Beyond Client Clustering: Fine-Grained Preference Alignment in Federated RLHF via Self-Evolving Routing

Ke Wang[1], **Shaojing Fu**[1*], **Yuchuan Luo**[1*], **Guilin Deng**[1], **Silong Chen**[1], **Zheng Yuan**[1] and **Lin Liu**[1]

[1]College of Computer Science, National University of Defense Technology
{wang_ke517, fushaojing, luoyuchuan09, dengguilin, chensilong, yuanz, liulin16}@nudt.edu.cn

Abstract

Federated Reinforcement Learning from Human Feedback (RLHF) enables the collaborative alignment of Large Language Models (LLMs) while preserving privacy, yet it faces critical bottlenecks arising from data heterogeneity. Existing approaches typically rely on rigid client-level clustering, which overlooks intra-client heterogeneity and fails to adapt to the multifaceted needs of individual users. To address this, we propose FedPrism, a novel framework that shifts alignment granularity from the coarse client level to the precise instance level. Similar to an optical prism dispersing mixed light into distinct spectral components, FedPrism decomposes complex, intra-client heterogeneous data streams by dynamically routing individual samples to specialized experts based on semantic features. Crucially, we devise a Posterior-Guided Performance Distillation (PGPD) mechanism that leverages experts' actual training loss as self-supervision to autonomously refine routing policies without explicit labels. Extensive experiments show that FedPrism not only establishes new state-of-the-art results but also effectively mitigates the negative transfer prevalent in non-IID settings, ensuring superior alignment fidelity.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide spectrum of tasks, ranging from complex reasoning to creative generation [Achiam *et al.*, 2023; Gao *et al.*, 2025]. To ensure these models align with human values and intent, Reinforcement Learning from Human Feedback (RLHF) has emerged as a standard paradigm [Christiano *et al.*, 2017], typically utilizing methods like Proximal Policy Optimization (PPO) [Schulman *et al.*, 2017] or Direct Preference Optimization (DPO) [Rafailov *et al.*, 2023]. However, the efficacy of RLHF heavily relies on high-quality, diverse human preference data. In real-world scenarios, such data is often inherently decentralized and privacy-sensitive (e.g., personal chat history, user-specific feedback), making centralized collection impractical due to privacy regulations such as GDPR [Voigt and Von dem Bussche, 2017]. Consequently, Federated Learning (FL) has been introduced to the RLHF pipeline, enabling collaborative alignment without exposing raw user data [Ye *et al.*, 2024].

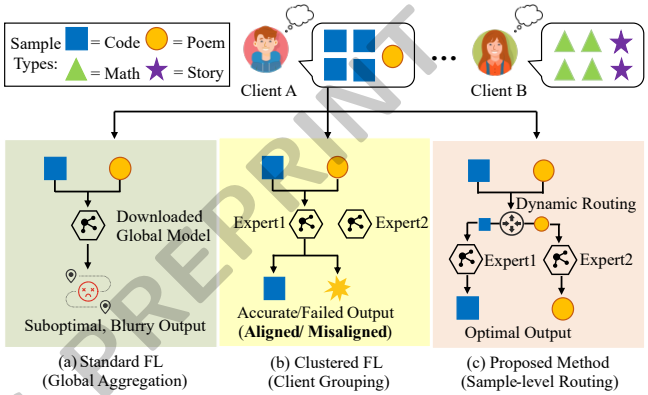

Figure 1: Comparison of decision granularities in federated RLHF. (a) Standard FL aggregates a single global model. (b) Existing clustered methods (e.g., FedBiscuit) assume intra-client homogeneity, assigning all samples from a user to a static expert, leading to mismatches. (c) Our FedPrism introduces instance-wise dynamic routing, dispatching individual samples to the most suitable experts regardless of client identity.

