USING CONTEXT IS NOT ENOUGH: TEST-TIME TRAINING FOR PERSONALIZED REWARD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning from human feedback (RLHF) aligns large language models (LLMs) with human preferences, yet most pipelines learn a single reward model that overlooks individual differences in preferences. Personalized reward models (PRMs) address this by conditioning rewards on user-specific feedback, most commonly through in-context learning (ICL), where a user's historical comparisons are supplied as contextual preference pairs. However, we identify a key limitation of ICL-based PRMs: they fail to capture the preference relations conveyed by contextual pairs. To address this, we propose **Preference-Aligned Test-Time Training (P-TTT)**, which explicitly encodes these relations into user-specific fast weights for personalized reward prediction. P-TTT introduces sequence-level update and apply operations to match the response-level granularity of preference feedback, together with a preference-aligned objective that directly uses pairwise preference relations to guide fast-weight adaptation. Notably, P-TTT is simple to implement and computationally efficient, updating fast weights within a single forward pass without inference-time backpropagation. Extensive experiments show that P-TTT more effectively captures historical preference relations and outperforms state-of-the-art methods by a large margin.

1 INTRODUCTION

Reinforcement learning from human feedback (RLHF) plays a central role in aligning large language models (LLMs) with human preferences (Christiano et al., 2017; Ouyang et al., 2022). Most existing RLHF pipelines learn one shared reward model from human preference annotations over pairs of preferred and rejected responses, and use it to guide policy optimization. Such a formulation implicitly assumes that all humans share a universal preference (Bradley & Terry, 1952). However, this assumption rarely holds in practice: human preferences are inherently heterogeneous, and different users may prefer different responses to the same prompt (Guan et al., 2025). A single reward model fitted to such data averages over conflicting preferences, which introduces a systematic bias toward the majority. More importantly, it leaves individual preferences unmodeled, and yet personalization has become essential for modern LLM services (Zhang et al., 2024).

Personalized reward models (PRMs) meet this need by conditioning reward estimation on user-specific information, so that the same response can receive different rewards for different users (Poddar et al., 2024). The central question is how a PRM leverages user information. Since new users and new feedback arrive continually, a PRM must infer a personalized reward function on the fly from a few historical pairs rather than through per-user retraining. In-context learning (ICL) offers the most straightforward way to do so and has been widely adopted (Zollo et al., 2025; Ryan et al., 2025): a user's historical comparisons are concatenated into the prompt as *contextual preference pairs*, from which the reward model is expected to infer the user's preference and adjust its reward for a new *target response* accordingly.

However, we identify a key limitation of ICL-based PRMs: *they may fail to effectively capture the preference relations conveyed by contextual pairs*. Concretely, they do not reliably distinguish preferred responses from rejected ones (*cf.* Figure 1). We verify this with a counterfactual test: we reverse the preference direction of the contextual pairs by swapping its chosen and rejected labels, and find that the model's predictions on the same target pairs remain unchanged in 85.95% of cases on average. The result is striking— reversed labels describe a user with exactly the opposite

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

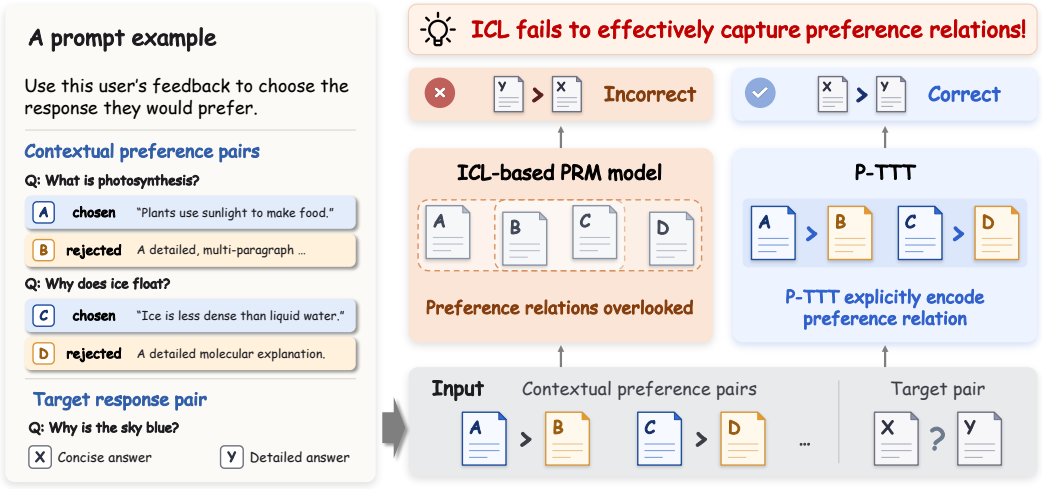

Figure 1: Illustration of personalized reward modeling with ICL and P-TTT (Ours). While an ICL-based PRM can utilize the content of contextual preference pairs, it may fail to capture their preference relations, resulting in an incorrect target prediction. P-TTT explicitly encodes these relations to better guide target response reward evaluation.

interests, yet the model makes almost no corresponding adjustment. It strongly indicates that ICL-based PRMs cannot effectively digest the preference signals expressed in the context. These findings motivate our core research question: ***How can we enable personalized reward models to better capture preference relations from contextual preference pairs?***

Towards this end, we propose **Preference-Aligned Test-Time Training (P-TTT)**. Our approach builds on test-time training (Sun et al., 2024), which adapts a small set of *fast weights* at inference time under an explicit learning objective. Unlike ICL, it writes contextual information into parameters rather than inferring it implicitly from the input. Directly applying existing TTT methods to PRMs, however, mismatches the task in two ways: token-level operations are misaligned with response-level preference feedback and reward prediction, while objectives designed for context memorization or language modeling do not explicitly capture preference relations. P-TTT resolves both with two designs: (1) *sequence-level update and apply operations* to align test-time adaptation with the response-level nature of preference annotations and reward prediction; (2) *preference-aligned objective* that directly uses pairwise preference relations as the learning signal for fast-weight adaptation. Together, these designs enable P-TTT to explicitly internalize such relations and use them to guide target reward prediction.

Notably, P-TTT is easy to implement and computationally efficient. It reuses the MLP down-projection of a standard reward model as fast weights (Feng et al., 2026), and updates only this small set of user-specific parameters within a single forward pass, without inference-time back-propagation. Moreover, because user history is stored in these parameters, P-TTT avoids including historical examples in the context when scoring target responses, reducing self-attention overhead for lengthy contexts and enabling faster inference than ICL. Our theoretical analysis shows that P-TTT captures the direction of preference relations and aligns with the reward modeling objective. Extensive experiments demonstrate that P-TTT consistently outperforms state-of-the-art baselines by up to 5.85 percentage points in accuracy.

In summary, this work makes the following contributions:

- We highlight the importance of modeling preference relations and identify a key limitation of conventional ICL-based strategies: they may fail to effectively capture these relations.
- We propose **Preference-Aligned Test-Time Training (P-TTT)**, a novel strategy for personalized reward modeling that enhances the model's ability to capture preference relations.
- Extensive experiments demonstrate that P-TTT effectively captures preference relations and achieves state-of-the-art performance.

