PERSPL: PERSONALIZED SEMI-SUPERVISED PREFERENCE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized reward models require user-specific preference comparisons, but collecting such feedback is costly and provides only limited supervision for each user. At the same time, large pools of responses can be generated without user identities or preference labels, yet existing personalized reward models leave this data unused. Directly learning from these responses is difficult because pairwise objectives constrain relative scores within a prompt, whereas pseudo-labeling individual responses requires scores that are comparable across prompts for the same user. We propose **PerSPL** (Personalized Semi-supervised Preference Learning), a backbone-agnostic framework that uses pair-mean score centering to stabilize this cross-prompt score coordinate and iteratively turns unlabeled responses into user-specific training signals. Across three primary benchmarks with a full suite of baselines, including Reddit TLDR with real-user preference comparisons, PerSPL achieves the highest mean accuracy for both seen and unseen users. A focused evaluation on Chatbot Arena further shows gains over the two strongest personalized baselines from the primary suite on real-user open-domain preferences. These results show that unlabeled responses can provide additional supervision for personalized reward modeling when per-user preference labels are scarce.

1 INTRODUCTION

Large language models are expected to serve users with different goals, values, and communication preferences. The best response for one user may therefore not be the best for another Bakker et al. (2022); Durmus et al. (2023); Kirk et al. (2024); Ryan et al. (2025). Alignment pipelines commonly learn what people prefer by asking them to compare two responses to the same prompt. A reward model learns to assign higher scores to preferred responses, and these scores guide model training or response selection Christiano et al. (2017); Ouyang et al. (2022); Bai et al. (2022). Most reward models, however, pool all comparisons and learn a single, population-wide notion of a good response. When users disagree systematically, this aggregate reward can obscure their differences and poorly reflect individual preferences. Personalized reward models instead make the score depend on the user and their prior feedback Poddar et al. (2024); Chen et al. (2025); Bose et al. (2025); Cai et al. (2026).

Personalizing the score creates a more demanding data requirement. A population model can pool judgments from all annotators, whereas a personalized model must learn about each user from that user's limited feedback. Every preference label requires the user to inspect and compare responses, so even seen users provide only short histories and a new user must be modeled from a handful of examples. Responses themselves are easier to obtain because they can be generated or collected without additional user feedback. This creates an opportunity to supplement scarce per-user labels with a much larger, shared pool of unpaired responses. Existing personalized reward models improve learning from sparse comparisons by sharing information across users Poddar et al. (2024); Chen et al. (2025); Bose et al. (2025); Cai et al. (2026), but they do not use this pool as additional supervision. What is missing is a way to turn the same unpaired response into potentially different training signals for different users.

Pairwise supervision alone does not determine how to pseudo-label individual responses across prompts. It requires a preferred response to score above the rejected response for the same prompt, but it does not fix where either score lies relative to scores for other prompts. Shifting both scores for

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.

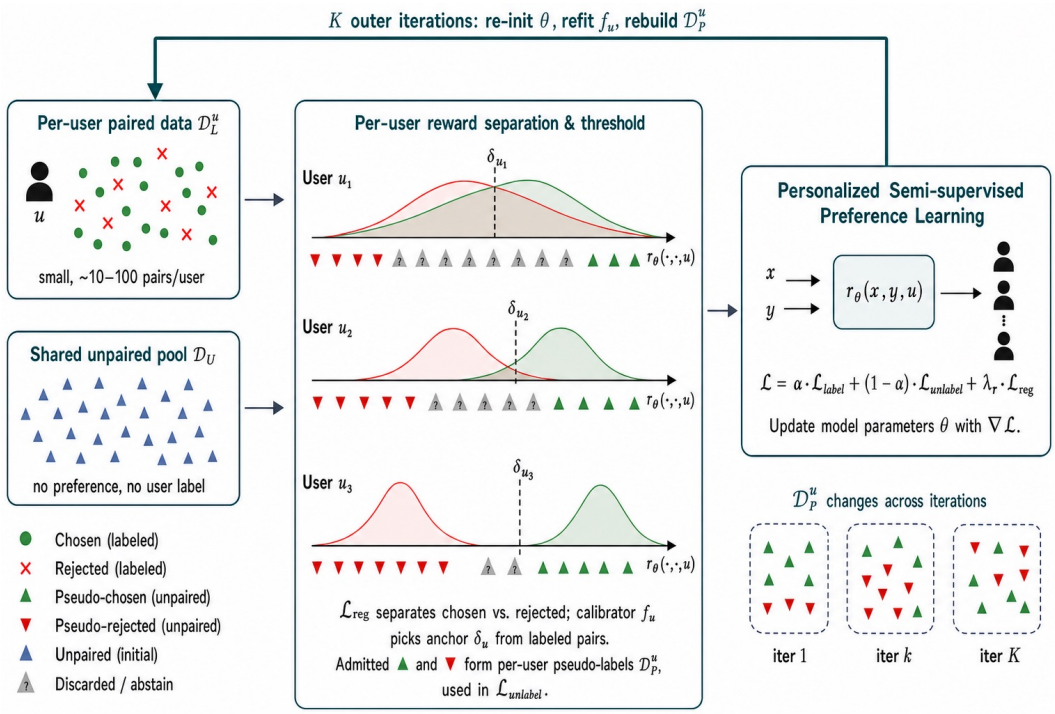

Figure 1: Overview of PERSPL. A small set of labeled comparisons identifies each user's preferences, while pair-mean score centering makes reward scores more comparable across prompts. A calibrator fitted separately for each user assigns pseudo-labels to confident responses from a shared unpaired pool and discards ambiguous ones. The reward model is then trained iteratively on both labeled and pseudo-labeled data.

a prompt by the same amount preserves the pairwise prediction while changing these cross-prompt comparisons. This ambiguity is irrelevant when ranking responses to the same prompt. It matters for an unpaired example, which contains one response and must be judged against a threshold fitted across prompts. Reliable pseudo-labeling therefore requires comparable scores across prompts for each user. Existing semi-supervised preference-learning methods use unlabeled data to train a population-wide reward or policy He et al. (2024); Kim et al. (2024); Lee et al. (2025). However, these methods do not provide this user-specific score comparability.

We address these challenges with **PerSPL** (Personalized Semi-supervised Preference Learning), which combines cross-prompt score stabilization with user-specific pseudo-labeling. To make scores more comparable across prompts, PerSPL encourages the two scores in each labeled pair to have mean zero, a mechanism we call pair-mean score centering. It then fits a per-user mapping from reward scores to preference probabilities, so the same response can receive different pseudo-labels across users. PerSPL retains confident pseudo-labels and iteratively trains on them alongside labeled data, extracting additional supervision without collecting new user feedback. Figure 1 shows how labeled and unpaired data interact in this process.

We make three contributions:

- **Personalized learning from unpaired responses.** We formulate semi-supervised personalized reward modeling, in which limited per-user comparisons are supplemented by a shared pool of responses without user identities or preference labels. PerSPL converts this pool into user-specific supervision for seen users and for new users given a small labeled support set.

- **Score centering for cross-prompt pseudo-labeling.** We identify a score-coordinate ambiguity that prevents pairwise reward models from directly pseudo-labeling individual responses across prompts. We introduce pair-mean centering to address this ambiguity, connect it theoretically to

cross-prompt and single-response risk, and show experimentally that it enables unpaired training to improve personalized reward learning.

- **Evaluation across complementary personalization settings.** PerSPL achieves the highest mean seen- and unseen-user accuracy in every primary setting across three benchmarks and a full suite of baselines. A focused Chatbot Arena evaluation provides further evidence on real-user open-domain preferences, while controlled generator-shift experiments show that the overall unlabeled-data benefit persists across all tested mixtures.

2 Related Work

Personalized reward modeling. Existing personalized reward models address per-user feedback scarcity through architectural sample efficiency, falling into the three paradigms identified by MRM Cai et al. (2026): (1) *personalized input* methods that condition a shared reward on user-specific signals such as personas or learned embeddings Zhao et al. (2023), induced personas Ryan et al. (2025), in-context preferences Singh et al. (2025), or a user-specific latent Poddar et al. (2024); (2) *personalized parameter* methods that add user-specific modules over a shared backbone Li et al. (2024); Chen et al. (2025); Bose et al. (2025), deliberately keeping the per-user parameter count small to remain tractable on tens of pairs; and (3) *meta-learning* methods that learn an adaptable initialization for few-shot inner-loop updates Cai et al. (2026), with related downstream work on heterogeneous direct alignment at the policy level Shirali et al. (2025); Zhang et al. (2024). All three paradigms attack per-user scarcity through architecture, sharing the implicit assumption that the only available signal is each user's labeled preference pairs. In contrast, PerSPL attacks per-user scarcity at the data level, turning each user's unpaired responses into pseudo-labeled supervision.

Semi-supervised preference learning under limited feedback. A separate line, semi-supervised reward modeling, leverages unlabeled data to mitigate annotation cost, but in the *homogeneous*-reward setting. SSRM He et al. (2024) self-trains on unlabeled *paired* responses, SPA Kim et al. (2024) iteratively generates responses from the current policy and DPO-self-labels them, and SSPO Lee et al. (2025) extends pseudo-labeling to *unpaired* SFT responses around a global threshold KDE-estimated from labeled data (see also Park et al. (2022)). Each method learns one population-wide reward model or policy and applies one global pseudo-labeling rule, with no mechanism for cross-user preference variation. PerSPL is purpose-built for personalized reward modeling with unpaired data, a problem these homogeneous-reward methods cannot address; we therefore omit them as empirical baselines.

3 Problem Setup

Personalized preference learning. For a user u, prompt x, and response y, a personalized reward model assigns a scalar score $r_\theta(x, y, u)$. Among responses to the same prompt, the model predicts that the user prefers the one with the higher score. Each labeled comparison (x, y^+, y^-) records that user u prefers y^+ to y^- for the same prompt. Writing $r^\pm = r_\theta(x, y^\pm, u)$, the Bradley–Terry model assigns the preference probability and loss

$$\Pr(y^+ \succ y^- \mid x, u) = \sigma(r^+ - r^-), \qquad \mathcal{L}_{\mathrm{BT}} = -\log \sigma(r^+ - r^-), \tag{1}$$

where $\sigma(t) = (1 + e^{-t})^{-1}$.

For each user u, $\mathcal{D}_L^u$ denotes the labeled comparisons available for that user. For a seen user, these comparisons are available during joint training. For a previously unseen user, they form the small support set used at adaptation, and evaluation uses disjoint comparisons. Thus, we study few-shot personalization, where every user provides some preference feedback, rather than zero-shot personalization. All users also share an unpaired pool $\mathcal{D}_U$ of prompt–response examples (x, y). Each example contains one response to a prompt and has neither a preference label nor a user identity. We focus on the regime in which this shared response pool is much larger than the labeled comparison set available for any individual user. Our goal is to use the shared pool to improve personalized ranking for both seen and unseen users without collecting additional user feedback. Because an unpaired example has no alternative response, using it requires interpreting its score relative to responses from other prompts.

Cross-prompt score ambiguity. The BT loss constrains the score difference between two responses to the same prompt. It does not constrain their common score location. For any prompt–user offset $c(x, u)$, the transformed reward function

$$r'(x, y, u) = r_\theta(x, y, u) + c(x, u) \tag{2}$$

produces the same preference probability and BT loss. This invariance is harmless for same-prompt ranking because the offset cancels, but it changes how a score compares with responses from other prompts and whether it crosses a pseudo-labeling threshold. The pairwise objective therefore does not identify the cross-prompt score coordinate required for single-response pseudo-labeling.