Despite the promise of federated RLHF, a fundamental challenge remains: data heterogeneity. Different users exhibit distinct preferences and interact with LLMs in diverse contexts. As illustrated in Fig. 1(a), traditional FL methods like FedAvg [McMahan *et al.*, 2017] aggregate diverse local updates into a single global model. This "one-size-fits-all" approach often fails to capture the nuances of diverse preferences, resulting in a mediocre model that produces suboptimal or blurry outputs satisfying no one. To mitigate this, recent works such as FedBiscuit [Wu *et al.*, 2025a] have proposed grouping clients into clusters and training separate preference selectors (Experts) for each group, as shown in Fig. 1(b). While this clustered FL approach addresses inter-client differences to some extent, it relies on a coarse-grained assumption of intra-client homogeneity—assuming a single client's needs are uniform enough to be handled by one expert. Fig. 1(b) visualizes the critical flaw in this assumption:

*Corresponding author

a real-world user is multifaceted. For instance, Client A may query the LLM for coding tasks at one moment and creative poetry at another. If the system statically categorizes Client A as a developer based on dominant data, the coding tasks may be aligned, but the creative tasks will inevitably suffer from misalignment. Assigning such a user to a single, static expert fundamentally limits the model's flexibility to address the instance-wise diversity of user queries.

We argue that the decision granularity in federated RLHF must shift from the client level to the instance level, ensuring that the right expert handles the right sample, regardless of which client the sample originates from. To bridge this granularity gap, we propose FedPrism, a novel framework that introduces instance-wise dynamic routing into federated RLHF. Functioning analogously to an optical prism that disperses mixed light into distinct spectral components, FedPrism maintains a global router and a set of specialized experts. As depicted in Fig. 1(c), the local router dynamically decomposes the heterogeneous user data stream by dispatching each input sample to the most suitable expert—for instance, steering rigorous coding tasks to a technical expert while channeling creative poetry to an artistic expert.

Crucially, to refine this routing policy without explicit labels, we devise a Posterior-Guided Performance Distillation (PGPD) mechanism. Specifically, we evaluate the actual loss of all experts on local samples to identify the ground-truth best expert. This posterior knowledge is then distilled back into the router via cross-entropy minimization, creating a self-reinforcing cycle where the router continuously learns to predict the optimal expert for any given input. By decoupling expert selection from client identity, FedPrism effectively handles both inter-client and intra-client heterogeneity. Finally, to stabilize the training dynamics and mitigate the "cold start" problem, we introduce a comprehensive initialization and update strategy: the router is pre-trained on public unlabeled prompts to acquire preliminary allocation capabilities, and is further safeguarded by warm-up and lazy update protocols to prevent collapse due to the initial immaturity of the experts. Our main contributions are summarized as follows:

- We identify the critical oversight of intra-client heterogeneity in existing federated RLHF approaches and propose FedPrism, the first framework to enable instance-wise dynamic routing in a privacy-preserving manner. By functioning as an alignment prism, it decouples expert selection from client identity, ensuring precise customization for multifaceted user needs.
- We devise a comprehensive router optimization strategy that overcomes the cold-start problem and lack of explicit labels. This includes a public-data initialization mechanism to bridge the semantic gap between unlabeled and user-specific prompts, followed by a novel PGPD strategy that leverages experts' actual training loss to autonomously refine routing policies.
- Comprehensive experiments are conducted on federated human preference benchmarks. Empirical results demonstrate that FedPrism outperforms state-of-the-art baselines, yielding superior alignment fidelity and effectively handling diverse preference distributions.

2 Related Work

2.1 Federated Instruction Tuning and Alignment

The integration of LLMs with FL offers a path to privacy-preserving intelligence [Zhao *et al.*, 2023; Jiang *et al.*, 2025], though it faces prohibitive resource burdens [Wu *et al.*, 2025b]. To mitigate this, Parameter-Efficient Fine-Tuning (PEFT) methods like Low-Rank Adaptation (LoRA) [Hu *et al.*, 2022] and Adapters [Houlsby *et al.*, 2019] have been extensively adopted. Frameworks such as FedIT [Zhang *et al.*, 2024], FedPETuning [Zhang *et al.*, 2023], and others [Sun *et al.*, 2023; Yi *et al.*, 2023] utilize these efficient infrastructures to enable Federated Instruction Tuning (FIT) [Lin *et al.*, 2023; Kuang *et al.*, 2024; Fan *et al.*, 2023]. While benchmarks like OpenFedLLM [Ye *et al.*, 2024] have evaluated these algorithms, the domain of preference alignment remains underexplored. Early attempts apply direct optimization to decentralized data but rely on naive aggregation, thereby failing to capture the intricate, non-IID nature of diverse human preferences [Santurkar *et al.*, 2023].