2 PRELIMINARIES

Conventional reward model. Reward models are a key component of RLHF for LLMs, providing scalable approximations of human judgments to guide policy optimization while reducing the need for costly manual evaluation. A reward model $r_\phi(x, y)$ assigns a scalar score to a response y given a prompt x. It is typically implemented as an LLM backbone equipped with a linear reward head (Ouyang et al., 2022):

$$r_\phi(x, y) = \mathbf{v}_r^\top \mathbf{e}_\theta(x, y), \tag{1}$$

where $\mathbf{e}_\theta(x, y)$ is the embedding produced by the LLM, $\mathbf{v}_r$ is the reward-head weight vector, and ϕ denotes the trainable model parameters.

To learn this reward function from pairwise preferences, we consider a dataset $\mathcal{D} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$, where y_i^+ and y_i^- are the chosen and rejected responses to prompt x_i, respectively. The BTL model is then trained by minimizing the negative log-likelihood of the observed preferences:

$$\mathcal{L}_{\mathrm{RM}}(\phi) = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}} \left[\log \sigma \left(r_\phi(x, y^+) - r_\phi(x, y^-)\right)\right] \tag{2}$$

where $\sigma(\cdot)$ denotes the sigmoid function. This formulation learns a shared reward function across users, without explicitly accounting for their individual preferences.

Personalized reward model. A personalized reward model extends this formulation by conditioning the reward function on user-specific information z (Poddar et al., 2024):

$$\mathcal{L}_{\mathrm{PRM}}(\phi) = -\mathbb{E}_{(x,y^+,y^-,z)\sim\mathcal{D}} \left[\log \sigma \left(r_\phi(x, y^+ \mid z) - r_\phi(x, y^- \mid z)\right)\right]. \tag{3}$$

In practice, explicit descriptions of user preferences are often unavailable. Instead, preferences can be inferred from each user u's historical preference dataset $\mathcal{D}_u = \{(x_{u,i}, y_{u,i}^+, y_{u,i}^-)\}_{i=1}^m$, and $y_{u,i}^+$ and $y_{u,i}^-$ are the responses chosen and rejected by user u for prompt $x_{u,i}$, respectively. A widely adopted approach to incorporating this information is in-context learning (ICL), where historical examples from $\mathcal{D}_u$ are provided as *contextual preference pairs* to guide reward prediction for a new *target response* (Jin et al.; Ryan et al., 2025; Zollo et al., 2025). The model is expected to learn to capture the preference information encoded in these contextual pairs through training.

Beyond ICL, other strategies have also been explored, but each faces practical limitations. *Embedding-based methods* (Poddar et al., 2024) encode users' preference information into compact representations for reward prediction. However, compressing complex preferences into a single embedding may discard fine-grained information. Moreover, LLMs pretrained primarily on natural language may struggle to effectively utilize such representations (Nam et al., 2026). Consequently, these methods often exhibit limited performance (*cf.* Section 5.2). *Parameter-based methods* (Kim et al., 2026; Liu et al., 2025) adapt model parameters to individual users based on their preference feedback. Such methods typically require computationally expensive backpropagation to optimize model parameters, incurring substantial computational overhead. This makes repeated adaptation costly in dynamic settings where user preferences and interactions evolve frequently. Moreover, adapting model parameters from limited user-specific feedback can increase the risk of overfitting. Further discussion is provided in Appendix D.

Test-Time Training. Test-time training (TTT) enables models to adapt to context at inference time by updating a set of parameters W, termed *fast weights* (Sun et al., 2024). These weights serve as a memory that retains contextual information for use in subsequent predictions.

Given a sequence of token representations $\{\mathbf{h}_t\}_{t=1}^T$, where $\mathbf{h}_t \in \mathbb{R}^d$, TTT constructs a query, key, and value $(\mathbf{q}_t, \mathbf{k}_t, \mathbf{v}_t)$ for each token and uses them in two core operations: *update* and *apply*. Starting from an initialization W_0, an *update operation* writes contextual information into the fast weights by learning an association between each key $\mathbf{k}_t$ and its corresponding value $\mathbf{v}_t$:

$$W_t = W_{t-1} - \eta \, \nabla_W \mathcal{L}_{\mathrm{TTT}}\big(f_W(\mathbf{k}_t), \mathbf{v}_t\big)\big|_{W=W_{t-1}}, \tag{4}$$

where η is the learning rate and $\mathcal{L}_{\mathrm{TTT}}$ is the adaptation objective. An *apply operation* then uses the query $\mathbf{q}_t$ to retrieve contextual information stored in the updated fast weights W_t by computing $\mathbf{o}_t = f_{W_t}(\mathbf{q}_t)$. These operations allow contextual information to influence subsequent predictions through parameter updates. However, existing TTT approaches are commonly designed either to memorize contextual information (Sun et al., 2024) or to predict next tokens (Feng et al., 2026), rather than modeling pairwise preference relations in PRMs.

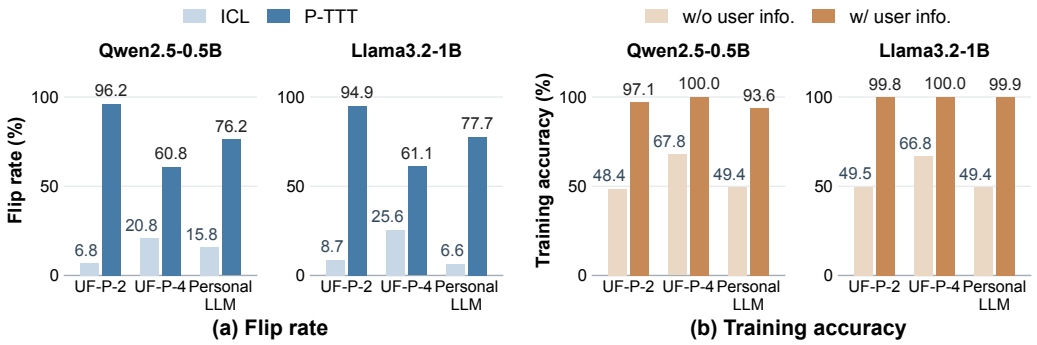

Figure 2: (a) Flip rates of ICL and P-TTT. (b) Training accuracy of ICL-based PRMs with and without contextual preference examples.

3 Empirical Analysis

In this section, we identify a key limitation of ICL-based PRMs: they fail to effectively capture the preference relations expressed in contextual examples.

Preference-flipping analysis. To isolate the role of contextual preference relations, we conduct a counterfactual experiment: we reverse the chosen and rejected labels within selected contextual pairs while leaving the response content unchanged. We then assess whether the model's predictions change in response to these reversals. Specifically, we define the *flip rate* as the fraction of initially correct predictions that change after label reversal. A higher value indicates that contextual preference relations have a greater influence on the model's predictions. Further details are provided in Appendix D. As shown in Figure 2(a), ICL-based PRMs exhibit an average flip rate of only 14.05%, retaining their original predictions in most cases. This low flip rate suggests that the relations expressed in the context exert only limited influence on model predictions.

Do the models simply ignore contextual examples? The low flip rate may arise if the models make limited use of contextual examples. To examine this possibility, we compare training accuracy with and without contextual preference examples. As shown in Figure 2(b), the inclusion of these examples yields substantial accuracy gains across all three datasets and both backbones (e.g., on UF-P-2, from 48.44% to 97.06% for Qwen and from 49.55% to 99.81% for Llama). These results indicate that the models exploit information provided by contextual examples, suggesting that the low flip rate is not attributable solely to a failure to use context.