The usual held-out metric cannot reveal this ambiguity. For held-out comparison i, let $r_i^{\pm} = r_\theta(x_i, y_i^{\pm}, u)$. PairAcc measures whether the preferred response beats its paired alternative, whereas GlobalAcc diagnoses the cross-prompt ordering relevant to single-response pseudo-labeling:

$$\mathrm{PairAcc}_u = \Pr(r_i^+ > r_i^-), \qquad \mathrm{GlobalAcc}_u = \Pr(r_i^+ > r_j^-), \tag{3}$$

where i and j are independently sampled held-out comparisons for the same user. PairAcc evaluates the target same-prompt ranking task. GlobalAcc is a diagnostic of whether preferred and rejected responses remain ordered across prompts. It requires comparability within each user, not across users.

For example, consider two prompts with reward scores

$$x_a : (r_a^+, r_a^-) = (101, 100), \qquad x_b : (r_b^+, r_b^-) = (1, 0).$$

Both same-prompt decisions are correct, so PairAcc is 1, but the comparison $r_b^+ = 1 < r_a^- = 100$ fails and GlobalAcc is $3/4$. Centering each pair around zero maps both score pairs to $(0.5, -0.5)$, preserving PairAcc and yielding GlobalAcc $= 1$ in this example. This contrast motivates the centering mechanism in Section 4.1.

4 PERSPL: PERSONALIZED SEMI-SUPERVISED PREFERENCE LEARNING

PerSPL turns the shared unpaired pool $\mathcal{D}_U$ into user-specific training data through cross-prompt score centering, per-user calibration and pseudo-labeling, and iterative self-training. Figure 1 shows their interaction, and Appendix A.2 gives the full training procedure.

4.1 PAIR-MEAN SCORE CENTERING

The Problem Setup shows that pairwise supervision leaves the common score location of each labeled pair unidentified. Using the rewards r^+ and r^- defined in the Problem Setup, the BT loss depends only on their difference. PerSPL supplies a training signal for their mean with

$$\mathcal{L}_{\mathrm{reg}} = \mathbb{E}\left[\left(\frac{r^+ + r^-}{2}\right)^2\right], \tag{4}$$

where the expectation is over users and their labeled comparisons. This penalty acts directly on pair centers rather than output-coordinate margins. Because both are produced by the same reward model, training can still change them jointly. Appendix B characterizes this score-space role through cross-prompt risk, an illustrative Gaussian model, a one-sided fixed-budget analysis, and real-data diagnostics.

4.2 PER-USER CALIBRATION AND PSEUDO-LABELING

With this centering signal, PerSPL learns how each user's reward scores relate to their observed preferences. For each labeled pair in $\mathcal{D}_L^u$, we assign target 1 to the chosen-response reward and target 0 to the rejected-response reward. We fit a non-decreasing isotonic regressor $f_u : \mathbb{R} \to [0, 1]$ to these labeled rewards with the Pool-Adjacent-Violators algorithm Zadrozny & Elkan (2002); Niculescu-Mizil & Caruana (2005). Thus, $f_u(r)$ is a monotone empirical estimate of the probability that a score r appears on the chosen side of user u's comparisons. The fit is a stop-gradient target and also defines the per-user score anchor

$$\delta_u = \inf\{s : f_u(s) \geq \tfrac{1}{2}\}. \tag{5}$$

For an unpaired response, write $r = r_\theta(x, y, u)$ and choose $\tau \in (1/2, 1)$. PerSPL assigns $z = 1$ if $f_u(r) > \tau$, assigns $z = 0$ if $f_u(r) < 1 - \tau$, and otherwise abstains. The retained triples (x, y, z) form $\mathcal{D}_P^u$, the pseudo-labeled set for user u. Their average pointwise loss is

$$\mathcal{L}_{\text{unlabel}} = -\mathbb{E}[z \log \sigma(r - \delta_u) + (1 - z) \log \sigma(\delta_u - r)], \tag{6}$$

where the expectation is over users and their admitted responses. Only confident responses contribute to this loss; all original comparisons remain in the labeled objective.

4.3 Iterative self-training

Because pseudo-labels depend on the current reward model, PerSPL alternates between training the model and updating the pseudo-labels. Within each outer iteration, every $\mathcal{D}_P^u$ remains fixed while the model trains for a fixed number of inner epochs using

$$\mathcal{L}^{(k)} = \begin{cases} \mathcal{L}_{\text{label}} + \lambda_r \mathcal{L}_{\text{reg}}, & k = 1, \\ \alpha \mathcal{L}_{\text{label}} + (1 - \alpha)\mathcal{L}_{\text{unlabel}} + \lambda_r \mathcal{L}_{\text{reg}}, & k > 1, \end{cases} \qquad \alpha \in [0, 1], \tag{7}$$

where $\mathcal{L}_{\text{label}}$ is the average BT loss from Eq. (1) over the labeled comparisons. The first iteration uses only labeled comparisons to produce the initial reward model and calibrators; subsequent iterations combine them with the current pseudo-labeled set.

After each inner epoch, PerSPL refits f_u and δ_u using only the current rewards on $\mathcal{D}_L^u$, so the calibration fit itself uses only labeled examples. At the end of each outer iteration, PerSPL applies $f_u \circ r_\theta$ to $\mathcal{D}_U$ and replaces every $\mathcal{D}_P^u$ using the confidence rule above. The reward model is randomly reinitialized before the next iteration. Algorithm 1 specifies the minibatch sampling, update order, and complete schedule.

5 Experiments

Our primary evaluation compares PerSPL with a full suite of baselines on three benchmarks spanning synthetic and real responses and simulated and real users. We then use ablations and controlled studies to examine the contribution of unlabeled training, its interaction with pair-mean centering, and its behavior under different label budgets, preference noise, and response-distribution shift. Appendix C.2 provides an additional focused evaluation on Chatbot Arena.

5.1 Datasets

The three primary benchmarks separate response realism from user realism. **PersonalLLM** Zollo et al. (2024) uses synthetic responses and Dirichlet-sampled simulated users. **HelpSteer2** Wang et al. (2024) uses human-rated responses with simulated users defined by weights over rating attributes. **Reddit TLDR** Stiennon et al. (2020) contains real users and human preferences on summaries. PersonalLLM uses $\alpha_{\text{div}} \in \{0.001, 0.01, 0.1\}$, while HelpSteer2 uses $\alpha_{\text{div}} \in \{0.01, 0.1\}$. Appendix A.1 gives the complete construction and split definitions for both the primary suite and the additional Chatbot Arena evaluation.

5.2 Baselines and Evaluation Protocol

Baselines and architecture. In the primary suite, we compare against a homogeneous Bradley–Terry reward model (**BT**), personalized-input methods (**GPO**, **VPL**), personalized-parameter methods (**PAL**, **LoRe**), and the meta-learning method **MRM** with its RPO objective Zhao et al. (2023); Poddar et al. (2024); Chen et al. (2025); Bose et al. (2025); Cai et al. (2026). Each baseline uses its released implementation and a benchmark-specific configuration. PerSPL uses the same reward architecture as LoRe, providing an architecture-controlled comparison between the two methods. The architecture maps frozen pre-trained reward representations to a shared low-dimensional basis and combines that basis using user-specific weights. We optimize its trainable reward-head parameters with Adam. The default PerSPL configuration uses $\lambda_r = 0.01$ across benchmarks; ablations explicitly mark deviations. Complete benchmark- and stage-specific configurations are provided in the supplementary material.

Table 1: Full baseline comparison on PersonalLLM, HelpSteer2, and Reddit TLDR. We report pairwise preference accuracy (%) on seen and unseen users as mean ± std over five matched seeds. PersonalLLM and HelpSteer2 use α_{div}=0.1; Reddit TLDR uses 100 or 150 labeled examples per seen user. Best per column is in bold.

	PersonalLLM		**HelpSteer2**		**Reddit TLDR**			
	α_{div}=0.1		α_{div}=0.1		100 examples/seen user		150 examples/seen user	
Method	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
Skywork V2	81.6±1.6	82.4±1.4	41.6±0.3	41.4±0.2	64.4±0.7	64.5±0.7	64.4±0.7	64.5±0.7
BT	88.8±3.2	89.0±3.8	65.3±3.1	65.2±3.4	67.8±1.1	67.8±0.8	68.1±1.1	68.2±0.8
VPL	89.6±1.2	89.8±1.2	61.4±0.9	61.2±1.2	67.6±1.1	67.7±1.0	67.7±1.2	67.9±1.0
GPO	89.1±6.7	89.6±6.8	68.1±1.5	68.0±1.5	68.0±1.1	68.0±1.2	68.5±1.1	68.6±1.2
PAL	84.7±1.4	85.2±1.5	57.2±1.5	56.9±1.3	65.8±1.0	65.9±1.1	66.3±1.2	66.7±0.9
LoRe	94.6±0.5	94.0±0.9	69.6±0.8	70.1±1.0	68.1±1.1	68.6±1.2	68.5±1.0	68.8±1.1
MRM	91.6±0.6	92.0±0.5	70.7±0.3	70.9±0.4	70.0±1.0	69.8±0.6	70.0±1.1	69.6±0.7
PerSPL	**95.7±0.2**	**95.5±0.2**	**71.6±0.3**	**71.2±0.8**	**70.7±1.2**	**70.1±0.6**	**70.9±0.9**	**70.3±0.7**

Table 2: PerSPL minus the baseline with the highest overall mean in each setting, measured in accuracy percentage points with paired-bootstrap 95% confidence intervals over five matched seeds.

Setting	Seen	Unseen	Overall
HelpSteer2 $\alpha_{\text{div}} = 0.1$	+0.91 [0.62, 1.21]	+0.35 [0.04, 0.65]	+0.62 [0.34, 0.93]
HelpSteer2 $\alpha_{\text{div}} = 0.01$	+1.22 [0.57, 2.01]	+0.37 [−0.43, 1.18]	+0.81 [0.11, 1.62]
Reddit, 100/seen	+0.78 [0.28, 1.35]	+0.38 [0.15, 0.60]	+0.56 [0.39, 0.76]
Reddit, 150/seen	+0.91 [0.31, 1.51]	+0.74 [0.40, 1.08]	+0.81 [0.48, 1.05]

Evaluation protocol. Seen users are trained jointly, whereas each unseen user is adapted only from its designated support set. All test comparisons are held out. To ensure consistent comparisons, methods use the same splits, labeled budgets, frozen response representations, and seed identities. The supervised baselines receive the same labeled comparisons, while PerSPL additionally receives $\mathcal{D}_U$, whose responses carry neither user identities nor preference labels. We report within-prompt pairwise accuracy separately for seen and unseen users. Unless noted, results are mean ± standard deviation over five matched seeds, while the controlled shift study uses ten. Table 2 additionally reports paired-bootstrap 95% confidence intervals obtained from 20,000 resamples of the matched seed-level differences; further details are provided in Appendix C.3.

Compute. The primary three-benchmark suite used a mix of eight NVIDIA A100 80GB and five RTX A5000 24GB GPUs and required approximately seven hours of wall-clock time across its runs. This estimate excludes the focused Chatbot Arena evaluation and the distribution-shift analysis.

5.3 Main Results

Table 1 summarizes the full baseline comparison on PersonalLLM, HelpSteer2, and Reddit TLDR.