2.2 RLHF

Aligning LLMs typically involves Supervised Fine-Tuning (SFT) and reinforcement learning [Ouyang *et al.*, 2022], utilizing algorithms like PPO [Schulman *et al.*, 2017] or the more recent DPO [Rafailov *et al.*, 2023]. To address privacy concerns in decentralized scenarios, federated RLHF has emerged, as established by FedBis and FedBiscuit [Wu *et al.*, 2025a]. This approach generally follows a two-stage paradigm: decentralized clients collaboratively train preference experts, which are subsequently used to guide server-side global LLM alignment. However, existing implementations often treat the federation as a collection of static entities, producing experts that capture only broad, average tendencies. This lack of instance-specific adaptation intrinsically limits alignment efficacy when users exhibit complex, multifaceted preferences.

2.3 Personalization and Mixture of Experts in FL

To tackle statistical heterogeneity, personalized FL often employs client clustering [Sattler *et al.*, 2020; Ghosh *et al.*, 2020]. FedBiscuit [Wu *et al.*, 2025a] adopts this for alignment, binding clients to static preference experts. However, this rigid assignment assumes intra-client homogeneity, failing when users exhibit multifaceted interests. Conversely, Mixture of Experts (MoE) architectures enable dynamic token dispatching [Shazeer *et al.*, 2017; Zec *et al.*, 2020]. While proven in centralized LLMs [Jiang *et al.*, 2024], federated MoE applications remain limited to general classification for efficiency [Yi *et al.*, 2024]. Our work bridges this gap by shifting from FedBiscuit's static client-level clustering to fine-grained, instance-wise routing, decoupling expert selection from client identity.

3 Methodology

3.1 Problem Formulation

Considering a federated preference alignment scenario comprising N clients, where each client i holds a private