Conclusion. These results reveal a notable discrepancy: ICL-based PRMs can use contextual examples for prediction, yet struggle to capture the preference relations expressed in them. This suggests that simply placing preference pairs in the context as model input may be insufficient for models to reliably identify the their relations. In particular, the critical preference labels (*chosen* or *rejected*) may not be effectively associated with the corresponding responses, especially when they are embedded in lengthy contextual examples. As a result, rather than modeling the preference relations themselves, models may rely on more readily accessible shortcut cues in historical prompts or responses, such as previously asked questions, to make predictions. Such reliance may limit model generalization to new target samples. We also explored simple prompting strategies to mitigate this issue (see the appendix E), but observed no improvement, suggesting that the difficulty may reflect an intrinsic limitation of how ICL captures preference relations. These findings motivate us to develop a method that strengthens the preference modeling capability of PRMs.

4 Method

In this section, we propose Preference-Aligned Test-Time Training (P-TTT) to address the limited ability of ICL to capture preference relations. We first provide an overview of the method (Section 4.1), then describe its two key components: sequence-level update and apply operations (Sec-

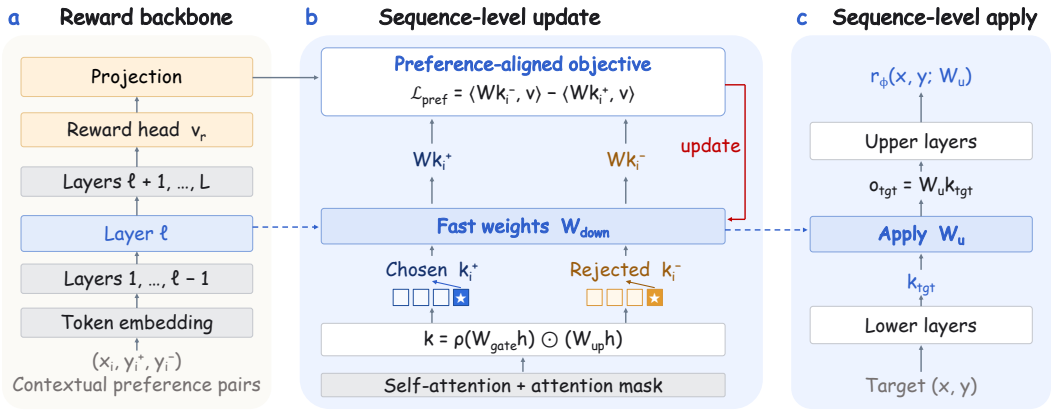

Figure 3: Overview of P-TTT.

tion 4.2) and a preference-aligned objective (Section 4.3). Finally, we present a theoretical analysis of P-TTT (Section 4.4). Figure 3 illustrates the overall framework of P-TTT.

4.1 Motivation

Motivation for Test-Time Training. Our empirical analysis shows that ICL-based PRMs largely underutilize the preference relations conveyed by contextual preference pairs. A key limitation is that preference labels are supplied solely as contextual input, requiring the model to infer how they should inform reward predictions for target responses. This motivates a mechanism that explicitly incorporates contextual preference relations into the model used for prediction. Test-time training (TTT) offers a natural framework for this purpose by adapting fast weights to contextual examples at inference time. With an adaptation objective defined over pairwise preferences, these relations can directly guide fast-weight updates. Building on this insight, we propose **Preference-Aligned TTT (P-TTT)**, which learns user-specific fast weights from contextual preference pairs for personalized reward prediction.

Adapting TTT to personalized reward modeling. However, directly applying TTT to PRMs introduces two key mismatches. First, a *granularity mismatch*: existing TTT methods typically perform update and apply operations at the token level, whereas preference anontations and reward predictions concern complete responses. We address this mismatch through *sequence-level update and apply operations* (*cf.* Section 4.2), performing both operations on the level of complete response sequences. Second, an *objective mismatch*: existing TTT methods often use reconstruction objectives to memorize context or LM-aligned objectives for next-token prediction, neither of which explicitly captures preference relations. We introduce a *preference-aligned objective* (*cf.* Section 4.3) that makes preference relations an explicit learning signal for fast-weight adaptation. We detail these two designs in the following sections.

4.2 Sequence-Level Update and Apply Operations

To resolve the granularity mismatch discussed above, P-TTT both updates and applies the fast weights at the response-sequence level.

User-specific fast weights. We repurpose the existing MLP blocks in selected Transformer layers for test-time adaptation without modifying the backbone architecture, following prior work (Feng et al., 2026). Specifically, given an MLP input $\mathbf{h} \in \mathbb{R}^d$, a gated MLP computes

$$\mathbf{k} = \rho(W_{\text{gate}}\mathbf{h}) \odot (W_{\text{up}}\mathbf{h}), \qquad \mathbf{o} = W_{\text{down}}\mathbf{k}, \tag{5}$$

where $\rho(\cdot)$ denotes the activation function, $\mathbf{k} \in \mathbb{R}^{d_{\text{ff}}}$ is the intermediate activation, and $\mathbf{o} \in \mathbb{R}^d$ is the MLP output. We use the down-projection matrix $W_{\text{down}} \in \mathbb{R}^{d \times d_{\text{ff}}}$ as the fast weight. For each user u, we initialize the fast weights from the shared model parameters W_0 and adapt them using contextual preference pairs $H_u = \{(x_i, y_i^+, y_i^-)\}_{i=1}^m$ to obtain the user-specific fast weights W_u.

Update operation. To align fast-weight adaptation with the response-level nature of preference supervision, we construct each update from complete-response representations rather than token-level. For each contextual preference pair (x_i, y_i^+, y_i^-), we take the MLP intermediate activations at the final tokens of the chosen and rejected responses as their corresponding keys, denoted by $\mathbf{k}_i^+$ and $\mathbf{k}_i^-$, respectively. Under causal attention, each key summarizes the shared prompt together with its corresponding response, providing a sequence-level representation for adaptation. We further apply an attention mask to prevent cross-response attention, so that the chosen and rejected responses are encoded separately.

Given the two response keys, each contextual preference pair induces a single fast-weight update:

$$W_i = W_{i-1} - \eta \, \nabla_W \mathcal{L}_{\text{pref}}\big(W; \mathbf{k}_i^+, \mathbf{k}_i^-, \mathbf{v}^+, \mathbf{v}^-\big)\big|_{W=W_{i-1}}, \tag{6}$$

where η denotes the test-time learning rate, and $\mathbf{v}^+$ and $\mathbf{v}^-$ are preference-related values associated with the chosen and rejected responses, respectively. The update encodes the preference relation into the fast weights by associating $\mathbf{k}_i^+$ with $\mathbf{v}^+$ and $\mathbf{k}_i^-$ with $\mathbf{v}^-$. We define these values and the preference-aligned objective $\mathcal{L}_{\text{pref}}$ in Section 4.3.

With contextual information encoded in the fast weights W_u, we remove them from the input during prediction to reduce the computational cost of attention. Empirically, this removal has a negligible effect on prediction performance (*cf.* Appendix G).

Apply operation. Similarly, we perform the apply operation on complete-response representations. For each target candidate (x, y), we take the MLP intermediate activation at the final token of the response as the target query $\mathbf{k}_{\text{tgt}}$. We then apply the user-specific fast weights W_u at this position:

$$\mathbf{o}_{\text{tgt}} = W_u \mathbf{k}_{\text{tgt}}. \tag{7}$$

The preference relations encoded in W_u thus directly modulate the final-token representation used for response-level scoring. To maintain sequence-level adaptation, fast-weight application is restricted to the final-token position, while all other token continue to use the shared weights W_0.