Among the methods evaluated in each setting, PerSPL has the highest mean in every seen- and unseen-user column. On PersonalLLM, it reaches 95.6 ± 0.1 overall, compared with 94.3 ± 0.7 for LoRe. On HelpSteer2 and Reddit TLDR, PerSPL has higher mean seen- and unseen-user accuracy than the strongest baseline in every reported setting. The Reddit TLDR result extends this pattern to real users and human preferences in summarization. Complete results across all diversity and label-budget settings appear in Appendix C.1. In an additional focused evaluation on Chatbot Arena, PerSPL also achieves higher mean seen- and unseen-user accuracy than LoRe and MRM; full results and paired intervals are provided in Appendix C.2.

To assess seed-level uncertainty, Table 2 reports absolute accuracy differences between PerSPL and the baseline with the highest overall mean in each setting. The intervals exclude zero for every seen-user and overall comparison and for three of the four unseen-user comparisons.

Table 3: Component ablations on HelpSteer2 ($\alpha_{\text{div}} \in \{0.1, 0.01\}$) and Reddit TLDR (100 or 150 labeled examples per seen user), forming a regularization $\times$ unpaired-training factorial comparison. Each variant changes only λ_r, K, or both, in both training phases; all other hyperparameters equal those of the full model. We report seen and unseen accuracy (%), mean $\pm$ std over five seeds. Best per column in bold.

	HelpSteer2				**Reddit TLDR**			
	α_{div}=0.1		α_{div}=0.01		100 examples/seen user		150 examples/seen user	
Objective	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen
PerSPL ($\mathcal{L}_{\text{label}} + \mathcal{L}_{\text{unlabel}} + \mathcal{L}_{\text{reg}}$)	**71.6±0.3**	**71.2±0.8**	**66.0±0.4**	**65.0±0.7**	**70.7±1.2**	**70.1±0.6**	**70.9±0.9**	**70.3±0.7**
$\mathcal{L}_{\text{label}} + \mathcal{L}_{\text{unlabel}}$ (λ_r=0)	66.5±0.5	69.7±0.9	62.5±1.2	63.2±0.5	63.7±1.1	68.6±0.7	62.5±2.1	62.5±1.4
$\mathcal{L}_{\text{label}} + \mathcal{L}_{\text{reg}}$ (K=1)	70.9±0.6	71.0±0.5	65.5±0.4	64.6±0.8	69.4±1.4	69.5±1.0	69.1±1.0	**70.3±0.7**
$\mathcal{L}_{\text{label}}$ only (λ_r=0, K=1)	70.9±0.6	71.1±0.7	65.4±0.3	64.5±0.9	69.5±1.3	69.7±1.1	69.0±1.3	69.6±0.7

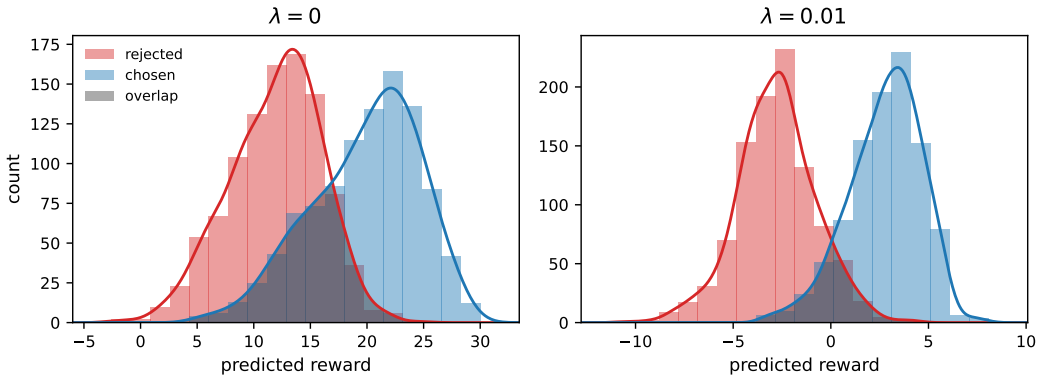

Figure 2: Reward distributions for an illustrative user. Without pair-mean regularization ($\lambda_r = 0$), the chosen and rejected distributions overlap substantially. With $\lambda_r = 0.01$, they show less overlap. Appendix B provides the corresponding score-space analysis and regularization-path diagnostics.

5.4 Ablations and Analysis

Component ablations. Table 3 reports a 2×2 factorial comparison of pair-mean regularization and unpaired training in all four HelpSteer2 and Reddit TLDR settings. Each variant keeps the configuration of the full model and changes only λ_r, K, or both, in both training phases. With unpaired training off, adding the regularizer changes seen accuracy by between -0.1 and $+0.1$ points. Without regularization, adding unpaired training changes seen accuracy by -6.5 to -2.9 points; with regularization, it changes seen accuracy by $+0.5$ to $+1.8$ points. The resulting interactions range from $+3.4$ to $+8.3$ points. For seen users, these contrasts show that centering changes the observed contribution of unpaired training from negative to positive. The large drop after removing regularization therefore reflects the interaction between centering and unpaired training rather than a standalone regularization gain. Appendix C.3 reports paired intervals for the component comparisons.

Centering diagnostics. Figure 2 illustrates how pair-mean centering changes the chosen and rejected reward distributions for one user. Its direct score-space action is on pair centers; shared parameters determine the learned centers and margins jointly. Appendix B characterizes this geometry and the labeled-only regularization path. Moderate centering reduces within-user pair-mean variation and cross-prompt loss while preserving held-out pairwise fit.

Noise and label-budget sweeps. We compare PerSPL with the strongest baseline, MRM, while varying preference noise and the labeled budgets of seen and unseen users on HelpSteer2 ($\alpha_{\text{div}} = 0.01$, Fig. 3).

Preference noise. We apply a label flip ratio from 0 to 0.5 uniformly across users (Fig. 3a). Both methods become less accurate as the noise increases, while PerSPL remains ahead throughout the

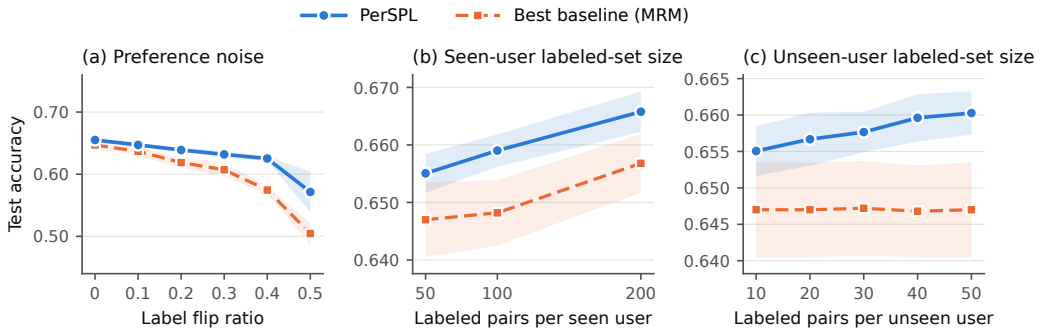

Figure 3: Performance under preference noise and varying label budgets on HelpSteer2 ($\alpha_{\text{div}} = 0.01$). PerSPL is compared with the strongest baseline, MRM. *(a)* Test accuracy as the label flip ratio increases. *(b)* Test accuracy with 50–200 labeled pairs per seen user. *(c)* Test accuracy with 10–50 labeled pairs per unseen user while fixing the seen-user budget at 50. Shaded bands show ± 1 standard deviation over five seeds.

tested range. The gap is largest at high noise, reaching 5.07 points at a 0.4 flip ratio and 6.69 points at 0.5.

Labeled pairs per seen user. Across 50, 100, and 200 labeled pairs per seen user (Fig. 3b), PerSPL improves as more labels become available and remains 0.81–1.08 points above MRM. The gain therefore persists across the displayed seen-user label budgets.

Labeled pairs per unseen user. Holding the seen-user budget fixed at 50 pairs and varying the unseen-user support set from 10 to 50 (Fig. 3c), PerSPL improves from 65.51% to 66.03%. Its gain over MRM increases from 0.81 to 1.33 points, supporting transfer to new users across the tested adaptation budgets.

Hyperparameter sensitivity. A controlled one-at-a-time sweep of the iso-confidence threshold τ, learning rate η, and labeled-loss weight α on HelpSteer2 ($\alpha_{\text{div}} = 0.01$) shows that accuracy is stable across the tested τ and α values. The selected learning rate $\eta = 0.01$ is also stable to nearby choices. Full sweep curves and quantitative comparisons are provided in Appendix C.4.

5.5 Controlled Response-distribution Shift

We test whether unlabeled responses remain useful when their generators differ from those represented in the labeled comparisons. Clean-label PersonalLLM is close to ceiling (Table 1), leaving little room to measure the additional contribution of unlabeled data. We therefore use a fixed 0.2 symmetric label-flip rate to create a non-saturated setting. The labeled data and flip rate remain fixed across all conditions and for the labeled-only reference, so label noise changes the task difficulty without changing the response-shift comparison.

We vary only the generator composition of the unlabeled pool, which follows $(1-\rho)P_{\text{in}} + \rho P_{\text{out}}$. The six in-domain generators and two held-out generators are disjoint, and the labeled and unpaired pools do not overlap. We compare PerSPL with the same regularized labeled-only model over ten matched seeds. Because this experiment uses corrupted labels, its absolute accuracies are not directly comparable with the clean-label PersonalLLM row in Table 1.

Across the tested mixtures, PerSPL's overall gain over the labeled-only reference ranges from 0.66 to 3.21 points (Table 4), and every paired interval is above zero. The overall benefit therefore persists even when the unlabeled pool consists entirely of the tested held-out generators. This evidence is specific to the controlled generator shift studied here. Appendix C.3 reports the generator identities and seen/unseen breakdown.

Table 4: Controlled generator shift on PersonalLLM with a 0.2 symmetric label-flip rate (overall accuracy, ten matched seeds). The matched regularized labeled-only reference is 81.49 ± 0.79.

OOD fraction ρ	PerSPL	Gain over labeled-only [95% CI]
0	84.70 ± 1.09	$+3.21$ [2.58, 3.82]
0.25	84.55 ± 1.02	$+3.07$ [2.31, 3.73]
0.50	83.63 ± 1.12	$+2.14$ [1.32, 2.90]
0.75	82.71 ± 0.89	$+1.22$ [0.57, 1.84]
1	82.15 ± 0.67	$+0.66$ [0.10, 1.18]

6 Conclusion

We introduced PerSPL to turn a shared pool of responses without user or preference labels into user-specific supervision. The method combines pair-mean score centering, per-user isotonic calibration, confidence filtering, and iterative self-training. Across three primary benchmarks with a full suite of baselines, PerSPL has the highest mean accuracy for both seen and unseen users. Paired-bootstrap intervals support its overall gains in every evaluated HelpSteer2 and Reddit TLDR setting. A focused evaluation on Chatbot Arena provides additional evidence on real-user open-domain preferences against LoRe and MRM.

The controlled generator-shift study further defines the scope of the unlabeled-data benefit. Across all tested mixtures, the overall gain over regularized labeled-only training ranges from 0.66 to 3.21 points, with every paired interval above zero. This evidence applies to the tested generators. Per-user calibration requires a labeled anchor, so the current setting studies few-shot rather than zero-shot personalization. Active selection and shift-aware calibration are natural next steps.

AI use statement

In preparing this revision, we used generative AI tools to help interpret experimental results and reviewer feedback, run diagnostic scripts, restructure and edit the manuscript, redraw figures, and check LaTeX formatting. The tools did not generate datasets. The authors verified the reported measurements and take responsibility for the final text, claims, and artifacts in this submission.