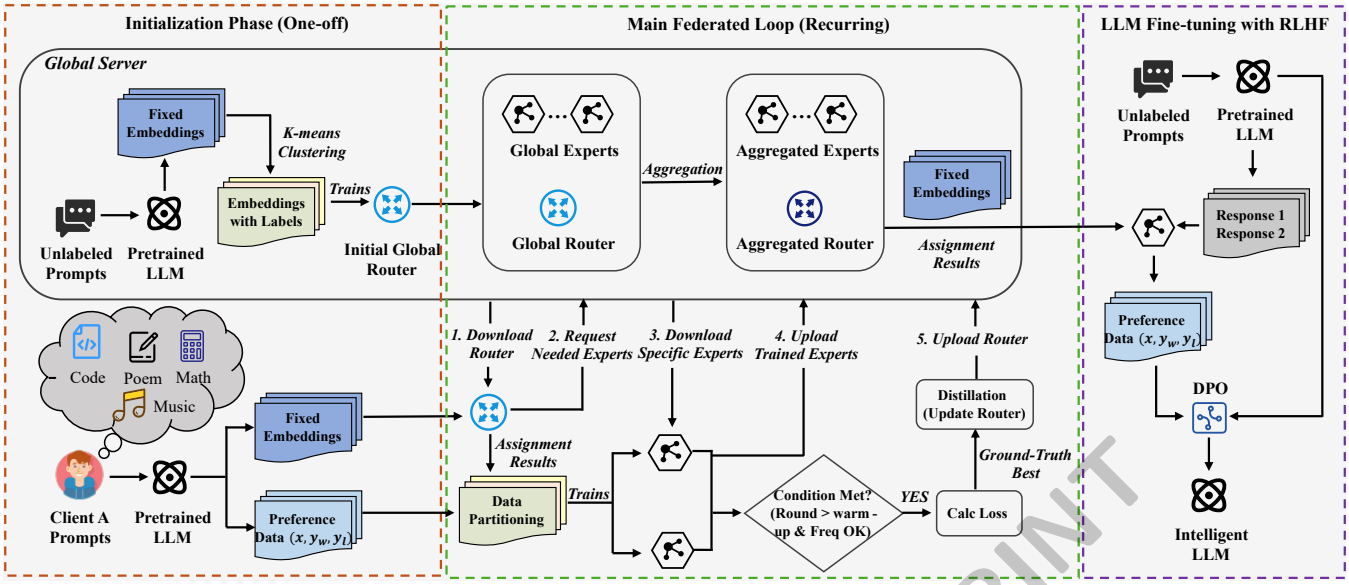

Figure 2: Overview of the FedPrism framework. The workflow consists of three parts: (1) Initialization Phase (One-off): The global router is warm-started on the server, and a fixed local cache of embeddings is generated on clients to reduce computational overhead. (2) Main Federated Loop: In each round, the local router utilizes cached embeddings to dynamically dispatch inputs and download only the specific required experts. Crucially, the router is trained via PGPD, using experts' training losses as supervision. (3) LLM Fine-tuning with RLHF: After federated training, the aggregated router and experts are used to construct a high-quality preference dataset from unlabeled prompts. This synthetic data is then used to fine-tune the pre-trained LLM via DPO, yielding the final intelligent model.

dataset $\mathcal{D}_i = \{(x, y)\}_{j=1}^{|\mathcal{D}_i|}$ containing instruction-response pairs drawn from a local distribution $\mathcal{P}_i$. Standard federated approaches typically aim to learn a global policy or static client-specific models by minimizing the aggregate loss $\min_\Theta \sum_i \mathcal{L}(\Theta; \mathcal{D}_i)$. However, this formulation often implicitly assumes intra-client homogeneity, treating each client as an atomic unit with a single dominant preference. In reality, a user's local distribution is often multimodal, governed by latent intent variables z (e.g., specific domains like coding, reasoning, or creative writing), meaning $\mathcal{D}_i$ is a mixture of diverse tasks. Consequently, a critical granularity mismatch arises in existing clustered methods, which assign a single static expert $E_{c(i)}$ to the entire client i. This creates a fundamental bottleneck: for a heterogeneous client, a single expert cannot simultaneously be optimal for all latent intents $z \in \mathcal{D}_i$, inevitably leading to misalignment on divergent tasks.

To address this limitation, we reformulate the federated alignment problem by shifting the decision granularity from the client level to the instance level. We assume a global mixture of M specialized experts parameterized by a set $\Theta = \{\Theta_1, \ldots, \Theta_M\}$, and introduce a dynamic routing policy $\mathcal{R}_\phi(x)$ that maps each specific input x to the most suitable expert index, regardless of the client identity. Our objective is to collaboratively learn the optimal experts and routing parameters ϕ to minimize the instance-wise alignment loss across the federation:

$$\min_{\Theta, \phi} \sum_{i=1}^{N} \sum_{(x,y)\in\mathcal{D}_i} \mathcal{L}_{\text{align}} \left(E(x; \Theta_{\mathcal{R}_\phi(x)}), y \right) \tag{1}$$

where $\mathcal{R}_\phi(x)$ denotes the gating mechanism that outputs the index of the selected expert for input x, making the expert assignment $E_{\mathcal{R}_\phi(x)}$ dependent solely on instance semantics rather than client identity. The parameters Θ and ϕ are jointly optimized to minimize the alignment loss $\mathcal{L}_{\text{align}}$, enabling the router learns to dispatch samples to the experts most capable of handling them. By solving this objective, the system ensures that the right expert handles the right sample, effectively decoupling preference alignment from static client clustering and adapting to the instance-wise diversity of user queries.

3.2 FedPrism Algorithm

To effectively solve the instance-wise optimization objective formulated in Eq. (1), we propose FedPrism, a framework designed to decouple preference alignment from client identity via dynamic routing. As illustrated in Fig. 2, the lifecycle of FedPrism is structured into three distinct phases to address the challenges of cold-start routing, label-free optimization, and downstream application.

