FEDPDPO: FEDERATED PERSONALIZED DIRECT PREFERENCE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct Preference Optimization (DPO) has emerged as an efficient alternative to reinforcement learning from human feedback (RLHF) for aligning large language models (LLMs), eliminating the need for explicit reward models and online policy optimization. However, applying DPO in federated settings introduces a fundamental challenge: different clients may exhibit heterogeneous preference distributions, making a single implicit reward function insufficient for personalized alignment. In this work, we identify that federated preference alignment requires jointly preserving shared linguistic knowledge and adapting to client-specific preference signals. To address this challenge, we propose `FedPDPO` (Federated Personalized Direct Preference Optimization), a personalized federated framework that decouples global representation learning from local preference calibration. Specifically, `FedPDPO` employs a LoRA-adapted global backbone shared across clients and personalized local preference heads that capture client-specific reward variations without sharing private preference data. Furthermore, we introduce **PDPO**, a personalized preference optimization objective that augments DPO's implicit reward with explicit client-specific reward corrections, providing a client-specific pathway for modeling preference differences. We provide a probabilistic interpretation of the reward correction mechanism based on the Bradley–Terry preference model. Extensive experiments on diverse preference alignment benchmarks demonstrate that `FedPDPO` consistently outperforms existing federated alignment and personalized FL baselines, on the reported held-out preference-pair accuracy across the evaluated client partitions.

INTRODUCTION

Federated learning (FL) enables collaborative model training without centralizing client data (McMahan et al., 2017; Li et al., 2020; Kairouz et al., 2021). For large language models (LLMs), however, collaboration raises a question beyond where training data reside: *whose preferences should the shared model follow?* Clients may benefit from common language capabilities while applying different criteria to judge the same response. This distinction matters in applications such as healthcare, finance, and education, where alignment must reflect the requirements of the deployment context (Ouyang et al., 2022; Bai et al., 2022; Singhal et al., 2023; Kasneci et al., 2023). Federated alignment must therefore support knowledge sharing without treating diverse preference criteria as interchangeable.

Two widely used approaches to preference alignment are reinforcement learning from human feedback (RLHF) and direct preference optimization (DPO). A common RLHF pipeline learns a reward model from human comparisons and optimizes the policy using proximal policy optimization (PPO) (Christiano et al., 2017; Schulman et al., 2017). Maintaining a reward model and generating online samples introduce substantial computational and optimization overhead. DPO (Rafailov et al., 2023) instead trains directly on offline preference pairs through a likelihood-based objective, avoiding a separately trained reward model and an online reinforcement learning stage. These properties make DPO an appealing foundation for federated alignment, but computational efficiency alone does not resolve disagreement among clients.

Recent work has extended preference alignment to federated settings through several mechanisms. FedRLHF (Fan et al., 2025) studies personalized RLHF, FedBis and FedBiscuit (Wu et al., 2025) use preference selectors, and FedDPO (Ye et al., 2024b) combines federated LoRA fine-tuning with DPO.

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.

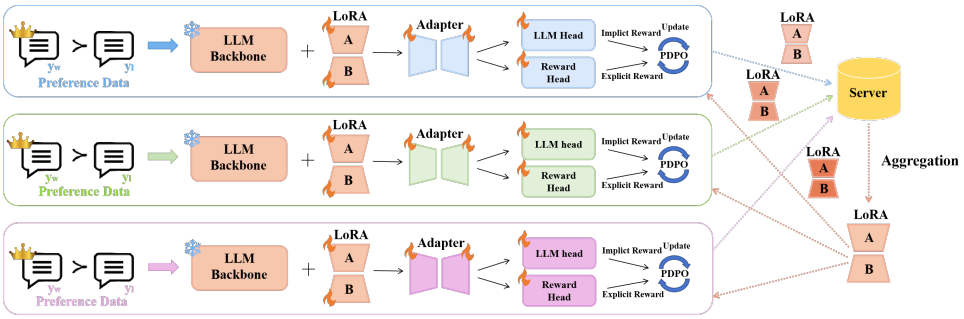


Figure 1: Overview of FedPDPO. Each client initializes its LoRA factors (A, B) from the server and retains two personalized heads: a language-model head h_1 for response likelihoods and a reward head h_2 for preference correction. Local training alternates between calibrating the heads and updating the LoRA factors under the PDPO objective. Only the LoRA factors are uploaded for weighted aggregation; both heads persist locally across rounds.

These approaches motivate a complementary question: how can an offline DPO-based framework retain client-specific preference signals while using them to train a shared representation? The central tension is that shared language capabilities need not imply shared judgments of response quality.

> *Clients can share the representations needed to understand a response while disagreeing on whether that response should be preferred.*

This tension concerns conditional preference orderings, not merely differences in the prompts that clients observe. For the same prompt x and responses (y_a, y_b), clients may assign different probabilities to $y_a \succ y_b$, or even favor opposite responses. DPO represents preference comparisons through differences in the implicit score $\beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$. When the policy and reference are client-independent, this score induces the same ordering for every client. Consequently, a shared policy must compromise when local orderings conflict, and reducing aggregate preference loss need not improve alignment for every client. This limitation arises from imposing a client-independent policy, rather than from DPO itself. Consistent with the broader difficulty of heterogeneous training, Figure 2 shows lower DPO performance under our non-IID setting than under IID training, although this comparison alone does not isolate preference disagreement from other distribution shifts.

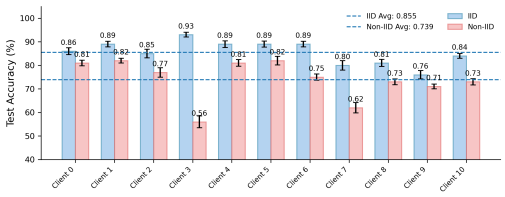


Figure 2: DPO performance under IID and non-IID client data distributions in our experimental setting.

Personalized federated learning offers a starting point by combining shared representations with local parameters (Tan et al., 2022). For preference alignment, however, the design question extends beyond *which parameters remain local* to *how local preferences shape the comparison signal used for training*. A personalized output head allows client-specific response likelihoods, but a DPO objective still expresses preferences entirely through policy-reference likelihood ratios. A local correction provides an additional pathway for fitting preference differences. Coupling this correction to backbone training allows local calibration to influence the representations learned through collaboration.

We introduce FedPDPO, a federated framework that couples shared representation learning with personalized preference calibration (Figure 1). Clients train a LoRA-adapted backbone while retaining a local language-model head and a local reward head. The language-model head personalizes response likelihoods, whereas the reward head supplies a client-specific correction to the preference margin. Our Personalized Direct Preference Optimization (PDPO) objective combines these signals with adaptive magnitude normalization, yielding a preference logit. The correction can reinforce or counteract the likelihood-based ordering while retaining offline pairwise optimization.

Crucially, calibration is integrated into shared training. Each client first updates its local heads with the backbone fixed, then updates its LoRA factors through both heads under the same PDPO objective. Only the LoRA factors are aggregated, while the local heads retain client-specific parameters across rounds. The resulting design personalizes both response generation and the comparison signal that shapes backbone adaptation: clients share representation parameters while maintaining distinct mechanisms for expressing and learning preferences.

In summary, **main contributions** are as follows:
• We propose `FedPDPO`, a personalized federated framework for offline LLM preference alignment, which decouples shared language representation learning from client-specific preference calibration through a LoRA-based global backbone and local preference heads.
• We introduce **PDPO**, a personalized extension of DPO that incorporates client-specific reward corrections into the implicit likelihood-based preference objective, enabling more robust alignment under heterogeneous preference distributions.
• We provide a probabilistic interpretation of the proposed reward correction mechanism under the Bradley–Terry preference model and conduct extensive experiments on homogeneous and heterogeneous preference benchmarks, demonstrating the effectiveness of FedPDPO over existing federated alignment and personalized FL baselines.