Ethics statement

This work studies personalization, which may better represent heterogeneous preferences but can also amplify filter bubbles, targeted persuasion, or privacy risks if user models are deployed without safeguards. We use existing public benchmarks and collect no new human-subject data. Deployment would require consent, access controls, and audits for disparate impact and misuse.

Reproducibility statement

The method objective and training procedure are specified in Section 4 and Algorithm 1. Appendix A.1 documents data construction and label budgets; Appendices C.1, C.2, and C.3 report complete results, uncertainty, and additional analyses. Code and configuration files are included in the supplementary material.

REFERENCES

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Michiel Bakker, Martin Chadwick, Hannah Sheahan, Michael Tessler, Lucy Campbell-Gillingham, Jan Balaguer, Nat McAleese, Amelia Glaese, John Aslanides, Matt Botvinick, et al. Fine-tuning language models to find agreement among humans with diverse preferences. *Advances in neural information processing systems*, 35:38176–38189, 2022.

Avinandan Bose, Zhihan Xiong, Yuejie Chi, Simon Shaolei Du, Lin Xiao, and Maryam Fazel. Lore: Personalizing llms via low-rank reward modeling. *arXiv preprint arXiv:2504.14439*, 2025.

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.

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

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference. *arXiv preprint arXiv:2403.04132*, 2024.

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.

Yijiang River Dong, Tiancheng Hu, Yinhong Liu, Ahmet Üstün, and Nigel Collier. When personalization meets reality: A multi-faceted analysis of personalized preference learning. *arXiv preprint arXiv:2502.19158*, 2025.

Esin Durmus, Karina Nguyen, Thomas I Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, et al. Towards measuring the representation of subjective global opinions in language models. *arXiv preprint arXiv:2306.16388*, 2023.

Yifei He, Haoxiang Wang, Ziyan Jiang, Alexandros Papangelis, and Han Zhao. Semi-supervised reward modeling via iterative self-training. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7365–7377, 2024.

Dongyoung Kim, Kimin Lee, Jinwoo Shin, and Jaehyung Kim. Spread preference annotation: Direct preference judgment for efficient llm alignment. *arXiv preprint arXiv:2406.04412*, 2024.

Hannah R Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, et al. The prism alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. *Advances in Neural Information Processing Systems*, 37:105236–105344, 2024.

Seonggyun Lee, Sungjun Lim, Seojin Park, Soeun Cheon, and Kyungwoo Song. Semi-supervised preference optimization with limited feedback. *arXiv preprint arXiv:2511.00040*, 2025.

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

Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pp. 625–632, 2005.

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.

Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. *arXiv preprint arXiv:2203.10050*, 2022.

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 Barr 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.

Ali Shirali, Arash Nasr-Esfahany, Abdullah Alomar, Parsa Mirtaheri, Rediet Abebe, and Ariel Procaccia. Direct alignment with heterogeneous preferences. *arXiv preprint arXiv:2502.16320*, 2025.

Anikait Singh, Sheryl Hsu, Kyle Hsu, Eric Mitchell, Stefano Ermon, Tatsunori Hashimoto, Archit Sharma, and Chelsea Finn. Fspo: Few-shot optimization of synthetic preferences personalizes to real users. *arXiv preprint arXiv:2502.19312*, 2025.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh N Sreedhar, and Oleksii Kuchaiev. Helpsteer 2: Open-source dataset for training top-performing reward models. *Advances in Neural Information Processing Systems*, 37:1474–1501, 2024.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699, 2002.

Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. Beyond bradley-terry models: A general preference model for language model alignment. *arXiv preprint arXiv:2410.02197*, 2024.

Siyan Zhao, John Dang, and Aditya Grover. Group preference optimization: Few-shot alignment of large language models. *arXiv preprint arXiv:2310.11523*, 2023.

Thomas P Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. Personalllm: Tailoring llms to individual preferences. *arXiv preprint arXiv:2409.20296*, 2024.

A Experimental Setup and Reproducibility Details

This appendix documents the dataset construction, data splits, and complete PerSPL training procedure.

A.1 Dataset Construction Details

PersonalLLM. PersonalLLM Zollo et al. (2024) provides a controllable synthetic benchmark for personalized preference learning. Each user is constructed by sampling a Dirichlet(α_{div}) weight vector over 10 reward heads from RewardBench, and pairwise preferences are generated by the corresponding weighted reward. We use 100 seen and 100 unseen users at three diversity levels $\alpha_{\text{div}} \in \{0.001, 0.01, 0.1\}$, where smaller α_{div} yields sharper, more disagreeing user weights and thus a harder personalization regime. The unpaired pool $\mathcal{D}_U$ is built from the source-dataset response candidates with their chosen and rejected labels removed, so no preference signal leaks from $\mathcal{D}_U$ into training.

Table 5: Dataset summary. PersonalLLM and HelpSteer2 use simulated users; Reddit TLDR and Chatbot Arena use real users and human preferences. Label budgets distinguish histories of seen users from support for unseen users.

Dataset	Responses	Users	# users	Labeled pairs (seen/unseen)	$\lvert\mathcal{D}_U\rvert$
PersonalLLM	synthetic	simulated	200	$45/9$	5,000
HelpSteer2	real	simulated	100	$50/10$–50	50,000
Reddit TLDR	real	real	40	100 or $150/50$	149,632
Chatbot Arena	real	real	38	$25/12$	training-pool remainder

Reddit TLDR. Reddit TLDR Stiennon et al. (2020) provides real human preferences on summary quality and serves as our fully real benchmark. Following the MRM benchmark protocol Cai et al. (2026), we use 20 seen and 20 unseen users, with 100 or 150 labeled pairs per seen user and 50 support pairs per unseen user. Evaluation comparisons are disjoint from these support sets. The unpaired pool $\mathcal{D}_U$ consists of the remaining users' summaries with their preference labels stripped, so the unlabeled signal is drawn from the same response distribution as the labeled set without overlapping preferences.

HelpSteer2. HelpSteer2 Wang et al. (2024) is the real-human multi-attribute counterpart to PersonalLLM, replacing LLM-judge reward heads with human attribute ratings. The dataset supplies $\sim$7K prompt–response pairs annotated by humans on five attributes (helpfulness, correctness, coherence, complexity, verbosity) on a 0–4 scale. We construct personalized users by sampling Dirichlet(α_{div}) weight vectors over the five attributes; each user's preference between two responses is determined by their attribute-weighted score. We use 50 seen and 50 unseen users, 50 labeled pairs per seen user, and 10–50 support pairs per unseen user. Remaining prompt–response pairs (with attribute scores stripped) form $\mathcal{D}_U$. We sweep $\alpha_{\text{div}} \in \{0.01, 0.1\}$.

Chatbot Arena. We use the active-user subset of Chatbot Arena distributed with the Personalized Benchmarking evaluation Dong et al. (2025); Chiang et al. (2024). After removing ties, we split each user's ordered comparisons into an 80% training pool and a held-out 20% test set and retain the 38 users with at least 25 training comparisons. Seen users receive 25 labeled pairs and unseen users 12; all methods use identical splits and budgets. The held-out test set is used only for evaluation.

A.2 PERSPL TRAINING PROCEDURE

The full PerSPL training procedure is given in Algorithm 1.

Algorithm 1 PerSPL: Personalized Semi-supervised Preference Learning

Require: Labeled sets $\{\mathcal{D}_L^u\}_{u\in\mathcal{U}}$, shared unpaired set $\mathcal{D}_U$, reward model r_θ, confidence threshold $\tau > 1/2$, weights λ_r and α, outer iterations K, inner epochs E

Initialize $\mathcal{D}_P^u \leftarrow \emptyset$ for every $u \in \mathcal{U}$
for $k = 1, \dots, K$ **do**
  Reinitialize $\theta \sim \mathrm{Init}(\cdot)$
  for epoch $e = 1, \dots, E$ **do**
    for each mini-batch step t **do**
      Sample $\mathcal{U}_t \subset \mathcal{U}$ and $\mathcal{B}_L^u \subset \mathcal{D}_L^u$ for each $u \in \mathcal{U}_t$
      Compute $\mathcal{L}_{\mathrm{label}}$ and $\mathcal{L}_{\mathrm{reg}}$ on $\{\mathcal{B}_L^u\}$ using Eqs. (1) and (4)
      if $k = 1$ **then**
        $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{label}} + \lambda_r \mathcal{L}_{\mathrm{reg}}$
      else
        Sample $\mathcal{B}_P^u \subset \mathcal{D}_P^u$ for each $u \in \mathcal{U}_t$ and compute $\mathcal{L}_{\mathrm{unlabel}}$ using Eq. (6)
        $\mathcal{L} \leftarrow \alpha\mathcal{L}_{\mathrm{label}} + (1-\alpha)\mathcal{L}_{\mathrm{unlabel}} + \lambda_r \mathcal{L}_{\mathrm{reg}}$
      end if
      Update θ with one Adam step on $\mathcal{L}$
    end for
    for each user $u \in \mathcal{U}$ **do**
      $\mathcal{C}_u \leftarrow \{(r_\theta(x, y^+, u), 1), (r_\theta(x, y^-, u), 0) : (x, y^+, y^-) \in \mathcal{D}_L^u\}$
      Fit isotonic regressor f_u on $\mathcal{C}_u$ and set $\delta_u \leftarrow \inf\{s : f_u(s) \geq 1/2\}$
    end for
  end for
  for each user $u \in \mathcal{U}$ **do**
    $\mathcal{D}_P^u \leftarrow \emptyset$
    for each $(x, y) \in \mathcal{D}_U$ **do**
      $q \leftarrow f_u(r_\theta(x, y, u))$
      if $q > \tau$ **then**
        $\mathcal{D}_P^u \leftarrow \mathcal{D}_P^u \cup \{(x, y, 1)\}$
      else if $q < 1 - \tau$ **then**
        $\mathcal{D}_P^u \leftarrow \mathcal{D}_P^u \cup \{(x, y, 0)\}$
      end if
    end for
  end for
end for

B THEORY AND DIAGNOSTICS FOR PAIR-MEAN CENTERING

This appendix characterizes the score-space role of pair-mean centering and its effect on labeled-only reward-head training. We first relate pair means to cross-prompt risk. We then use an idealized Gaussian setting to exhibit how center covariance can change the optimized scoring direction. Finally, we derive the one-sided response of a fixed-budget Adam checkpoint and measure it on real data. The factorial ablation in Section 5.4 separately measures how centering interacts with iterative unpaired training.

B.1 PAIR-MEAN GEOMETRY AND CROSS-PROMPT RISK

Fix one user and draw a labeled pair (X^+, X^-). Let

$$d = r(X^+) - r(X^-), \qquad m = \frac{r(X^+) + r(X^-)}{2}, \qquad \ell(z) = \log(1 + e^{-z}). \tag{8}$$

The paired risk is $L_{\mathrm{BT}} = \mathbb{E}\ell(d)$ and the pair-mean penalty is $R_M = \mathbb{E}m^2$. Let $(\widetilde{d}, \widetilde{m})$ be an independent pair from the same user. The cross-prompt logistic risk is

$$L_{\mathrm{cross}} = \mathbb{E}\ell\Big(r(X^+) - r(\widetilde{X}^-)\Big) = \mathbb{E}\ell\left(\frac{d + \widetilde{d}}{2} + m - \widetilde{m}\right). \tag{9}$$

This risk is a smooth surrogate for GlobalAcc. It compares a chosen response and a rejected response drawn from different pairs.

