, E_t). \tag{10}$$

The update prompt (provided in Appendix E.2) instructs π_{meta} to follow the predefined policy schema, make targeted revisions based on the feedback, and avoid example-specific rules that merely memorize support instances.

The revised policy may adjust the refinement goal, the types of user evidence to preserve, the compression strategy, the error patterns to avoid, the output style, or the priority order among these instructions. An example policy is provided in Appendix F.2. In this way, the meta learner converts rollout diagnostics into a reusable task-level refinement policy rather than producing direct answers or per-example corrections.

After T rounds, AlignXada selects and returns the best evaluated policy as the final task-specific refinement policy:

$$\phi^{(\tau)} = \arg\max_{(\phi_t, J_{D^{(\tau)}}(\phi_t)) \in \mathcal{H}} J_{D^{(\tau)}}(\phi_t). \tag{11}$$

4 Experiments

4.1 Experimental Setup

Benchmark. To evaluate profile refinement in multi-domain, lifelong personalization settings, we construct a composite benchmark by integrating PersonaMem-v2 [Jiang et al., 2025] and MemoryCD [Zhang et al., 2026]. We first derive task-agnostic user summaries from both datasets and extract semantic signatures capturing stable interests, preferences, aversions, and contextual constraints. We then match users one-to-one based on semantic compatibility, excluding pairs with explicit preference conflicts. For each matched pair, we construct a shared universal profile by interleaving and summarizing their PersonaMem and MemoryCD histories. Each composite user is evaluated with the same universal profile across 13 downstream tasks: nine PersonaMem-v2 conversational tasks and four MemoryCD tasks—item ranking, rating prediction, review-title generation, and review generation. For each task, we construct user-disjoint support, development, and evaluation sets, ensuring that users involved in policy induction do not appear in the held-out evaluation set. Detailed dataset statistics are provided in Appendix A. We also evaluate the two original benchmarks separately and report the results in Section 4.3.

Evaluation Metrics. We evaluate four-way choice questions with exact accuracy, item ranking with Hit@1, rating prediction with *Rating-Score*, and both generation tasks with ROUGE-L. User profiles are constructed under the all-history memory setting, which summarizes user behavior from the complete historical context. Given the ground-truth rating y and model prediction $\hat{y} \in \{1, 2, 3, 4, 5\}$, Rating-Score is defined as $\text{S}(\hat{y}, y) = \max\left(0, 1 - \frac{|\hat{y}-y|}{4}\right)$.

Alongside task-level scores, we report context compression using the token ratio (TR), defined as the average refined-profile length divided by the average universal-profile length.

Implementation Details. In the main experiments, we use Gemini-2.5-Pro [Comanici et al., 2025] to generate a universal user preference from each user's raw history provided by the benchmark. Unless otherwise stated, both the meta learner and the preference refiner use Gemini-2.5-Pro. We set the support batch size to $b = 20$ and the number of policy-update rounds to $T = 5$, subject to our experimental budget. When more demonstrations are available, we select the support examples using the adaptive sampling method described in Appendix B to mitigate sampling bias. The universal-preference baseline directly passes the universal preference to the downstream model without refinement. The RAG baseline uses BM25 [Robertson and Zaragoza, 2009].

Table 1: Overall performance (%) on the composite benchmark with different downstream models. Higher metric is better unless otherwise specified. Token ratio (TR) denotes the ratio of refined profile token length to those in the original profile. Task-level gains over Raw are significant under a two-sided exact sign test ($p = 2.44 \times 10^{-4}$).

Task	Qwen3-8B				DeepSeek-V4-Flash				GPT-5-mini			
	Raw	RAG	AlignXada	TR(%)↓	Raw	RAG	AlignXada	TR(%)↓	Raw	RAG	AlignXada	TR(%)↓
Chat Acc.	43.46	40.78$_{-2.68}$	**45.63**$_{+2.17}$	18.9	55.34	41.75$_{-13.59}$	**64.08**$_{+8.74}$	18.1	43.40	40.78$_{-2.62}$	**49.51**$_{+6.11}$	19.6
Creative Acc.	55.68	46.59$_{-9.09}$	**56.82**$_{+1.14}$	21.7	59.09	54.55$_{-4.54}$	**73.86**$_{+14.77}$	22.5	59.32	61.36$_{+2.04}$	**65.91**$_{+6.59}$	28.9
Knowledge Acc.	72.19	61.54$_{-10.65}$	**75.00**$_{+2.81}$	14.3	**82.90**	71.01$_{-11.89}$	82.84$_{-0.06}$	33.5	**75.21**	71.01$_{-4.20}$	73.63$_{-1.58}$	18.7
Personal Acc.	**34.52**	33.33$_{-1.19}$	33.33$_{-1.19}$	24.9	45.24	39.29$_{-5.95}$	**54.76**$_{+9.52}$	20.7	47.62	41.67$_{-5.95}$	**50.00**$_{+2.38}$	17.1
Prof.Email Acc.	**39.17**	35.00$_{-4.17}$	37.67$_{-1.50}$	25.1	46.67	35.83$_{-10.84}$	**59.17**$_{+12.5}$	25.8	**41.50**	38.33$_{-3.17}$	40.83$_{-0.67}$	26.4
Prof.Writing Acc.	31.82	27.27$_{-4.55}$	**40.00**$_{+8.18}$	14.6	50.00	40.91$_{-9.09}$	**57.27**$_{+7.27}$	27.8	44.55	36.36$_{-8.19}$	**49.09**$_{+4.54}$	16.8
Social Acc.(%)	36.36	34.09$_{-2.27}$	**40.91**$_{+4.55}$	20.3	56.82	53.41$_{-3.41}$	**62.50**$_{+5.68}$	37.9	48.41	48.86$_{+0.45}$	**51.14**$_{+2.73}$	19.2
Translation Acc.	38.64	34.23$_{-4.41}$	**42.24**$_{+3.60}$	24.4	54.95	45.95$_{-9.00}$	**63.06**$_{+8.11}$	18.6	54.95	50.45$_{-4.50}$	**59.46**$_{+4.51}$	21.7
Trouble Acc.	49.51	46.60$_{-2.91}$	**53.40**$_{+3.88}$	22.3	60.19	55.34$_{-4.85}$	**62.14**$_{+1.94}$	24.3	**55.19**	**55.19**$_{+0.00}$	54.37$_{-0.82}$	18.5
Ranking Hit@1	59.61	55.77$_{-3.84}$	**61.53**$_{+1.92}$	20.8	69.23	78.85$_{+9.62}$	**82.69**$_{+13.46}$	31.9	78.69	76.92$_{-1.77}$	**80.77**$_{+2.08}$	26.7
Rating Score	75.84	75.48$_{-0.36}$	**77.40**$_{+1.56}$	28.2	74.04	76.44$_{+2.40}$	**78.37**$_{+4.33}$	27.2	77.88	**80.29**$_{+2.41}$	79.33$_{+1.45}$	26.4
Title ROUGE-L	13.46	12.99$_{-0.47}$	**15.38**$_{+1.92}$	13.1	12.78	10.76$_{-2.02}$	**16.02**$_{+3.24}$	20.4	13.04	13.17$_{+0.13}$	**13.80**$_{+0.76}$	30.3
Review ROUGE-L	13.94	13.90$_{-0.04}$	**14.37**$_{+0.43}$	22.2	13.89	13.86$_{-0.03}$	**15.32**$_{+1.43}$	22.8	11.61	12.03$_{+0.42}$	**12.06**$_{+0.45}$	18.4
Avg. Δ	–	-3.59	**+2.27**	20.8	–	-4.86	**+7.00**	25.5	–	-1.92	**+2.19**	22.2