Related Work

Federated Fine-Tuning for LLMs. Federated fine-tuning adapts large language models to decentralized data (Sun et al., 2024b; Zhang et al., 2024; Yi et al., 2023; Qin et al., 2024; Bai et al., 2024). Parameter-efficient methods, including black-box fine-tuning and offsite-tuning (Sun et al., 2024a; Lin et al., 2023; Wu et al., 2024a; Kuang et al., 2024), reduce communication and computation costs but primarily target task adaptation rather than preference alignment. FedRLHF (Fan et al., 2025) supports personalized alignment with convergence guarantees but incurs additional overhead from separate reward models. FedDPO (Ye et al., 2024b) combines DPO with LoRA-based federated fine-tuning, avoiding explicit reward modeling. However, a shared implicit reward may inadequately capture heterogeneous client preferences, motivating client-specific preference calibration that preserves DPO's efficiency.

RLHF and DPO. RLHF aligns language models through supervised fine-tuning, reward modeling, and policy optimization, often with PPO (Christiano et al., 2017; Ouyang et al., 2022; Schulman et al., 2017). DPO instead optimizes offline preference pairs directly, avoiding a separately trained reward model and online policy optimization (Rafailov et al., 2023). Extensions include Step-DPO, DreamBooth-DPO, β-DPO, and Offset-DPO (Lai et al., 2024; Ayupov et al., 2025; Wu et al., 2024b; Amini et al., 2024). `FedPDPO` brings client-specific reward correction to federated DPO without training a separate full reward-model backbone. Relative to FedDPO (Ye et al., 2024a), it achieves higher reported held-out pairwise accuracy on the evaluated partitions; these experiments do not isolate conflicting preferences on shared prompts.

Personalized Federated Learning (PFL). PFL combines shared knowledge with client-specific models to address statistical heterogeneity (Tan et al., 2022). Representative methods use meta-learning (Per-FedAvg (Fallah et al., 2020)), regularization (FedProx and Ditto (Li et al., 2020; 2021)), personalized aggregation (FedAMP and FedALA (Huang et al., 2021; Zhang et al., 2023)), or local heads (FedPer and FedRep (Arivazhagan et al., 2019; Collins et al., 2021)). `FedPDPO` builds on the shared-representation, local-head paradigm: clients aggregate LoRA factors while retaining heads that personalize response likelihoods and preference comparisons.

Methodology

Problem Setting

We study personalized federated preference alignment of large language models (LLMs) across N clients. Client i holds a private dataset $\mathcal{D}_i = \{(x, y_w, y_l)\}$ sampled from a distribution P_i, where x is a prompt and y_w is preferred to y_l. Beyond differences in prompt and response distributions, clients may disagree on how the same responses should be ranked. To distinguish this preference

heterogeneity from distribution shift alone, let

$$q_i(x, y_a, y_b) := \Pr_i(y_a \succ y_b \mid x, y_a, y_b)$$

denote client i's conditional preference probability. Heterogeneous preference criteria allow $q_i(x, y_a, y_b) \neq q_j(x, y_a, y_b)$ on shared support; in particular, clients may favor opposite responses to the same prompt. Federated alignment faces a central tension: clients can share language representations while disagreeing on preference orderings. A client-independent model must compromise under such disagreement, so minimizing aggregate loss need not align each client effectively. We therefore combine shared representations with client-specific mechanisms for learning preference differences.

We instantiate this goal through a shared LoRA-adapted backbone and personalized local heads. Let W_0 denote the frozen pretrained backbone, $\omega = (A, B)$ the shared trainable LoRA factors, and $\phi_i = (h_{1,i}, h_{2,i})$ the personalized parameters of client i. Here, $h_{1,i}$ denotes the local policy pathway, including its bottleneck adapter and language-model output head, which defines the client's response likelihoods; $h_{2,i}$ denotes the local reward pathway and its reward head. Shared local adapters, if any, are included in the local parameter block ϕ_i. With W_0 omitted from the trainable parameter notation, the personalized model is $\Theta_i = (\omega, \phi_i)$. We optimize the coupled objective

$$\left(\omega^\star, \{\phi_i^\star\}_{i=1}^N\right) \in \underset{\omega, \{\phi_i\}_{i=1}^N}{\arg\min} \sum_{i=1}^N k_i \, \mathcal{L}_i^{\text{pref}}(\omega, \phi_i), k_i = \frac{|\mathcal{D}_i|}{\sum_{j=1}^N |\mathcal{D}_j|}, \tag{1}$$

where $\mathcal{L}_i^{\text{pref}}(\omega, \phi_i) := \mathbb{E}_{(x, y_w, y_l) \sim \widehat{P}_i} \left[\ell_{\text{PDPO}}(\omega, \phi_i; x, y_w, y_l)\right]$, and $\widehat{P}_i$ is the empirical distribution of $\mathcal{D}_i$. The personalized models are $\hat{\Theta}_i = (\omega^\star, \phi_i^\star)$. The shared variable ω couples local objectives, allowing clients to benefit from collaborative training without requiring identical preference criteria.

Following the shared-backbone, local-head principle of FedPer and FedRep (Arivazhagan et al., 2019; Collins et al., 2021), we personalize the preference comparison signal: the local language-model head defines the implicit likelihood-based margin, and the reward head provides a client-specific correction under PDPO. Each round, clients initialize LoRA factors $\omega_i = (A_i, B_i)$ from the server and upload their updates for aggregation, while retaining both heads locally. This couples collaborative representation learning with persistent personalization of response generation and preference calibration.

THE FEDPDPO FRAMEWORK

`FedPDPO` shares representations without imposing a common preference ordering. A client-independent policy must compromise when clients disagree, potentially limiting individual alignment. This constraint stems from policy sharing, not DPO itself. We couple *local preference calibration* with *collaborative representation learning*. Each client retains a language-model head $h_{1,i}$ for personalized response likelihoods and a reward head $h_{2,i}$ that corrects the implicit preference margin. Both pathways shape backbone updates through PDPO. The framework combines: (*i*) a calibrated objective integrating implicit and explicit preference signals, and (*ii*) alternating head and backbone updates with federated aggregation of LoRA factors.

PERSONALIZED DIRECT PREFERENCE OPTIMIZATION

Implicit preference signal. Let $\omega = (A, B)$ denote the LoRA factors and $\phi_i = (h_{1,i}, h_{2,i})$ the local heads. With the frozen pretrained backbone omitted from the notation, client i's policy is $\pi_{\omega, h_{1,i}}$. For a preference tuple (x, y_w, y_l), its reference-adjusted log-likelihood margin is

$$\Delta_{\text{logp},i} = \log \frac{\pi_{\omega, h_{1,i}}(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_{\omega, h_{1,i}}(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}, \tag{2}$$

where π_{ref} is a frozen reference policy. Standard DPO minimizes $-\mathbb{E}_{\mathcal{D}_i}[\log \sigma(\beta \Delta_{\text{logp},i})]$, where σ is the sigmoid function and $\beta > 0$ scales the preference logit. Although the local language-model head already personalizes this margin, its preference signal remains entirely determined by policy-reference likelihood ratios. We introduce an additional pathway for expressing client-specific preference differences.