First, to overcome the lack of initial routing capabilities and ensure feature consistency, we introduce a initialization phase. This phase not only leverages a public proxy dataset to activate the server-side router but also simultaneously aligns client-side data representations via semantic embedding extraction. Second, the core of our method lies in the main federated loop, where we employ a novel PGPD strategy. This allows the router and experts to collaboratively evolve in a decentralized manner without requiring ground-truth routing labels. Finally, in the LLM fine-tuning phase, the fully trained mixture-of-experts is utilized as a high-precision preference selector to align the server-side LLM via RLHF. The details of these three components are elaborated below.

Phase I: Initialization and Feature Alignment

To mitigate the cold-start problem and ensure semantic consistency across the federation, this phase involves synchronized initialization steps on both the global server and local clients, as depicted in the left panel of Fig. 2.

Server-Side: Router Activation via Public Data. To endow the router with initial semantic discrimination capabilities without accessing private client data, the server utilizes a public unlabeled dataset consisting of diverse prompts. First, a frozen pretrained LLM extracts fixed semantic embeddings from these prompts. Next, we apply K-means clustering to these embeddings to discover M latent semantic clusters, assigning a pseudo-label $c(x)$ to each sample. Finally, the global router, architected as a lightweight Multi-Layer Perceptron (MLP), is supervised to predict these cluster assignments, resulting in an initial global router that captures coarse-grained semantic structures.

Client-Side: Semantic Feature Extraction. Simultaneously, each client i processes its local private instructions, which may span various domains such as code, poems, or math, to align its data representation with the router's input space. Specifically, clients employ the same frozen pretrained LLM used by the server to convert their raw instruction prompts x into fixed semantic embeddings. This step is crucial because the router, trained on server-side embeddings, requires inputs in the same latent space to function correctly. These pre-computed embeddings, along with the local preference pairs denoted as (x, y_w, y_l) (where y_w represents the response preferred over y_l), serve as the input for the subsequent federated loop.

Phase II: Main Federated Loop

Following the initialization, the system transitions into the main federated loop, which operates in iterative communication rounds $t = 1, \ldots, T$. As depicted in the middle panel of Fig. 2, this phase constitutes the core lifecycle of FedPrism, where the router and experts co-evolve to adapt to user preferences. To balance instance-wise personalization performance with the strict communication constraints of edge devices, we decompose the workflow of each round into four stages:

Step 1: Instance-Wise Routing and Sparse Retrieval. At the beginning of round t, each participating client i first downloads the current global router parameters $\phi^{(t)}$ from the server. Crucially, the router is designed as a lightweight module, incurring negligible communication cost compared to the LLM backbone. Using the fixed semantic embeddings extracted in Phase I, the client computes the gating probabilities for every local instruction $x \in \mathcal{D}_i$ via the downloaded router:

$$p(x) = \text{Softmax}\left(\mathcal{R}_{\phi^{(t)}}(h_x)\right) \tag{2}$$

where h_x is the pre-computed embedding of x. Based on these probabilities, the client selects the single most relevant expert for each instance by identifying the index with the highest probability:

$$k^*(x) = \underset{k \in \{1,\ldots,M\}}{\text{argmax}} \; [p(x)]_k \tag{3}$$

The client i then aggregates these indices to form a request list of unique experts required for its local dataset, denoted as $\mathcal{I}_i = \bigcup_{x \in \mathcal{D}_i} \{k^*(x)\}$. The server responds by transmitting only the parameters of the experts in $\mathcal{I}_i$, specifically their lightweight LoRA adapters, rather than the full mixture $\Theta^{(t)}$. This sparse retrieval mechanism significantly reduces the downlink bandwidth consumption, ensuring that the communication cost remains proportional to the diversity of local tasks rather than the total number of experts.

Step 2: Expert Specialization via Partitioned Training. Once the specific active experts are retrieved, the client organizes the training process by partitioning its local dataset $\mathcal{D}_i$ into disjoint subsets based on the routing decisions. Let $\mathcal{D}_i^{(k)}$ denote the subset of preference pairs assigned to expert k. Fundamentally, each expert is a LLM configured as a binary selector to output a scalar probability logit. However, fine-tuning and storing $|\mathcal{I}_i|$ full-parameter LLMs is computationally prohibitive for edge clients. To address this, we adopt LoRA and formulate the k-th expert as $E_k(x, y_A, y_B) = \text{LLM}(x, y_A, y_B; \Theta_{\text{base}}, \delta_k)$, where Θ_{base} is the frozen pretrained backbone shared by all experts, and δ_k represents the lightweight, trainable LoRA adapters specific to expert k. This design implies that the system physically maintains one LLM body but logically switches between different expert personalities by swapping small adapters. Since the expert acts as a binary classifier, we must format the raw preference pairs into a labeled classification dataset. To eliminate position bias, we augment the local partition $\mathcal{D}_i^{(k)}$ by splitting each preference pair (x, y_w, y_l) into two distinct samples with ground-truth labels:

