The Reward Model Selection Crisis in Personalized Alignment

Anonymous ACL submission

Abstract

Personalized alignment from preference data has focused primarily on improving personal reward model (RM) accuracy, with the implicit assumption that better preference ranking translates to better personalized behavior. However, in deployment, computational constraints necessitate inference-time adaptation such as reward-guided decoding (RGD) rather than per-user policy fine-tuning. This creates a critical but overlooked requirement: reward models must not only rank preferences accurately but also effectively guide generation. We demonstrate that standard RM accuracy fails catastrophically as a selection criterion for deployment-ready personalized rewards. We introduce *policy accuracy*—a metric quantifying whether RGD-adapted LLMs correctly discriminate between preferred and dispreferred responses—and show that upstream RM accuracy correlates only weakly with downstream policy accuracy (Kendall's $\tau = 0.08$–0.31). More critically, we introduce PREF-LAMP, the first personalized alignment benchmark with ground-truth user completions, enabling direct behavioural evaluation. On PREF-LAMP, we expose a complete decoupling between discriminative ranking and generation metrics: methods with 20-point RM accuracy differences produce almost identical output quality, and methods with high ranking accuracy can fail to generate behaviorally aligned responses. These findings reveal that the field has been optimizing for proxy metrics that do not predict deployment performance, and that current personalized alignment methods fail to operationalize preferences into behavioral adaptation under realistic deployment constraints. In contrast, we find simple in-context learning (ICL) to be highly effective - dominating all reward-guided methods for models $\geq$3B parameters, achieving $\sim$3 point ROUGE-1 gains over the best reward method at 7B scale.

1 Introduction

Recent advances in aligning large language models (LLMs) with human preferences have primarily focused on learning from aggregated feedback across diverse user populations (Rafailov et al., 2023; Ouyang et al., 2022). However, preferences are inherently pluralistic—varying across individuals, communities, and contexts (Santurkar et al., 2023; Sorensen et al., 2024). This reality motivates *personalized alignment*: adapting model behavior to heterogeneous, sometimes conflicting, user preferences rather than collapsing them into a single consensus objective.

Current personalized alignment research has converged on a common paradigm: collect user-specific preference data (pairwise comparisons), train personalized *ranking*/reward models to capture individual preferences, and assume that better reward models naturally translates to better policies (Bose et al., 2025; Chen et al., 2025a; Shenfeld et al., 2025; Li et al., 2024; Poddar et al., 2024). The last assumption is likely to break as suggested by Goodhart's law (El-Mhamdi and Hoang, 2024). Unlike standard RLHF, personalized alignment lacks downstream benchmarks that measures policy performance such as MMLU (Hendrycks et al., 2021) and GSM8k (Cobbe et al., 2021).

Practical Deployment and the End-to-End Perspective Per-user policy fine-tuning using personal rewards via RL is computationally infeasible at scale. RL-based personalization requires per-user dynamic adapter management, RL instability mitigation, and orders of magnitude more compute than inference-time alternatives. One key scalable deployment path is *inference-time adaptation* through reward-guided decoding (RGD) (Khanov et al., 2024), maintaining a single base policy while using personalized rewards to guide generation. Another option is or Best-of-N sampling (Ichihara et al., 2025) but the high latency of BoN makes it

unfit for personalization text generation.

This deployment reality demands we adopt an *end-to-end behavioral perspective*: personalized alignment is not merely reward modeling, but the complete process from preference data to actual generation behavior. We propose a key principle:

> *A personalized alignment method must specify not only how preferences are modeled, but how they are operationalized into behavioral adaptation.*

A direct corollary of this is that papers proposing reward models are responsible for evaluating if improved RM accuracy translates to improved generation. Current evaluations ignore this responsibility, treating reward modeling and policy adaptation as independent. This obscures whether methods actually achieve their objective: making models generate responses aligned with user preferences.

Our Investigation We adopt an end-to-end perspective, studying the complete chain from preference modeling to generation behavior. We ask three questions: (1) Does RM accuracy predict policy accuracy under RGD? (2) Does policy accuracy predict generation quality? (3) How do reward-based alignment methods compare to simpler baselines? To answer these, we introduce (A) policy accuracy, measuring whether RGD scoring assigns higher scores to preferred responses, and (B) PREF-LAMP, a preference learning benchmark with ground-truth user completions enabling direct behavioral evaluation.

Key Findings Our findings reveal a fundamental selection crisis: practitioners cannot reliably choose deployment-ready methods because standard metrics do not predict actual performance.

Finding 1: Upstream RM accuracy does not predict downstream policy accuracy under RGD (Kendall's $\tau = 0.08$–0.31). Methods with 20-point RM accuracy differences achieve nearly identical policy performance.

Finding 2: Response *ranking* quality does not predict response *generation* quality. On PREF-LAMP, methods with similar generation quality vary dramatically in RM and policy accuracy.

Finding 3: ICL dominates at scale. At 7B parameters, ICL-RAG, with RAG selected preference demonstrations (Salemi et al., 2024), outperforms best personalized reward model by ~3 ROUGE-1 points.

Implications: For practitioners, use simple ICL-RAG in preference to published personal reward methods. For researchers, take an end-to-end perspective: co-design and co-evaluate reward personalization with policy adaptation strategies and evaluate generation quality as well as ranking. Use PREF-LAMP and develop more benchmarks with ground-truth completions, analogous to GSM8K/MMLU for general RLHF.

Contributions To summarize, we (1) contribute PREF-LAMP—the first benchmark with ground-truth user completions, (2) demonstrate the standard RM accuracy metric fails as a selection criterion across three datasets and four scales, (3) demonstrate that a simple ICL baseline outperforms published personal alignment work in end-to-end adaptation, and (4) provide actionable recommendations for practitioners and researchers.

2 Related Work

Personalized Alignment Recent work has focused on learning user-specific reward models or policies for alignment under limited supervision (e.g., PAL, PReF, LoRE, P-DPO, VPL) (Chen et al., 2025a; Poddar et al., 2024; Bose et al., 2025; Shenfeld et al., 2025; Li et al., 2024). These approaches largely target reward-modeling accuracy (e.g relative preference ranking) as proxies for personalization quality (Chen et al., 2025a; Bose et al., 2025; Shenfeld et al., 2025). However, such metrics often fail to capture (i) whether RM adaptation translates to downstream policy adaptation, and (ii) whether, under realistic resource-constrained settings, a personalized policy is able to go beyond response ranking and actually *generate* responses reflective of a user's preferences. This evaluation limitation leaves open the null hypothesis that prior personal alignment results, measured by RM accuracy, are due to unintended overoptimization, also known as reward-hacking (Pan et al., 2022).

Multi-Objective Alignment Multi-objective alignment (MOA) addresses the challenge of optimizing language models across multiple known and predefined reward dimensions simultaneously. Unlike personalized alignment, where the goal is to learn individual user preferences under limited supervision, MOA assumes access to distinct reward models for each objective dimension (e.g., helpfulness, harmlessness, factuality) and focuses on finding optimal policy trade-offs among these objec-

tives. Prior work has explored weighted reward optimization (Zhou et al., 2024), model merging (Jang et al., 2024; Rame et al., 2023), auxiliary correction models (Ji et al., 2024; Yang et al., 2024a), test-time reward-guided decoding (Chen et al., 2025b) among other methods (Yang et al., 2024b).

Evaluation Challenges in RLHF Existing evaluation practices in RLHF and personalized alignment rely on proxy metrics such as reward-model scores which are susceptible to reward hacking and circularity (Tien et al., 2023; Pan et al., 2022). These methods assess optimization success rather than behavioral quality (Wen et al., 2025; Gao et al., 2023; El-Mhamdi and Hoang, 2024). In contrast, our work introduces a framework for *direct behavioral evaluation*, measuring whether generated responses match user-provided completions (Section 5). Extended discussion of related evaluation pathologies appears in Appendix A. Our work addresses these limitations by introducing direct behavioral evaluation on ground-truth user completions, measuring actual generation quality rather than relying on proxy metrics.

Inference-Time Alignment Reward-guided decoding (Khanov et al., 2024) and Best-of-N sampling (Ichihara et al., 2025) enable policy steering without fine-tuning, making them computationally attractive for personalization. Recent work has explored their effectiveness (Wu, 2025), but standard personal alignment evaluation remains limited to reward-based metrics. Our work is the first to systematically evaluate test-time alignment (reward-guided decoding in particular) for personalized alignment with ground-truth behavioral assessment, revealing fundamental limitations in their ability to operationalize user preferences.