Algorithm 1: `FedPDPO`: Federated Personalized Direct Preference Optimization

Require: Initial LoRA factors $\omega^0 = (A^0, B^0)$; local heads $\{\phi_i^0 = (h_{1,i}^0, h_{2,i}^0)\}_{i=1}^N$; datasets $\{\mathcal{D}_i\}_{i=1}^N$; learning rates η_h, η_w; communication rounds R.

Set $k_i \leftarrow |\mathcal{D}_i| / \sum_{j=1}^N |\mathcal{D}_j|$ for each client i.
for $r = 0, \ldots, R-1$ **do**
  Broadcast shared LoRA factors ω^r to all clients.
  for each client $i = 1, \ldots, N$ **in parallel do**
    Initialize $\omega_i^r \leftarrow \omega^r$; retain personalized heads ϕ_i^r.
    Calibrate local preferences (freeze ω_i^r):
      $\phi_i^{r+1} \leftarrow \phi_i^r - \eta_h \nabla_{\phi_i} \mathcal{L}_{\mathrm{PDPO},i}(\omega_i^r, \phi_i^r)$.
    Adapt shared representations (freeze ϕ_i^{r+1}):
      $\omega_i^{r+1} \leftarrow \omega_i^r - \eta_w \nabla_{\omega_i} \mathcal{L}_{\mathrm{PDPO},i}(\omega_i^r, \phi_i^{r+1})$.
    Upload ω_i^{r+1}; keep ϕ_i^{r+1} locally.
  end for
  Aggregate shared LoRA factors: $A^{r+1} \leftarrow \sum_{i=1}^N k_i A_i^{r+1}$, $B^{r+1} \leftarrow \sum_{i=1}^N k_i B_i^{r+1}$.
  Set $\omega^{r+1} \leftarrow (A^{r+1}, B^{r+1})$.
end for

Ensure: Personalized models $\{\hat{\Theta}_i = (\omega^R, \phi_i^R)\}_{i=1}^N$, with shared LoRA factors and locally retained heads.

Explicit preference correction. The local reward head operates on backbone representations and assigns a scalar score $r_{\omega,h_{2,i}}(x, y)$. Its pairwise correction is

$$\Delta_{r_i} = r_{\omega,h_{2,i}}(x, y_w) - r_{\omega,h_{2,i}}(x, y_l). \tag{3}$$

PDPO combines the two signals through $\Delta_{\mathrm{PDPO},i} = \Delta_{\mathrm{logp},i} + w_r s_i \Delta_{r_i}$, where $w_r \geq 0$ controls the strength of the correction. The client-specific scale $s_i = \frac{\mathbb{E}_{\mathcal{D}_i}[|\Delta_{\mathrm{logp},i}|]}{\mathbb{E}_{\mathcal{D}_i}[|\Delta_{r_i}|]+\epsilon}, \epsilon > 0$, adjusts for differences in the magnitudes of the two signals. The expectation specifies the intended client-level statistic; its estimator, update timing, and treatment in automatic differentiation must be matched to the implementation. All probability interpretations below condition on a fixed scale. The expectation specifies the intended client-level statistic; its estimator, update timing, and treatment in automatic differentiation must be matched to the implementation. All probability interpretations below condition on a fixed scale. Thus, w_r controls their relative contribution more directly than an unnormalized additive correction would. The resulting local objective is

$$\mathcal{L}_{\mathrm{PDPO},i}(\omega, \phi_i) = -\mathbb{E}_{(x,y_w,y_l)\sim\widehat{P}_i}\left[\log \sigma\left(\beta \Delta_{\mathrm{PDPO},i}\right)\right], \tag{4}$$

where $\widehat{P}_i$ is the empirical distribution of $\mathcal{D}_i$.

Calibration in preference log-odds. For a fixed local normalization scale, the combined margin defines the modeled preference probability $p_i(y_w \succ y_l \mid x) = \sigma(\beta \Delta_{\mathrm{PDPO},i})$, and hence

$$\log \frac{p_i(y_w \succ y_l \mid x)}{1 - p_i(y_w \succ y_l \mid x)} = \beta \Delta_{\mathrm{logp},i} + \beta w_r s_i \Delta_{r_i}. \tag{5}$$

The reward pathway therefore provides an additive correction in preference log-odds: a positive correction reinforces the likelihood-based ordering, whereas a negative correction attenuates or reverses it. This gives personalization a direct role in the pairwise comparison used for optimization. Setting $w_r = 0$ recovers the DPO loss for the same personalized policy parameterization. PDPO retains an offline pairwise training objective and requires neither online response sampling nor a separate PPO stage. Its additional reward head shares the adapted backbone with the policy, rather than introducing a separate full reward-model backbone.

Federated Optimization Process

We optimize the shared LoRA factors and local heads through alternating parameter-block updates. At communication round r, the server broadcasts $\omega^r = (A^r, B^r)$. Client i initializes $\omega_i^r \leftarrow \omega^r$ and retains its current local heads ϕ_i^r. For clarity, the following equations and Algorithm 1 display one gradient step per block; stochastic local optimization replaces the corresponding gradients with minibatch estimates.

Local preference calibration. With the LoRA factors fixed, client i updates both heads:

$$\phi_i^{r+1} \leftarrow \phi_i^r - \eta_h \nabla_{\phi_i} \mathcal{L}_{\mathrm{PDPO},i}(\omega_i^r, \phi_i^r). \tag{6}$$

This step adapts the policy likelihoods and reward correction to local comparisons before updating the shared representation parameters. **Shared representation learning.** The client then fixes the updated head parameters and optimizes its LoRA factors:

$$\omega_i^{r+1} \leftarrow \omega_i^r - \eta_w \nabla_{\omega_i} \mathcal{L}_{\mathrm{PDPO},i}(\omega_i^r, \phi_i^{r+1}). \tag{7}$$

Although the head parameters remain fixed in this step, gradients propagate through both heads to the backbone. Consequently, the shared representation update is informed by the client's calibrated preference signal, including both its likelihood-based margin and explicit reward correction. The local heads therefore participate in shaping collaborative learning rather than serving only as personalized readouts. **Server aggregation.** After local optimization, clients upload only their LoRA factors. The server performs data-size-weighted factor averaging: $A^{r+1} = \sum_{i=1}^{N} k_i A_i^{r+1}, B^{r+1} = \sum_{i=1}^{N} k_i B_i^{r+1}$, where $k_i = |\mathcal{D}_i| / \sum_j |\mathcal{D}_j|$. The aggregated factors initialize the next communication round, while ϕ_i^{r+1} remains on client i. This procedure combines shared backbone adaptation with persistent local preference parameters, allowing collaboration without averaging the personalized heads.

Addressing Preference Heterogeneity in Federated DPO

Federated preference alignment requires deciding not only *what to share*, but also *which preference signals should shape the shared parameters*. Clients can benefit from common language representations while disagreeing on the relative desirability of responses. The resulting challenge is therefore not simply to fit different data distributions, but to support different preference orderings within a collaboratively trained model. `FedPDPO` addresses this challenge at two complementary levels: the parameterization of personalized policies and the comparison signal used to optimize them.

Challenge 1: Sharing representations without imposing a common preference ordering. Preference heterogeneity can persist even for the same prompt and candidate responses:

$$\Pr_i(y_a \succ y_b \mid x, y_a, y_b) \neq \Pr_j(y_a \succ y_b \mid x, y_a, y_b).$$

A client-independent policy cannot represent conflicting preference orderings through a likelihood-based margin. This limitation stems from policy sharing, not DPO. `FedPDPO` aggregates LoRA factors (A, B) while retaining the language-model head $h_{1,i}$ and reward head $h_{2,i}$ locally. This enables representation sharing with client-specific response likelihoods and preference corrections.