For each query, we segment the universal profile at structural boundaries and further divide it into overlapping windows of 120 words with a 30-word overlap. We use the downstream task prompt as the retrieval query and rank all profile chunks with BM25 ($k_1 = 1.5$, $b = 0.75$). The top eight chunks are restored to their original order and concatenated within an approximate 768-token profile budget ($TR \approx 20\%$), which is comparable to the length of profiles refined by our method. The downstream models are Qwen3-8B [Yang et al., 2025], DeepSeek-V4-Flash [DeepSeek-AI, 2026], and GPT-5-mini [OpenAI, 2025]. We evaluate generalization with DeepSeek-V4-Flash as the meta model in Appendix C, provide further analyses beyond performance comparisons in Appendix D, and present case studies in Appendix G. Unless otherwise specified, Qwen3-8B is used as the default downstream model for these analyses.

4.2 Main Results

We evaluate whether AlignXada improves the performance–context trade-off across heterogeneous personalization tasks and downstream models. An effective preference refinement method should reduce preference context while preserving or improving downstream performance. We compare AlignXada with the raw universal-preference and RAG baselines across 39 task–model cells.

The results in Table 1 yield three observations. **(1) AlignXada consistently improves the performance–efficiency trade-off across models and tasks.** Across the 39 task–model cells, AlignXada improves 33, with an average gain of +3.82 points over the raw universal preference. All three downstream models improve on average: +2.27 points for Qwen3-8B, +7.00 for DeepSeek-V4-Flash, and +2.19 for GPT-5-mini. The gains also span task formats, covering all 12 MemoryCD cells. Meanwhile, the refined profiles retain only 22.8% of the original tokens, and performance gains are nearly uncorrelated with token ratio ($r = 0.06$). Thus, AlignXada improves the utility of retained evidence rather than relying on longer profiles. **(2) The advantage over RAG shows that task-oriented preference adaptation goes beyond query-level retrieval.** AlignXada outperforms RAG in 36 cells, ties in one, and underperforms it in only two, with an average margin of +7.28 points. In contrast, RAG reduces the raw-profile score by 3.46 points on average. For example, on professional email with DeepSeek-V4-Flash, RAG lowers accuracy from 46.67% to 35.83%, whereas AlignXada raises it to 59.17%. Retrieval may surface locally relevant records but fragment preferences whose relevance is indirect or distributed across interactions. AlignXada instead reorganizes the consolidated profile into a coherent decision context. **(3) The optimal preference representation depends on both the task and the downstream model.** For professional email, the changes are −1.50, +12.50, and −0.67 points for Qwen3-8B, DeepSeek-V4-Flash, and GPT-5-mini, respectively; for

Table 2: Task-level performance and token ratios on separate PersonaMem-v2 and MemoryCD (values ×100%).

Task	Raw	RAG	AlignXada	TR↓
Benchmark=PersonaMem-v2				
Chat Message	30.11	27.42	**30.91**	31.4
Creative Writing	31.13	27.13	**32.78**	37.0
Knowledge Query	**49.92**	43.98	48.71	43.3
Personal Email	30.93	29.83	**32.39**	40.4
Professional Email	31.09	27.72	**33.83**	38.7
Professional Writing	**28.46**	24.73	27.66	51.6
Social Media Post	25.99	29.66	**29.94**	39.7
Translation	31.80	29.68	**32.43**	38.5
Trouble Consult	32.77	33.06	**35.29**	46.3
Avg. Δ	–	-2.11	**+1.30**	40.8
Benchmark=MemoryCD				
Item Ranking	57.14	54.25	**61.14**	62.1
Rating Prediction	78.86	75.81	**80.00**	59.5
Review Title	13.47	14.32	**14.36**	44.7
Review Generation	13.72	14.04	**14.46**	25.8
Avg. Δ	–	-1.19	**+1.69**	48.0

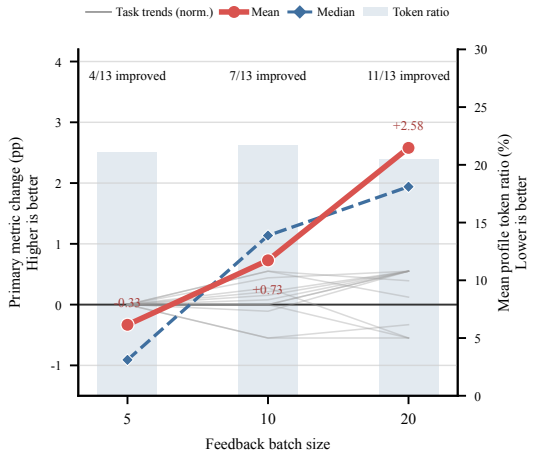

Figure 3: Effect of feedback batch size on the composite benchmark. Solid and dashed lines show the mean and median primary-metric changes, respectively. Gray lines show normalized task-level trends, bars show the mean profile token ratio, and annotations indicate the number of tasks outperforming the raw-profile baseline.

knowledge query, they are +2.81, −0.06, and −1.58 points. These sign reversals indicate that no single compression strategy is optimal for all downstream models, motivating the use of downstream feedback for policy induction. Despite this heterogeneity, all regressions remain within 1.58 points, whereas the largest gain reaches +14.77 points.

4.3 Results on Source-Native Benchmarks

The composite benchmark merges heterogeneous histories from PersonaMem-v2 and MemoryCD into a universal user profile, exposing each downstream task to both relevant and unrelated evidence. Although this setting reflects the noisy histories of lifelong agents, AlignXada may benefit mainly from removing cross-domain noise introduced by benchmark construction. We therefore evaluate AlignXada separately on the two source benchmarks, where profiles are built from more domain-coherent native histories, to test whether it remains effective from a cleaner starting point.

The results in Table 2 show two patterns. **AlignXada remains effective on cleaner, source-native profiles.** It improves the average primary metric by 1.30 points on PersonaMem-v2, outperforming the raw profile on seven of nine tasks, and by 1.69 points on MemoryCD, improving all four tasks. These gains indicate that AlignXada goes beyond removing cross-domain noise by using support-set feedback to induce preference representations better aligned with downstream requirements. **Feedback-based adaptation provides a better performance–compression trade-off than query-level retrieval.** AlignXada outperforms RAG on every task, whereas RAG reduces the average score by 2.11 points on PersonaMem-v2 and 1.19 points on MemoryCD. Meanwhile, AlignXada reduces the profiles to average token ratios of 40.8% and 48.0%, respectively. This suggests that support-set feedback reorganizes user evidence into a compact representation that is more useful to downstream models than locally retrieved excerpts.