3 Preliminaries and Problem Setup

3.1 Personalized Preference Learning

Consider a preference dataset $\mathcal{D} = \{(u_i, x_i, y_i^{(w)}, y_i^{(l)})\}_{i=1}^{N}$, where $u_i \in \{1..K\}$ is a user identifier, x_i is a prompt, and $y_i^{(w)}, y_i^{(l)}$ are chosen/winning and rejected/loosing completions. We partition users into $\mathcal{U}_{\text{train}}$ (for learning shared preference structure) and $\mathcal{U}_{\text{adapt}}$ (for evaluating few-shot personalization). Users in $\mathcal{U}_{\text{adapt}}$ are further split into support sets $\mathcal{D}_k^{\text{support}}$ (for adaptation) and query sets $\mathcal{D}_k^{\text{query}}$ (for evaluation). For user k, we denote their full dataset as $\mathcal{D}_k = \{(x_i, y_i^{(w)}, y_i^{(l)}) : u_i = k\}$.

Personalized Reward Modeling. The reward model conditions on user identity: $r_{\theta,z_k}(y \mid x)$, decomposing into shared parameters θ (general preference structure) and user-specific parameters z_k (individual preferences). Training on $\mathcal{U}_{\text{train}}$ learns both θ and $\{z_k\}_{k\in\mathcal{U}_{\text{train}}}$. At deployment, for user $k \in \mathcal{U}_{\text{adapt}}$ with support data $\mathcal{D}_k^{\text{support}}$, we adapt:

$$z_k = \mathcal{A}\big(\mathcal{D}_k^{\text{support}}; \theta\big)$$

where $\mathcal{A}$ is the adaptation algorithm.

3.2 Deployment via Reward-Guided Decoding

Given computational constraints prohibiting per-user policy fine-tuning, we deploy personalized alignment through inference-time guidance using Reward-Guided Decoding (Khanov et al., 2024).

Reward-Guided Decoding (ARGS). At each generation step t, ARGS scores the top-k tokens $V_t^{(k)}$ retrieved from an off-the-shelf prior LLM policy, π, using

$$\text{score}(v \mid x, y_{<t}; z_k, \lambda) = \log \pi(v \mid x, y_{<t}) + \lambda \cdot r_{\theta,z_k}(v \mid x, y_{<t}), \tag{1}$$

selecting $y_t = \arg\max_{v\in V_t^{(k)}} \text{score}(v)$. This balances base model fluency ($\log \pi$) with personalized alignment (r_{θ,z_k}).

3.3 The Evaluation Gap

Standard practice evaluates personalization methods by reward model accuracy, defined below.

Definition 1 (Reward Model Ranking Accuracy)**.** *For user k with evaluation set $\mathcal{D}_k^{query} = \{(x_i, y_i^{(w)}, y_i^{(l)})\}$, we define the Reward Model Ranking Accuracy as*

$$\frac{1}{|\mathcal{D}_k^{eval}|} \sum_{\mathcal{D}_k^{eval}} \mathbb{I}\Big[r_{\theta,z_k}(y_i^{(w)} \mid x_i) > r_{\theta,z_k}(y_i^{(l)} \mid x_i)\Big]. \tag{2}$$

This measures pairwise ranking on complete responses of the reward model. Later we will show that this standard metric has several issues and is not predictive of deployment performance.

4 Policy Accuracy: Measuring Preference Ranking Under RGD

We introduce a metric quantifying whether the RGD scoring function—not just the reward model in isolation—correctly ranks preferred over dispreferred responses.

Definition 2 (Policy Ranking Accuracy)**.** *Let $s : \mathcal{Y} \times \mathcal{X} \to \mathbb{R}$ be the scoring function used at generation time. The policy accuracy for user k is given by*

$$\frac{1}{|\mathcal{D}_k^{query}|} \sum_{\mathcal{D}_k^{query}} \mathbb{I}\Big[s(y_i^{(w)} \mid x_i) > s(y_i^{(l)} \mid x_i)\Big], \quad (3)$$

where $y^{(w)}$ and $y^{(l)}$ denote the chosen (winning) and rejected (losing) completions.

We instantiate s with three scoring functions, each revealing different aspects of the personalization pipeline.

Base Policy. The base policy's length-normalized log-likelihood, its off-the-shelf non-personalized zero-shot ranking ability,

$$s_{\text{base}}(y \mid x) = \frac{1}{|y|} \sum_{t=1}^{|y|} \log \pi(y_t \mid x, y_{<t}). \quad (4)$$

Global RGD. RGD with a non-personalized reward model r_θ trained by aggregating data across all users denoted $s_{\text{global}}(y \mid x)$,

$$\sum_{t=1}^{|y|} \Big[\log \pi(y_t \mid x, y_{<t}) + \lambda \cdot r_\theta(y_t \mid x, y_{<t}) \Big]. \quad (5)$$

Personalized RGD. RGD with personalized reward model r_{θ,z_k} denoted $s_{\text{personal}}(y \mid x; z_k)$,

$$\sum_{t=1}^{|y|} \Big[\log \pi(y_t \mid x, y_{<t}) + \lambda \cdot r_{\theta,z_k}(y_t \mid x, y_{<t}) \Big]. \quad (6)$$

Comparing these reveals: (1) whether reward guidance improves ranking over the base policy, and (2) whether personalization provides gains beyond a global reward model.

In-Context Learning. An alternative personalization mechanism conditions the base policy on user-specific demonstrations $\mathcal{D}_k^{\text{demo}} = \{(x_i, y_i^{(w)}, y_i^{(l)}\} \subset \mathcal{D}_k^{\text{support}}$ rather than learning reward parameters. The ICL scoring function, denoted $s_{\text{ICL}}(y \mid x; \mathcal{D}_k^{\text{demo}})$ is

$$\frac{1}{|y|} \sum_{t=1}^{|y|} \log \pi(y_t \mid \mathcal{D}_k^{\text{demo}}, x, y_{<t}), \quad (7)$$

where demonstrations are prepended to the input prompt. Both personalized RGD and ICL leverage user-specific information—z_k versus $\mathcal{D}_k^{\text{demo}}$—but through different mechanisms: learned reward shaping versus direct context conditioning. This allows us to compare whether parametric reward models or demonstration-based adaptation better capture user preferences, particularly as model scale increases.

Outstanding Limitation. Policy accuracy measures how well the scoring function ranks static responses—not whether the policy will actually *generate* outputs that actually align with user preferences. A method might rank existing responses correctly while producing generations that differ substantially from what users would write. This motivates our behavioral evaluation in Section 5.

5 Pref-LaMP: A Benchmark for Direct Behavioral Evaluation

To enable direct measurement of behavioral alignment without circular reward-based metrics, we introduce Pref-LaMP—a personalized alignment benchmark providing both pairwise preferences and ground-truth user-authored completions.

Dataset Construction Pref-LaMP derives from LaMP-5 (Salemi et al., 2024), pairing researchers' abstracts with their titles. Both are author-written, capturing individual style. We construct preferences via hard negative mining: (1) encode abstracts with Qwen3-Embedding-0.6B, (2) retrieve top-k similar abstracts, (3) sample one retrieved abstract as x and use its title as $y^{(l)}$, (4) use original title as $y^{(w)}$. This ensures rejections are topically relevant but different in title formulation[1].

Pref-LaMP is the first benchmark enabling direct behavioural evaluation of personalization through user-authored completions, measurable via ROUGE and BERTScore.

Behavioral Alignment Metric We evaluate end-to-end behavioural alignment by comparing user-generated and personalized model responses.

Definition 3 (Behavioral Alignment)**.** *Let $\mathcal{G} : \mathcal{X} \to \mathcal{Y}$ be a generation operator and $\mathcal{S} : \mathcal{Y} \times \mathcal{Y} \to \mathcal{R}$ be a similarity measure. For user k with test set $\mathcal{P}_k = \{(x_j, y_j^{GT})\}_{j=1}^{M_k}$:*

$$\mathcal{A}_{\mathcal{S}}(\mathcal{G}, k) = \frac{1}{|\mathcal{P}_k|} \sum_{(x_j, y_j^{GT}) \in \mathcal{P}_k} \mathcal{S}\Big(\mathcal{G}(x_j), y_j^{GT}\Big) \quad (8)$$

We instantiate $\mathcal{G}$ with ARGS decoding (Eq. 1), zero-shot generation and ICL generation. Meanwhile, $\mathcal{S}$ in instantiated with ROUGE-1 (lexi-

[1] Human-written rather than LLM-written negatives avoid shortcut learning. Initial LLM-generated rejections let linear probes detect generation artifacts rather than preference signals.

Dataset	Reward Model	Base Policies
TLDR	SmolLM2-180M	180M, 360M, 1.7B
PRISM	Qwen2.5-0.5B	0.5B, 1.5B, 3B, 7B
Pref-LaMP	Qwen2.5-0.5B	0.5B, 1.5B, 3B, 7B

Table 1: Model configurations.

cal overlap), ROUGE-L (longest common subsequence), and BERTScore-F1 (semantic similarity). This measures whether generated outputs match what users actually write, providing ground-truth assessment of behavioral alignment.