Challenge 2: Making personalization shape the alignment signal. A personalized architecture alone does not specify how client-specific preference information should enter the optimization objective. With a DPO loss, the comparison signal is determined entirely by the policy-reference likelihood margin. PDPO augments this signal with a local reward correction:

$$\Delta_{\mathrm{PDPO},i} = \Delta_{\mathrm{logp},i} + w_r s_i \Delta_{r_i}.$$

The correction modifies preference log-odds, reinforcing, attenuating, or reversing the likelihood-based ordering. The scale s_i matches signal magnitudes, while w_r controls the correction strength, providing a trainable pathway beyond policy-reference likelihood ratios. Local calibration also shapes shared representation learning. After updating its heads, each client fixes their parameters and optimizes its LoRA factors through both the likelihood and reward pathways under PDPO. Thus, local heads retain client-specific preferences across rounds while guiding shared backbone adaptation. This coupling preserves offline pairwise optimization without online sampling or a separate PPO.

Theoretical Interpretation of PDPO

We analyze how PDPO modifies preference modeling and the training signal received by the policy. Our results establish a random-utility interpretation, characterize the preference discrepancies that the correction can represent, and quantify its effect on optimization. Throughout, we fix a client i and treat the normalization scale s_i as constant within a comparison model. Let $\lambda_i := w_r s_i, \qquad u_i(x, y) := \beta \log \frac{\pi_{\omega, h_{1,i}}(y|x)}{\pi_{\mathrm{ref}}(y|x)}$, where the policy and reference probabilities are positive. Define the corrected

utility $v_i(x, y) := u_i(x, y) + \beta\lambda_i r_{\omega,h_{2,i}}(x, y)$. For a response pair (y_a, y_b), write $a_i := u_i(x, y_a) - u_i(x, y_b), c_i := \beta\lambda_i \left[r_{\omega,h_{2,i}}(x, y_a) - r_{\omega,h_{2,i}}(x, y_b)\right]$. The PDPO comparison logit is therefore $a_i + c_i$.

Proposition 1 (Random-utility interpretation). Let $U_{i,a} = v_i(x, y_a) + \varepsilon_a$ and $U_{i,b} = v_i(x, y_b) + \varepsilon_b$, where $\varepsilon_a, \varepsilon_b$ are independent $\mathrm{Gumbel}(0, 1)$ random variables. Then $\Pr(U_{i,a} > U_{i,b}) = \sigma(a_i + c_i)$. Consequently, for an observed preferred–rejected pair (y_w, y_l), the negative log-likelihood is exactly $-\log\sigma\left(\beta[\Delta_{\mathrm{logp},i} + \lambda_i\Delta_{r_i}]\right)$, which is the per-example PDPO loss.

This interpretation places the correction in preference log-odds rather than treating it as an auxiliary loss. If $p_i^0 = \sigma(a_i)$ and $p_i^c = \sigma(a_i + c_i)$, then $\frac{p_i^c}{1-p_i^c} = \exp(c_i)\frac{p_i^0}{1-p_i^0}$. Thus, the local reward pathway multiplicatively adjusts the odds induced by the policy. Because the correction is a difference of response scores, it also preserves pairwise consistency: reversing the responses changes the sign of the logit and complements the preference probability.

Proposition 2 (Residual utility representation). Suppose client i's true preference probabilities admit a Bradley–Terry representation $q_i(x, y_a, y_b) = \sigma\big(v_i^\star(x, y_a) - v_i^\star(x, y_b)\big)$. Fix the policy utility u_i and assume $\lambda_i > 0$. If the reward-head function class contains $r_i^\star(x, y) = \frac{v_i^\star(x,y) - u_i(x,y)}{\beta\lambda_i} + C_i(x)$, where $C_i(x)$ is arbitrary, then PDPO exactly represents q_i.

Moreover, let $\ell(q, p) := -q\log p - (1 - q)\log(1 - p)$ denote the conditional population preference loss. For any pair with $q_i \in (0, 1)$, $\ell(q_i, p_i^0) - \ell(q_i, q_i) = \mathrm{KL}\left(\mathrm{Bern}(q_i) \,\|\, \mathrm{Bern}(p_i^0)\right) \geq 0$, with strict inequality whenever $p_i^0 \neq q_i$. Proposition 2 identifies the role of the reward head: it can represent the residual between the client's preference utility and the utility currently expressed by its policy. Under the stated realizability condition, this additional pathway removes conditional preference-modeling error that remains at a fixed policy. The conclusion concerns representational capacity; it does not assume that optimization necessarily recovers the residual utility.

Proposition 3 (Calibration-dependent gradient). For a preferred–rejected pair, let $\ell_i = -\log\sigma(a_i + c_i)$. Its partial derivative with respect to the policy logit is $\frac{\partial\ell_i}{\partial a_i} = -\sigma(-a_i - c_i)$. Relative to DPO at the same logit, the gradient magnitude is rescaled by $\frac{|\partial\ell_i/\partial a_i|}{|\partial[-\log\sigma(a_i)]/\partial a_i|} = \frac{1+\exp(a_i)}{1+\exp(a_i+c_i)}$. Hence, a positive correction reduces the pressure to increase the policy margin, whereas a negative correction increases it. For shared backbone parameters ω, $\nabla_\omega\ell_i = -\sigma(-a_i - c_i)\left(\nabla_\omega a_i + \nabla_\omega c_i\right)$.

Local calibration changes the comparison weight and, through the reward pathway, the backbone update direction. Freezing head parameters preserves this pathway because reward predictions depend on backbone representations. Local heads shape the LoRA factors aggregated by the server. **Scope.** The results establish a coherent comparison model, conditional expressivity at a fixed policy, and calibration-dependent backbone updates, but not improved policy-only generation: reward corrections may absorb residuals without correcting the policy. The combined margin can be interpreted as the likelihood ratio of a reward-reweighted policy $q_i(y \mid x) \propto \pi_i(y \mid x)\exp(\lambda_i r_i(x, y))$ when λ_i is response-independent and the normalizer exists (Appendix C); ordinary decoding from π_i need not sample from q_i. Direct policy-generation evaluation remains necessary for generation claims.

EXPERIMENTS

Evaluation scope. Tables 2 and 3 reproduce the reported test accuracy and loss values. The reported results do not specify whether the pairwise prediction uses the policy likelihood margin, the reward score, or their combination; consequently these values cannot be interpreted as policy-generation quality. Appendix G defines the distinct scores required for a reproducible evaluation.

Datasets. We evaluate `FedPDPO` in two settings. (*i*) *Homogeneous clients:* clients use data from the same source dataset. We consider two benchmarks: IMDB (Maas et al., 2011) for sentiment-based preference alignment and Code-Vulnerability-Security[1] for security-oriented code preferences. (*ii*) *Heterogeneous clients:* three clients use distinct datasets: WebGPT (Nakano et al., 2022), PyDPO[2], and UltraFeedback (Cui et al., 2023). This setting introduces variation in both data domains and preference supervision, allowing us to assess personalized alignment under cross-dataset heterogeneity.

[1] https://huggingface.co/datasets/CyberNative/Code_Vulnerability_Security_DPO

[2] https://huggingface.co/datasets/jondurbin/py-dpo-v0.1

Table 1: Test accuracy (%) under the homogeneous setting. Left: Reward-Margin Based Partition (IMDB). Right: Language-Based Partition (Code-Vulnerability-Security). Results are mean ± std over 5 runs.