Proposition 1 (Cross-prompt control by pair means)**.** *If the scores have finite second moments, then*

$$L_{\mathrm{cross}} \le L_{\mathrm{BT}} + \frac{1}{4}\operatorname{Var}(m) \le L_{\mathrm{BT}} + \frac{1}{4}R_M. \tag{10}$$

The inequality allows arbitrary dependence between m and d within a pair.

Proof. Set $a = (d + \widetilde{d})/2$ and $z = m - \widetilde{m}$. Since $\ell'' \le 1/4$,

$$\ell(a + z) \le \ell(a) + \ell'(a)z + \frac{1}{8}z^2. \tag{11}$$

Exchanging the two sampled pairs keeps a fixed and changes the sign of z. The expected linear term is therefore zero, including when m and d are dependent within a pair. Convexity gives $\mathbb{E}\ell(a) \le L_{\mathrm{BT}}$, while independence across the two draws gives $\mathbb{E}z^2 = 2\operatorname{Var}(m)$. The second inequality follows from $\operatorname{Var}(m) \le \mathbb{E}m^2$. □

Why penalize the pair mean? The implemented second moment decomposes as

$$R_M = \mathbb{E}m^2 = \operatorname{Var}(m) + (\mathbb{E}m)^2. \tag{12}$$

It therefore controls the dispersion term in Eq. (10) and selects zero as the common score anchor. Writing a score pair as $(m + d/2, m - d/2)$ makes the direct action explicit: m^2 penalizes the common score component without directly penalizing the contrast d. It is therefore a simple pair-local quadratic signal for the score component absent from the BT objective. Shared model parameters still determine the learned centers and margins jointly.

The same coordinates relate centering to individual-response classification. For this fixed user, let $b \in \mathbb{R}$ be a score threshold shared across prompts. Draw Y uniformly from $\{-1, +1\}$, and set $X = X^+$ when $Y = +1$ and $X = X^-$ otherwise. Define the balanced single-response logistic risk

$$L_{\mathrm{single}}(b) = \mathbb{E}\ell(2Y(r(X) - b)) = \frac{1}{2}\mathbb{E}\big[\ell\big(2(r(X^+) - b)\big) + \ell\big(2(b - r(X^-))\big)\big]. \tag{13}$$

This population risk is an analysis surrogate for classifying one response as chosen or rejected; it is not an additional training objective. The factor two places its margin on the BT scale: when $m = b$, both logistic arguments equal d. In general, the two arguments are $d \pm 2(m - b)$. Convexity and the same smoothness argument used above therefore give

$$L_{\mathrm{BT}} \le L_{\mathrm{single}}(b) \le L_{\mathrm{BT}} + \frac{1}{2}\mathbb{E}(m - b)^2. \tag{14}$$

Equation (10) shows that pair-center dispersion controls the gap between within-pair BT risk and cross-prompt ranking risk. Equation (14) gives the analogous link to single-response classification around a shared threshold b. Thus the midpoint penalty controls a score component that within-pair BT comparisons do not explicitly constrain, but that both cross-prompt and single-response decisions use.

Scope of the score ambiguity. An additive prompt offset cancels in same-prompt selection, including best-of-N, but remains when a shared threshold compares responses across prompts. Pair-mean centering supplies an explicit location signal for this second setting, and GlobalAcc measures the resulting cross-prompt ordering.

B.2 An illustrative Gaussian model

We next use an idealized Gaussian setting to illustrate in closed form how center covariance can change the optimized scoring direction. It provides a concrete regime where the midpoint penalty improves single-response decisions. For a linear scorer $r_\beta(x) = \beta^\top x$, write

$$C = \frac{X^+ + X^-}{2}, \qquad D = X^+ - X^-, \qquad X^\pm = C \pm \frac{D}{2}. \tag{15}$$

Then $m = \beta^\top C$, so for any distribution with $\mathbb{E}\|C\|^2 < \infty$, the penalty defined above becomes

$$R_M(\beta) = \mathbb{E}(\beta^\top C)^2 = \beta^\top Q_c \beta, \qquad Q_c = \mathbb{E}[CC^\top]. \tag{16}$$

Thus pair-mean centering is a data-dependent quadratic penalty on sensitivity to directions shared within a pair.

Assumption 2 (Gaussian center–contrast model)**.** *For one user,*

$$C \sim \mathcal{N}(0, \Sigma_c), \qquad D \sim \mathcal{N}(s, \Sigma_d), \qquad C \perp D, \tag{17}$$

where C and D are independent, Σ_c and Σ_d are positive-definite covariance matrices, and $s \neq 0$ is the mean contrast.

The condition $\mathbb{E}[C] = 0$ isolates covariance-driven variation in pair centers. If $\mathbb{E}[C] = \mu_c \neq 0$, the implemented penalty instead uses $\Sigma_c + \mu_c\mu_c^\top$, and the local sign depends on the combined covariance and anchor geometry. Proposition 1 does not require either the Gaussian or mean-zero condition.

Let $L_{\mathrm{BT}}(\beta) = \mathbb{E}\ell(\beta^\top D)$. For a pair-mean coefficient $\lambda \geq 0$, define

$$\beta_\lambda = \arg\min_\beta \left\{ L_{\mathrm{BT}}(\beta) + \lambda\, \beta^\top \Sigma_c \beta \right\}, \tag{18}$$

and write β_{BT} for the solution at $\lambda = 0$. For the balanced single-response task defined above, the zero-threshold accuracy of the linear scorer is

$$\mathrm{Acc}_{\mathrm{single}}(\beta) = \Pr(Y\beta^\top X > 0). \tag{19}$$

Below, $v \parallel w$ means that v is a positive scalar multiple of w.

Theorem 3 (Local single-response accuracy improvement under the Gaussian model)**.** *Under Assumption 2, the direction maximizing* $\mathrm{Acc}_{\mathrm{single}}$ *is* $(4\Sigma_c + \Sigma_d)^{-1}s$*. The solutions in Eq.* (18) *satisfy*

$$\beta_\lambda \;\parallel\; \left(\Sigma_d + \frac{2\lambda}{J(\beta_\lambda)}\Sigma_c\right)^{-1} s, \qquad J(\beta) = \mathbb{E}\ell''(\beta^\top D) > 0. \tag{20}$$

If $\Sigma_c\Sigma_d^{-1}s$ is not proportional to s, then

$$\frac{d}{d\lambda}\,\mathrm{Acc}_{\mathrm{single}}(\beta_\lambda)\bigg|_{\lambda=0} > 0, \qquad 0 \leq L_{\mathrm{BT}}(\beta_\lambda) - L_{\mathrm{BT}}(\beta_{\mathrm{BT}}) = O(\lambda^2). \tag{21}$$

Thus, for small positive λ, the midpoint penalty yields a first-order gain in single-response accuracy while incurring at most a second-order increase in paired BT risk. The nonalignment condition distinguishes a directional correction from a pure rescaling. If $\Sigma_c\Sigma_d^{-1}s$ is proportional to s, the penalty preserves the BT direction to first order. Otherwise, it moves the optimized direction toward higher single-response accuracy.

Proof. Gaussian integration by parts gives

$$\nabla L_{\mathrm{BT}}(\beta) = J(\beta)\Sigma_d\beta - a(\beta)s, \qquad a(\beta) = -\mathbb{E}\ell'(\beta^\top D) > 0. \tag{22}$$

The first-order condition for Eq. (18) yields Eq. (20). Assumption 2 also gives

$$\mathrm{Acc}_{\mathrm{single}}(\beta) = \Phi\left(\frac{\beta^\top s}{\sqrt{\beta^\top(4\Sigma_c + \Sigma_d)\beta}}\right), \tag{23}$$

where Φ is the standard normal cumulative distribution function. Its maximizing direction follows from the generalized Rayleigh quotient.

It remains to determine the initial direction of change. Define $u = \Sigma_d^{-1/2}s$ and $B = \Sigma_d^{-1/2}\Sigma_c\Sigma_d^{-1/2}$. For $\beta(t) \parallel (\Sigma_d + t\Sigma_c)^{-1}s$, direct differentiation of the signal-to-noise ratio gives

$$\frac{d}{dt}\log\frac{\beta(t)^\top s}{\sqrt{\beta(t)^\top(4\Sigma_c + \Sigma_d)\beta(t)}}\bigg|_{t=0} = \frac{4\left[(u^\top u)(u^\top B^2 u) - (u^\top Bu)^2\right]}{(u^\top u)(u^\top u + 4u^\top Bu)}. \tag{24}$$

By Cauchy–Schwarz, this derivative is positive exactly when the stated nonalignment condition holds. Along the regularized path, $t = 2\lambda/J(\beta_\lambda)$ has derivative $2/J(\beta_{\mathrm{BT}}) > 0$ at zero, which proves the first statement in Eq. (21). The gradient of L_{BT} vanishes at β_{BT}, so its Taylor expansion along the same path is $O(\lambda^2)$. Nonnegativity follows because β_{BT} minimizes L_{BT}. □

Finite-sample illustration. We simulate the two-dimensional center–contrast model in Assumption 2 under three settings:

$$\text{unequal center variance:} \quad s = (1,1)^\top, \quad \Sigma_c = \mathrm{diag}(9,1), \quad \Sigma_d = I_2,$$

$$\text{misaligned covariance axes:} \quad s = (1,-0.5)^\top, \quad \Sigma_c = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}, \quad \Sigma_d = \begin{pmatrix} 1.5 & -0.2 \\ -0.2 & 0.8 \end{pmatrix},$$

$$\text{direction-preserving control:} \quad s = (0.8,-0.3)^\top, \quad \Sigma_c = 4I_2, \quad \Sigma_d = I_2.$$

The first two settings satisfy the nonalignment condition in Theorem 3; the control does not. For each $n \in \{50, 200, 1000\}$, we draw n independent training pairs (C_i, D_i) and fit

$$\widehat{\beta}_\lambda = \arg\min_\beta \left\{\frac{1}{n}\sum_{i=1}^n \ell(\beta^\top D_i) + \frac{\lambda}{n}\sum_{i=1}^n (\beta^\top C_i)^2\right\}.$$

We compare $\lambda = 0$ with the oracle population coefficient from Eq. (20), $\lambda^\star$ satisfying $2\lambda^\star/J(\beta_{\lambda^\star}) = 4$. By Eq. (20), this gives $\beta_{\lambda^\star} \parallel (4\Sigma_c + \Sigma_d)^{-1}s$, the single-response-optimal direction. Its values are 0.4688, 0.4748, and 0.4980 in the three settings, respectively, and are fixed before drawing the finite samples. We repeat each fit 200 times and evaluate $\widehat{\beta}_\lambda$ using the exact population accuracy in Eq. (19).

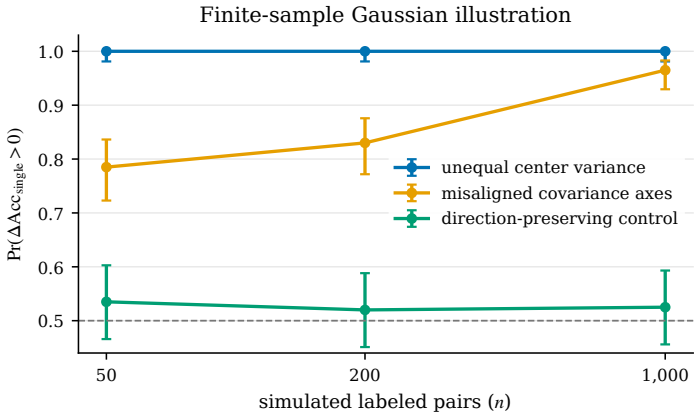

Figure 4: Finite-sample illustration of Theorem 3. The horizontal axis is the number n of simulated labeled pairs used to fit each model. The plot shows the fraction of 200 fits with $\Delta\mathrm{Acc}_{\mathrm{single}} > 0$, where the change compares $\lambda^\star$ with $\lambda = 0$; bars are Wilson 95% intervals. The direction-preserving setting is the null control.