6 Experimental Setup

Datasets and Models We consider datasets: **TLDR** (Stiennon et al., 2020): Binary stylistic preferences, 10 training users (2,097 prefs/user), 31 adaptation users (100 prefs/user). **PRISM** (Kirk et al., 2024): Pluralistic preferences, 1,232 training users (22.1 prefs/user), 139 adaptation users (14.5 prefs/user). **Pref-LaMP (ours):** User-authored completions, 485 training users (48.8 prefs/user), 126 adaptation users (49.2 prefs/user).

Models and Methods All reward models use LoRA rank 8, trained on $\mathcal{U}_{\text{train}}$ to learn shared θ and user-specific $\{z_k\}$. We evaluate six personalization methods: LoRE (Bose et al., 2025) (learn reward bases and user-specific convex combination), LoRE-Alt (same as LoRE but alternates between bases and user specific parameter gradient steps), PReF (Shenfeld et al., 2025) (collaborative filtering), PAL (Chen et al., 2025a), VPL (Poddar et al., 2024), MPU/MPU-Avg (a simple baselines of per-user MLPs), and P-DPO (Li et al., 2024) (personalized direct preference optimization). Baselines include Global-RM (non-personalized Bradley-Terry using last token embedding as input to RM decoder), Global-RM-V2 (a sequence reward is average reward for all tokens), GenARM (Xu et al., 2025) (autoregressive RM for token-level guidance), zero-shot generation, ICL (random demonstrations), and ICL-RAG (retrieved demonstrations).

Evaluation Protocol We measure: (1) RM accuracy on adaptation users' held-out preferences (Eq. 2), (2) Adaption users' policy accuracy vs prior (no-reward) and global reward baselines (Eq. 3), (3) generation quality on PREF-LAMP via ROUGE-1/L and BERTScore against ground-truth (Eq. 8), and (4) win rates where each method's RM judges its own outputs versus zero-shot baseline.

	Method	RM Accuracy	Policy Accuracy 135M	360M	1.7B
Personalized	LoRE	82.56 ± 1.12	81.09 ± 0.76	81.09 ± 0.89	81.01 ± 1.10
	MPU	76.76 ± 0.26	49.96 ± 3.24	54.36 ± 7.36	55.12 ± 8.07
	MPU-Avg	77.83 ± 0.19	52.91 ± 4.01	52.96 ± 3.97	52.94 ± 3.98
	P-DPO	-	-	-	-
	PAL	87.00 ± 0.05	-	-	-
	PReF	65.27 ± 8.15	46.68 ± 7.18	49.56 ± 11.77	49.65 ± 11.95
	VPL	68.67 ± 1.94	59.78 ± 4.34	60.40 ± 4.82	60.67 ± 4.92
Baselines	GenARM	53.67 ± 3.15	53.06 ± 0.02	61.56 ± 0.03	62.59 ± 0.02
	Global	73.67 ± 2.45	58.80 ± 4.79	59.58 ± 1.55	60.34 ± 1.80
	Global-V2	38.66 ± 1.40	53.05 ± 0.02	61.26 ± 0.07	62.54 ± 0.04
	Prior	-	62.00 ± 0.00	61.57 ± 0.00	64.60 ± 0.00
	Pearson r	-	0.41	0.15	0.11
	Spearman ρ	-	0.36	0.09	0.09
	Kendall τ	-	0.31	0.08	0.09

Table 2: Policy accuracy and reward model performance on TLDR + SmolLM. Base reward model is SmolLM2-135M. Top: Personal rewards. Mid: Global alignment.

7 Result 1: Reward Model Accuracy Does Not Imply Policy Accuracy

We first investigate whether reward model accuracy predicts policy accuracy under reward-guided decoding. I.e., whether personal rewards that rank preferences well can guide policies to do the same.

TLDR: Weak Correlation on Simple Data We first evaluate the popular TDLR dataset's simple binary style preferences in Table 2. The main observation is that *upstream RM accuracy correlates weakly with downstream policy accuracy* (Kendall's $\tau = 0.08$–0.31), degrading as scale increases (Pearson r: $0.41 \rightarrow 0.11$ from 180M to 1.7B).

Additionally, most personalization methods (MPU, MPU-Avg, PReF) fail to adapt policies, achieving policy accuracies below the prior baseline (62-65%), particularly at smaller scales. PAL achieves highest RM accuracy (87.0%) but doesn't support RGD, while LoRE—with lower RM accuracy (82.6%)—achieves superior policy accuracy (~81%). Overall, LoRE substantially outperforms all competitors as well as prior and global reward policies, demonstrating genuine effectiveness for inference-time adaptation and validating our evaluation framework.

PRISM: Personalization Fails on Pluralistic Data Results for the more complex PRISM benchmark are summarized in Table 3 and Figure 1. All personalization methods fail in terms of their policy accuracy underperforming the global RM baseline (77.9% vs. 52.8-74.3%) in Table 3. Meanwhile, RM-policy correlation, as shown in Fig 1, remains weak (Kendall's $\tau = 0.17$-0.29), slightly strengthening with scale (r: $0.31 \rightarrow 0.48$). Critically, VPL achieves highest policy accuracy (63.8-66.4%) despite 9.5 points lower RM accuracy

	Personalized							Global	
	LoRE	LoRE-Alt	MPU	MPU-Avg	PAL	PReF	VPL	Global RM	Global RM-v2
RM Acc.	$65.95^{\pm 4.57}$	$67.15^{\pm 0.42}$	$52.76^{\pm 5.40}$	$60.43^{\pm 1.25}$	$70.74^{\pm 0.42}$	$74.34^{\pm 2.20}$	$68.35^{\pm 2.59}$	$\mathbf{77.94}^{\pm 3.55}$	$62.11^{\pm 4.15}$

Table 3: Reward model accuracies of various personal alignment methods using Qwen2.5-0.5B backbone on PRISM dataset. All personal alignment methods underperform non-personal Global Reward Model V1.

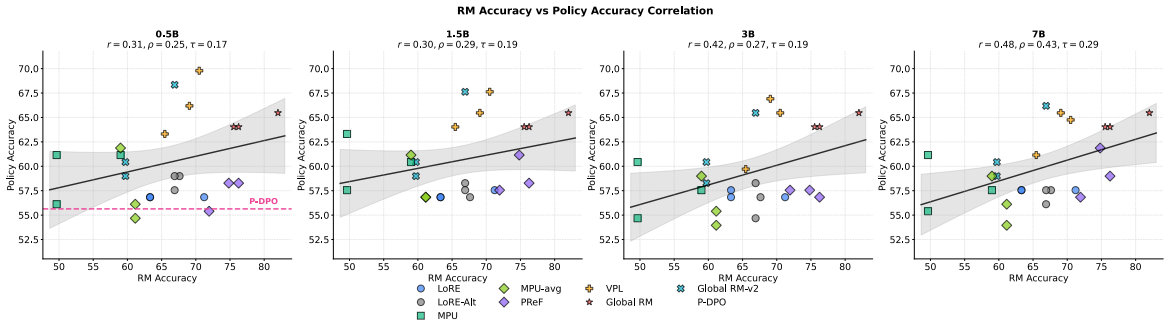

Figure 1: RM vs Policy Accuracy Correlation on PRISM across model scales. Correlations remain consistently weak: Pearson r ranges from 0.30–0.48, Spearman ρ from 0.25–0.43, and Kendall τ from 0.17–0.29. While correlations slightly strengthen with scale, they remain far below what would be needed for RM accuracy to reliably predict policy performance. Notably, methods with similar RM accuracy can have substantially different policy accuracy and vice-versa, demonstrating that the standard RM metric is not a reliable proxy for deployment performance.

than Global RM—a complete ranking inversion.

In terms of scale, methods show minimal scaling gains, remaining in narrow bands (LoRE-Alt: 57.1-58.5%, VPL: 63.8-66.4%). Unlike TLDR/SmolLM2 where correlations degraded with scale (r = 0.41→0.11), PRISM/Qwen2.5 shows strengthening correlations (r = 0.31→0.48). Whether this reflects dataset differences, model architecture, or their interaction remains unclear. Regardless, even at 7B, correlations remain too weak for choosing methods based on RM accuracy.

Discussion Our careful control evaluation shows wide failure of prior personal alignment methods both in terms of beating global alignment baselines, and in terms of the standard metric of RM accuracy not corresponding to downstream policy accuracy. We attribute this to a mixture of released code not reproducing results, missing non-personal baselines, and inconsistent non-comparable choice of datasets in prior evaluations. See Appendix D and E for further discussion.