Method	IMDB				Code-Vulnerability-Security			
	Client0	Client1	Client2	Avg	Client0	Client1	Client2	Avg
FedDPO	$95.54_{\pm 0.18}$	$83.19_{\pm 0.22}$	$69.47_{\pm 0.25}$	82.73	$98.21_{\pm 0.14}$	$93.45_{\pm 0.17}$	$88.80_{\pm 0.21}$	93.49
Per-FedAvg (+DPO)	$96.13_{\pm 0.21}$	$83.33_{\pm 0.25}$	$70.73_{\pm 0.30}$	83.40	$97.62_{\pm 0.18}$	$94.64_{\pm 0.21}$	$90.36_{\pm 0.27}$	94.21
FedAMP (+DPO)	$95.54_{\pm 0.20}$	$82.56_{\pm 0.27}$	$70.14_{\pm 0.33}$	82.75	$97.45_{\pm 0.19}$	$92.86_{\pm 0.23}$	$89.84_{\pm 0.30}$	93.38
FedPer (+DPO)	$95.24_{\pm 0.24}$	$80.60_{\pm 0.29}$	$69.15_{\pm 0.36}$	81.66	$98.21_{\pm 0.16}$	$95.24_{\pm 0.20}$	$92.19_{\pm 0.28}$	95.21
FedRep (+DPO)	$96.13_{\pm 0.22}$	$81.37_{\pm 0.28}$	$69.05_{\pm 0.32}$	82.18	$98.21_{\pm 0.15}$	$94.05_{\pm 0.22}$	$92.71_{\pm 0.25}$	95.00
FedPDPO	$\mathbf{98.21}_{\pm 0.16}$	$\mathbf{89.23}_{\pm 0.20}$	$\mathbf{74.80}_{\pm 0.19}$	**87.41**	$\mathbf{98.81}_{\pm 0.12}$	$\mathbf{98.21}_{\pm 0.15}$	$\mathbf{93.75}_{\pm 0.18}$	**96.92**

Model architectures. All experiments use Llama-2-7B as the backbone. FedPDPO shares LoRA factors across clients while keeping bottleneck adapters and output heads local. Each client uses a bottleneck adapter of width 256 with dropout 0.1, a language-model head for token-level likelihoods, and a reward head with hidden width 256 that maps mean-pooled representations to a scalar score.

Baselines. Within the FedDPO framework (Ye et al., 2024b), we compare five methods: FedAvg (McMahan et al., 2017) for weighted parameter averaging; Per-FedAvg (Fallah et al., 2020) for adaptation-oriented initialization; FedAMP (Huang et al., 2021) for attentive collaboration; and FedPer (Arivazhagan et al., 2019) and FedRep (Collins et al., 2021) for shared representations with personalized components. All baselines share the backbone, LoRA, and DPO objective, differing in aggregation and personalization. FedPDPO uses PDPO for local preference calibration.

Hyperparameter Settings. For all methods, we use AdamW as the local optimizer and adopt cosine learning rate decay within each local epoch. The learning rate is selected from $\{1 \times 10^{-4}, 5 \times 10^{-5}, 3 \times 10^{-5}, 8 \times 10^{-5}, 1 \times 10^{-3}\}$. The DPO temperature β is chosen from $\{0.1, 0.05, 0.01\}$, and the weight ratio w_r is linearly increased from 0.5 to 1.5 over communication rounds. All experiments were conducted on an NVIDIA GeForce RTX 3090 GPU with 24GB memory, with clients executed sequentially to simulate the federated learning environment, with each training run taking approximately 8-12 hours depending on the dataset size and number of communication rounds.

Experiments with Homogeneous Datasets

In the first experimental block, clients receive partitions of the same dataset, with statistical heterogeneity induced by dataset-specific non-IID splitting strategies.

Reward-margin-based partition. For IMDB, we rank samples by their precomputed reward margin (chosen minus rejected reward) in descending order and divide them into equal-sized contiguous shards. With three clients, Client0–Client2 receive high-, medium-, and low-margin subsets, with mean margins of approximately 4.12, 1.47, and 0.25, respectively. These subsets represent increasing preference ambiguity, with margin serving as a proxy for comparison difficulty. For ten clients, we apply the same procedure to obtain ten shards assigned to Client0–Client9 in descending margin order. Unless otherwise specified, we partition the training and test sets separately using this rule, preserving each client's relative margin profile across splits.

Programming-language-based partition. For Code-Vulnerability-Security, we construct non-IID partitions using programming-language identifiers. In the three-client setting, languages are grouped into Web & Scripting, Enterprise & Mobile, and Systems & HPC, assigned to Client0–Client2, respectively. This partition models clients specializing in different programming domains. In the eleven-client setting, each client receives a disjoint shard corresponding to one retained language. Each client's data are split into training and test sets with a fixed 90%/10% ratio.

Results under same-dataset heterogeneity. Tables 1 and 2 evaluate performance across partition strategies and client counts. On IMDB, FedPDPO exceeds FedDPO by 4.68 and 7.67 percentage points with three and ten clients, respectively, and outperforms the personalized baselines. On Code-Vulnerability-Security, the corresponding gains are 3.43 and 8.71 points with three and eleven clients. These results support the effectiveness of FedPDPO under both reward-margin and programming-domain heterogeneity, with larger gains over FedDPO in the evaluated settings with more clients.

Table 2: Test accuracy (%) under homogeneous and heterogeneous federated alignment settings. For homogeneous partitions, we report average accuracy across $K = 10$ clients on IMDB and $K = 11$ clients on Code-Vulnerability-Security. For the heterogeneous setting, three clients use distinct datasets; we report per-client accuracies and their unweighted average (Avg.). Error bars indicate standard deviations over 5 runs.

Method	Homogeneous		Heterogeneous			
	IMDB $K = 10$	Code-Vulnerability-Security $K = 11$	C1: WebGPT	C2: PyDPO $K = 3$	C3: UltraFeedback	Avg.
FedDPO	$76.45_{\pm 0.23}$	$84.09_{\pm 0.19}$	$52.92_{\pm 0.16}$	$85.53_{\pm 0.09}$	$66.45_{\pm 0.24}$	68.30
Per-FedAvg (+DPO)	$78.83_{\pm 0.22}$	$85.44_{\pm 0.19}$	$54.96_{\pm 0.23}$	$87.36_{\pm 0.18}$	$67.48_{\pm 0.27}$	69.93
FedAMP (+DPO)	$77.39_{\pm 0.24}$	$87.12_{\pm 0.17}$	$54.58_{\pm 0.21}$	$83.93_{\pm 0.22}$	$65.47_{\pm 0.30}$	67.99
FedPer (+DPO)	$79.95_{\pm 0.19}$	$86.37_{\pm 0.15}$	$55.38_{\pm 0.20}$	$88.42_{\pm 0.19}$	$68.02_{\pm 0.28}$	70.61
FedRep (+DPO)	$80.43_{\pm 0.26}$	$88.25_{\pm 0.13}$	$53.96_{\pm 0.22}$	$86.71_{\pm 0.21}$	$67.25_{\pm 0.29}$	69.31
FedPDPO	$\mathbf{84.12}_{\pm 0.14}$	$\mathbf{92.80}_{\pm 0.13}$	$\mathbf{58.92}_{\pm 0.14}$	$\mathbf{91.47}_{\pm 0.12}$	$\mathbf{69.92}_{\pm 0.19}$	**73.44**

Experiments with Heterogeneous Datasets

We further evaluate cross-dataset heterogeneity by assigning WebGPT, PyDPO, and UltraFeedback to Client0–Client2, respectively. These datasets emphasize question answering, code generation, and instruction following, introducing variation in both task domains and preference supervision.