4.3 Preference-Aligned Objective

To encode preference relations in the fast weights through key–value associations, we first construct a preference-related value from the reward-head weight vector $\mathbf{v}_r$. The reward head is learned to assign higher rewards to chosen responses than to rejected ones. Its weight vector thus captures a direction associated with higher rewards, making it a natural source of preference-related information. Accordingly, we define $\mathbf{v} = W_{\text{value}}\mathbf{v}_r$, where W_{value} is a linear projection, and associate the chosen response's key with $\mathbf{v}^+ = +\mathbf{v}$ and the rejected response's key with $\mathbf{v}^- = -\mathbf{v}$. Specifically, we define the preference-aligned objective as

$$\mathcal{L}_{\text{pref}}\big(W; \mathbf{k}_i^+, \mathbf{k}_i^-, \mathbf{v}^+, \mathbf{v}^-\big) = -\langle W\mathbf{k}_i^+, \mathbf{v}^+\rangle - \langle W\mathbf{k}_i^-, \mathbf{v}^-\rangle = \langle W\mathbf{k}_i^-, \mathbf{v}\rangle - \langle W\mathbf{k}_i^+, \mathbf{v}\rangle. \tag{8}$$

Optimizing this objective encourages the fast-weight output of the chosen response to align more strongly with $\mathbf{v}$ than that of the rejected response, thereby encoding the preference relation between the two responses into the fast weights. The gradient of this objective with respect to the fast weights can be computed efficiently in closed form, yielding the following gradient descent update:

$$W_i = W_{i-1} + \eta \, \mathbf{v}(\mathbf{k}_i^+)^\top - \eta \, \mathbf{v}(\mathbf{k}_i^-)^\top. \tag{9}$$

Training objective. To train the model to use the preference information encoded in W_u, we optimize the model parameters ϕ on target preference pairs using the loss:

$$\mathcal{L}_{\text{train}}(\phi) = -\mathbb{E}_{u,(x,y^+,y^-)}\left[\log \sigma\left(r_\phi(x, y^+; W_u) - r_\phi(x, y^-; W_u)\right)\right], \tag{10}$$

where $r_\phi(x, y; W_u)$ denotes the reward assigned to response y for prompt x using the user-specific fast weights W_u. The two objectives therefore serve complementary roles: $\mathcal{L}_{\text{pref}}$ guides the encoding of contextual preferences into W_u through fast-weight updates, while $\mathcal{L}_{\text{train}}$ trains the model to use these adapted weights to predict preferences on target pairs.

4.4 Theoretical Analysis

Intuitively, P-TTT encodes preference relations by associating each response's key with a value representing the corresponding preference direction. In this section, we provide a theoretical analysis of two questions: how reversing contextual preferences affects model predictions, and whether the objective of P-TTT aligns with the reward modeling objective. Detailed assumptions and proofs are provided in Appendices B and C.

We first establish a direct connection between contextual preference direction and the adaptation-induced change in the target reward margin.

Theorem 1. *Let W_u and W_u^{flip} denote the fast weights adapted from the same initialization W_0 before and after reversing the preference labels in every contextual pair, respectively. For any fixed target response pair (x, y^A, y^B), define the reward margin*

$$M(W) = r_\phi(x, y^A; W) - r_\phi(x, y^B; W). \tag{11}$$

Let $\Delta M(W) = M(W) - M(W_0)$ denote the change in reward margin relative to the initialization. Then reversing all contextual preference labels negates this change:

$$\Delta M(W_u^{\mathrm{flip}}) = -\Delta M(W_u). \tag{12}$$

The proof is given in Appendix B. Theorem 1 shows that reversing contextual preferences induces an equal and opposite change to the target reward margin, demonstrating that P-TTT incorporates the direction of contextual preference relations into its predictions.

We next characterize how the preference information encoded in the fast weights affects the reward of a target response.

Theorem 2. *For a target response (x, y) with query $\mathbf{k}_{\mathrm{tgt}}$, define its reward correction as*

$$\Delta r_\phi(x, y) = r_\phi(x, y; W_u) - r_\phi(x, y; W_0).$$

Under the specified assumptions, with $\eta > 0$ and $\mathbf{v}_r \neq \mathbf{0}$,

$$\begin{aligned} &\Delta r_\phi(x, y) \geq \eta \|\mathbf{v}_r\|_2^2 c_{\mathrm{match}} > 0, \quad \textit{if matched to } \mathbf{k}_{i^*}^+, \\ &\Delta r_\phi(x, y) \leq -\eta \|\mathbf{v}_r\|_2^2 c_{\mathrm{match}} < 0, \quad \textit{if matched to } \mathbf{k}_{i^*}^-. \end{aligned} \tag{13}$$

Here, $i^\star$ indexes the matching contextual response, and $c_{\mathrm{match}} > 0$ denotes the corresponding similarity lower bound.

The proof is provided in Appendix C. Theorem 2 shows that retrieving a contextual response from the chosen side increases the target reward, whereas retrieving one from the rejected side decreases it. This behavior is directly aligned with the reward modeling objective, which assigns higher rewards to chosen responses than to rejected ones. Consequently, when the target responses match contextual keys on their corresponding preference sides, the P-TTT update increases the reward margin between the chosen and rejected target responses.

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate on three benchmarks for personalized reward modeling: **UF-P-2**, **UF-P-4**, and **PersonalLLM**, which have been widely used in prior work (Choi et al., 2025; Nam et al., 2026; Kim & Kim, 2026). UF-P-2 and UF-P-4 construct user preferences along different response quality dimensions (Poddar et al., 2024), while PersonalLLM simulates diverse users through user-specific combinations of pretrained reward models (Zollo et al., 2025). Across all three benchmarks, the task is to predict a user's preference between two target responses given their historical preference pairs. Further dataset details and statistics are provided in Appendix H.

Baselines. We compare against eight baselines grouped into five categories. **(1) No personalization.** BTL (Ouyang et al., 2022) learns a shared reward function without user-specific information. **(2) ICL-based PRMs.** ICL and PLUS (Nam et al., 2026) condition reward prediction on

Table 1: Pairwise preference accuracy (%). We report the mean and standard deviation over 3 seeds. Best results are bold.

Model	Method	UF-P-2	UF-P-4	PersonalLLM
Qwen2.5-0.5B	BTL (NIPS 2022)	49.88 ± 0.12	58.35 ± 0.84	48.83 ± 1.03
	ICL	53.33 ± 1.92	60.46 ± 0.31	51.05 ± 1.05
	PLUS (ICLR 2026)	51.76 ± 0.07	58.86 ± 0.39	50.65 ± 0.48
	GPO (ICLR 2024)	50.52 ± 0.07	55.44 ± 0.64	50.25 ± 0.05
	VPL (NIPS 2024)	50.16 ± 0.07	57.73 ± 1.09	50.07 ± 0.06
	SPL (ICLR 2026)	52.68 ± 1.98	58.80 ± 0.40	50.18 ± 0.03
	MRM (SIGIR 2026)	69.55 ± 0.37	59.40 ± 0.39	58.23 ± 0.29
	In-Place TTT (ICLR 2026)	50.32 ± 0.56	52.73 ± 0.61	50.07 ± 1.15
	P-TTT	$\mathbf{75.40 \pm 0.54}$	$\mathbf{64.00 \pm 0.57}$	$\mathbf{61.27 \pm 0.55}$
Llama3.2-1B	BTL (NIPS 2022)	49.92 ± 0.07	58.47 ± 0.10	49.97 ± 0.06
	ICL	51.88 ± 0.50	60.08 ± 0.82	56.02 ± 3.72
	PLUS (ICLR 2026)	51.36 ± 0.28	58.67 ± 0.37	50.80 ± 1.34
	GPO (ICLR 2024)	50.76 ± 0.42	56.57 ± 0.16	50.10 ± 0.09
	VPL (NIPS 2024)	50.24 ± 0.21	58.57 ± 0.30	50.02 ± 0.03
	SPL (ICLR 2026)	50.60 ± 0.32	59.20 ± 0.33	50.13 ± 0.06
	MRM (SIGIR 2026)	70.71 ± 0.39	62.29 ± 0.68	57.40 ± 0.35
	In-Place TTT (ICLR 2026)	50.28 ± 0.87	52.77 ± 0.97	49.52 ± 0.43
	P-TTT	$\mathbf{74.48 \pm 0.18}$	$\mathbf{64.60 \pm 0.97}$	$\mathbf{62.50 \pm 0.26}$