4.4 Hyperparameter Analysis

Impact of Support Batch Size b. Figure 3 compares feedback batch sizes $b \in \{5, 10, 20\}$ while keeping the induction pool and update-round budget fixed. **(1) Larger feedback batches improve the reliability of policy induction.** As b increases from 5 to 20, the mean primary-metric change rises from −0.33 to +2.58 points, the median from −0.91 to +1.94 points, and the number of improved tasks from 4/13 to 11/13 (7/13 at $b = 10$). Smaller batches make each update more sensitive to the sampled examples, whereas larger batches provide broader and more balanced feedback, leading to more stable policies. **(2) The gains come from better feedback coverage rather than weaker compression.** $b = 20$ achieves the highest mean and median gains while producing the lowest mean token ratio (20.5%, compared with 21.1% at $b = 5$ and 21.7% at $b = 10$). Thus, its advantage does not result from retaining more context. Instead, broader feedback helps distinguish recurring evidence-loss patterns from isolated failures and prioritize more useful preference evidence. We therefore use $b = 20$ as the default, as it provides the best accuracy–compression trade-off among the evaluated settings. Larger batches are not evaluated because they exceed the context budget.

Impact of the Number of Update Rounds T. Figure 4 shows when the final selected policy first appears in diagnostic runs with $T \in \{5, 10\}$. **(1) Effective refinement policies are typically learned within a few updates.** Eight of the thirteen task-specific policies emerge by the second round, and

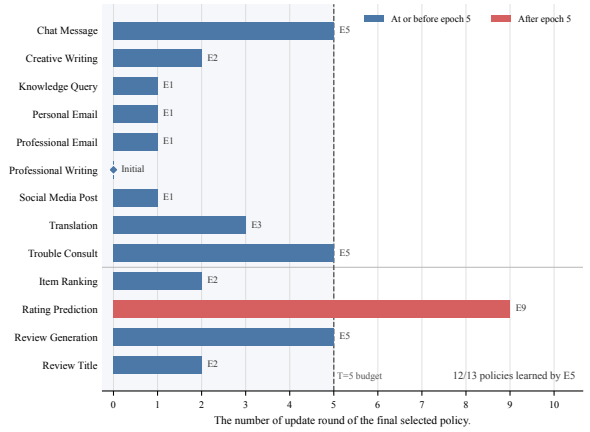

Figure 4: Update round of the final selected policy in the $T \in \{5, 10\}$ diagnostic runs. Blue bars denote policies obtained by round five, red bars denote later policies.

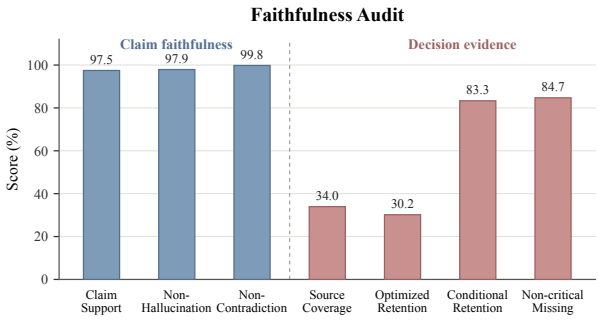

Figure 5: Faithfulness audit result on the PersonaMem-v2.

twelve within five rounds, indicating that the meta-learner can quickly translate support-set feedback into effective policies. **(2) Additional rounds offer task-dependent benefits.** Rating prediction is the only task whose final policy first appears after round five, at round nine. Moreover, among the seven tasks for which $T = 10$ performs better, six select policies generated within the first five rounds. Thus, most tasks converge early, while additional rounds mainly benefit challenging or unstable tasks. Balancing policy quality and inference cost, we use $T = 5$ in the main experiments.

4.5 Faithfulness Audit

Performance and compression alone do not establish whether a refined preference can serve as a faithful substitute for the user universal preference. A refiner may reduce the context budget while removing the decisive preference evidence or introduce fictional traits. We therefore audit the refined preferences along two dimensions: **claim-level faithfulness** to the input profile and **preference evidence retention**. Experimental details are elaborated in Appendix D.2.

We have two main observations about AlignXada from Figure 5. (1) **AlignXada is highly faithful to the source profile.** Across all scenarios, 97.5% of extracted claims are supported by the original profile, the non-hallucination rate is 97.9%, and the non-contradiction rate reaches 99.8%. These results show that the verbal policy does not turn the refiner into an unconstrained generator. Instead, AlignXada largely functions as a controlled compression-and-reorganization operator that preserves the factual content of the universal preference representation while adapting it for downstream use. (2) **The main bottleneck is incomplete decision-evidence availability rather than profile hallucination.** Only 34.0% of benchmark preference evidence is recoverable from the original source profiles, while refined profiles retain 30.2% overall. This small gap from the source-evidence ceiling suggests that many errors arise from insufficient evidence in the universal profile itself, rather than from the refiner deleting available evidence. Conditional on the original profile containing the golden preference, AlignXada preserves it in 83.3% of cases, with a critical missing rate of 15.3%. Therefore, the audit provides a more precise interpretation of AlignXada's downstream limitations: AlignXada can faithfully compress and adapt the profile it receives, but its performance ceiling depends on whether the universal profile contains the cross-topic evidence needed for the decision. The remaining headroom mainly lies in improving source-profile coverage and making decision-relevant evidence more consistently available to the refiner.

5 Conclusion

In this work, we study LLM personalization from the perspective of universal user preference interfaces. We propose AlignXada, which induces a reusable natural-language refinement policy through verbal reinforcement learning, improving the performance–budget trade-off while preserving source-supported claims and critical evidence. By enabling effective use of universal preferences, our work offers a new perspective on memory adaptation for lifelong personalized agents.

Limitations

Despite its advantages, AlignXada has two main limitations. First, although our evaluation spans PersonaMem-v2 and MemoryCD across multiple task formats, both benchmarks are constructed or curated evaluation settings and may not fully capture the evolving, noisy interactions of real-world lifelong agents. Future work should develop broader datasets that provide both universal user preferences and diverse downstream task formats, enabling more comprehensive evaluation of task-specific preference adaptation. Second, AlignXada assumes that the support set used for policy induction is representative of the target task distribution. When the support set is biased or insufficiently diverse, the induced refinement policy may not generalize well. More effective sampling and support-set construction strategies

remain important directions for future work.

Ethics Statement

All experimental datasets used in this study are derived from previously published work and obtained either through official APIs or by synthetic construction based on these sources. All user-related information in the datasets has been anonymized, and no personally identifiable information is involved. We do not use any non-open-source data. All data are used solely for scientific research, rather than for commercial purposes or for profiling or decision-making about individuals, and their acquisition and use comply with relevant ethical guidelines and standards of academic integrity. All existing resources used in our experiments, including datasets, pretrained models, and APIs, are accessed and used in accordance with their original licenses and terms of use. The datasets and models we rely on were filtered and processed by their original authors prior to public release to mitigate potential ethical risks.