Results under cross-dataset heterogeneity. Table 2 shows that FedPDPO achieves 73.44% average accuracy, exceeding FedDPO by 5.14 percentage points and the strongest personalized baseline, FedPer, by 2.83 points. Its gains over FedAMP and FedRep are 5.45 and 4.13 points, respectively. The improvements extend across all three clients: FedPDPO surpasses the strongest baseline on WebGPT, PyDPO, and UltraFeedback by 3.54, 3.05, and 1.90 points, respectively. These results support the effectiveness of combining shared representation learning with local preference calibration across diverse data sources.

Ablation Study

Component ablation. Table 3 evaluates four variants on IMDB with ten clients: (A1) standard DPO without reward correction; (A2) PDPO without bottleneck adapters; (A3) PDPO without the exclusion rule, retaining samples despite consecutive misclassifications; and (A4) the complete FedPDPO framework. A1 yields the lowest accuracy (80.96%), while A2 provides only a marginal improvement, indicating that removing the adapters reduces accuracy in this configuration. A3 has lower accuracy than A4. The loss column requires an identical evaluation definition across variants before it supports a direct comparison. The complete framework achieves the highest reported accuracy (84.12%) and lowest reported loss (0.3211), improving over A1 by 3.16 percentage points. These single-component ablations are consistent with contributions from the correction, adapters, and sample exclusion in the evaluated setting; they do not isolate interactions between components.

Table 3: Ablation study of FedPDPO on IMDB with 10 clients. Results are averaged over 5 runs.

Variant	Avg. acc. (%) ↑	Reported loss ↓
A1: w/o PDPO (DPO)	$80.96_{\pm 0.18}$	0.3616
A2: w/o adapter	$81.06_{\pm 0.19}$	0.3523
A3: w/o exclusion rule	$82.21_{\pm 0.21}$	0.3414
A4: FedPDPO (Full)	$\mathbf{84.12}_{\pm 0.16}$	**0.3211**

Conclusion

In this work, we introduced FedPDPO, a federated preference alignment framework that integrates a principled reward correction mechanism into distributed DPO training. By coupling LoRA-based backbone sharing with client-specific dual heads, FedPDPO achieves both collaborative generalization and personalized alignment. We further provided a theoretical justification of the reward correction through a Gumbel-based analysis, and designed strategies to stabilize learning under heterogeneous data. Extensive experiments on both homogeneous and heterogeneous client settings show higher reported held-out pairwise accuracy than the evaluated baselines on the studied partitions. These comparisons do not establish improvements in generated responses or isolate conditional preference conflict. Our study motivates client-specific reward correction for federated preference

alignment and points to future directions on scaling to larger language models and aligning them with more complex, human-centered preference datasets.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*, 2024.

Manoj Ghuhan Arivazhagan, Vinay Aggarwal, Aaditya Kumar Singh, and Sunav Choudhary. Federated learning with personalization layers. *arXiv preprint arXiv:1912.00818*, 2019.

Shamil Ayupov, Maksim Nakhodnov, Anastasia Yaschenko, Andrey Kuznetsov, and Aibek Alanov. Dreamboothdpo: Improving personalized generation using direct preference optimization. *arXiv preprint arXiv:2505.20975*, 2025.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *Advances in Neural Information Processing Systems*, volume 37, pp. 14457–14483, 2024.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: harmlessness from ai feedback. 2022. *arXiv preprint arXiv:2212.08073*, 8(3), 2022.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. On the weaknesses of reinforcement learning for neural machine translation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eCw3EKvH.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4299–4307, 2017.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *International conference on machine learning*, pp. 2089–2099. PMLR, 2021.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep RL: A case study on PPO and TRPO. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1etN1rtPB.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. *Advances in neural information processing systems*, 33:3557–3568, 2020.

Flint Xiaofeng Fan, Cheston Tan, Yew-Soon Ong, Roger Wattenhofer, and Wei-Tsang Ooi. FedRLHF: A convergence-guaranteed federated framework for privacy-preserving and personalized RLHF. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 713–721, 2025.

Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 7865–7873, 2021.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. FederatedScope-LLM: A comprehensive package for fine-tuning large language models in federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 5260–5271, 2024. doi: 10.1145/3637528.3671573.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*, 2024.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *International conference on machine learning*, pp. 6357–6368. PMLR, 2021.

Zihao Lin, Yan Sun, Yifan Shi, Xueqian Wang, Lifu Huang, Li Shen, and Dacheng Tao. Efficient federated prompt tuning for black-box large pre-trained models. *arXiv preprint arXiv:2310.03123*, 2023.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. Webgpt: Browser-assisted question-answering with human feedback, 2022. URL https://arxiv.org/abs/2112.09332.

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.

Zhen Qin, Daoyuan Chen, Bingchen Qian, Bolin Ding, Yaliang Li, and Shuiguang Deng. Federated full-parameter tuning of billion-sized language models with communication cost under 18 kilobytes. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41473–41497. PMLR, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yudong Liu, Zhixu Du, Yiran Chen, and Holger R. Roth. FedBPT: Efficient federated black-box prompt tuning for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47159–47173. PMLR, 2024a.

Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving LoRA in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*, 2024b.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE transactions on neural networks and learning systems*, 34(12):9587–9603, 2022.

Feijie Wu, Zitao Li, Yaliang Li, Bolin Ding, and Jing Gao. FedBiOT: LLM local fine-tuning in federated learning without full model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3345–3355, 2024a. doi: 10.1145/3637528.3671897.

Feijie Wu, Xiaoze Liu, Haoyu Wang, Xingchen Wang, Lu Su, and Jing Gao. Towards federated RLHF with aggregated client preference for LLMs. In *International Conference on Learning Representations*, 2025.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-dpo: Direct preference optimization with dynamic β. *Advances in Neural Information Processing Systems*, 37:129944–129966, 2024b.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. OpenFedLLM: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6137–6147. Association for Computing Machinery, 2024a. doi: 10.1145/3637528.3671582.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. Openfedllm: Training large language models on decentralized private data via federated learning, 2024b. URL https://arxiv.org/abs/2402.06954.

Liping Yi, Han Yu, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pFedLoRA: Model-heterogeneous personalized federated learning with LoRA tuning. *arXiv preprint arXiv:2310.13283*, 2023.

Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11237–11244, 2023.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedGPT: Federated instruction tuning. In *Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6915–6919, 2024. doi: 10.1109/ICASSP48485.2024.10447454.

Ethics Statement

This work studies personalized federated preference alignment using existing datasets: IMDB, Code-Vulnerability-Security, WebGPT, PyDPO, and UltraFeedback. Our experiments simulate federated clients using benchmark partitions rather than collecting new preference data from users. Although FedPDPO keeps preference examples and personalized heads local, communicated LoRA updates may still reveal information about local data. The proposed protocol does not provide a formal privacy guarantee or establish resistance to inference attacks. Personalized models may inherit biases and harmful content from pretrained models and preference datasets, and alignment with local preferences does not necessarily ensure safety or fairness. Deployment therefore requires appropriate privacy safeguards and application-specific safety evaluation.

Reproducibility Statement

The paper presents the FedPDPO architecture, personalized preference objective, and federated optimization procedure. The experimental section describes the datasets, client partitioning strategies, backbone model, baseline methods, hyperparameter search ranges, and computational setup. Reported results are summarized over five runs. We will publicly release the implementation and experiment scripts upon acceptance.

AI Use Statement

Large language models were used to assist with language editing, manuscript organization, and drafting supplementary material. They also assisted in identifying potential inconsistencies, examining mathematical formulations, and suggesting additional experiments and evaluation protocols. AI-generated content does not constitute independent evidence for theoretical or empirical claims. The authors are responsible for verifying all incorporated material and for the accuracy and integrity of the final manuscript, including its proofs, experimental results, and citations.