historical preference pairs or textual summaries of user preferences. **(3) Embedding-based PRMs.** GPO (Zhao et al., 2024), VPL (Poddar et al., 2024), and SPL (Kim & Kim, 2026) encode historical feedback into user-specific representations for preference prediction. **(4) Parameter-based PRMs.** MRM (Cai et al., 2026) fine-tunes user-specific parameters from a meta-learned initialization using historical preference feedback. **(5) Conventional test-time training.** In-Place TTT (Feng et al., 2026) is designed for language modeling using a next-token prediction objective.

Implementation Details. Following prior reward modeling studies (Nam et al., 2026), we use Qwen2.5-0.5B-Instruct (Hui et al., 2024) and Llama3.2-1B-Instruct (Grattafiori et al., 2024) as reward model backbones for all methods. For P-TTT, we perform full-parameter fine-tuning using the AdamW optimizer for 3 epochs, with a learning rate of 5×10^{-6} and a batch size of 128. During test-time training, we use a learning rate of $\eta = 0.5$, with fast weights introduced every 6 layers. To ensure fair comparisons, we utilize the source code provided by the original authors and tune the hyperparameters of all baseline methods according to the guidelines specified in their respective publications. We evaluate performance using pairwise preference accuracy, defined as the proportion of evaluation pairs for which the model assigns a higher reward to the chosen response than to the rejected response, and report the mean accuracy over runs with three random seeds. Our code is available at `https://anonymous.4open.science/r/P-TTT-C7FE`.

5.2 Main Results

As shown in Table 1, P-TTT achieves the highest mean accuracy across all three datasets with both backbones, outperforming the strongest baseline in each setting by an average of 3.94 percentage points. Among ICL-based methods, ICL's weaker performance is consistent with our analysis in Section 3: its difficulty in capturing preference relations limits its ability to infer user interests and generalize effectively. Although PLUS learns a summarizer, compressing feedback into textual summaries may still discard fine-grained preference information. Embedding-based methods also offer limited gains over BTL, potentially due to information loss when encoding complex users' histories into compact representations. MRM benefits from meta-learned adaptation, but constraining adaptation to low-dimensional combinations of shared reward functions may limit its ability to capture complex preferences patterns. Finally, the poor performance of In-Place TTT is consistent with the mismatch between its language modeling aligned objective and the goal of PRMs. In contrast, P-TTT directly aligns adaptation with preference relations, enabling the model to capture useful preference information from user feedback and better generalize to unseen samples.

To further evaluate whether P-TTT captures preference relations, we revisit the preference-flipping experiment in Section 3. As shown in Figure 2(a), P-TTT achieves substantially higher flip rates than ICL across all settings, with an average of 77.81% versus 14.05% for ICL. These results suggest that P-TTT's predictions are more responsive to changes in the preference relations expressed in user feedback. These findings provide evidence that P-TTT more effectively models user-specific preference relations.

5.3 Ablation Studies

In this section, we ablate the two core components of P-TTT using Qwen2.5-0.5B-Instruct as the backbone: *sequence-level update and apply operations* (SLUA) and the *preference-aligned objective* (PAO). For *w/o SLUA*, we perform fast-weight updates at every token of the contextual responses and apply the adapted weights at every token of the target responses. For *w/o PAO*, we replace the preference-aligned update objective with the LM-aligned objective used in In-Place TTT. As shown in Table 2, removing either component reduces accuracy across all three benchmarks. Without SLUA, accuracy drops substantially (e.g. 64.00% to 50.24% on UF-P-4), supporting the importance of matching adaptation granularity to response-level preference feedback. Without PAO, accuracy falls to near-BTL levels across all datasets (e.g. 75.40% to 50.12% on UF-P-2), suggesting that updates under a misaligned objective fail to encode useful user-specific preference information into the fast weights. These results highlight the complementary contributions of P-TTT's two modules and underscore the importance of aligning both the granularity and the objective of adaptation with user-specific preferences.

Table 2: Ablation study. SLUA: sequence-level update and apply operations; PAO: preference-aligned objective. Best results are bold.

Method	UF-P-2	UF-P-4	PersonalLLM
BTL	49.88	58.35	48.83
ICL	53.33	60.46	51.05
w/o SLUA	59.50	50.24	53.65
w/o PAO	50.12	58.19	49.95
P-TTT	**75.40**	**64.00**	**61.27**

5.4 Efficiency Comparison

In this section, we evaluate the inference efficiency of P-TTT against ICL and per-user fine-tuning (*cf.* Appendix F for details), as shown in Figure 4. We measure the total time required to evaluate rewards for a batch of 32 samples. P-TTT is consistently the fastest across both backbones and all three datasets, requiring 2.446–4.694 s per batch and achieving 1.70–1.83× and 6.21–9.72× speedups over ICL and fine-tuning, respectively. Unlike per-user fine-tuning, P-TTT updates fast weights within a single forward pass without inference-time backpropagation. Moreover, encoding contextual information in fast weights allows P-TTT to omit lengthy contextual preference examples from the model input during target prediction, reducing attention computation relative to ICL. These results demonstrate that P-TTT achieves strong predictive performance while maintaining high inference efficiency.

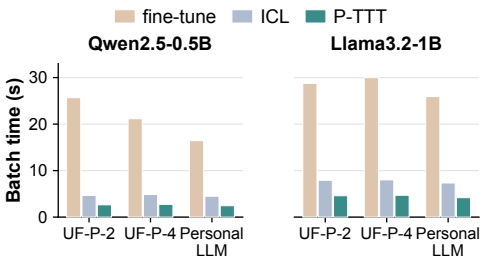

Figure 4: Inference time per batch samples for different methods.

6 Conclusion

In this paper, we identify a key limitation of ICL-based personalized reward models: they underutilize the preference relations expressed in contextual examples. To address this limitation, we propose Preference-Aligned Test-Time Training (P-TTT), which explicitly encodes these relations into user-specific fast weights through sequence-level update and apply operations and a preference-aligned objective. Empirical results demonstrate that P-TTT outperforms existing methods while reducing inference latency, highlighting its effectiveness and efficiency for personalized reward modeling. Future work could explore continual personalization, enabling LLMs to adapt to evolving user preferences while selectively retaining and updating long-term user information.

AI DISCLOSURE

Generative AI tools were used to assist with literature retrieval and discovery, polishing the writing, and drafting portions of the manuscript. All AI-assisted content was carefully reviewed and revised by the authors, and relevant references were verified against their original sources. The authors take full responsibility for the accuracy, integrity, and final content of the manuscript.