- Sample 1: Input (x, y_w, y_l), label $c = 0$ (indicating the first response y_w is better).
- Sample 2: Input (x, y_l, y_w), label $c = 1$ (indicating the second response y_w is better).

This results in an augmented dataset $\hat{\mathcal{D}}_i^{(k)}$. During training, the adapter δ_k is updated to minimize the Cross-Entropy (CE) loss over these labeled samples:

$$\mathcal{L}_{\text{expert}}(\delta_k) = \mathbb{E}_{(x, y_A, y_B, c) \sim \hat{\mathcal{D}}_i^{(k)}} \left[\mathcal{L}_{\text{CE}}\left(E_k(x, y_A, y_B; \delta_k), c\right)\right] \tag{4}$$

By optimizing this objective, each expert specializes in identifying the superior response within its specific sub-domain while keeping the communication overhead minimal.

Step 3: PGPD. This step in the loop addresses the challenge of optimizing the router without ground-truth routing labels. We propose a PGPD strategy that allows the router to learn from the experts' evolved capabilities. However, directly updating the router based on randomly initialized experts would introduce severe noise. To mitigate this, we impose a stability constraint consisting of a warm-up period and a regulated update frequency. The router update is executed only when the current round t exceeds the warm-up threshold T_{warm} and satisfies the update frequency F_{freq}. During the warm-up phase, the router remains frozen, acting solely as a dispatcher to facilitate expert specialization. When the update conditions are met, the client leverages the set of locally retrieved experts $\mathcal{I}_i$ to identify the post-hoc optimal expert for each sample. Although the router initially assigned instruction x to a specific expert, the client effectively holds a small pool of experts $\mathcal{I}_i$.

For each sample (x, y_w, y_l), the client evaluates the CE loss on all available experts in $\mathcal{I}_i$ and identifies the one yielding the minimum loss:

$$k_{\text{best}}(x) = \underset{k \in \mathcal{I}_i}{\operatorname{argmin}} \, \mathcal{L}_{\text{CE}}\left(E_k(x, y_w, y_l; \delta_k), 0\right) \tag{5}$$

This index $k_{\text{best}}(x)$ serves as the posterior pseudo-label, representing the expert that actually performed best on this instance. Finally, the local router parameters ϕ are updated to predict these posterior optimal indices. We minimize the Kullback-Leibler (KL) divergence between the router's prior distribution and the one-hot distribution of the posterior best expert:

$$\mathcal{L}_{\text{router}}(\phi) = -\mathbb{E}_{x \sim \mathcal{D}_i}\left[\log\left([\text{Softmax}(\mathcal{R}_\phi(h_x))]_{k_{\text{best}}(x)}\right)\right] \tag{6}$$

By recursively applying this distillation, the router's decision-making logic progressively aligns with the experts' actual performance, forming a positive feedback loop that enhances routing accuracy over time.

Step 4: Selective Upload and Global Aggregation. At the conclusion of local training, the server synchronizes the dispersed knowledge via a unified weighted aggregation mechanism. Let ϵ denote a specific trainable component (i.e., $\epsilon = \phi$ for the global router, or $\epsilon = \delta_k$ for an expert adapter where $k \in \{1, \ldots, M\}$), and $\mathcal{S}_\epsilon$ be the set of clients who updated it. The global aggregation is formulated as:

$$\epsilon^{(t+1)} = \sum_{i \in \mathcal{S}_\epsilon} \frac{n_i(\epsilon)}{\sum_{j \in \mathcal{S}_\epsilon} n_j(\epsilon)} \epsilon_i^{(t)} \tag{7}$$