A Notation and Scope

Let $W = (A, B)$ denote the shared LoRA parameters. Client i maintains local adapter parameters U_i, policy-head parameters ψ_i, and reward-head parameters ϕ_i. The local policy is $\pi_i = \pi_{W,U_i,\psi_i}$, and the scalar reward is $r_i = r_{W,U_i,\phi_i}$. This notation allows both branches to depend on shared hidden representations; it does not assume independent gradients through the backbone. The frozen reference distribution is $\pi_{\mathrm{ref},i}$. A common reference is the special case in which these distributions are identical across clients.

For a preference pair (x, y_w, y_l), define

$$a_i = \log \frac{\pi_i(y_w \mid x)}{\pi_{\mathrm{ref},i}(y_w \mid x)} - \log \frac{\pi_i(y_l \mid x)}{\pi_{\mathrm{ref},i}(y_l \mid x)}, \tag{8}$$

$$b_i = r_i(x, y_w) - r_i(x, y_l), \qquad c_i = w_r s_i, \tag{9}$$

$$m_i = a_i + c_i b_i, \qquad z_i = \beta m_i, \qquad \ell_i = \log(1 + e^{-z_i}). \tag{10}$$

Throughout the probabilistic interpretation, $\beta > 0$, $c_i \geq 0$, and all log probabilities are well-defined on the compared responses. Unless otherwise stated, c_i is fixed when interpreting a comparison and is the same for all candidate responses to a given prompt. The client risk is $L_i = \mathbb{E}_{\mathcal{D}_i} \ell_i$.

A shared/local formulation of the empirical objective is

$$\min_{W, \{U_i, \psi_i, \phi_i\}_{i=1}^N} \sum_{i=1}^N k_i L_i(W, U_i, \psi_i, \phi_i), \qquad k_i \geq 0, \quad \sum_i k_i = 1. \tag{11}$$

If the scale is recomputed or detached during training, the implemented gradient must follow that convention; the fixed-coefficient interpretation below does not define this update rule.

Preference heterogeneity. Differences in $P_i(x, y_w, y_l)$ include domain shift, difficulty shift, and preference shift. Preference conflict is more specifically characterized by different conditional preferences on common candidate pairs:

$$p_i(y_a \succ y_b \mid x) \neq p_j(y_a \succ y_b \mid x). \tag{12}$$

A shared, client-independent policy may compromise under this conflict. DPO itself does not prohibit local or client-conditioned policies.

B Random-Utility Interpretation

Proposition B.1 (Gumbel comparison with the correct temperature)**.** *Fix a client, a prompt, model parameters, and a coefficient c_i. Define*

$$u_i(x, y) = \beta \log \frac{\pi_i(y \mid x)}{\pi_{\mathrm{ref},i}(y \mid x)} + \beta c_i r_i(x, y). \tag{13}$$

Let ϵ_w, ϵ_l be independent standard Gumbel random variables, with CDF $F(t) = \exp(-e^{-t})$, and let $R_j = u_i(x, y_j) + \epsilon_j$. Then

$$\Pr(R_w > R_l \mid x, y_w, y_l) = \sigma\big(\beta(a_i + c_i b_i)\big). \tag{14}$$

Thus the pairwise negative log-likelihood is exactly ℓ_i.

Proof. Write $d = u_i(x, y_w) - u_i(x, y_l)$. Independence yields

$$\Pr(R_w > R_l) = \int_{-\infty}^{\infty} F(t+d) f(t)\, dt \tag{15}$$

$$= \int_{-\infty}^{\infty} e^{-e^{-(t+d)}} e^{-t} e^{-e^{-t}}\, dt \tag{16}$$

$$= \int_{0}^{\infty} \exp\{-v(1 + e^{-d})\}\, dv \tag{17}$$

$$= \frac{1}{1 + e^{-d}}. \tag{18}$$

The substitution is $v = e^{-t}$. By definition, $d = \beta a_i + \beta c_i b_i$, which proves the result. □

Threshold-shift form. If the Gumbel locations contain only the implicit utilities $\beta \log(\pi_i / \pi_{\mathrm{ref},i})$, adding a deterministic threshold correction $\delta = \beta c_i b_i$ gives

$$\Pr(R_w - R_l + \delta > 0) = \sigma(\beta a_i + \delta). \tag{19}$$

This is a corollary of the same calculation rather than an independent performance guarantee. The factor β must multiply both terms to recover the manuscript's objective.

Scope. This result specifies a coherent likelihood model. It neither identifies a true reward correction nor proves better generalization, convergence, or generation quality. If the coefficient is computed separately from each candidate pair, the comparison can still be logistic, but a single response-level utility shared across all comparisons need not exist. Client-level frozen statistics avoid this particular ambiguity at evaluation time.

C Which Policy Does the Corrected Margin Represent?

Proposition C.1 (Equivalent exponentially reweighted policy)**.** *For a fixed prompt, suppose c_i is independent of the candidate response and*

$$0 < Z_i(x) = \sum_y \pi_i(y \mid x) \exp(c_i r_i(x, y)) < \infty. \tag{20}$$

Define

$$q_i(y \mid x) = \frac{\pi_i(y \mid x) \exp(c_i r_i(x, y))}{Z_i(x)}. \tag{21}$$

Then

$$\log \frac{q_i(y_w \mid x)}{\pi_{\mathrm{ref},i}(y_w \mid x)} - \log \frac{q_i(y_l \mid x)}{\pi_{\mathrm{ref},i}(y_l \mid x)} = a_i + c_i b_i. \tag{22}$$

Consequently, the corrected comparison loss has the DPO form for q_i, not necessarily for the autoregressive policy π_i used for decoding.

Proof. Substitute $\log q_i = \log \pi_i + c_i r_i - \log Z_i(x)$. The prompt-dependent normalizer cancels between the two responses. □

Interpretation and implementation. This identity is analytical; it does not provide a tractable exact sampler from q_i. Ordinary decoding from π_i does not generally sample from q_i. Selecting the best of finitely many candidates using a reward score also need not sample from q_i. Accordingly, the deployed inference procedure must be specified and evaluated directly.

A ranking counterexample. Let the implicit margin on a pair be $a_i = -1$. For a one-pair magnitude estimate with $b_i = 1$ and $s_i = 1/(1+\epsilon)$, the manuscript's $w_r = 1.5$ gives

$$m_i = -1 + \frac{1.5}{1+\epsilon} > 0 \quad \text{whenever } 0 < \epsilon < 0.5. \tag{23}$$

The corrected classifier ranks the preferred response correctly even though the implicit margin has the wrong sign. With equal reference probabilities on these two responses, the policy itself also assigns the preferred response lower probability. This is a possible configuration, not evidence that it occurs in the reported runs.

Non-identifiability in an unrestricted model. For fixed $c_i > 0$ and any positive distribution q_i with compatible support, set

$$r_i(x, y) = \frac{1}{c_i} \log \frac{q_i(y \mid x)}{\pi_i(y \mid x)}. \tag{24}$$

The reweighting then exactly recovers q_i, with $Z_i(x) = 1$. Thus a combined likelihood alone does not uniquely determine the policy/reward decomposition when the reward function is unrestricted. Restricted head capacity and adaptive normalization may constrain this construction; the statement is not a proof of arbitrary decomposition under every implementation.

Implications for deployment. The combined scorer can be evaluated as a preference-ranking system. Claims about generated responses require direct evaluation of the policy and its inference procedure. An auxiliary policy loss or distillation would constitute a different method.