Figure 4 measures how reliably the sign of the population improvement is recovered from a finite training sample. With unequal center variance, $\lambda^\star$ improves population single-response accuracy in all 200 repetitions at each training size. With misaligned covariance axes, the improvement probability rises from 0.785 at $n = 50$ to 0.965 at $n = 1000$. In the direction-preserving control, where the penalty cannot correct the population scoring direction, this probability remains between 0.520 and 0.535 and the mean accuracy change is at most 0.00018.

B.3 Turning on centering under a fixed training budget

The Gaussian calculation concerns an optimized population model. PerSPL instead returns the Adam checkpoint after a fixed training budget. We analyze the local dependence of this checkpoint on the centering coefficient. Fix the initialization, minibatch sequence, learning-rate schedule, and Adam hyperparameters. At step t, let

$$g_t(\lambda) = \nabla B_t(\theta_t(\lambda)) + \lambda \nabla R_t(\theta_t(\lambda)), \tag{25}$$

where B_t is the minibatch BT loss and R_t is the corresponding pair-mean penalty. With first and second moments μ_t and ν_t, Adam updates

$$\mu_{t+1} = \beta_1 \mu_t + (1-\beta_1) g_t, \qquad \nu_{t+1} = \beta_2 \nu_t + (1-\beta_2) g_t^{\odot 2}, \tag{26}$$

$$\bar{\mu}_{t+1} = \frac{\mu_{t+1}}{1-\beta_1^{t+1}}, \quad \bar{\nu}_{t+1} = \frac{\nu_{t+1}}{1-\beta_2^{t+1}},$$

$$\theta_{t+1} = \theta_t - \eta_t \frac{\bar{\mu}_{t+1}}{\sqrt{\bar{\nu}_{t+1}} + \epsilon}, \tag{27}$$

where $\odot$ denotes coordinatewise multiplication and the square root and division are also coordinatewise, $0 \le \beta_1 < 1$, $0 < \beta_2 < 1$, $\epsilon > 0$, and $\mu_0 = \nu_0 = 0$. A superscript 0 denotes the trajectory at $\lambda = 0$.

Proposition 4 (One-sided effect of turning on centering under Adam)**.** *Fix a finite training horizon K. Suppose B_t, R_t, and the evaluation target T are twice continuously differentiable near the unregularized parameter trajectory, and θ_0 is independent of λ. Define $H_t = \nabla^2 B_t(\theta_t^0)$ and $q_t = \nabla R_t(\theta_t^0)$. For any quantity $x_t(\lambda)$ along the training path, write*

$$\dot{x}_t = \left.\frac{dx_t(\lambda)}{d\lambda}\right|_{\lambda=0^+}.$$

These right derivatives exist. Starting from $\dot{\theta}_0 = \dot{\mu}_0 = \dot{\nu}_0 = 0$, they satisfy

$$\dot{g}_t = H_t \dot{\theta}_t + q_t, \tag{28}$$

$$\dot{\mu}_{t+1} = \beta_1 \dot{\mu}_t + (1-\beta_1)\dot{g}_t, \qquad \dot{\nu}_{t+1} = \beta_2 \dot{\nu}_t + 2(1-\beta_2) g_t^0 \odot \dot{g}_t.$$

The bias-corrected derivatives are $\dot{\bar{\mu}}_{t+1} = \dot{\mu}_{t+1}/(1-\beta_1^{t+1})$ and $\dot{\bar{\nu}}_{t+1} = \dot{\nu}_{t+1}/(1-\beta_2^{t+1})$. For each coordinate i, the parameter tangent obeys

$$\dot{\theta}_{t+1,i} = \dot{\theta}_{t,i} - \eta_t \begin{cases} \dfrac{\dot{\bar{\mu}}_{t+1,i}}{\sqrt{\bar{\nu}^0_{t+1,i}} + \epsilon} - \dfrac{\bar{\mu}^0_{t+1,i} \dot{\bar{\nu}}_{t+1,i}}{2\sqrt{\bar{\nu}^0_{t+1,i}}(\sqrt{\bar{\nu}^0_{t+1,i}} + \epsilon)^2}, & \bar{\nu}^0_{t+1,i} > 0, \\ \dfrac{\dot{\bar{\mu}}_{t+1,i}}{\epsilon}, & \bar{\nu}^0_{t+1,i} = 0, \end{cases} \tag{29}$$

As the centering coefficient approaches zero from the right, $\lambda \downarrow 0$,

$$T(\theta_K(\lambda)) = T(\theta_K^0) - \lambda A^{0+}_{K,\mathrm{Adam}} + o(\lambda), \tag{30}$$

$$A^{0+}_{K,\mathrm{Adam}} = -\nabla T(\theta_K^0)^\top \dot{\theta}_K. \tag{31}$$

Consequently, $A^{0+}_{K,\mathrm{Adam}} > 0$ is sufficient for the evaluation target to decrease for all sufficiently small $\lambda > 0$.

Proof. Suppose the state has a right derivative through step t. Differentiating Eq. (25) gives $\dot{g}_t = H_t \dot{\theta}_t + q_t$. Differentiating Eq. (26) gives the remaining recursions in Eq. (28).

It remains to differentiate the normalized Adam direction. If $\bar{\nu}^0_{t+1,i} > 0$, ordinary differentiation gives the first branch of Eq. (29). If $\bar{\nu}^0_{t+1,i} = 0$, the nonnegative second-moment recursion and its zero initialization imply that the unregularized gradients of coordinate i have been zero through step t. Its first moment is also zero. Smoothness and the induction hypothesis therefore give

$$\bar{\mu}_{t+1,i}(\lambda) = \lambda\dot{\bar{\mu}}_{t+1,i} + o(\lambda), \qquad \bar{\nu}_{t+1,i}(\lambda) = O(\lambda^2).$$

For $\lambda \downarrow 0$,

$$\frac{\bar{\mu}_{t+1,i}(\lambda)}{\sqrt{\bar{\nu}_{t+1,i}(\lambda)} + \epsilon} = \lambda\frac{\dot{\bar{\mu}}_{t+1,i}}{\epsilon} + o(\lambda),$$

which proves the second branch. Equation (27) now gives $\dot{\theta}_{t+1}$, completing the induction. Finally, a one-sided expansion of T gives $T(\theta_K(\lambda)) = T(\theta^0_K) + \lambda\nabla T(\theta^0_K)^\top\dot{\theta}_K + o(\lambda)$, which proves Eqs. (30)–(31). □

The recurrence traces each midpoint gradient q_t through the subsequent curvature and Adam moment updates. Equation (31) then gives the exact initial change in a smooth evaluation target. Its sign is determined by the data, representation, and training path. The matched one-sided runs below estimate this quantity for the Adam training path used by PerSPL.

B.4 Real-data checks

We isolate the effect of centering on labeled-only reward heads. We use HelpSteer2 with 50 seen users, rank five, and 50 labeled pairs per user, and Reddit TLDR with 20 seen users, rank two, and 150 labeled pairs per user. Every setting uses seeds 42–46 and last-epoch checkpoints.

Regularization path. We evaluate the same matched training path at two scales. To measure the initial effect of centering, we first train with Adam at $\lambda_r \in \{0, 10^{-8}, 10^{-7}\}$ while holding the initialization, minibatch sequence, and training horizon fixed. For a positive coefficient λ_r, the matched forward difference is

$$\widehat{A}^{0+}_{K,\mathrm{Adam}}(\lambda_r) = \frac{T(\theta_K(0)) - T(\theta_K(\lambda_r))}{\lambda_r}. \tag{32}$$

The held-out pairs are deterministically divided into two halves, with at most 256 pairs per user in each half. We set T to L_{cross} on the primary held-out half and repeat the measurement on the disjoint half. Figure 5 shows these slopes together with the corresponding changes in pair-mean variation. Error bars are paired 95% t intervals over five seeds; faint lines show individual seeds.

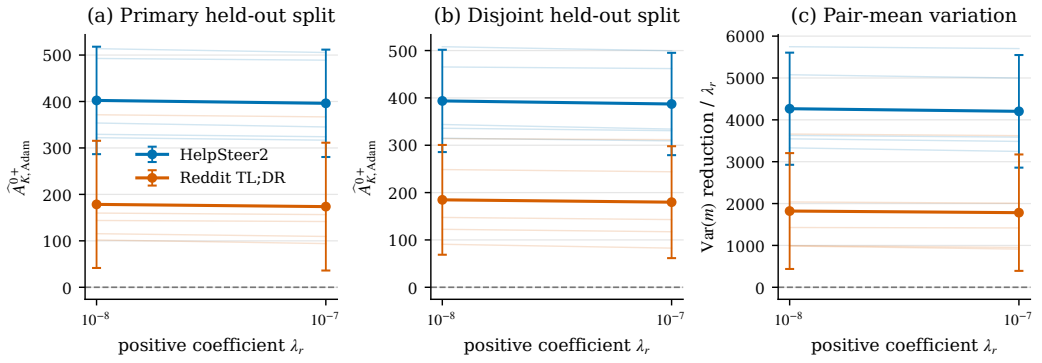

Figure 5: Matched one-sided differences at $\lambda_r = 0$ for the actual fixed-budget Adam training path. Panels (a) and (b) estimate $\widehat{A}^{0+}_{K,\mathrm{Adam}}$ on two disjoint held-out splits. Panel (c) reports the reduction in held-out within-user pair-mean variance. Positive values denote improvement.

The primary held-out alignment is positive for every seed, dataset, and positive coefficient. At $\lambda_r = 10^{-8}$, $\widehat{A}^{0+}_{K,\mathrm{Adam}}$ is 402.3 ± 115.7 on HelpSteer2 and 178.5 ± 136.9 on Reddit. The disjoint-half estimates are 393.6 ± 108.0 and 184.7 ± 115.9, respectively. Increasing the coefficient to 10^{-7}

gives primary-half estimates of 396.1 ± 115.6 and 173.7 ± 137.6. Thus the mean primary-half slopes change by 1.6% and 2.8% across the two coefficients. Held-out $\mathrm{Var}(m)$ also decreases for every seed and coefficient.

These slopes characterize the neighborhood of zero. To measure finite changes, we extend the same matched path to $\lambda_r \in \{0, 10^{-4}, 10^{-3}, 3\times10^{-3}, 10^{-2}, 3\times10^{-2}, 10^{-1}, 1\}$. Held-out pairs are split deterministically into the same two halves. Error bars in Figure 6 report paired 95% t intervals over the five seeds; faint lines show the individual seeds.