References

Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar. Knowledge-augmented large language models for personalized contextual query suggestion. In *Proceedings of the ACM Web Conference 2024*, pages 3355–3366, 2024. doi: 10.1145/3589334.3645404. URL https://doi.org/10.1145/3589334.3645404.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. TALLRec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1007–1014, 2023. doi: 10.1145/3604915.3608857. URL https://doi.org/10.1145/3604915.3608857.

Yizhuo Chen, Xin Liu, Ruijie Wang, Zheng Li, Pei Chen, Changlong Yu, Qingyu Yin, Priyanka Nigam, Meng Jiang, and Bing Yin. POPI: Personalizing LLMs via optimized natural language preference inference. *arXiv preprint arXiv:2510.17881*, 2025. URL https://arxiv.org/abs/2510.17881.

Chuanqi Cheng, Quan Tu, Wei Wu, Shuo Shang, Cunli Mao, Zhengtao Yu, and Rui Yan. "in-dialogues we learn": Towards personalized dialogue without pre-defined profiles through in-dialogue learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10408–10422, 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.

Christopher Clarke, Yuzhao Heng, Lingjia Tang, and Jason Mars. PEFT-U: Parameter-efficient fine-tuning for user personalization. *arXiv preprint arXiv:2407.18078*, 2024. URL https://arxiv.org/abs/2407.18078.

Gheorghe Comanici et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL https://arxiv.org/abs/2507.06261.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/DeepSeek_V4.pdf.

Yi Dong, Zhilin Wang, Makesh Narsimhan Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Linfeng Du, Ye Yuan, Zichen Zhao, Fuyuan Lyu, Emiliano Penaloza, Xiuying Chen, Zipeng Sun, Jikun Kang, Laurent Charlin, Xue Liu, and Haolun Wu. Optimizing user profiles via contextual bandits for retrieval-augmented LLM personalization, 2026. URL https://arxiv.org/abs/2601.12078. Accepted to ACL 2026.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (RLP): A unified pretrain, personalized prompt and predict paradigm (P5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 299–315, 2022. doi: 10.1145/3523227.3546767. URL https://doi.org/10.1145/3523227.3546767.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. EvoPrompt: Connecting LLMs with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2309.08532.

Hermes. Hermes agent: The agent that grows with you, 2026. URL https://hermes-agent.nousresearch.com/.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023. URL https://arxiv.org/abs/2310.11564.

Bowen Jiang, Yuan Yuan, Maohao Shen, Zhuoqun Hao, Zhangchen Xu, Zichen Chen, Ziyi Liu, Anvesh Rao Vijjini, Jiashu He, Hanchao Yu, Radha Poovendran, Gregory Wornell, Lyle Ungar, Dan Roth, Sihao Chen, and

Camillo Jose Taylor. PersonaMem-v2: Towards personalized intelligence via learning implicit user personas and agentic memory. *arXiv preprint arXiv:2512.06688*, 2025. URL https://arxiv.org/abs/2512.06688.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263. URL https://doi.org/10.1109/MC.2009.263.

Jia-Nan Li, Jian Guan, Songhao Wu, Wei Wu, and Rui Yan. From 1,000,000 users to every user: Scaling up personalized preference for user-level alignment. *ArXiv*, abs/2503.15463, 2025a. URL https://api.semanticscholar.org/CorpusID:277113478.

Jia-Nan Li, Jian Guan, Wei Wu, and Rui Yan. Extended inductive reasoning for personalized preference inference from behavioral signals. *ArXiv*, abs/2505.18071, 2025b. URL https://api.semanticscholar.org/CorpusID:278886858.

Lei Li, Yongfeng Zhang, and Li Chen. Personalized prompt learning for explainable recommendation. *ACM Transactions on Information Systems*, 41(4), 2023. doi: 10.1145/3580488. URL https://doi.org/10.1145/3580488.

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 *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9373–9385, 2025a. doi: 10.18653/v1/2025.acl-long.461. URL https://aclanthology.org/2025.acl-long.461/.

Shuai Liu, Hyundong Cho, Marjorie Freedman, Xuezhe Ma, and Jonathan May. RECAP: Retrieval-enhanced context-aware prefix encoder for personalized dialogue response generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8404–8419, 2023. doi: 10.18653/v1/2023.acl-long.468. URL https://aclanthology.org/2023.acl-long.468/.

Yuting Liu, Jinghao Zhang, Yizhou Dang, Yuliang Liang, Qiang Liu, Guibing Guo, Jianzhe Zhao, and Xingwei Wang. CoRA: Collaborative information perception by large language model's weights for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12246–12254, 2025b. doi: 10.1609/aaai.v39i12.33334. URL https://doi.org/10.1609/aaai.v39i12.33334.

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. *arXiv preprint arXiv:2601.04963*, 2026. URL https://arxiv.org/abs/2601.04963.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. LLM-rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 583–612. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.39. URL https://aclanthology.org/2024.findings-naacl.39/.

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. *arXiv preprint arXiv:2507.13579*, 2025. URL https://arxiv.org/abs/2507.13579.

OpenAI. GPT-5 system card, 2025. URL https://openai.com/index/gpt-5-system-card/.

OpenClaw. Openclaw: The ai that actually does things, 2026. URL https://openclaw.ai/.

Atsushi Otsuka, Kazuya Matsuo, Ryo Ishii, Narichika Nomoto, and Hiroaki Sugiyama. User-specific dialogue generation with user profile-aware pre-training model and parameter-efficient fine-tuning. *arXiv preprint arXiv:2409.00887*, 2024. URL https://arxiv.org/abs/2409.00887.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Zhiyuan Peng, Xuyang Wu, Huaixiao Tou, Yi Fang, and Yu Gong. Memrerank: Preference memory for personalized product reranking, 2026. URL https://arxiv.org/abs/2603.29247.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, page 333–389, 2009.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. Optimization methods for personalizing large language models through retrieval augmentation. *arXiv preprint arXiv:2404.05970*, 2024. URL https://arxiv.org/abs/2404.05970.

Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. Retrieval augmented generation with collaborative filtering for personalized text generation. *arXiv preprint arXiv:2504.05731*, 2025. URL https://arxiv.org/abs/2504.05731.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 8634–8652, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/1b44b878bb782e6954cd888628510e90-Abstract-Conference.html.

Chenkai Sun, Ke Yang, Revanth Gangi Reddy, Yi Fung, Hou Pong Chan, Kevin Small, ChengXiang Zhai, and Heng Ji. Persona-DB: Efficient large language model personalization for response prediction with collaborative data refinement. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 281–296, 2025. URL https://aclanthology.org/2025.coling-main.20/.

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. URL https://arxiv.org/abs/2402.04401. Accepted to EMNLP 2024 Main.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2306.07174.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2502.12110. arXiv:2502.12110.