Implication: Personal RM accuracy does not reflect performance during policy inference and cannot guide choice of reward model for deployment: Methods with 10+ point RM gaps can perform identically as adapted policies; methods with near identical RM accuracy can have 10+ point gaps in policy accuracy; and alignment methods can invert in ranking between reward and policy evaluations.

Recommendation. Future personal alignment methods must specify a policy adaptation strategy, and assess downstream policy understanding across multiple datasets and scales—not just upstream personal reward accuracy. The RM-policy disconnect demands new metrics measuring reward models' suitability for guiding generation, rather than pairwise ranking accuracy alone.

8 Result 2: Preference Discrimination Does Not Imply Generation Quality

Given the weak correlation between RM accuracy and policy discrimination ability under RGD, we now ask: even when methods achieve high policy accuracy—demonstrably preferring chosen over rejected responses—do they actually generate outputs that behaviorally align with user preferences?

8.1 Pref-LaMP: Complete Decoupling

We first study Pref-LaMP with preference ranking evaluation in Table 4. They key observation is that for this challenging task, similarly to PRISM (Table 3), personal alignment methods struggle to surpass Global RM baselines – for both the standard proxy metric of upstream RM accuracy, as well as our downstream policy accuracy. Only LoRE-Alt come close to the global baselines in RM accuracy.

We next move to analysing behavioural generation quality of the policies – as uniquely enabled

	Global			Personalized							
Acc	GenARM	Global RM	Global RM-v2	LoRE	LoRE-Alt	MPU	MPU-Avg	PAL	PReF	VPL	ICL
RM	$83.89^{\pm 0.60}$	$84.96^{\pm 0.13}$	$84.69^{\pm 0.07}$	$65.60^{\pm 7.89}$	$84.96^{\pm 0.48}$	$65.26^{\pm 1.03}$	$67.30^{\pm 0.30}$	$53.77^{\pm 0.28}$	$51.46^{\pm 3.84}$	$43.63^{\pm 3.77}$	-
Policy	$71.34^{\pm 0.56}$	$68.28^{\pm 0.00}$	$78.02^{\pm 0.94}$	$69.65^{\pm 0.31}$	$65.76^{\pm 6.60}$	$63.46^{\pm 8.11}$	$63.16^{\pm 6.70}$	-	$68.79^{\pm 0.70}$	$66.75^{\pm 2.33}$	$75.67^{\pm 0.44}$

Table 4: Pref-LaMP preference ranking accuracy for RMs and adapted policies (Qwen2.5-3B). RM personalization does not clearly outperform global RMs for either upstream RM or downstream policy preference ranking.

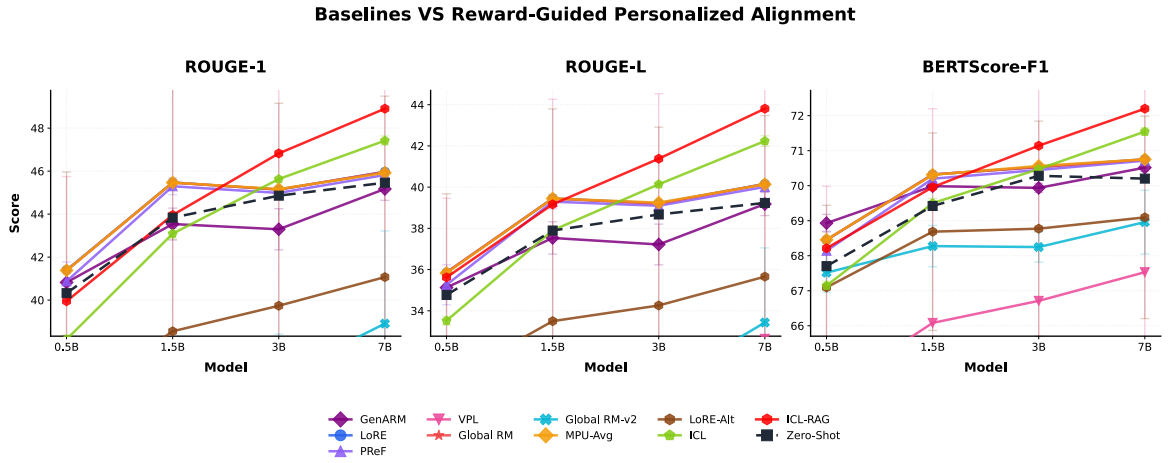

Figure 2: Generation quality (ROUGE-1, ROUGE-L, BERTScore-F1) under RGD across model scales. At 0.5B-1.5B, personalized RMs marginally improve over zero-shot; at 3B-7B, ICL baselines dominate all reward-guided methods.

by our Pref-LAMP dataset, in Figure 2 and raw results in Appendix C. We see that: (1) The top personal alignment methods for upstream ranking accuracy (LoRE-Alt in Table 4) tend to underperform in downstream generation quality. (2) A few personal alignment methods can surpass the zero-shot baseline but produce comparative performance to Global RM baseline. However, the better methods for generation (e.g, MPU-Avg and LoRe) are worse for upstream ranking (Table 4). Both these observations reflect decoupling between upstream RM accuracy and downstream behavioural generation. This shows that downstream generation quality evaluation is a crucial missing component of standard evaluation practice.

Our end-to-end behavioural evaluation also allows direct comparison between existing RM-focused personal alignment approaches and ICL. From Table 4, we can see that ICL actually achieves better policy preference ranking than the personal RMs. In terms of generation quality, Figure 2 shows that direct application of ICL surpasses both the baselines and prior personal alignment methods at 3-7B scale. This suggests that practitioners today should use simple ICL in favour of complex RM-based alignment approaches.

Discussion: Reconciling Prior Claims How

	Method	0.5B	1.5B	3B	7B
Global	GenARM	100.0%	100.0%	100.0%	99.8%
	Global RM	99.9%	99.9%	99.9%	99.9%
	Global RM-v2	98.4%	97.2%	89.4%	84.4%
Personalized	LoRE	73.7%	72.9%	73.8%	73.7%
	LoRE-Alt	97.0%	96.0%	96.3%	96.3%
	MPU	61.9%	61.8%	61.5%	61.5%
	MPU-Avg	69.8%	69.3%	70.1%	69.9%
	PReF	54.9%	54.8%	56.1%	55.5%
	VPL	51.1%	50.1%	50.5%	48.4%

Table 5: Win rate: fraction of examples where each RM judges its own RGD output as better than zero-shot. High win rates reveal severe reward hacking—methods claim near-perfect improvement despite ground-truth metrics showing minimal or negative gains (Fig 2).

can we reconcile prior papers' claims of successful RM+RGD based non-personal alignment with the often negative results from our experiments? The evaluation protocol of prior RGD-based analyses involved guiding generation with a RM, and then evaluating the resulting generations using the same RM (Khanov et al., 2024). The issue with win-rates scored in this way is circularity. If RGD-adaptation can hack the RM (find a 'false positive' response that the RM accepts, while not actually reflecting user preferences), using the same RM to evaluate the result produces overly optimistic results.

To study this protocol, we report win Rate vs zero-shot in Table 5, which confirms the risk of 'circular' evaluation. Using the same RM to guide decod-

Scale	RM Acc	Policy Acc	Win Rate
Kendall's τ with ROUGE-1			
0.5B	-0.126	-0.017	-0.135
1.5B	-0.126	-0.054	-0.188
3B	-0.148	-0.158	-0.112
7B	-0.114	-0.112	-0.034

Table 6: Correlations between metrics and generation quality (ROUGE-1) on Pref-LaMP. All correlations are negligible to negative across model scales, demonstrating no standard metric predicts behavioral alignment.

ing and judge completions vs a baseline suffers from reward hacking/overoptimization (El-Mhamdi and Hoang, 2024). GenARM claims 100% improvement over zero-shot despite ROUGE-1 being marginally worse. Global RM claims 99.9% superiority while producing outputs identical to methods 20 points lower in RM accuracy. The circular evaluation protocol is thus vulnerable to reward hacking, and only appears to solve personal alignment.

Despite their limitations, ranking held-out preferences (Table 2,4) does not suffer from this – because the RM is not used to generate; neither does our ground-truth evaluation (Figure. 2) – because the RM generation is compared to ground-truth.

Additional Analysis: ICL We provide further analysis in Appendix C.1 showing ICL and ICL-RAG improve further with shots higher than 8 (as used in the main text).

No Metric Predicts Generation Quality To summarise, we considered the standard RM accuracy, RGD-win rate, and our policy-accuracy metrics, all of which are discriminative ranking metrics. Table 6 correlates each of these against our end-to-end generation quality metric. All exhibit negligible to negative correlations with generation quality (Kendall's τ = -0.188 to -0.017), with the negative correlations suggesting reward hacking.