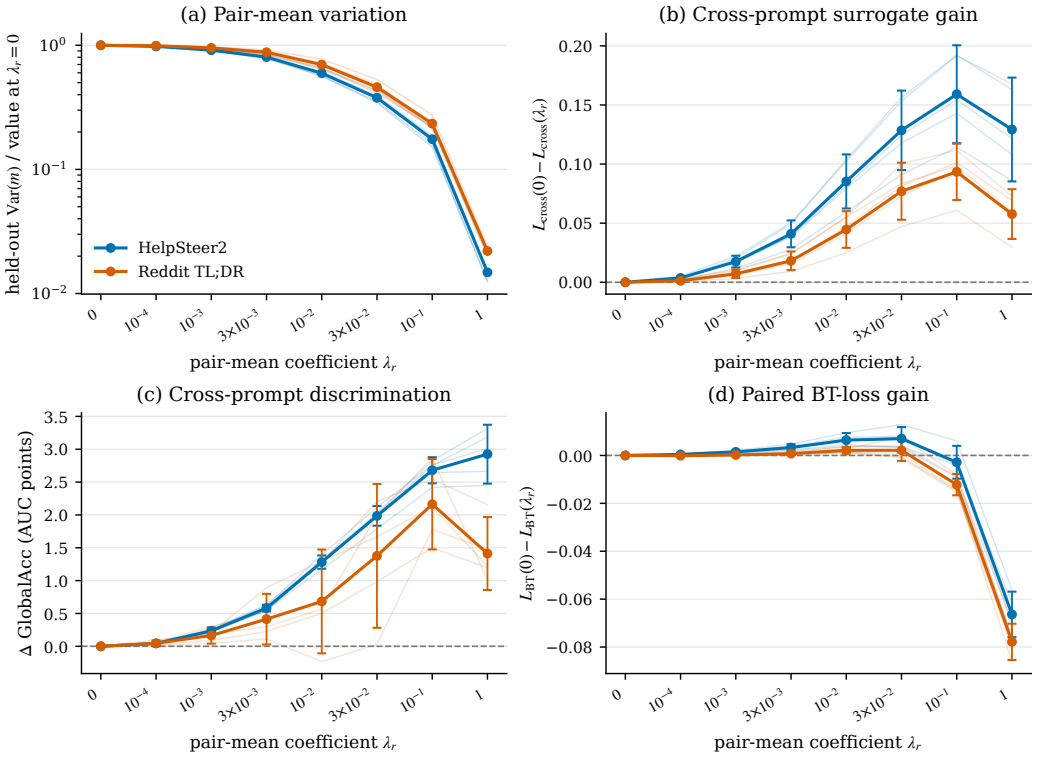

Figure 6: Real-data regularization paths for labeled-only reward heads. The held-out within-user pair-mean variance $\mathrm{Var}(m)$ in (a) is normalized within seed; the thick line is its geometric mean. Panels (b)–(d) show paired changes from $\lambda_r = 0$ with mean and 95% intervals over five seeds. Positive values denote improvement.

This diagnostic sweep isolates the effect of λ_r; all main comparisons use the fixed coefficient $\lambda_r = 0.01$ reported in Section 5.2.

At this coefficient, held-out PairAcc changes by -0.02 ± 0.14 points on HelpSteer2 and 0.00 ± 0.36 points on Reddit. This is consistent with the penalty acting on a cross-prompt score coordinate rather than directly changing same-prompt comparisons. Precision among the highest-confidence 10% of candidate pseudo-labels rises by 2.16 ± 0.29 points on HelpSteer2. The corresponding held-out-response proxy on Reddit changes by -1.18 ± 3.67 points, with an interval containing zero. All intervals are paired 95% intervals over five seeds.

Figure 6(a) shows the intended direct effect. At $\lambda_r = 0.1$, the held-out within-user pair-mean variance falls from 1.74 to 0.30 on HelpSteer2 and from 1.17 to 0.27 on Reddit. Figure 6(b) reports the held-out cross-prompt loss reduction $L_{\mathrm{cross}}(0) - L_{\mathrm{cross}}(\lambda_r)$ relative to the matched $\lambda_r = 0$ model. Positive values therefore indicate improved cross-prompt ranking. All 70 nonzero-λ_r seed comparisons are positive. At $\lambda_r = 0.1$, the mean reductions are 0.159 on HelpSteer2 and 0.093 on Reddit. Figure 6(c) shows the corresponding improvement in GlobalAcc, equivalently the AUC for classifying chosen versus rejected responses. At $\lambda_r = 0.1$, it rises by 2.68 ± 0.20 points on HelpSteer2 and 2.16 ± 0.69 points on Reddit, where the intervals are paired 95% intervals.

Figure 6(d) examines whether centering compromises held-out pairwise prediction. It reports $L_{\mathrm{BT}}(0) - L_{\mathrm{BT}}(\lambda_r)$, so positive values indicate lower loss and negative values indicate higher loss. At $\lambda_r = 0.01$, the mean gains are 0.0064 on HelpSteer2 and 0.0021 on Reddit. For comparison, the values at $\lambda_r = 1$ are -0.0664 and -0.0778. At $\lambda_r = 10^{-8}$, the per-unit BT-loss reductions are 35.4 ± 28.6 and -55.2 ± 39.9, respectively; after multiplication by λ_r, both are absolute changes below 10^{-6}. Both datasets therefore improve at the coefficient used in the paper.

Together, these results characterize pair-mean centering at three levels. The distribution-free bounds identify the score component used by cross-prompt and single-response decisions. The Gaussian model gives a concrete regime in which this component changes the optimized direction favorably. The Adam recurrence and matched runs establish the initial and finite effects along the actual labeled-only training path. Table 3 supplies the complementary full-procedure evidence: centering has little standalone effect on PairAcc, but changes the observed contribution of iterative unpaired training from negative to positive.

C Additional Results and Robustness Analyses

C.1 Full Per-dataset Results

We report full pairwise preference accuracy (%) on each primary benchmark for every method, including the per-dataset *Overall* column (per-seed average of seen and unseen accuracy) that is omitted from Table 1 for space. All numbers are mean ± std over five random seeds. The seen/unseen entries here match Table 1 on the matched settings (α_{div}=0.1 for PersonalLLM and HelpSteer2; 100 and 150 labeled examples/seen user for Reddit TLDR), and we additionally report the harder diversity levels ($\alpha_{\text{div}} \in \{0.001, 0.01\}$ for PersonalLLM, α_{div}=0.01 for HelpSteer2) and the Reddit baselines omitted from Section 5.3 for space (Skywork-Reward V1, SynthesizeMe variants, and MRM with a V1 backbone). Best per column is in bold; "% Improvement" is PerSPL's relative gain over the second-best method per column.

Table 6: Full PersonalLLM results across $\alpha_{\text{div}} \in \{0.001, 0.01, 0.1\}$ (45 pairs/seen user). Mean ± std over five seeds.

	α_{div}=0.001			α_{div}=0.01			α_{div}=0.1		
Method	Seen	Unseen	Overall	Seen	Unseen	Overall	Seen	Unseen	Overall
Skywork-Reward V2	76.9±1.1	76.8±2.6	76.8±1.7	77.8±2.4	76.1±1.9	77.0±1.7	81.6±1.6	82.4±1.4	82.0±1.2
BT	81.7±3.8	81.8±4.0	81.7±3.8	82.5±4.5	81.6±4.2	82.0±4.2	88.8±3.2	89.0±3.8	88.9±3.4
VPL	82.8±1.7	82.6±0.9	82.7±1.2	83.6±2.1	82.9±2.0	83.3±1.8	89.6±1.2	89.8±1.2	89.7±1.1
GPO	85.4±2.2	84.3±2.9	84.9±2.5	79.4±16.1	77.6±15.1	78.5±15.6	89.1±6.7	89.6±6.8	89.4±6.7
PAL	81.0±0.8	80.5±0.7	80.8±0.6	80.5±2.8	79.7±2.0	80.1±1.9	84.7±1.4	85.2±1.5	85.0±1.1
LoRe	92.7±1.0	91.2±0.7	91.9±0.6	93.0±0.8	90.8±1.1	91.9±0.9	94.6±0.5	94.0±0.9	94.3±0.7
MRM	86.0±0.4	86.1±1.1	86.0±0.7	87.0±1.1	85.3±0.7	86.1±0.7	91.6±0.6	92.0±0.5	91.8±0.4
PerSPL	**93.1±1.0**	**92.6±0.6**	**92.9±0.8**	**93.7±0.5**	**92.9±0.3**	**93.3±0.4**	**95.7±0.2**	**95.5±0.2**	**95.6±0.1**
% Improvement	0.4%	1.5%	1.1%	0.8%	2.3%	1.5%	1.2%	1.6%	1.4%

Table 7: Full HelpSteer2 results across $\alpha_{\text{div}} \in \{0.01, 0.1\}$ (50 pairs/seen user). Mean ± std over five seeds.

	α_{div}=0.01			α_{div}=0.1		
Method	Seen	Unseen	Overall	Seen	Unseen	Overall
Skywork-Reward V2	44.5±0.4	44.5±0.3	44.5±0.2	41.6±0.3	41.4±0.2	41.5±0.1
BT	59.8±2.7	59.5±2.8	59.7±2.8	65.3±3.1	65.2±3.4	65.2±3.3
VPL	57.7±0.9	57.4±1.2	57.6±1.0	61.4±0.9	61.2±1.2	61.3±1.1
GPO	63.2±1.4	62.7±1.3	62.9±1.3	68.1±1.5	68.0±1.5	68.1±1.5
PAL	55.4±0.4	55.6±1.2	55.5±0.6	57.2±1.5	56.9±1.3	57.0±1.4
LoRe	64.5±0.4	64.8±1.0	64.6±0.5	69.6±0.8	70.1±1.0	69.9±0.6
MRM	64.8±0.5	64.6±0.8	64.7±0.6	70.7±0.3	70.9±0.4	70.8±0.2
PerSPL	**66.0±0.4**	**65.0±0.7**	**65.5±0.4**	**71.6±0.3**	**71.2±0.8**	**71.4±0.5**
% Improvement	1.9%	0.3%	1.2%	1.3%	0.4%	0.8%

C.2 Additional Chatbot Arena Evaluation

We additionally evaluate PerSPL on Chatbot Arena, which contains real users and human preferences over open-domain chatbot responses Chiang et al. (2024); Dong et al. (2025). This focused evaluation compares PerSPL with LoRe and MRM. These are the strongest personalized baselines in the primary suite, with LoRe leading on PersonalLLM and MRM on HelpSteer2 and Reddit TLDR.

We use the active-user subset distributed with the Personalized Benchmarking evaluation. After removing ties, we split each user's ordered comparisons into an 80% training pool and a held-out 20% test set. We retain 38 users with at least 25 training comparisons. Seen users receive 25 labeled pairs, while unseen users receive 12 support pairs. Only the training pool supplies labeled, support, or unpaired responses, and all methods use the same splits, budgets, and seeds 42–46.

Table 8: Full Reddit TLDR results at 100 and 150 labeled examples/seen user. Mean ± std over five seeds. SynthesizeMe variants are inference-time-personalization methods (no in-domain training); MRM is reported with both Skywork-Reward V1 and V2 backbones.