An Yang et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Bufang Yang, Lilin Xu, Yixuan Li, Kaiwei Liu, Xiaofan Jiang, and Zhenyu Yan. Sensorpersona: An llm-empowered system for continual persona extraction from longitudinal mobile sensor streams, 2026. URL https://arxiv.org/abs/2604.06204.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations (ICLR)*, 2024a. URL https://arxiv.org/abs/2309.03409.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. In *Proceedings of the 41st International Conference on Machine Learning*, pages 56276–56297, 2024b.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. TextGrad: Automatic "differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024. URL https://arxiv.org/abs/2406.07496.

Jinghao Zhang, Yuting Liu, Wenjie Wang, Qiang Liu, Shu Wu, Liang Wang, and Tat-Seng Chua. Personalized text generation with contrastive activation steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7128–7141, 2025. doi: 10.18653/v1/2025.acl-long.353. URL https://aclanthology.org/2025.acl-long.353/.

Weizhi Zhang, Xiaokai Wei, Wei-Chieh Huang, Zheng Hui, Chen Wang, Michelle Gong, and Philip S Yu. Memorycd: Benchmarking long-context user memory of llm agents for lifelong cross-domain personalization. *arXiv preprint arXiv:2603.25973*, 2026.

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, pages 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://doi.org/10.1609/aaai.v38i17.29946.

Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. HYDRA: Model factorization framework for black-box LLM personalization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.02888. arXiv:2406.02888.

Table 3: Statistics of the constructed composite benchmark.

Task	Support	Query	Total
Personal Email	40	84	210
Professional Email	40	120	256
Professional Writing	40	110	269
Creative Writing	40	88	239
Translation	40	111	264
Trouble Consult	40	103	243
Chat Message	40	103	226
Social Media Post	40	88	217
Knowledge Query	40	169	416
Item Ranking	40	52	131
Rating Prediction	40	52	131
Review Title Generation	40	52	131
Review Generation	40	52	131
Total	**520**	**1,184**	**2,864**

Table 4: Statistics of the PersonaMem-v2 benchmark.

Task	Support	Query	Total
Personal Email	49	424	473
Professional Email	50	485	535
Professional Writing	63	453	516
Creative Writing	52	457	509
Translation	52	478	530
Trouble Consult	49	452	501
Chat Message	49	444	493
Social Media Post	56	426	482
Knowledge Query	61	661	722
Total	**481**	**4,280**	**4,761**

A Benchmark Statistics

Tables 3–5 summarize the task distributions used in our main and source-native evaluations. Since a user may contribute multiple targets and PersonaMem-v2 users do not necessarily have an example in every conversational scenario, the number of available records varies across tasks. In Table 3, *Support* denotes the fixed induction budget used for each task, *Query* denotes the held-out instances used for final evaluation, and *Total* denotes all constructed task records before runtime support and development subsampling; the latter therefore also includes candidate induction and development records not displayed in separate columns. Overall, the composite benchmark contains 2,864 records across 13 tasks. For the source-native evaluations, PersonaMem-v2 contributes 4,761 examples across nine conversational tasks, while MemoryCD contributes 2,240 examples across four recommendation and generation tasks. All experiments use user-disjoint induction and evaluation splits to prevent policy learning from exploiting evaluation-user histories.

Table 5: Statistics of the MemoryCD benchmark.

Task	Support	Query	Total
Item Ranking	160	400	560
Rating Prediction	160	400	560
Review Title Generation	160	400	560
Review Generation	160	400	560
Total	**640**	**1600**	**2240**

B Adaptive Sampling for Policy Learning

The meta learner has a limited context budget, so each policy update can use at most b support examples even when a larger candidate pool is available. This makes the choice of support examples important. Fixed or randomly sampled batches may over-represent easy successes, persistent failures, or isolated regressions, leading to updates that are either too conservative or too reactive. To obtain a more balanced diagnostic view, we use adaptive diagnostic sampling, which selects support examples from multiple outcome-transition states.

In the main experiments, we reserve a candidate pool $C^{(\tau)} = \{(P_{u_i}, x_i, y_i)\}_{i=1}^{m}$ for adaptive support selection and a development set $D^{(\tau)}$ for final policy selection. Here, m denotes the number of candidate examples available in the experimental split rather than a method-level budget. At each update, the sampler selects b examples from $C^{(\tau)}$ to form the support set $S^{(\tau)}$, where $b \le m$. This allows each update to remain within the context budget while preserving broader coverage across users, topics, and difficulty levels. In deployment, the candidate-pool size is determined by the available task-specific interactions.

For each support example, we compare two binary outcomes: the outcome under the raw universal preference and the most recent outcome under the current refinement policy. This comparison assigns each example to one of four transition states:

- IMPROVED: the raw profile is incorrect, but the refined profile is correct, indicating useful rewritten-profile representations;
- REGRESSED: the raw profile is correct, but the refined profile is incorrect, exposing evidence that may have been removed or made less usable;
- STABLE-SUCCESS: both the raw and refined profiles are correct, identifying safe compression behavior;
- PERSISTENT-FAILURE: both the raw and refined profiles are incorrect, revealing evidence requirements that neither profile makes accessible to the downstream model.

At each update round, the sampler partitions the candidate pool by these transition states and allocates the support budget uniformly across them. If b is not divisible by

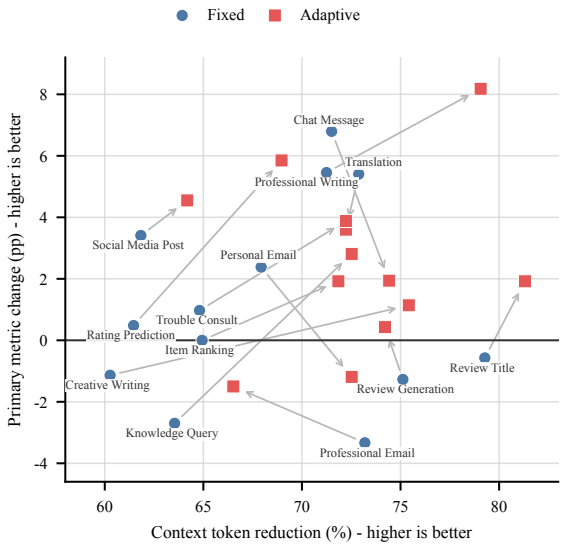

Figure 6: Fixed versus adaptive support sampling on the composite benchmark. Each arrow connects the same task under fixed sampling (circle) and adaptive sampling (square). The horizontal axis reports context-token reduction and the vertical axis reports the change in the task-specific primary metric; higher is better on both axes.

four, the remaining slots are assigned deterministically, with priority given to IMPROVED and REGRESSED examples. If a transition state has too few examples, its unused slots are redistributed to the remaining states. This strategy provides the meta learner with both corrective signals from failures and stabilizing signals from successes in each update.