Takeaway: No existing metric predicts whether personalization methods will generate aligned outputs. Ground-truth evaluation on user-authored completions is a necessary evaluation component.

9 Discussion and Conclusion

Our findings reveal an evaluation crisis in personal alignment research: RM accuracy is uncorrelated with policy accuracy (τ = 0.08-0.31), and method rankings can completely invert between upstream and downstream evaluations. Using Pref-LaMP—the first benchmark with ground-truth user completions, we show discriminative metrics fail to predict generation quality (Table 6): reward models claiming 99% win rates show no improvement over baselines in ground-truth similarity. The field has been optimizing proxy metrics divorced from deployment objectives.

On a more positive note, we highlight that in contrast to these issues with personal rewards and their evaluations, simple in-context learning dominates reward-guided methods for models $\geq$3B parameters, while being easy and reliable to implement.

Practical Recommendations. Practitioners should use ICL with retrieval for 3B+ models; reward modeling adds complexity without benefit at scale. Researchers should: (1) evaluate complete pipelines end-to-end, not just reward model accuracy, (2) include policy accuracy and ground-truth behavioral metrics, (3) test across model scales to detect scale-dependent effects, (4) build behavioral benchmarks with user-authored completions and (5) compare against ICL baselines and focus future research effort on developing such amortized approaches to personal alignment.

9.1 Limitations

We focus on RGD because it represents a key scalable deployment path—per-user RL fine-tuning remains computationally infeasible for realistic populations. A fundamental challenge with RGD is that it assumes reward models can be token-wise factorized to provide local guidance at each generation step, which is a known source of error when this assumption is violated (Li et al., 2025). While GenARM is specifically designed to address this limitation through token-level autoregressive reward training, it still exhibits the same performance gaps we observe across other methods. This suggests the problem runs deeper than factorization alone—the disconnect between preference learning and generation guidance may be fundamental to the inference-time adaptation paradigm.

Our use of three datasets goes beyond most prior work, which often used only one or contrived datasets. However, our results do show some facets of dataset dependence, so future work should aim to establish larger multi-dataset benchmark suits to thoroughly test personalization across more dimensions of interest.

References

Avinandan Bose, Zhihan Xiong, Yuejie Chi, Simon Shaolei Du, Lin Xiao, and Maryam Fazel. 2025.

Lore: Personalizing LLMs via low-rank reward modeling. In *2nd Workshop on Models of Human Feedback for AI Alignment*.

Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. 2025a. PAL: Sample-efficient personalized reward modeling for pluralistic alignment. In *The Thirteenth International Conference on Learning Representations*.

Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. 2025b. PAD: Personalized alignment at decoding-time. In *The Thirteenth International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

El-Mahdi El-Mhamdi and Lê-Nguyên Hoang. 2024. On goodhart's law, with an application to value alignment. *Preprint*, arXiv:2410.09638.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Kenshi Abe, Kaito Ariu, Mitsuki Sakamoto, and Eiji Uchibe. 2025. Evaluation of best-of-n sampling strategies for language model alignment. *Transactions on Machine Learning Research*.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2024. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*.

Jiaming Ji, Boyuan Chen, Hantao Lou, Donghai Hong, Borong Zhang, Xuehai Pan, Juntao Dai, Tianyi Qiu, and Yaodong Yang. 2024. Aligner: Efficient alignment by learning to correct. In *Advances in Neural Information Processing Systems*, volume 37, pages 90853–90890. Curran Associates, Inc.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. 2024. ARGS: Alignment as reward-guided search. In *The Twelfth International Conference on Learning Representations*.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Michael Bean, Katerina Margatina, Rafael Mosquera, Juan Manuel Ciro, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. 2024. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Bolian Li, Yifan Wang, Anamika Lochab, Ananth Grama, and Ruqi Zhang. 2025. Cascade reward sampling for efficient decoding-time alignment. In *Second Conference on Language Modeling*.

Xinyu Li, Ruiyang Zhou, Zachary Chase Lipton, and Liu Leqi. 2024. Personalized language modeling from personalized human feedback. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*.

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. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. 2024. Personalizing reinforcement learning from human feedback with variational preference learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. 2023. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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

pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? *Preprint*, arXiv:2303.17548.

Idan Shenfeld, Felix Faltings, Pulkit Agrawal, and Aldo Pacchiano. 2025. Language model personalization via reward factorization. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. 2024. Position: a roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Jeremy Tien, Jerry Zhi-Yang He, Zackory Erickson, Anca Dragan, and Daniel S. Brown. 2023. Causal confusion and reward misidentification in preference-based reward learning. In *The Eleventh International Conference on Learning Representations*.

Xueru Wen, Jie Lou, Yaojie Lu, Hongyu Lin, XingYu, Xinyu Lu, Ben He, Xianpei Han, Debing Zhang, and Le Sun. 2025. Rethinking reward model evaluation: Are we barking up the wrong tree? In *The Thirteenth International Conference on Learning Representations*.

Xiaobao Wu. 2025. A comprehensive survey on learning from rewards for large language models: Reward models and learning strategies. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17847–17875, Suzhou, China. Association for Computational Linguistics.

Yuancheng Xu, Udari Madhushani Sehwag, Alec Koppel, Sicheng Zhu, Bang An, Furong Huang, and Sumitra Ganesh. 2025. GenARM: Reward guided generation with autoregressive reward model for test-time alignment. In *The Thirteenth International Conference on Learning Representations*.

Kailai Yang, Zhiwei Liu, Qianqian Xie, Jimin Huang, Tianlin Zhang, and Sophia Ananiadou. 2024a. Metaaligner: Towards generalizable multi-objective alignment of language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 34453–34486. Curran Associates, Inc.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. 2024b. Rewards-in-context: multi-objective alignment of foundation models with dynamic preference adjustment. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. 2024. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10586–10613, Bangkok, Thailand. Association for Computational Linguistics.

A Extended Related Work and Limitations

Incomparable Accuracy Metrics RLHF and DPO both use pairwise preference accuracy, but these metrics measure fundamentally different things. In RLHF, reward model accuracy measures how well R_θ ranks response pairs. However, the reward model is not the final artifact—it guides a policy through ARGS or RL fine-tuning. The critical question is: *does the resulting policy generate aligned responses?* Reward model accuracy cannot answer this. A reward model might perfectly rank static pairs while the derived policy fails to generate appropriate responses. In DPO, policy accuracy measures whether π_ϕ assigns higher probability to preferred responses, but only at the likelihood level—not generation quality. These metrics are not comparable across methods, and neither directly measures the ultimate goal: whether generated outputs align with user preferences.

Circular Evaluation Under Frozen Rewards A common RLHF practice adapts policies using reward models, then evaluates by measuring if adapted policies achieve higher rewards than baselines. This creates circularity: the reward model serves as both the training signal and evaluation metric. High performance only confirms the policy learned to exploit the reward model's scoring function—not that it captures actual user preferences. If the reward model is misspecified, this circular evaluation systematically hides the failure. A policy could achieve high reward scores while generating responses users would disprefer, and the evaluation cannot detect this because both training and evaluation use the same potentially-flawed reward model.

Proxy-Based Evaluation with LLM-as-a-Judge Recent work uses frontier LLMs as judges, conditioning them on few-shot user examples to rank policy outputs. While appealing, LLM-as-judge remains a learned proxy, not a direct measure of user satisfaction. It provides only relative rankings between methods and cannot quantify whether even the best-ranked method produces satisfactory outputs for individual users.

Toward Comprehensive Evaluation These limitations motivate our evaluation framework, which: (1) introduces comparable metrics for both reward model quality and policy understanding, (2) breaks circular evaluation by measuring behavioral alignment against ground-truth user completions rather than reward scores, (3) moves beyond proxies to evaluate actual generation quality, and (4) disentangles where personalization succeeds or fails across the reward modeling, policy guidance, and generation stages.

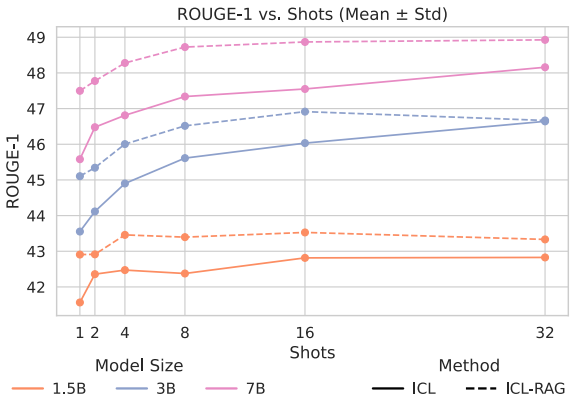

Figure 3: Pref-LAMP generation quality. ROUGE-1 vs. number of in-context examples. ICL-RAG consistently outperforms random ICL, with performance scaling with both model size and context length.