ETHICS STATEMENT

This work uses publicly available datasets and open-weight models and does not collect private or sensitive personal information. Personalized reward models may reinforce biases or harmful preferences, and adapted weights may retain sensitive information. Real-world deployment therefore requires privacy safeguards, fairness evaluation, and independent safety constraints.

REPRODUCIBILITY STATEMENT

We describe our method in Section 4 and provide implementation details and evaluation settings in Section 5.1. Dataset construction and statistics are detailed in Appendix H, and theoretical assumptions and proofs are provided in Appendices B and C. Our code is available at `https://anonymous.4open.science/r/P-TTT-C7FE`.

REFERENCES

Rachit Bansal, Aston Zhang, Rishabh Tiwari, Lovish Madaan, Venkata Sai Surya Subramanyam Duvvuri, Devvrit Khatri, David Brandfonbrener, David Alvarez-Melis, Prajjwal Bhargava, Mihir Kale, et al. Let's (not) just put things in context: Test-time training for long-context llms. In *International Conference on Learning Representations*, volume 2026, pp. 112130–112153, 2026.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *Advances in Neural Information Processing Systems*, 38:113506–113543, 2026.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Hongru Cai, Yongqi Li, Tiezheng Yu, Fengbin Zhu, Wenjie Wang, Fuli Feng, and Wenjie Li. One adapts to any: Meta reward modeling for personalized llm alignment. *arXiv preprint arXiv:2601.18731*, 2026.

Youngbin Choi, Seunghyuk Cho, Minjong Lee, MoonJeong Park, Yesong Ko, Jungseul Ok, and Dongwoo Kim. Copl: Collaborative preference learning for personalizing llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12886–12904, 2025.

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

Guhao Feng, Shengjie Luo, Kai Hua, Ge Zhang, Wenhao Huang, Di He, and Tianle Cai. In-place test-time training. In *International Conference on Learning Representations*, volume 2026, pp. 114452–114472, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jian Guan, Junfei Wu, Jia-Nan Li, Chuanqi Cheng, and Wei Wu. A survey on personalized alignment—the missing piece for large language models in real-world applications. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5313–5333, 2025.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Xisen Jin, Zheng Li, Zhenwei Dai, Hui Liu, Xianfeng Tang, Chen Luo, Rahul Goutam, Xiang Ren, and Qi He. In-context personalized alignment with feedback history under counterfactual evaluation. In *2nd Workshop on Models of Human Feedback for AI Alignment*.

Gihoon Kim and Euntai Kim. Swap-guided preference learning for personalized reinforcement learning from human feedback. *arXiv preprint arXiv:2603.12595*, 2026.

Seongyoon Kim, Boryeong Cho, Jihwan Oh, Seokhyun Chung, and Se-Young Yun. Rethinking personalized reward modeling for llms under preference heterogeneity via group-debiased federated learning. *arXiv preprint arXiv:2608.01556*, 2026.

Junchen Liu, Sven Elflein, Or Litany, Zan Gojcic, and Ruilong Li. Test-time training with kv binding is secretly linear attention. *arXiv preprint arXiv:2602.21204*, 2026.

Renpu Liu, Peng Wang, Donghao Li, Cong Shen, and Jing Yang. A shared low-rank adaptation approach to personalized rlhf. *arXiv preprint arXiv:2503.19201*, 2025.

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. In *International Conference on Learning Representations*, volume 2026, pp. 122761–122784, 2026.

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

Xuan Ouyang, Zefan Cai, and Junjie Hu. Test-time training with next-token prediction. *arXiv preprint arXiv:2606.21803*, 2026.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning. *Advances in Neural Information Processing Systems*, 37:52516–52544, 2024.

Michael J Ryan, Omar Shaikh, Aditri Bhagirath, Daniel Frees, William Held, and Diyi Yang. Synthesizeme! inducing persona-guided prompts for personalized reward models in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8045–8078, 2025.

Idan Shenfeld, Felix Faltings, Pulkit Agrawal, and Aldo Pacchiano. Language model personalization via reward factorization. *arXiv preprint arXiv:2503.06358*, 2025.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Arnuv Tandon, Karan Dalal, Xinhao Li, Daniel Koceja, Marcel Rød, Sam Buchanan, Xiaolong Wang, Jure Leskovec, Sanmi Koyejo, Tatsunori Hashimoto, et al. End-to-end test-time training for long context, 2025. *URL https://arxiv. org/abs/2512.23675*.

Changshuo Zhang, Xiao Zhang, Teng Shi, Jun Xu, and Ji-Rong Wen. Test-time alignment for tracking user interest shifts in sequential recommendation. *arXiv preprint arXiv:2504.01489*, 2025.

Zhehao Zhang, Ryan A Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, et al. Personalization of large language models: A survey. *arXiv preprint arXiv:2411.00027*, 2024.

Siyan Zhao, John Dang, and Aditya Grover. Group preference optimization: Few-shot alignment of large language models. In *International conference on learning representations*, volume 2024, pp. 57965–57987, 2024.

Thomas Zollo, Andrew Siah, Naimeng Ye, Li Li, and Hongseok Namkoong. Personalllm: Tailoring llms to individual preferences. In *International Conference on Learning Representations*, volume 2025, pp. 66949–66971, 2025.

A Related Work

Personalized Reward Model Personalized reward models (PRMs) capture heterogeneous user preferences rather than learning a single shared reward function (Poddar et al., 2024; Shenfeld et al., 2025). Existing approaches broadly fall into three categories: **(1) Embedding-based methods** encode historical feedback into compact user representations for reward prediction (Poddar et al., 2024; Zhao et al., 2024; Kim & Kim, 2026). However, compressing complex preference patterns can obscure fine-grained information, while LLM-based reward models pretrained on natural language may struggle to utilize these embeddings effectively (Nam et al., 2026). **(2) Parameter-based methods** adapt model parameters using individual feedback (Kim et al., 2026; Liu et al., 2025). Full-model adaptation incurs substantial inference-time overhead due to per-user backpropagation. Parameter-efficient approaches reduce this cost by representing preferences in a low-dimensional space and optimizing a small set of user-specific parameters (Shenfeld et al., 2025; Cai et al., 2026). However, this restriction may limit expressiveness. Moreover, when user feedback is scarce, these methods are prone to overfitting the observed feedback, limiting generalization to unseen inputs. **(3) ICL-based methods** provide historical preference pairs as textual context (Jin et al.; Ryan et al., 2025; Zollo et al., 2025), with related approaches learning compact, interpretable user summaries (Nam et al., 2026). These methods rely on the model to infer how historical comparisons should influence new judgments, while lengthy histories increase processing costs. Prior work reports limited performance of ICL-based methods and broadly attributes this limitation to models' difficulty in leveraging lengthy contexts (Nam et al., 2026). Our work focuses on this ICL-based PRM setting and provides a more fine-grained analysis: trained models can leverage contextual content while making limited use of the preference relations between chosen and rejected responses in historical pairs.