We compare fixed and adaptive sampling in Figure 6 and make two observations. **(1) Adaptive sampling improves the overall performance–compression trade-off rather than trading additional context for better predictions.** The mean task-level gain in the primary metric increases from +1.22 points under fixed sampling to +2.58 points under adaptive sampling, while the number of tasks that improve over the raw-profile baseline rises from 7/13 to 11/13. At the same time, the mean context-token reduction increases from 68.3% to 72.7%. Ten of the thirteen arrows move upward, ten move rightward, and eight move in both directions. Adaptive sampling therefore yields a broadly more favorable shift in performance–compression space by improving the selection of feedback examples used for policy induction, rather than by retaining longer refined profiles. **(2) The primary benefit is the recovery of tasks for which fixed sampling induces unstable or incomplete policies.** Adaptive sampling changes the gains for creative writing, knowledge query, review title generation, and review generation from negative to positive. It also increases the gain for rating prediction from +0.48 to +5.85 points, professional writing from +5.45 to +8.18 points, and trouble consultation from +0.97 to +3.88 points. These tasks benefit from repeatedly exposing the meta-learner to both regressions, which reveal decision-relevant evidence discarded by the current policy, and persistent failures, which reveal evidence that the policy still fails to surface. The effect is not uniform: fixed sampling remains stronger for chat message, personal email, and translation, while professional email remains below the raw-profile baseline. Adaptive sampling should therefore be interpreted as a more reliable strategy for allocating feedback across heterogeneous tasks, rather than as a guarantee of improvement on every task.

Table 6: Performance and token ratio with DeepSeek-V4-Flash as the meta model, rewrite model, and the target model (values ×100%).

Task	Raw	RAG	AlignXada-D	TR↓
Chat Message	55.34	41.75	60.24	16.2
Creative Writing	59.09	54.55	68.71	23.1
Knowledge Query	82.90	71.01	79.16	30.7
Personal Email	45.24	39.29	48.81	16.9
Professional Email	46.67	35.83	56.67	24.6
Professional Writing	50.00	40.91	52.13	26.5
Social Media Post	56.82	53.41	61.06	31.8
Translation	54.95	45.95	61.48	20.4
Trouble Consult	60.19	55.34	60.91	22.3
Item Ranking	69.23	78.85	82.42	33.7
Rating Prediction	74.04	76.44	77.83	26.2
Review Title	12.78	10.76	15.35	20.5
Review Generation	13.89	13.86	14.50	18.4

C Generalization Across Different Meta Models

To assess whether AlignXada generalizes beyond its default model configuration, we replace Gemini-2.5-Pro with DeepSeek-V4-Flash for policy induction and profile rewriting. We also use DeepSeek-V4-Flash as the downstream model, yielding an all-DeepSeek variant, AlignXada-D, while keeping the learning procedure and experimental protocol unchanged.

AlignXada remains effective when all model roles use DeepSeek-V4-Flash, indicating that its gains are not specific to Gemini-2.5-Pro. AlignXada-D improves 12 of the 13 tasks over the raw universal preference, with an average gain of +4.47 points. It improves eight of the nine PersonaMem-v2 tasks, averaging +4.22 points, and all four MemoryCD tasks, averaging +5.04 points. The refined profiles retain only 23.95% of the original tokens. These results demonstrate that AlignXada transfers across model families and task formats while maintaining a favorable performance–efficiency trade-off.

Its consistent advantage over RAG suggests that this transfer stems from preference-level adaptation rather

than model-specific retrieval behavior. AlignXada-D outperforms RAG on all 13 tasks by 9.33 points on average, whereas RAG falls 4.86 points below the raw-profile baseline. Its largest gains over the raw profile occur on item ranking (+13.19), professional email (+10.00), and creative writing (+9.62), spanning ranking, classification, and open-ended generation. Thus, feedback-guided preference reorganization remains effective even when policy induction, rewriting, and downstream inference use the same model family.

Different policy-induction and rewriting models nevertheless produce distinct performance–compression trade-offs. For the same DeepSeek-V4-Flash downstream model, the default Gemini-based configuration achieves an average gain of +7.00 points with a token ratio of 25.5%, compared with +4.47 points and 23.95% for AlignXada-D. AlignXada-D therefore produces slightly more compact profiles but smaller performance gains, with knowledge query as its only regression (−3.74 points). These results distinguish framework generality from model interchangeability: AlignXada transfers across model families, but the choice of policy-induction and rewriting models still affects how effectively decision-relevant evidence is identified and retained.

D Diagnostic Analysis on PersonaMem-v2

D.1 Preference Source Robustness

To examine whether AlignXada's behavior depends on the quality and format of the universal preference representation, we construct three sources of universal preference information on PersonaMem-v2. **Source A** summarizes a user's identity, background, interests, communication style, and other attributes using GPT-5 [OpenAI, 2025], which is provided by the benchmark. We use A to test whether AlignXada can still improve a compact, curated persona preference that has already removed much of the conversational detail. **Source B** uses the persona's raw structured JSON record from the benchmark. This record is the generator-side persona artifact used to create the benchmark's preferences, conversation snippets, and answers. To avoid direct conversation leakage, we remove the benchmark conversation branch while retaining the structured persona and preference fields. We use B as a high-information structured-source stress test to measure whether a verbal refiner can replace hand-designed field selection when explicit preferences are available. **Source C** summarizes each user's raw chat history into a comprehensive natural language preference description using Gemini-2.5-Pro. Unlike A and B, C is therefore a history-derived preference rather than a persona-generation artifact. We use C as the default source in the main experiments because it best matches our target setting: a system first summarizes long user–assistant interactions into a rich but task-agnostic preference summary, and AlignXada then adapts that profile into a task-specific preference representation.

The results are shown in Table 7, from which we draw the following observations. **(1) AlignXada works best as a task-specific refiner for natural-language preference profiles, rather than as a replacement for upstream profile construction.** Both Source A and Source C improve after refinement, with absolute gains of +0.61% and +1.30%, respectively, whereas Source B starts from the strongest raw baseline but decreases after refinement. This pattern suggests that AlignXada benefits from sources whose evidence is already expressed in natural language, while raw structured records may require a more field-aware transformation. **(2) The gains on A and C come from reorganizing available preference evidence into a more task-oriented form.** Source A has the lowest raw accuracy (25.67%), because the dataset-provided expanded persona is compact and often lacks the full cross-topic evidence required by downstream tasks. AlignXada still improves it to 26.28%, showing that even a limited persona can benefit from task-specific reorganization. Source C starts from a stronger raw baseline (32.47%), as the comprehensive profile contains richer user history, and AlignXada further improves it to 33.77%. This makes Source C the best match for our target setting: it is sufficiently evidence-rich for personalization while remaining in natural language, allowing the refiner to reliably compress and reorganize it. **(3) The drop on B suggests a format mismatch rather than a lack of useful information.** Source B's high raw accuracy (32.91%) is expected, because it is the structured persona used to synthesize the benchmark and often contains explicit preference aligned with the query-required evidence. However, after refinement, accuracy drops to 32.07%. One likely reason is that many useful signals are stored as fine-grained structured fields or leaf values; rewriting them into a compact natural-language profile can remove exact decision evidence that the downstream model can directly read. From the compression perspective, this result also shows that strong compression alone is insufficient. The best trade-off is achieved by Source C, where AlignXada obtains a positive gain while reducing the profile to 40.8% of its original token length.