D Gradient and Scale Diagnostics

For any trainable parameter block v, differentiation gives

$$\nabla_v \ell_i = -\beta\sigma(-\beta m_i)\left(\nabla_v a_i + c_i \nabla_v b_i + b_i \nabla_v c_i\right). \tag{25}$$

When the scale is detached, the last term is zero. If a policy-only parameter ψ_i does not affect b_i, then

$$\nabla_{\psi_i} \ell_i = -\beta\sigma(-\beta(a_i + c_i b_i)) \nabla_{\psi_i} a_i. \tag{26}$$

For the same a_i and a detached nonnegative correction $c_i b_i \geq 0$, the scalar coefficient is no larger than the DPO coefficient $\beta\sigma(-\beta a_i)$. A successful reward correction can therefore reduce the policy-only learning signal. Backbone gradients can behave differently because the backbone may affect both branches.

Freezing parameters is not detaching features. Holding the head parameters fixed during the backbone phase still permits gradients to propagate through the fixed head to hidden states and LoRA parameters. Detaching those hidden states removes this path and defines a different update. This distinction must be consistent between pseudocode and code.

Initialization. For the proposed scale

$$s_i = \frac{\mathbb{E}|a_i|}{\mathbb{E}|b_i| + \epsilon}, \tag{27}$$

identical initial policy and reference imply $a_i = 0$ and hence $s_i = 0$. Under detached scaling, reward-head parameters that only affect b_i receive zero gradient from the correction at that instant. The policy branch can still update and subsequently enable a nonzero correction. The behavior depends on update ordering, normalization, and any auxiliary losses.

Magnitude normalization limits a simple scaling argument. Suppose reward margins are all multiplied by $k > 0$ and write $A = \mathbb{E}|a_i|$, $B = \mathbb{E}|b_i|$. The corrected contribution becomes

$$w_r \frac{A}{kB + \epsilon} k b_i. \tag{28}$$

For $B > 0$, it converges to $w_r A b_i / B$ as $k \to \infty$. Therefore, arbitrarily scaling the reward head alone does not necessarily make the normalized corrected margin unbounded. Analysis that assumes a fixed scale should not be presented as a proof about this adaptive normalization.

E A Conditional Comparison-Risk Bound

This section states precisely what an accurate correction could imply. It does not assume that a learned reward head satisfies the condition.

Proposition E.1 (Excess logistic risk from logit error)**.** *Consider a fixed orientation of candidate pairs and a random preference label $Y \in \{0, 1\}$. Suppose its true conditional probability is $p^* = \sigma(z^*)$ with finite logit z^*. Define*

$$\mathcal{R}(z; p^*) = -p^* \log \sigma(z) - (1 - p^*) \log(1 - \sigma(z)). \tag{29}$$

Then, pointwise,

$$0 \le \mathcal{R}(z; p^*) - \mathcal{R}(z^*; p^*) \le \frac{1}{8}(z - z^*)^2. \tag{30}$$

If $z = \beta(a_i + c_i b_i)$ and $z^* = \beta(a_i + \delta_i^*)$, the expected excess risk is at most

$$\frac{\beta^2}{8}\mathbb{E}[(c_i b_i - \delta_i^*)^2], \tag{31}$$

provided the expectation exists.

Proof. The first derivative is $\sigma(z) - p^*$, which is zero at z^*. The second derivative is $\sigma(z)(1-\sigma(z)) \le 1/4$. Taylor's theorem yields the upper bound; optimality of z^* yields nonnegativity. Substitution and expectation prove the last statement. □

Limits of the result. The residual $\delta_i^* = z^*/\beta - a_i$ is unknown. A smaller upper bound alone does not establish that the actual risk is lower than that of DPO. The statement also gives no finite-sample generalization bound and no generation-quality guarantee. It motivates measuring calibration and correction error where a controlled preference model supplies ground truth.

F Federated Update and Communication Accounting

Write $\xi_i = (U_i, \psi_i, \phi_i)$ for local parameters. The following round-index convention extends Algorithm 1 to a selected set S_r. The main-text algorithm assumes all-client participation:

1. Initialize W^0 and each ξ_i^0 before round zero.
2. At round r, select S_r and broadcast W^r. Each selected client sets $W_i^{r,0} = W^r$ and loads ξ_i^r.
3. Perform the specified number of local-head steps with W_i fixed, producing ξ_i^{r+1}.
4. Perform the specified number of LoRA steps with ξ_i^{r+1} fixed, producing W_i^{r+1}; retain gradients through fixed heads where required.
5. Upload only W_i^{r+1}. With $\alpha_i^r = n_i / \sum_{j \in S_r} n_j$, set
$$A^{r+1} = \sum_{i \in S_r} \alpha_i^r A_i^{r+1}, \qquad B^{r+1} = \sum_{i \in S_r} \alpha_i^r B_i^{r+1}. \tag{32}$$
6. Retain unselected clients' local states. Optimizer and normalization state persistence is an implementation choice that must be reported alongside the executed configuration.

This describes sample-weighted factor aggregation. For partial participation, the selected-client estimator generally differs from the full-client objective without an additional sampling argument.

Factor averaging differs from product averaging. Let $\bar{A} = \sum_i \alpha_i A_i$ and $\bar{B} = \sum_i \alpha_i B_i$, with weights summing to one. Expansion gives

$$\sum_i \alpha_i B_i A_i - \bar{B}\bar{A} = \sum_i \alpha_i (B_i - \bar{B})(A_i - \bar{A}). \tag{33}$$

Thus averaging LoRA factors is generally not equivalent to averaging the induced weight updates. The identity does not make factor-wise aggregation invalid, but its behavior should not be described as exact full-update averaging.

Parameter and byte counts. For target matrices indexed by l, with dimensions $d_{\mathrm{out},l} \times d_{\mathrm{in},l}$ and LoRA rank r_l,

$$P_{\mathrm{LoRA}} = \sum_l r_l (d_{\mathrm{in},l} + d_{\mathrm{out},l}). \tag{34}$$

At b bytes per communicated parameter, a selected client's upload and download total approximately $2bP_{\mathrm{LoRA}}$ bytes per round, excluding protocol overhead and one-time model distribution. This is per-client unicast accounting; actual server traffic depends on the deployment.

A full vocabulary projection contains Vd parameters, excluding bias. A residual bottleneck with width h has approximately $2dh$ parameters per insertion, excluding biases. Thus local-head retention reduces communication but does not alone establish low local memory. Trainable-parameter and optimizer-state memory depends on the instantiated architecture; the formula above accounts only for communicated LoRA factors.

G Evaluation Definitions and Limitations

For each held-out pair, the policy, reward, and combined scores are respectively

$$s_i^{\mathrm{policy}} = a_i, \qquad s_i^{\mathrm{reward}} = b_i, \qquad s_i^{\mathrm{combined}} = a_i + c_i b_i. \tag{35}$$

Their accuracies need not agree. In particular, a correct combined ranking does not imply a correct policy ranking (Appendix C). A per-client score accuracy requires an explicit tie convention and a stated aggregation over clients and independent runs. The existing tables do not identify the underlying score or tie convention, so no deployment or policy-only accuracy is inferred from them. The two different standard deviations attached to 84.12% in Tables 2 and 3 likewise require verification against run logs.

The IMDB margin partitions vary comparison difficulty; language partitions vary domain; and the three-dataset experiment also changes task and annotation source. These partitions do not isolate disagreement on the same prompts and response pairs. The probabilistic results concern modeled comparisons and provide neither a finite-sample generalization guarantee nor a guarantee about generated responses. The method transmits LoRA factors rather than raw preference examples or local heads; this alone does not provide differential privacy or resistance to inference attacks.