where $n_i(\epsilon)$ represents the effective sample count used by client i for component ϵ. Specifically, for the router ($\epsilon = \phi$), the update depends on the full local dataset, so $n_i(\phi) = |\mathcal{D}_i|$; for a specific expert ($\epsilon = \delta_k$), the update relies solely on the routed subset, so $n_i(\delta_k) = |\mathcal{D}_i^{(k)}|$. This unified mechanism ensures that every module, whether global or specialized, evolves proportionally to its relevant domain data support. The updated global router $\phi^{(t+1)}$ and expert adapters $\{\delta_k^{(t+1)}\}_{k=1}^M$ are then ready for the next communication round. We provide the complete pseudocode for the phase II execution in Algorithm 1.

Phase III: LLM Fine-tuning with RLHF

After the federated training concludes, the server obtains a fully evolved MoE system, comprising the global router $\mathcal{R}_\phi$ and the set of aggregated domain-specific experts $\{\delta_k\}_{k=1}^M$. In this final phase, we leverage this collaborative intelligence as a high-precision, privacy-preserving preference annotator to align the server-side LLM. As illustrated in the right panel of Fig. 2, this phase consists of two key steps:

1. Synthetic Preference Dataset Construction. Since real user preference data is private and remains local, the server utilizes a public dataset of unlabeled prompts $\mathcal{D}_{\text{unlabeled}}$ to distill the federated knowledge. For each prompt $x \in \mathcal{D}_{\text{unlabeled}}$, the server's base LLM Λ generates a pair of candidate responses (y_1, y_2). To determine which response is better without human intervention, we employ a hierarchical annotation strategy. First, the global router $\mathcal{R}_\phi$ analyzes the prompt embedding h_x and identifies the most suitable expert index $k^* = \text{argmax}_k[\mathcal{R}_\phi(h_x)]_k$. Next, the selected expert E_{k^*} evaluates the pair. Since the expert is trained as a binary selector, it computes the probability logits for both response orders. The response with the higher probability is labeled as the winner y_w, and the other as the loser y_l. By repeating this process, we construct a synthetic, high-quality preference dataset $\mathcal{D}_{\text{syn}} = \{(x, y_w, y_l)\}$.

2. Policy Optimization via DPO. Finally, we fine-tune the target LLM Λ on $\mathcal{D}_{\text{syn}}$ by directly minimizing the standard DPO loss [Rafailov *et al.*, 2023]. Through this process, the server-side LLM effectively absorbs the diverse, domain-specific preferences learned from decentralized clients, achieving "one model, all preferences" alignment.

Algorithm 1 Main Federated Loop

Input: Rounds T, Warm-up T_{warm}, Update freq F_{freq}, Local data $\mathcal{D}_i$, Initial router $\phi^{(0)}$, Initial experts (LoRA) $\Delta^{(0)}$
Output: Updated $\Delta^{(T)}$, Updated $\phi^{(T)}$

▷ **Server Side**
for round $t = 0, \ldots, T-1$ **do**
  Sample clients $\mathcal{S}^{(t)}$ and broadcast Router $\phi^{(t)}$
  for each client $i \in \mathcal{S}^{(t)}$ **in parallel do**
    $\mathcal{I}_i \leftarrow$ **ClientRoute**$(i, \phi^{(t)})$
    $\Delta_{\text{sparse}} \leftarrow \{\delta_k^{(t)} \mid k \in \mathcal{I}_i\}$
    $\Delta_i, (\phi_i) \leftarrow$ **ClientTrain**$(i, \Delta_{\text{sparse}}, \mathcal{I}_i)$
  end for
  $\delta_k^{(t+1)}, (\phi^{(t+1)}) \leftarrow$ **FedAvg**$(\{\delta_{i,k} \mid k \in \mathcal{I}_i\}, (\{\phi_i\}))$
end for
return Updated $\Delta^{(T)}$, Updated $\phi^{(T)}$
▷ **Client Side**
Determine required expert indices $\mathcal{I}_i$ via Router ϕ
Download active experts $\{\delta_k\}_{k \in \mathcal{I}_i}$ from Δ_{global}
for each expert $k \in \mathcal{I}_i$ **do**
  Assign data subset $\mathcal{D}_i^{(k)}$ via router choice
  Update δ_k as a binary selector on $\mathcal{D}_i^{(k)}$
end for
if $t > T_{\text{warm}}$ **and** time to update **then**
  Find best expert $k_{\text{best}}(x)$ for each sample via loss
  Update Router ϕ to predict $k_{\text{best}}(x)$
  return Updated Δ_i, Updated ϕ
else
  return Updated Δ_i
end if