D.2 Faithfulness Audit Details

In Section 4.5, we audit refined preferences along two dimensions: **claim-level faithfulness** to the input profile and **preference evidence retention**. The experimental setup and implementation details are as follows:

Claim-level faithfulness audit. For each unique raw–refined preference pair, we use DeepSeek-V4-Pro [DeepSeek-AI, 2026] as the judge model to decompose the refined preference into atomic claims and label each claim with respect to the universal preference as *supported*, *contradicted*, *not found*, or *too vague*. Let N_c denote the total number of atomic claims across all audited pairs. The three claim-level metrics—Claim Support, Non-Hallucination, and Non-

Table 7: Ablation on the source of the universal preference (values ×100%).

	Source A			Source B			Source C		
Task	Raw	AlignXada	TR(↓)	Raw	AlignXada	TR(↓)	Raw	AlignXada	TR(↓)
Chat Message	28.37	28.64	36.8	30.19	29.73	16.4	30.11	30.91	31.4
Creative Writing	24.58	24.13	49.7	33.42	31.89	31.2	31.13	32.78	37.0
Knowledge Query	33.21	37.84	35.3	52.76	50.18	23.6	49.92	48.71	43.3
Personal Email	22.73	23.46	53.2	31.95	30.27	28.7	30.93	32.39	40.4
Professional Email	23.64	21.37	45.9	29.58	28.42	27.3	31.09	33.83	38.7
Professional Writing	24.82	25.16	45.6	26.94	27.31	26.5	28.46	27.66	51.6
Social Media Post	26.35	26.78	45.2	30.67	29.14	23.8	25.99	29.94	39.7
Translation	25.91	27.32	41.5	34.28	34.76	21.4	31.80	32.43	38.5
Trouble Consult	21.46	21.83	34.7	26.37	26.92	30.1	32.77	35.29	46.3
Avg. Δ	–	+0.61	43.1	–	-0.84	25.4	–	+1.30	40.8

Contradiction—are defined as follows:

$$\begin{aligned} \text{Claim Support} &= \frac{\#\text{supp.}}{N_c}, \\ \text{Non-Hallucination} &= 1 - \frac{\#\text{notf.} + \#\text{cont.}}{N_c}, \\ \text{Non-Contradiction} &= 1 - \frac{\#\text{cont.}}{N_c}. \end{aligned} \tag{12}$$

Here, *supported* indicates that a claim is explicitly stated or semantically entailed by the universal preference, *not found* indicates unsupported new information, and *contradicted* indicates inconsistency with the source. Claims labeled *too vague* are included in N_c but are not counted as supported. We define the judge prompt as follows:

Claim-Level Faithfulness Audit

```
Audit the refined profile against the original profile.

Labels:
- supported: the claim is explicitly stated or semantically entailed by the original profile.
- contradicted: the claim conflicts with the original profile.
- not_found: the claim adds specific information not present in the original profile.
- too_vague: the claim is so generic that it cannot be checked as a concrete profile claim.

Return exactly one JSON object with this schema:
{
  "claims": [
    {
      "claim": "atomic claim from the refined profile",
      "label": "supported | contradicted | not_found | too_vague",
      "evidence": "short supporting or contradicting evidence from the original profile, or empty string",
      "rationale": "brief reason"
    }
  ]
}

Split the refined profile into concise atomic claims. Do not invent claims that are not in the refined profile.

<OriginalProfile>
{original_profile}
</OriginalProfile>

<RefinedProfile>
{refined_profile}
</RefinedProfile>
```

Preference-evidence audit. PersonaMem-v2 provides a "preference" field for each benchmark query, which we treat as ground-truth decision evidence. For each audited query, the judge determines whether this evidence is preserved in the universal preference and in the refined preference. Each preference is labeled as *exact/specific*, *semantically generalized*, *missing*, or *contradicted*, where the first two labels are counted as evidence **retained**. Let N_e denote the number of audited query examples. We define four evidence-level metrics: Source Coverage, Refined Retention, Conditional Retention, and Non-Critical Missing:

$$\begin{aligned} \text{Source Coverage} &= \frac{\#\text{source ret.}}{N_e}, \\ \text{Refined Retention} &= \frac{\#\text{refined ret.}}{N_e}, \\ \text{Conditional Retention} &= \frac{\#\text{source ret. \& refined ret.}}{\#\text{source ret.}}, \\ \text{Non-Critical Missing} &= 1 - \frac{\#\text{critical missing}}{\#\text{source ret.}}. \end{aligned} \tag{13}$$

Here, *source ret.* indicates that the universal preference contains the ground-truth evidence, and *refined ret.* indicates that the refined preference preserves it. A *critical missing* case occurs when the universal preference retains the evidence but the refined preference is labeled *missing*. The judge prompt is defined as follows:

Preference-Evidence Audit

```
Audit whether each profile preserves the benchmark preference.

Labels:
- exact_or_specific: the profile explicitly preserves the preference or a highly specific equivalent.
- semantic_generalized: the profile preserves the preference at a broader but still useful level.
- missing: the profile does not contain usable evidence for the preference.
- contradicted: the profile states the opposite or says the preference was retracted/should not be used.

Return exactly one JSON object with this schema:
{
  "original": {
    "label": "exact_or_specific | semantic_generalized | missing | contradicted",
    "evidence": "short evidence span from the original profile, or empty string",
    "rationale": "brief reason"
  },
  "optimized": {
    "label": "exact_or_specific | semantic_generalized | missing | contradicted",
    "evidence": "short evidence span from the refined profile, or empty string",
    "rationale": "brief reason"
  }
}

<BenchmarkPreference>
{benchmark_preference}
</BenchmarkPreference>

<OriginalProfile>
{original_profile}
</OriginalProfile>

<RefinedProfile>
{refined_profile}
</RefinedProfile>
```

E Prompt Templates

E.1 Rewrite Prompt Template

The refiner receives the current policy ϕ_t, the downstream task family description, and the original profile P_u.

Prompt Template for Preference Generation

```
Current rewrite pattern phi_t:
{current_policy}

Downstream task family:
{task_description}

Original profile P_u:
{original_profile}

Rewrite the profile strictly according to phi_t.

Requirements:
1. Execute phi_t exactly, including its preservation, compression, abstraction, structure, and style rules.
2. Preserve factual faithfulness to P_u; do not add unsupported facts, preferences, or assumptions.
3. Keep the rewrite query-agnostic and do not optimize it for a specific prompt, option, answer, or evaluation instance.
4. Output only the rewritten profile P*_u.
```

E.2 Meta-Learner Prompt Template

The meta learner receives the current policy and mini-batch feedback, then emits only the next policy.