Test Time Training Test-time training (TTT) enables models to adapt to context through inference-time parameter updates. TTT layers and neural memory architectures incorporate contextual information into fast weights through self-supervised updates (Sun et al., 2024; Behrouz et al., 2026; Liu et al., 2026). Recent work further aligns adaptation with language modeling: TTT-E2E optimizes next-token prediction over the observed context and meta-learns an initialization for adaptation (Tandon et al.), while In-Place TTT repurposes existing MLP down-projections as fast weights with an LM-aligned objective and efficient chunk-wise updates (Feng et al., 2026). TTT-NTP uses next-position contextual hidden states to supervise local fast-weight updates in pretrained LLMs, connecting adaptation more directly to next-token prediction (Ouyang et al., 2026). Context-specific gradient updates have also been shown to improve long-context performance by addressing limitations of static self-attention (Bansal et al., 2026). Beyond language modeling, test-time alignment has been explored in sequential recommendation to track user interest shifts through incremental self-supervised updates (Zhang et al., 2025). These studies motivate adaptation objectives tailored to the target task. Our work focuses on personalized reward modeling, where adaptation must capture comparative preferences over complete responses (Poddar et al., 2024; Nam et al., 2026), and uses historical chosen–rejected pairs to guide fast-weight updates.

B Proof of Theorem 1

For analytical tractability, following Feng et al. (2026), we focus our analysis on a single Transformer block equipped with P-TTT.

Theorem 1. *Let W_u and W_u^{flip} denote the fast weights adapted from the same initialization W_0 before and after reversing the preference labels in every contextual pair, respectively. For any fixed target response pair (x, y^A, y^B), define the reward margin*

$$M(W) = r_\phi(x, y^A; W) - r_\phi(x, y^B; W). \tag{14}$$

Let $\Delta M(W) = M(W) - M(W_0)$ denote the change in reward margin relative to the initialization. Then reversing all contextual preference labels negates this change:

$$\Delta M(W_u^{\text{flip}}) = -\Delta M(W_u). \tag{15}$$

Proof. Let $\mathbf{k}(x, y)$ denote the MLP intermediate activation at the final token of response y to prompt x. For contextual pair i, let $\mathbf{k}_i^{\pm} = \mathbf{k}(x_i, y_i^{\pm})$ and define $\mathbf{d}_i = \mathbf{k}_i^{+} - \mathbf{k}_i^{-}$. The preference-aligned

objective satisfies

$$\mathcal{L}_{\text{pref}}(W; \mathbf{k}_i^+, \mathbf{k}_i^-, \mathbf{v}^+, \mathbf{v}^-) = -\mathbf{v}^\top W \mathbf{d}_i, \qquad \nabla_W \mathcal{L}_{\text{pref}} = -\mathbf{v}\mathbf{d}_i^\top.$$

Thus, accumulating the m updates yields

$$W_u - W_0 = \sum_{i=1}^{m} (W_i - W_{i-1}) = \eta \mathbf{v} \sum_{i=1}^{m} \mathbf{d}_i^\top. \tag{16}$$

Reversing all contextual preference labels replaces each $\mathbf{d}_i$ by $-\mathbf{d}_i$. Consequently,

$$W_u^{\text{flip}} - W_0 = -\eta \mathbf{v} \sum_{i=1}^{m} \mathbf{d}_i^\top = -(W_u - W_0).$$

For the fixed target pair, let

$$\mathbf{k}_{\text{tgt}}^A = \mathbf{k}(x, y^A), \qquad \mathbf{k}_{\text{tgt}}^B = \mathbf{k}(x, y^B), \qquad \Delta\mathbf{k}_{\text{tgt}} = \mathbf{k}_{\text{tgt}}^A - \mathbf{k}_{\text{tgt}}^B.$$

The reward model gives

$$r_\phi(x, y; W) - r_\phi(x, y; W_0) = \mathbf{v}_r^\top (W - W_0)\mathbf{k}_{\text{tgt}}. \tag{17}$$

It follows that

$$\Delta M(W) = \mathbf{v}_r^\top (W - W_0)\Delta\mathbf{k}_{\text{tgt}}.$$

Since $\Delta\mathbf{k}_{\text{tgt}}$ is unchanged under contextual label reversal,

$$\begin{aligned} \Delta M(W_u^{\text{flip}}) &= \mathbf{v}_r^\top (W_u^{\text{flip}} - W_0)\Delta\mathbf{k}_{\text{tgt}} \\ &= -\mathbf{v}_r^\top (W_u - W_0)\Delta\mathbf{k}_{\text{tgt}} \\ &= -\Delta M(W_u). \end{aligned}$$

□

C Proof of Theorem 2

Theorem 2. *For a target response (x, y) with query $\mathbf{k}_{\text{tgt}}$, define its reward correction as*

$$\Delta r_\phi(x, y) = r_\phi(x, y; W_u) - r_\phi(x, y; W_0).$$

Under the specified assumptions, with $\eta > 0$ and $\mathbf{v}_r \neq \mathbf{0}$,

$$\begin{aligned} &\Delta r_\phi(x, y) \geq \eta \|\mathbf{v}_r\|_2^2 c_{\text{match}} > 0, \quad \textit{if matched to } \mathbf{k}_{i^*}^+, \\ &\Delta r_\phi(x, y) \leq -\eta \|\mathbf{v}_r\|_2^2 c_{\text{match}} < 0, \quad \textit{if matched to } \mathbf{k}_{i^*}^-. \end{aligned} \tag{18}$$

Here, i^* indexes the matching contextual response, and $c_{\text{match}} > 0$ denotes the corresponding similarity lower bound.

Assumption. We assume that W_{value} is the identity transformation, so that $\mathbf{v} = \mathbf{v}_r$. We further make the following two assumptions.

(i) Positive similarity to a matching key. For the key of target response $\mathbf{k}_{\text{tgt}}$, there exist an index $i^* \in \{1, \ldots, m\}$, a label $s \in \{+, -\}$, and a constant $c_{\text{match}} > 0$ such that

$$\langle \mathbf{k}_{i^*}^s, \mathbf{k}_{\text{tgt}} \rangle \geq c_{\text{match}}. \tag{19}$$

(ii) Zero aggregate contribution from the remaining keys. The remaining entries satisfy

$$\sum_{\substack{1 \leq i \leq m,\ t \in \{+,-\} \\ (i,t) \neq (i^*, s)}} \langle \mathbf{k}_i^t, \mathbf{k}_{\text{tgt}} \rangle \mathbf{v}^t = \mathbf{0}. \tag{20}$$

Proof. Equation 9 implies

$$W_u - W_0 = \sum_{i=1}^{m} (W_i - W_{i-1}) = \eta \sum_{i=1}^{m} \sum_{t \in \{+,-\}} \mathbf{v}^t (\mathbf{k}_i^t)^\top.$$

Consequently,

$$\Delta \mathbf{o}_{\mathrm{tgt}} := (W_u - W_0)\mathbf{k}_{\mathrm{tgt}} = \eta \sum_{i=1}^{m} \sum_{t\in\{+,-\}} \langle \mathbf{k}_i^t, \mathbf{k}_{\mathrm{tgt}} \rangle \mathbf{v}^t. \tag{21}$$

By Equation 20,

$$\Delta \mathbf{o}_{\mathrm{tgt}} = \eta \mathbf{v}^s \langle \mathbf{k}_{i^*}^s, \mathbf{k}_{\mathrm{tgt}} \rangle.$$

Then

$$\Delta r_\phi(x, y) = \mathbf{v}_r^\top \Delta \mathbf{o}_{\mathrm{tgt}} = \eta (\mathbf{v}_r^\top \mathbf{v}^s) \langle \mathbf{k}_{i^*}^s, \mathbf{k}_{\mathrm{tgt}} \rangle. \tag{22}$$

Using $\mathbf{v}^+ = \mathbf{v}_r$ and $\mathbf{v}^- = -\mathbf{v}_r$, Equations 19 and 22 give

$$\begin{aligned} \Delta r_\phi(x, y) &\geq \eta \|\mathbf{v}_r\|_2^2 c_{\mathrm{match}} > 0, \qquad s = +, \\ \Delta r_\phi(x, y) &\leq -\eta \|\mathbf{v}_r\|_2^2 c_{\mathrm{match}} < 0, \qquad s = -. \end{aligned}$$

□

D PREFERENCE-FLIPPING ANALYSIS

This section provides experimental details for the preference-flipping analysis in Section 3 and Figure 2(a). We conduct a counterfactual experiment to examine whether the model changes its prediction when contextual preference relations are reversed while the response content is kept unchanged.