B Raw Results: PRISM

Please note that we only evaluate on a subset of the test split of PRISM. This is because policy accuracy computation was expensive. Reward model's performance on the full test split is in Table 8. Global RM still outperforms all other methods so our conclusions in the main paper text does not change. Meanwhile, data plotted in Figure 1 can be found in Table 7.

C Raw Results: Pref-LaMP5

Raw results for Pref-LaMP5 dataset can be found in Tables 9, 10, 11 and 12.

C.1 ICL Further Analysis

We further analyse our strong ICL baselines in terms of number of demonstrations. ICL-RAG improves steadily with demonstrations and scale, reaching ~49 ROUGE-1 at 7B with 8 shots. Larger models show no saturation, effectively leveraging context without reward guidance. This shows that personal alignment is not only possible, but straightforward to implement. However, operationalizing standard but more complex RM-based personal alignment approaches with RGD is comparatively fraught. This is shown in Figure 3.

	LoRE	PReF	VPL	Global RM	Global RM-v2	MPU	MPU-Avg	LoRE-Alt
RM Accuracy	65.95 ± 4.57	74.34 ± 2.20	68.35 ± 2.59	77.94 ± 3.55	62.11 ± 4.15	52.76 ± 5.40	60.43 ± 1.25	67.15 ± 0.42
Policy accuracies								
0.5B	56.83 ± 0.00	57.31 ± 1.66	66.43 ± 3.24	64.51 ± 0.83	62.59 ± 5.04	59.47 ± 2.91	57.55 ± 3.81	58.51 ± 0.83
1.5B	57.07 ± 0.42	58.99 ± 1.90	65.71 ± 1.81	64.51 ± 0.83	62.35 ± 4.63	60.43 ± 2.88	58.27 ± 2.49	57.55 ± 0.72
3B	57.07 ± 0.42	57.31 ± 0.42	64.03 ± 3.81	64.51 ± 0.83	61.39 ± 3.69	57.55 ± 2.88	56.12 ± 2.59	56.59 ± 1.81
7B	57.55 ± 0.00	59.23 ± 2.53	63.79 ± 2.31	64.51 ± 0.83	61.87 ± 3.81	58.03 ± 2.91	56.35 ± 2.53	57.07 ± 0.83

Table 7: Reward model accuracy and policy accuracies per model and method for the PRISM dataset.

	LoRE	LoRE-Alt	MPU	MPU-Avg	PAL	PReF	VPL	Global RM	Global RM2
RM Accuracy	56.53 ±0.15	60.33 ±0.57	49.66 ±1.13	49.07 ±1.46	62.51 ±0.30	62.73 ±0.64	60.31 ±2.98	64.51 ±0.41	59.18 ±0.74

Table 8: Test set accuracy evaluated across all samples from unseen users. Note: The results reported in the main text use only a single test sample per user due to computational constraints.

D Architectural and Initialization Enhancements to PReF for Reward-Guided Decoding and Dataset Considerations

D.1 Motivation for Architectural Modification

In our work, we leverage the core principles of PReF but introduce a key architectural modification and a novel initialization scheme. These changes are motivated by the need to adapt PReF from a pairwise *preference* model into a pointwise *reward* model, making it suitable for advanced applications such as reward-guided decoding.

The original PReF model is designed to predict a user's preference for one complete response over another. It computes a single score for a *pair* of items, (r_1, r_2). However, reward-guided decoding requires a scalar reward score for a *single, often incomplete, sequence* at each step of the generation process. The original PReF formulation is therefore unsuitable for this task.

To address this, we modified the PReF architecture to explicitly compute a user-specific reward for an individual response, $R(u, r)$. This allows us to score single candidate sequences during decoding.

D.2 Original vs. Modified Reward Formulation

The original PReF model calculates a preference score s for a user u and a pair of responses (r_1, r_2) based on the difference of their feature representations:

$$s(u, r_1, r_2) = \mathbf{u}^\top \big(\phi(r_1) - \phi(r_2)\big)$$

Here, ϕ is a linear head that projects the LLM's response embeddings into a latent feature space, and $\mathbf{u}$ is the user's embedding vector.

Our modified architecture decomposes this calculation into two distinct reward computations:

$$R(u, r) = \mathbf{u}^\top \phi(r)$$

$$s(u, r_1, r_2) = R(u, r_1) - R(u, r_2)$$

While these two formulations are mathematically equivalent in the final forward pass, this decomposition presents a significant challenge for the model's initialization, which we address with a novel technique.

D.3 The Initialization Challenge and Our Solution

The PReF methodology uses Singular Value Decomposition (SVD) on a (response_pair × user) preference matrix to warm-start the model's parameters. A key challenge arises because the SVD process yields a single feature vector, $\mathbf{v}_p$, for each response pair p. This vector $\mathbf{v}_p$ serves as a proxy for the latent feature *difference*, $\phi(r_1) - \phi(r_2)$.

The SVD provides no direct information about the individual feature vectors $\phi(r_1)$ and $\phi(r_2)$. To initialize a linear head ϕ that operates on individual responses, we leverage the linearity of the projection and work directly in the difference space.

We achieve this with the following direct regression algorithm:

1. **Perform SVD:** Perform SVD on the preference matrix as in the original PReF to obtain the matrix of pairwise feature vectors U_S (representing $\phi(r_1) - \phi(r_2)$ for each pair) and user embeddings V_S.

2. **Compute LLM Embedding Differences:** For each unique response pair (r_1, r_2) in the training data, compute the difference of their frozen LLM embeddings: $\mathbf{e}_{\text{diff}} = \mathbf{e}(r_1) - \mathbf{e}(r_2)$.

Method	RM Acc	Policy Acc	ROUGE-1	ROUGE-L	BertScore-F1
Reward Models + ARGS					
GenARM	83.89 ± 0.60	71.62 ± 0.85	40.82 ± 0.50	35.13 ± 0.42	68.93 ± 0.25
Global RM	84.96 ± 0.13	66.41 ± 0.13	41.37 ± 0.02	35.87 ± 0.02	68.45 ± 0.00
Global RM-v2	84.69 ± 0.07	79.13 ± 0.57	31.41 ± 0.75	26.40 ± 0.79	67.52 ± 0.56
LoRE	65.60 ± 7.89	67.84 ± 0.50	41.40 ± 0.01	35.86 ± 0.01	68.45 ± 0.01
LoRE-Alt	84.96 ± 0.48	64.63 ± 5.61	35.14 ± 10.81	30.61 ± 9.07	67.10 ± 2.34
MPU	65.26 ± 1.03	60.96 ± 8.54	39.97 ± 2.35	34.43 ± 2.30	67.69 ± 1.26
MPU-Avg	67.30 ± 0.30	60.14 ± 5.50	41.38 ± 0.03	35.83 ± 0.02	68.46 ± 0.01
PAL	53.77 ± 0.28	nan ± nan	nan ± nan	nan ± nan	nan ± nan
PReF	51.46 ± 3.84	66.06 ± 2.55	40.88 ± 0.90	35.26 ± 0.97	68.14 ± 0.56
VPL	43.63 ± 3.77	63.58 ± 4.02	31.04 ± 14.71	26.68 ± 12.78	64.09 ± 5.90
DPO					
P-DPO	-	77.57 ± 0.92	40.65 ± 0.46	34.56 ± 0.43	68.04 ± 0.24
Baselines					
Zero-Shot	-	-	40.32 ± 0.00	34.77 ± 0.00	67.70 ± 0.00
ICL	-	69.46 ± 0.16	38.17 ± 0.20	33.52 ± 0.21	67.14 ± 0.13
ICL-RAG	-	-	39.95 ± 0.02	35.62 ± 0.02	68.21 ± 0.00

Table 9: ROUGE scores for Qwen2.5-0.5B-Instruct (mean ± std across seeds).

3. **Direct Regression on Differences:** Construct a regression problem where the input matrix X contains all LLM embedding differences $\mathbf{e}_{\text{diff}}$, and the target matrix Y contains the corresponding SVD-derived pairwise features $\mathbf{v}_p$ from U_S. Solve for the linear head weights W such that:

$$W \cdot (\mathbf{e}(r_1) - \mathbf{e}(r_2)) \approx \mathbf{v}_p$$

4. **Bias-Free Linear Regression:** A bias-free (intercept-free) regression is used to find the optimal initial weights W for the linear head ϕ. This approach is mathematically sound because:
   - If $\phi(r) = W \cdot \mathbf{e}(r)$, then $\phi(r_1) - \phi(r_2) = W \cdot (\mathbf{e}(r_1) - \mathbf{e}(r_2))$
   - Learning W from differences is equivalent to learning it from individual features
   - The lack of bias term means reward values have an arbitrary global offset, which cancels out in pairwise comparisons

D.4 End-to-End Training

Following this warm-start initialization of both the user embeddings (from V_S) and the linear reward head (from our direct regression method), the model is trained end-to-end using backpropagation. During training, the model computes individual rewards $R(u, r_1)$ and $R(u, r_2)$ for chosen and rejected responses. The Bradley-Terry preference learning loss is then computed:

$$\mathcal{L} = -\log \sigma(R(u, r_1) - R(u, r_2))$$

where σ is the sigmoid function. Gradients are backpropagated to fine-tune ϕ, $\mathbf{u}$, and optionally the LLM encoder if not frozen.