Prompt Template for Policy Update

```
Meta prompt version:
{prompt_version}

Downstream task family:
{task_description}

Primary optimization metric:
{primary_metric_name}

Current pattern phi_t:
{current_policy}

Support-set error information (JSON):
{feedback_payload}

Optimization objective:
- Improve future rewritten-profile performance for the same task family while updating only the rewrite policy, not the downstream answer behavior.
- Preserve or improve the primary metric first; reduce redundant context only after retaining decision evidence.
- Keep the policy query-agnostic so it can serve unseen examples rather than the current support prompts.
```

```
- Treat concrete user facts, constraints, sensitive or anti-stereotypical preferences, and privacy behavior as potential decision evidence unless the feedback shows they are harmful.

Use feedback:
- Prioritize regressions as evidence of omitted, distorted, or under-weighted signals; preserve behaviors that explain improvements and stable successes.
- Separate evidence loss from downstream confusion, ambiguity, or source-profile insufficiency before changing the policy.
- Balance latent evidence with ordinary concrete evidence such as roles, places, objects, language/register, named interests, and everyday preferences.

Constraints:
- Make the smallest useful update to phi_t and avoid overfitting to one example, user, topic, or surface form.
- next_pattern must describe how to rewrite profiles, not how to answer the downstream task.
- Return only the final JSON.
```

E.3 Structured Feedback Template

The structured feedback record E_t is serialized as a JSON list, where each entry corresponds to one support example in the mini-batch. By default, the feedback does not include the full raw preference; instead, the meta learner observes the rewritten profile, paired outcomes, scores, and task metadata.

Feedback Template

```
[
  {
    "idx": int,
    "prompt": string,
    "reference": string,
    "optimized_profile": string,
    "original_prediction": string,
    "optimized_prediction": string,
    "original_primary_score": float,
    "optimized_primary_score": float,
    "score_delta": float,
    "original_metrics": object,
    "optimized_metrics": object,
    "meta_summary": {
      "transition": "improved | regressed | persistent_failure | stable_success",
      "scenario": string,
      "pref_type": string,
      "topic_query": string,
      "reference_letter": string,
      "gold_signal": string,
      "original_prediction": string,
      "original_signal": string,
      "optimized_prediction": string,
      "predicted_signal": string,
      "prediction_changed": bool
    },
    "original_profile_words": int,
    "optimized_profile_words": int,
    "word_reduction": int
  }
]
```

F Policy Examples

F.1 Initial Policy

All experiments start from the same task-agnostic initial policy:

Initial Policy

```
Rewrite the profile into concise, stable preference rules that preserve relevant user signals while reducing unnecessary detail.
```

F.2 An example of Induced Policies

Example Induced Policy

```
Goal: Produce a compact, faithful refined profile that lets a Qwen3-8B downstream answer a 4-way personal-email choice question grounded in the user's relationship context, tone, and recurring email phrasing, while preserving cross-scenario evidence that disambiguates the choice.

Preserve:
- Relationship roles and names of recurring email recipients.
- Tone, greetings, signoffs, and recurring phrasing by relationship.
- Lifestyle constraints and cross-scenario preferences that affect email decisions.

Compress:
- Narrative chit-chat from chat history.
- Duplicated preferences or long descriptions that do not change the email decision.

Avoid:
- Removing cross-scenario evidence or confusing the user's preferences with those of family members.
- Copying exact answer-text phrasings from chat history.

Output Style:
```

```
- Bulleted, with explicit field labels and
relationship/scenario tags.
- No prose narrative.

Priority: When in doubt, preserve
cross-scenario preference evidence over
local email-style fluff.
```

G Case Study

We select a representative user case, together with its raw universal preference and refined preference, as a case study. The raw universal preference leads the downstream model to an incorrect answer, whereas the AlignXada-refined profile leads it to the correct one. The user asks for calming activities or audio options after a long day. Among the four candidate responses, the gold answer recommends a tranquil Bach harpsichord suite or softly recorded Gregorian chant. With the raw universal preference, the downstream model selects a response centered on Japanese shakuhachi and koto music; after refinement, it selects the gold response. The boxes below show lightly formatted excerpts copied from the corresponding JSON outputs, with omitted material marked by ellipses.

Raw Universal Preference Excerpt

This user is a highly educated, articulate, and reflective individual, likely an academic in the humanities, with a professional focus on history, theology, and liturgy, particularly within the Anglican tradition. Their name is Elaine H. Nakamura. They reside in or near Portland, Oregon, and are of East Asian, specifically Japanese, descent. Elaine is deeply engaged in both her professional life as a university lecturer and her personal life within a parish community, where she is a lay leader and choir member.

. . . The rhythms of the church calendar (Advent, feast days) are significant markers in her life.

Her aesthetic sensibilities are refined and consistent. She appreciates:
- **Music:** She is a choir member, rehearsing and performing works by composers like Tallis. She has a sophisticated appreciation for classical music (Bach's *Jesu, meine Freude*), choral traditions, and the emotional impact of music in different settings. She also enjoys karaoke with friends.
- **Literature:** She is an avid reader, enjoying 19th-century novels like *Middlemarch*, mystery novels (P.D. James, Sarah Waters), and poetry. Reading is a cherished ritual, often done in the evening with tea and a cozy blanket.
- **Visual Arts & Design:** She is drawn to art galleries, theater (both scripted and improvisational), and the aesthetics of interior design . . .

AlignXada-Refined Profile Excerpt

Tone & Style
- **Formality:** Prefers a polite, formal, and articulate communication style, even in personal emails.
- **Tone:** Warm, reflective, and slightly literary.
- **Vocabulary:** Uses precise, rich, sensory vocabulary (e.g., "faint patter of rain," "ancient cadences").

. . .

Content & Interests
- **Professional Identity:** University lecturer in Church History. This is a central part of her identity and informs her worldview.
- **Core Academic Interests:** Liturgical practices and history (Anglican tradition, medieval mystery plays, funeral liturgies); theology and its intersection with culture; 19th-century parish library development.
- **Cultural & Aesthetic Preferences:**
  - **Music:** Classical (Bach), choral traditions (Tallis). Participates in a parish choir. Also enjoys karaoke with friends.
  - **Literature:** 19th-century novels (*Middlemarch*), mystery novels (P.D. James, Sarah Waters), poetry. Reading is a cherished ritual.
  - **Arts:** Enjoys art galleries, theater (scripted and improv), and appreciates handmade crafts like greeting cards.

This example shows that AlignXada can improve downstream behavior without introducing new user facts. The raw preference already contains the relevant evidence, but it presents decision-relevant musical and liturgical preferences alongside many other salient identity and cultural details. In this case, the raw model appears to overweight the user's Japanese heritage and selects the candidate mentioning shakuhachi and koto. The refined profile reorganizes the same source-supported information into task-relevant clusters: classical music, choral practice, Anglican parish life, 19th-century reading, and quiet evening rituals. This makes the Bach/Gregorian-chant candidate more directly supported than the Japanese-instrument distractor, while reducing the profile by 46.6%.