Label reversal. For an evaluation sample j, let H_j denote its contextual preference examples, and let (x_j, y_j^A, y_j^B) denote the target prompt and its two candidate responses. We construct a counterfactual context $\widetilde{H}_j^{(v)}$ by swapping the chosen and rejected labels within selected contextual pairs, without necessarily reversing all pairs. The contextual prompts and response content, as well as the target prompt and both candidate responses, remain unchanged.

To ensure that the contextual label changes imply a reversal of the target preference, we validate each variant using the dataset-defined preference rules. We first verify that the original history implies an unambiguous target preference. We then identify all personas or generated users consistent with the modified history. A variant is retained only if this set is nonempty and every compatible user prefers the opposite target response. All other variants are excluded. Let $\mathcal{V}_j$ denote the resulting set of valid label-reversal variants for sample j.

Flip rate. Let $z_j \in \{0, 1\}$ denote the original target preference label, with $z_j = 1$ indicating that y_j^A is preferred to y_j^B and $z_j = 0$ indicating the reverse. Let $\hat{z}_j$ be the model's prediction under H_j, and $\hat{z}_j^{(v)}$ its prediction under $\widetilde{H}_j^{(v)}$. For each valid variant, the target preference label is $1 - z_j$. We evaluate samples that the model initially predicts correctly and that have at least one valid label-reversal variant:

$$\mathcal{C} = \{j : \hat{z}_j = z_j,\ |\mathcal{V}_j| > 0\}. \tag{23}$$

For each $j \in \mathcal{C}$, we compute its flip rate as the fraction of valid variants for which the prediction changes to the other candidate:

$$\text{Flip Rate}_j = \frac{1}{|\mathcal{V}_j|} \sum_{v\in\mathcal{V}_j} \mathbf{1}\left[\hat{z}_j^{(v)} = 1 - z_j\right], \tag{24}$$

where $\mathbf{1}[\cdot]$ is the indicator function. Since the original prediction is correct and each valid variant reverses the target preference, this prediction change agrees with the reversed target preference label. We then average the per-sample flip rates equally:

$$\text{Flip Rate} = \frac{1}{|\mathcal{C}|} \sum_{j\in\mathcal{C}} \text{Flip Rate}_j. \tag{25}$$

Each initially correct sample with at least one valid variant receives equal weight, regardless of its number of valid variants. A higher flip rate indicates that contextual preference relations have a greater influence on the model's predictions.

E Limitations of Prompt-Based Strategies

We investigate whether explicit instructions can improve the model's use of contextual preference relations, evaluating Qwen2.5-0.5B-Instruct on UF-P-2 and UF-P-4. Specifically, we prepend an instruction emphasizing the preference labels to the contextual preference examples during both training and evaluation:

> Pay explicit attention to the preference labels in every historical example. 'Chosen response' is the response this user preferred; 'Rejected response' is the response this user liked less. These labels, not the order in which responses appear, specify the user's preference. Compare the chosen and rejected responses to infer which traits the user favors and disfavors. When scoring the response to the current request, reward alignment with the labeled chosen responses and penalize alignment with the labeled rejected responses. Treat the labels shown in the examples as the evidence of this user's preferences.

As shown in Table 3, explicitly emphasizing preference labels does not yield consistent improvements. On UF-P-2, the instruction modestly increases the flip rate from 6.83% to 8.98% and test accuracy from 53.33% to 53.97%. On UF-P-4, however, the flip rate decreases from 20.82% to 8.17%, while test accuracy drops from 60.46% to 58.80%. These results suggest that, in the settings evaluated, explicit instructions alone are insufficient to reliably improve the model's use of contextual preference relations.

Table 3: Effect of explicit preference-label instructions on flip rate and test accuracy (%) using Qwen2.5-0.5B-Instruct.

Method	Metric	UF-P-2	UF-P-4
ICL	Flip rate	6.83	20.82
	Test accuracy	53.33	60.46
ICL + instruction	Flip rate	8.98	8.17
	Test accuracy	53.97	58.80

F Details of Per-User Fine-Tuning

In this section, we describe the experimental setup for per-user fine-tuning. We fine-tune the reward model using the m historical preference pairs associated with each target sample. Let $H_u = \{(x_i, y_i^+, y_i^-)\}_{i=1}^m$ denote these preference pairs. The tuning objective is

$$\mathcal{L}_{\mathrm{FT}}(\phi; H_u) = -\frac{1}{m}\sum_{i=1}^{m} \log \sigma\big(r_\phi(x_i, y_i^+) - r_\phi(x_i, y_i^-)\big)\,. \tag{26}$$

The target pair and its preference label are excluded from adaptation. After fine-tuning, the model scores the target responses without contextual examples in the input. We reset the model parameters and optimizer state before adapting to each target sample, including samples from the same user. We evaluate fine-tuning for 3 optimization steps for each sample.

G Retaining Context in Self-Attention

In this section, we examine whether P-TTT benefits from retaining historical context in the input during target prediction, to assess whether the adapted fast weights effectively retain contextual preference information. The variant *P-TTT w/ context* retains contextual tokens so that target responses can attend to them through self-attention, in addition to using the adapted fast weights. As shown in Table 4, retaining context has little effect on prediction accuracy. These results suggest that P-TTT effectively encodes and retains relevant contextual preference information in its fast weights, supporting its design of omitting contextual tokens during target prediction.

Table 4: Accuracy (%) with context accessible to self-attention.

Method	UF-P-2	UF-P-4	PersonalLLM
P-TTT	75.40	64.00	61.27
P-TTT w/ context	75.84	64.06	61.10

H Dataset Overview and Statistics

We evaluate on three personalized preference datasets: UF-P-2, UF-P-4, and PersonalLLM. For each dataset, we reserve 100 preference pairs as a shared survey pool and exclude them from the training targets. For each user–target sample, we sample contextual pairs from this pool and label them according to the user's preferences. The experiments use $m = 4$ contextual pairs. Table 5 summarizes their statistics.

Table 5: Processed dataset statistics.

Dataset	Split	Users	Samples	m
UF-P-2	Train	2	7,516	4
	Test	2	832	4
UF-P-4	Train	4	15,740	4
	Test	4	1,660	4
PersonalLLM	Train	100	10,000	4
	Test	120	2,000	4

UF-P-2 and UF-P-4. Following UF-P (Poddar et al., 2024), we construct user preferences from UltraFeedback ratings along different response quality dimensions. UF-P-2 includes two user types that prefer *helpfulness* and *honesty*, respectively; UF-P-4 adds two types that prefer *instruction following* and *truthfulness*. Each user prefers the response with the higher rating on their designated dimension. Training and testing share the same user types but use disjoint target prompts.

PersonalLLM. PersonalLLM (Zollo et al., 2025) simulates diverse users through user-specific weighted combinations of ten reward models, with pairwise preferences determined by the resulting scores. We use 100 users for training and evaluate on both these users and 20 unseen users. The test set contains 1,000 samples from each group, and we report accuracy over the combined 2,000 samples.