	100 examples/seen user			150 examples/seen user		
Method	Seen	Unseen	Overall	Seen	Unseen	Overall
Skywork-Reward V1	62.6±0.5	62.6±0.9	62.6±0.9	62.6±0.5	62.6±0.9	62.6±0.9
Skywork-Reward V2	64.4±0.7	64.5±0.7	64.5±0.7	64.4±0.7	64.5±0.7	64.5±0.7
BT	67.8±1.1	67.8±0.8	67.8±0.9	68.1±1.1	68.2±0.8	68.2±0.8
VPL	67.6±1.1	67.7±1.0	67.7±0.9	67.7±1.2	67.9±1.0	67.8±1.0
GPO	68.0±1.1	68.0±1.2	68.0±1.0	68.5±1.1	68.6±1.2	68.6±0.6
PAL	65.8±1.0	65.9±1.1	65.9±0.7	66.3±1.2	66.7±0.9	66.5±1.0
LoRe	68.1±1.1	68.6±1.2	68.4±0.8	68.5±1.0	68.8±1.1	68.7±0.7
SynthesizeMe (ICL)	64.5±0.5	64.5±0.6	64.5±0.4	64.5±0.5	64.5±0.6	64.5±0.4
SynthesizeMe (FT)	64.5±0.5	64.4±0.6	64.4±0.5	64.5±0.5	64.4±0.6	64.4±0.5
MRM (Skywork-Reward V1)	68.2±1.3	68.5±1.6	68.4±1.1	68.6±1.0	68.6±1.0	68.6±0.8
MRM (Skywork-Reward V2)	70.0±1.0	69.8±0.6	69.9±0.4	70.0±1.1	69.6±0.7	69.8±0.4
PerSPL	**70.7±1.2**	**70.1±0.6**	**70.4±0.5**	**70.9±0.9**	**70.3±0.7**	**70.6±0.5**
% Improvement	1.0%	0.4%	0.7%	1.3%	1.0%	1.1%

Table 9: Focused Chatbot Arena comparison with LoRe and MRM. We report pairwise accuracy (%, mean ± std over five matched seeds).

Method	Seen	Unseen	Overall
LoRe	81.27 ± 2.08	78.38 ± 2.05	79.83 ± 1.00
MRM	80.69 ± 1.44	77.17 ± 1.58	78.93 ± 0.46
PerSPL	$\mathbf{82.99 \pm 2.97}$	$\mathbf{81.04 \pm 2.43}$	$\mathbf{82.01 \pm 0.64}$

PerSPL has the highest mean accuracy for both seen and unseen users. Relative to LoRe, the stronger baseline in all three columns, its paired gains are 1.72 points [0.14, 3.45] for seen users, 2.66 points [1.13, 4.28] for unseen users, and 2.19 points [0.79, 3.20] overall. The brackets give paired-bootstrap 95% confidence intervals from 20,000 resamples of the five matched seed differences. All three intervals exclude zero, extending the primary evidence to real-user preferences in open-domain chatbot interaction.

C.3 Additional Statistical and Robustness Evidence

Paired uncertainty. For each comparison in Table 2, we retain the seed pairing between PerSPL and the baseline with the highest overall mean in that setting and draw 20,000 bootstrap resamples of the five paired seed differences. The interval is the 2.5th–97.5th percentile range and summarizes variation over the executed matched seeds.

Factorial analysis. Across the four settings in Table 3, the regularizer-only change ranges from -0.1 to $+0.1$ seen-accuracy points. Conditional on centering, adding the unpaired loss changes seen accuracy by $+0.7$ [0.25, 1.07] and $+0.5$ [0.46, 0.63] on HelpSteer2 ($\alpha_{\text{div}} = 0.1$ and 0.01), and by $+1.3$ [0.54, 2.62] and $+1.8$ [0.79, 2.81] on Reddit TLDR (100 and 150 examples/seen user). Brackets give paired-bootstrap 95% intervals over five matched seeds, and all four seen-user intervals exclude zero. The corresponding unseen-user changes are $+0.2$ [-0.10, 0.47], $+0.4$ [-0.13, 0.95], $+0.6$ [0.06, 1.40], and 0.0 [-0.45, 0.48]. Without centering, adding unpaired training changes seen accuracy by -4.4, -2.9, -5.8, and -6.5 points in the same four settings.

Controlled generator shift. For PersonalLLM, the in-domain generators are GPT-4o, Claude 3 Opus, Claude 3 Sonnet, Llama 3 70B, Gemini Pro 1.5, and Mixtral 8x22B; Command R+ and GPT-4 Turbo are held out as the response-generator shift. Labeled and unpaired sets are disjoint, labels

have a 0.2 symmetric flip rate, and all rows use ten matched seeds. Table 10 gives the seen/unseen breakdown.

Table 10: Seen- and unseen-user breakdown for the controlled generator-shift study in Table 4. Overall results and paired gains are reported in the main table.

Setting	Seen	Unseen
Labeled only	80.31 ± 1.37	82.66 ± 0.67
$\rho = 0$	84.56 ± 1.32	84.83 ± 1.07
$\rho = 0.25$	84.44 ± 1.14	84.66 ± 1.14
$\rho = 0.50$	83.55 ± 1.15	83.71 ± 1.38
$\rho = 0.75$	82.58 ± 1.00	82.83 ± 1.19
$\rho = 1$	81.85 ± 0.86	82.45 ± 0.84

C.4 Hyperparameter Sensitivity Analysis

To complement the qualitative sensitivity discussion in Section 5.4, we report a controlled one-at-a-time sweep of the iso-confidence threshold τ, learning rate η, and labeled-loss weight α. The sweep is run on HelpSteer2 ($\alpha_{\text{div}} = 0.01$). When one hyperparameter is varied, the other two are held at the selected values $\tau = 0.6$, $\eta = 0.01$, and $\alpha = 0.7$. The training schedule and number of self-training iterations are also fixed. Each point is the overall test accuracy averaged across seen and unseen users over five matched seeds; shaded bands show $\pm$ one standard deviation.

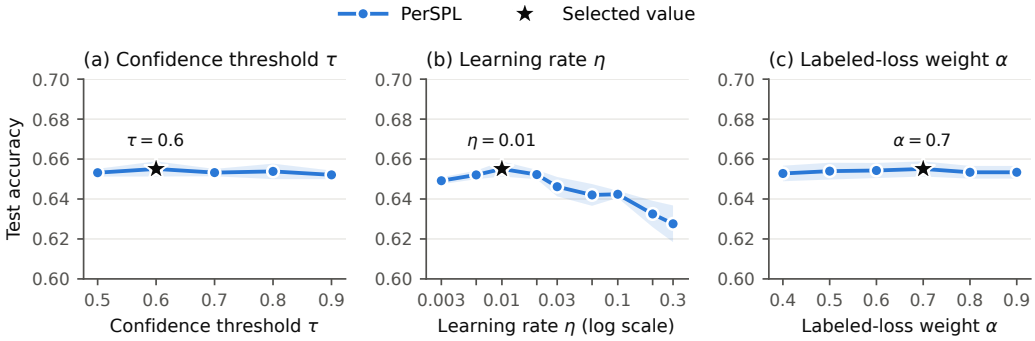

Figure 7: Hyperparameter sensitivity of PerSPL on HelpSteer2 ($\alpha_{\text{div}} = 0.01$). Stars mark the selected values ($\tau = 0.6$, $\eta = 0.01$, $\alpha = 0.7$). Overall accuracy changes little across the tested τ and α values and is stable near the selected learning rate.

Findings. The threshold and labeled-loss weight have little effect within the tested ranges (Fig. 7). Across $\tau \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, overall accuracy spans only 0.30 percentage points. In particular, $\tau = 0.5$, which effectively removes confidence-based abstention, is 0.19 points below the selected $\tau = 0.6$. Across $\alpha \in \{0.4, \ldots, 0.9\}$, the corresponding range is 0.23 points. The learning rate has a clearer effect. Values $\eta \in \{0.006, 0.01, 0.02\}$ remain within 0.30 points of one another, and the selected value is 2.75 points above $\eta = 0.3$. Thus, the confidence threshold, labeled/unlabeled loss weighting, and the learning-rate neighborhood used by PerSPL are stable in this setting.

D Limitations

Labeled anchors and information budgets. Per-user isotonic calibration requires labeled comparisons for every user to whom the method is applied. Zero-label cold start is therefore out of scope. Our seen-user histories and unseen-user support sets play different roles and must not be conflated: the former learn shared structure, while the latter adapt that structure to a target user. Mining user-generated content could reduce explicit annotation cost, but would introduce privacy and consent issues and would require an evaluation with a matched total information budget.

Selective pseudo-labeling. Confidence filtering intentionally abstains on ambiguous responses and retains all original labeled comparisons. It can nevertheless overrepresent easy or well-calibrated response regions. We test the all-response alternative and report accuracy, but do not yet measure subgroup-specific pseudo-label precision and coverage; deployment should audit both.

Distribution shift. The response-shift experiment holds labeled data fixed and varies a mixture of six in-domain and two held-out generators. It does not establish robustness to arbitrary semantic, demographic, adversarial, or temporal shift. Recalibration or abstention may be required when the unlabeled pool is far from the labeled support.

Architecture and scale. Experiments use a frozen reward-model representation with a learned low-rank reward basis. The findings may change with end-to-end fine-tuning or a much larger personalized backbone; a capacity-matched full-fine-tuning control is not included. The user populations contain at most 200 users, so computational and statistical behavior at production scale remains uncharacterized.

Language and tasks. The benchmarks are English-language and focus on instruction following, open-domain chatbot comparisons, and summarization. Cross-lingual transfer, multi-turn interaction, code, and long-horizon consequences are not evaluated.

E Broader Impacts

Positive impacts. PerSPL targets a core failure mode of homogeneous reward modeling: a single global reward fits the majority of annotators well and the long tail poorly. By training reward models that explicitly represent per-user preference structure under realistic label-scarcity, our work supports pluralistic alignment of language models, in which minority and underrepresented user populations are not averaged away by majority-vote-style aggregation. The semi-supervised component additionally reduces the labeled-data burden per user, lowering the annotation cost of including a new user in a personalized system and making fine-grained personalization more accessible to research and applications outside well-resourced organizations.

Potential negative impacts. Personalized preference models can in principle be misused: the same per-user accuracy that benefits alignment could also reinforce filter bubbles or enable targeted persuasion, and per-user reward heads encode fine-grained signal about individuals that raises privacy considerations. PerSPL is foundational research evaluated on public benchmarks and does not itself provide a deployment path for these harms; downstream practitioners adopting personalized reward models should adopt standard mitigations such as consent, access controls, and audits before deployment.

F Existing Assets, Repositories, and Licenses

We rely on three publicly available preference-learning benchmarks and on the released implementations of every baseline we compare against. All assets are used in accordance with their respective licenses, including the non-commercial restriction of LoRe Bose et al. (2025) (academic research use only). Repositories and licenses are listed as reported by each asset's official source at the time of writing.

Datasets. We use PersonalLLM Zollo et al. (2024) (`https://huggingface.co/datasets/namkoong-lab/PersonalLLM`; CC-BY-4.0), HelpSteer2 Wang et al. (2024) (`https://huggingface.co/datasets/nvidia/HelpSteer2`; CC-BY-4.0), and Reddit TLDR Stiennon et al. (2020) (`https://huggingface.co/datasets/openai/summarize_from_feedback`; CC-BY-4.0).

Baseline implementations. We use the official released code for GPO Zhao et al. (2023) (`https://github.com/jamqd/Group-Preference-Optimization`; MIT), VPL Poddar et al. (2024) (`https://github.com/WEIRDLabUW/vpl_llm/tree/release`; license not specified), PAL Chen et al. (2025) (`https://github.com/RamyaLab/`

pluralistic-alignment; Apache-2.0), LoRe Bose et al. (2025) (https://github.com/facebookresearch/LoRe; CC-BY-NC-4.0), and MRM Cai et al. (2026) (https://github.com/ModalityDance/MRM; MIT).

Pre-trained reward model. We extract reward-model embeddings from Skywork-Reward-V2-Llama-3.1-8B (https://huggingface.co/Skywork/Skywork-Reward-V2-Llama-3.1-8B; Llama 3.1 Community License), which is derived from meta-llama/Llama-3.1-8B-Instruct and inherits its license terms.