This procedure optimizes the true preference learning objective, with the SVD-based initialization serving as a high-quality starting point that accelerates convergence and improves stability. Any imprecision in initialization (such as the arbitrary offset in absolute reward values) is corrected during training. This enhanced methodology preserves the core insights of PReF's SVD-based initialization while adapting its architecture to support reward-guided decoding.

D.5 PReF's Acknowledged Synthetic Augmentation

The original PReF implementation uses a synthetically augmented version of PRISM rather than the natural conversational data. The authors state:

Method	RM Acc	Policy Acc	ROUGE-1	ROUGE-L	BertScore-F1
Reward Models + ARGS					
GenARM	83.89 ± 0.60	71.24 ± 0.52	43.54 ± 0.74	37.53 ± 0.79	69.99 ± 0.04
Global RM	84.96 ± 0.13	67.17 ± 0.04	45.47 ± 0.02	39.45 ± 0.03	70.32 ± 0.00
Global RM-v2	84.69 ± 0.07	78.73 ± 0.71	32.75 ± 0.78	27.77 ± 0.78	68.27 ± 0.59
LoRE	65.60 ± 7.89	68.49 ± 0.26	45.47 ± 0.03	39.45 ± 0.04	70.32 ± 0.01
LoRE-Alt	84.96 ± 0.48	65.11 ± 6.09	38.55 ± 11.99	33.50 ± 10.30	68.69 ± 2.82
MPU	65.26 ± 1.03	61.97 ± 8.37	44.53 ± 1.59	38.56 ± 1.46	69.76 ± 0.90
MPU-Avg	67.30 ± 0.30	61.15 ± 6.38	45.47 ± 0.04	39.45 ± 0.04	70.31 ± 0.01
PReF	51.46 ± 3.84	67.07 ± 1.74	45.30 ± 0.39	39.30 ± 0.35	70.20 ± 0.18
VPL	43.63 ± 3.77	64.72 ± 3.68	35.19 ± 15.65	30.32 ± 13.94	66.08 ± 6.12
DPO					
P-DPO	-	77.57 ± 0.92	40.65 ± 0.46	34.56 ± 0.43	68.04 ± 0.24
Baselines					
Zero-Shot	-	-	43.83 ± 0.00	37.89 ± 0.00	69.42 ± 0.00
ICL	-	72.98 ± 0.38	43.09 ± 0.25	37.90 ± 0.07	69.50 ± 0.14
ICL-RAG	-	-	43.96 ± 0.03	39.16 ± 0.01	69.96 ± 0.00

Table 10: ROUGE scores for Qwen2.5-1.5B-Instruct (mean ± std across seeds).

> "However, the original PRISM dataset cannot be used directly because it was collected in a way that prevents overlap between users and prompts, which is necessary for our method. Therefore, we augmented it with synthetic annotations via the protocol described in PERSONA, resulting in 50 user preferences per prompt."

They leverage the PERSONA protocol (TODO cite), which uses LLMs as judges to simulate user preferences. The code snippet confirms this synthetic generation approach:

```
def get_preference_prism(
                    user_description,
                    prompt,
                    response_1,
                    response_2):
  prompt1=prompts.PRISM_no_confidence.\
   format(
       prompt=prompt,
     user_description=user_description,
       response_1=response_1,
       response_2=response_2,
   )
   pref1 = get_completion(
       prompt1,
       system_prompt=None,
       model="gpt-4o-mini",
       temp=0.0
       )
```

Code can be found in github codebase idanshen/PReF_code on line 78 in file PReF_code/utils/data.py. This can be found here.

D.6 Data Structure Analysis

Analysis of their dataset reveals a perfectly uniform structure:

Total training samples: 90,450

Unique (prompt, r_1, r_2) triples: 1,809

Unique users: 1,200

Users per prompt triple: **Exactly 50** (zero variance)

Average samples per user: 75.4

This perfect uniformity demonstrates artificial construction.

D.7 Real PRISM Conversational Data

In contrast, the real PRISM dataset (TODO cite) consists of:

- Natural multi-turn conversations between users and AI assistants

Method	RM Acc	Policy Acc	ROUGE-1	ROUGE-L	BertScore-F1
Reward Models + ARGS					
GenARM	83.89 ± 0.60	71.34 ± 0.56	43.30 ± 0.96	37.22 ± 0.99	69.94 ± 0.13
Global RM	84.96 ± 0.13	68.28 ± 0.00	45.14 ± 0.03	39.20 ± 0.02	70.53 ± 0.01
Global RM-v2	84.69 ± 0.07	78.02 ± 0.94	36.43 ± 1.99	30.36 ± 1.77	68.25 ± 0.43
LoRE	65.60 ± 7.89	69.65 ± 0.31	45.14 ± 0.03	39.21 ± 0.02	70.53 ± 0.01
LoRE-Alt	84.96 ± 0.48	65.76 ± 6.60	39.74 ± 9.42	34.26 ± 8.65	68.77 ± 3.07
MPU	65.26 ± 1.03	63.46 ± 8.11	44.75 ± 0.75	38.62 ± 0.97	70.22 ± 0.49
MPU-Avg	67.30 ± 0.30	63.16 ± 6.70	45.16 ± 0.05	39.25 ± 0.03	70.56 ± 0.04
PReF	51.46 ± 3.84	68.79 ± 0.70	44.99 ± 0.29	39.10 ± 0.16	70.45 ± 0.12
VPL	43.63 ± 3.77	66.75 ± 2.33	36.00 ± 14.93	30.90 ± 13.64	66.71 ± 6.07
DPO					
P-DPO	-	77.57 ± 0.92	40.65 ± 0.46	34.56 ± 0.43	68.04 ± 0.24
Baselines					
Zero-Shot	-	-	44.86 ± 0.00	38.67 ± 0.00	70.28 ± 0.00
ICL	-	75.67 ± 0.44	45.63 ± 0.04	40.14 ± 0.05	70.47 ± 0.06
ICL-RAG	-	-	46.82 ± 0.03	41.38 ± 0.04	71.14 ± 0.00

Table 11: ROUGE scores for Qwen2.5-3B-Instruct (mean ± std across seeds).

- Real human preferences expressed through dialogue
- Sparse preference matrix (unique conversation contexts)
- No systematic overlap between users and prompts

Our implementation extracts genuine conversational preferences, yielding ~27K training samples.

D.8 Comparison and Implications

PReF's "PRISM" dataset differs fundamentally from real PRISM data across multiple dimensions. While PReF uses synthetic data generated by GPT-4o-mini with simulated demographic preferences, real PRISM captures authentic human conversations and actual user choices. The synthetic dataset exhibits a dense matrix structure with exactly 50 users per item, enabling high controllability and strong SVD performance, whereas real PRISM data is characterized by sparse, unique contexts with natural variation that yields weaker SVD results. PReF's dataset contains over 90K samples compared to 27K in the real data, but this larger volume comes at the cost of realism—the synthetic patterns may not generalize to authentic human behavior in the way that real PRISM's genuine user interactions do.

D.9 Implications for Reproducibility and Comparison

Key points:

1. Not an apples-to-apples comparison—PReF's dense setup differs fundamentally from real sparse data.
2. SVD initialization performs better in dense synthetic matrices.
3. Training dynamics differ due to uniform synthetic distribution.
4. Evaluation on simulated preferences may not generalize to real human data.

D.10 Methodological Considerations and Design Choices

PReF's reliance on dense user-item overlap is intrinsic to collaborative filtering. Sparse real data poses challenges but reflects real-world personalization.

Design options:

- **Match PReF:** Use synthetic PERSONA-style preferences for strong SVD initialization.
- **Use Real Data:** Accept weaker SVD signals, require stronger regularization and robust training.

Method	RM Acc	Policy Acc	ROUGE-1	ROUGE-L	BertScore-F1
Reward Models + ARGS					
GenARM	83.89 ± 0.60	72.08 ± 0.58	45.17 ± 0.53	39.18 ± 0.56	70.52 ± 0.11
Global RM	84.96 ± 0.13	69.08 ± 0.04	45.96 ± 0.04	40.15 ± 0.02	70.75 ± 0.01
Global RM-v2	84.69 ± 0.07	78.44 ± 0.90	38.91 ± 4.31	33.43 ± 3.62	68.96 ± 0.91
LoRE	65.60 ± 7.89	70.97 ± 0.35	45.97 ± 0.03	40.16 ± 0.03	70.75 ± 0.00
LoRE-Alt	84.96 ± 0.48	66.52 ± 7.25	41.07 ± 8.42	35.66 ± 7.82	69.10 ± 2.89
MPU	65.26 ± 1.03	64.82 ± 8.31	45.56 ± 0.66	39.78 ± 0.66	70.45 ± 0.47
MPU-Avg	67.30 ± 0.30	64.02 ± 6.99	45.92 ± 0.04	40.14 ± 0.02	70.75 ± 0.01
PReF	51.46 ± 3.84	70.04 ± 0.45	45.82 ± 0.18	40.01 ± 0.23	70.71 ± 0.08
VPL	43.63 ± 3.77	67.82 ± 2.50	37.57 ± 14.22	32.64 ± 12.70	67.53 ± 5.39
DPO					
P-DPO	-	77.57 ± 0.92	40.65 ± 0.46	34.56 ± 0.43	68.04 ± 0.24
Baselines					
Zero-Shot	-	-	45.46 ± 0.00	39.23 ± 0.00	70.20 ± 0.00
ICL	-	74.74 ± 0.25	47.42 ± 0.22	42.24 ± 0.26	71.54 ± 0.11
ICL-RAG	-	-	48.90 ± 0.02	43.81 ± 0.01	72.20 ± 0.00

Table 12: ROUGE scores for Qwen2.5-7B-Instruct (mean ± std across seeds).

- **Hybrid:** Augment sparse real data with synthetic overlap.

Our Choice: We prioritize authenticity by using real PRISM conversational preferences in their natural sparse form, tackling the more difficult—but more realistic—personalization problem.

E LoRe Architecture

LoRE is a pairwise preference learning method introduced in *LoRe: Personalizing LLMs via Low-Rank Reward Modeling* (**?**). It learns a reward function from preference data, where each datapoint consists of a user input and two responses, one preferred over the other.

Unlike methods that train a binary classifier to predict which response is better, LoRE optimizes a **logistic loss** over the **difference of reward values** assigned to the preferred and dispreferred responses.

E.1 Architecture

The LoRE architecture consists of two key components:

Feature Extractor A shared feature extractor ϕ (typically a pretrained language model) processes the input x and response y to produce K base reward scores: $R_\phi(x, y) \in \mathbb{R}^K$.

User-Specific Weights For each user, we learn a low-rank weight vector $w \in \mathbb{R}^K$ that linearly combines these base rewards to produce a personalized scalar reward:

$$R_w(x, y) = w^\top R_\phi(x, y) \tag{9}$$

This architecture allows the model to learn a shared representation of reward dimensions through ϕ, while capturing individual user preferences through the lightweight weight vectors w.

E.2 Original Loss Formulation

For a preference pair (x, y^+, y^-) where y^+ is preferred over y^-, the loss uses the difference of personalized rewards:

$$\mathcal{L}_{\text{LoRE}} = \log(1+\exp(-w^\top [R_\phi(x, y^+) - R_\phi(x, y^-)])) \tag{10}$$

This encourages the model to assign a higher personalized reward to the preferred response y^+ over the dispreferred one y^-.

E.3 Two Training Algorithms

The paper introduces two variants:

LoRE Trains both user-specific weights w and the feature extractor ϕ simultaneously in a single optimization step. This approach was used for the TL;DR dataset in the original implementation.

LoRE-Alt Uses an alternating optimization strategy: for each batch, it takes one gradient step on the user-specific weights w (freezing the feature extractor ϕ), then one step on the feature extractor ϕ (freezing the weights w). This approach was used for more complex datasets in the original implementation.

LoRE-Alt also leverages an off-the-shelf reward model (Skywork RM) and includes a regularization term to prevent the learned model from deviating too far from the pretrained baseline. However, since we train our Qwen2.5-0.5B model from scratch without a pretrained reward model, we omit this regularization.

Note: The original codebase does not successfully reproduce results on the PRISM dataset.

E.4 our Variant: Equivalent Log-Sigmoid Loss

In our implementation, we instead use:

```
loss = -F.logsigmoid(reward_diff).mean()
```

where:

```
reward_diff = w.T @ (
    R_phi(x, y^+) - R_phi(x, y^-)
)
```

This is mathematically equivalent to the original logistic loss, since:

$$-\log(\sigma(x)) = \log(1 + \exp(-x)) \tag{11}$$

The `logsigmoid` loss is a numerically stable, PyTorch-friendly implementation of the same core principle. This change does not affect the training dynamics or final optimization target—it is purely an implementation detail.

E.5 Architectural Equivalence

In the original LoRE paper, the reward model can be formulated to take the **difference of features** directly:

$$w^\top [\phi(x, y^+) - \phi(x, y^-)] \tag{12}$$

In our implementation, we compute the reward separately on each response using the K-dimensional feature extractor, then take the weighted difference:

$$w^\top R_\phi(x, y^+) - w^\top R_\phi(x, y^-) \tag{13}$$

These formulations are mathematically identical due to the linearity of the inner product:

$$w^\top [R_\phi(x, y^+) - R_\phi(x, y^-)] = \tag{14}$$
$$w^\top R_\phi(x, y^+) - w^\top R_\phi(x, y^-) \tag{15}$$

This equivalence holds because the personalization layer (the w weights) is linear in the feature space.

E.6 ARGS Support

LoRE also supports **Alignment as Reward-Guided Search (ARGS)**, where generation is guided at decoding time using the learned reward model. In our implementation, we enable ARGS as a runtime decoding strategy by plugging in the learned reward model as a plug-and-play scoring function.

This is implemented by scoring candidate continuations during beam or sampling-based decoding using the personalized reward:

$$R_w(x, y_{\text{candidate}}) = w^\top R_\phi(x, y_{\text{candidate}}) \tag{16}$$

This allows us to steer generation toward responses that maximize the learned user-aligned reward signal, without requiring reinforcement learning or sampling from a reward-shaped distribution.

E.7 Known Reproduction Issues

It is a known issue that LoRe released code does not reproduce their results on the PRISM dataset due to issues in dataset preparation that confalted reported results. [2]. This can be found here.

F Hyperparameters

F.1 Reward Modelling

For the TLDR dataset, all models were trained with a LoRA of rank 8 and LoRA alpha of 16. rsLora was used for initializaiton. The backbone (LoRA module) was 5×10^{-5}. Different models decoder used varying hyperparameters as listed below:

1. **LoRe:** decoder LR=0.5, latent dim=2
2. **MPU & MPU-Avg:** decoder LR=1×10^{-3}
3. **PAL:** default hyperparameters from original implementation. Topic/Query projectors with LR=1×10^{-4} and weight decay of 0.01 and user weights with LR=5×10^{-3} without weight decay. Finally, latent dimension was 2
4. **P-DPO:** Number of soft tokens = 8
5. **PReF:** decoder LR was 1×10^{-3} and weight decay of 0.02. Latent dimension was 2

[2] https://github.com/facebookresearch/LoRe/issues/1

6. **Vanilla & Vanilla-V2:** decoder (reward MLP) was trained with LR=1×10^{-4}

7. **VPL:** used original implementation hyperparameters. The VAE component was trained with LR=1×10^{-4}. MLP used same LR and weight decay of 0.001

For PRISM, and **Pref-LaMP5** dataset, same hyperparameters were used with the differences of:

1. **LoRe:** latent dimension = 8

2. **LoRe-Alt:** same learning rates but latent dimension = 20

3. **PReF:** latent dimension = 8

4. **PAL:** latent dimension = 8

F.2 ARGS

To find the best weight hyperparameter for ARGS on TLDR and PRISM dataset, we maximized the policy accuracy metric over the seen/train users test split for each trained model. We found that this is necessary since reward models trained using the same algorithm can converge to producing rewards of different magnitudes. For LaMP-5, we used the harmonic mean of rouge-1, inverse perplexity (coherence), and policy accuracy to find the best weight which is subsequently used during generation.

G Use of AI

We used AI assistance in two capacities during this work:

Code development: We used Cursor IDE with AI-assisted code completion during implementation. All AI-generated code suggestions were manually reviewed and verified before integration into the codebase.

Writing assistance: We used large language models to help articulate technical concepts and improve clarity of exposition. The conceptual content, experimental design, results, and conclusions are entirely our own work. AI assistance was limited to rephrasing and refining presentation of ideas we specified.

All scientific claims, experimental results, and intellectual contributions in this paper are the original work of the authors.