4 Experiment

4.1 Experimental Setup

Benchmarks and Heterogeneous Partitioning. Following the federated RLHF benchmark established by [Wu *et al.*, 2025a], we evaluate on two open-ended generation tasks. For summarization, we utilize the Reddit TL;DR dataset [Völske *et al.*, 2017] for local training and employ the OpenAI TL;DR dataset as unlabeled prompts for global alignment. To simulate natural preference heterogeneity, data is partitioned by unique worker IDs into 53 non-IID clients. For Question-Answering (QA), we employ the SHP dataset [Ethayarajh *et*

al., 2022]. We construct a challenging non-IID setting by partitioning data among 200 clients using a Dirichlet distribution with $\alpha = 0.3$, while utilizing SafeRLHF prompts [Dai *et al.*, 2023] for the alignment phase.

Base Models, Baselines, and Metrics. We employ Gemma-2B [Team *et al.*, 2024] and LLaMA-2-7B [Touvron *et al.*, 2023] as backbone LLMs. For preference scoring, we leverage heterogeneous binary selectors including Qwen-2-0.5B, Gemma-2B, and LLaMA-2-7B [Touvron *et al.*, 2023]. We compare FedPrism against representative methods: FedAvg using local LoRA [McMahan *et al.*, 2017; Sun *et al.*, 2024], FedDPO with local optimization [Ye *et al.*, 2024], FedBis aggregating a single global selector, and FedBiscuit employing clustered training [Wu *et al.*, 2025a]. Performance is assessed via Auto-J [Li *et al.*, 2023] for summarization win rates and AlpacaEval 2.0 [Dubois *et al.*, 2024] for length-controlled win rates in QA.

Implementation Details. Experiments are conducted on a single NVIDIA A100 GPU with 80GB memory using the PyTorch framework. We implement FedPrism with $M = 5$ global experts initialized identically. The global router is designed as a parameter-efficient MLP that maps input semantic embeddings to 128 hidden units and subsequently to 2 output logits with Tanh activation. Training spans 400 communication rounds and we adopt LoRA for efficient tuning, configured with a rank of 8, an alpha of 16, and a dropout rate of 0.05. Optimization employs a warm-up strategy where the router remains frozen for the initial 150 rounds. We also apply a lazy update mechanism for the router every 20 rounds with a learning rate of 0.001, while the experts are updated in every round using a learning rate of 1e-5 to ensure precise and stable specialization. Our code is available at link.

4.2 Quantitative Evaluation on Summarization Task

Table 1 presents a comprehensive comparative analysis of human-written and model-generated summaries on the Reddit TL;DR dataset, quantified by the win rate against human references via Auto-J. FedPrism consistently establishes a new state-of-the-art across all experimental settings. With the LLaMA-2-7B backbone, it achieves a peak win rate of 94.34%. More notably, on the Gemma-2B backbone, FedPrism demonstrates a significant performance leap with an 89.29% win rate, delivering substantial improvements of 11.95% over FedBis and 5.48% over FedBiscuit. These results underscore the effectiveness of our instance-wise routing mechanism, which successfully directs diverse prompts to the most suitable experts, thereby maximizing the utilization of heterogeneous client knowledge better than single-model aggregation. We observe that standard approaches like FedAvg and FedDPO suffer from catastrophic performance drops, confirming that directly averaging parameters in highly non-IID scenarios leads to severe model divergence and skill forgetting. In contrast, FedPrism demonstrates exceptional robustness. Even with the lightweight Qwen-2-0.5B expert, it maintains an impressive 87.82% win rate on Gemma-2B. Furthermore, while the clustered baseline FedBiscuit exhibits performance instability with weaker experts,

Expert	Methods	Win Rate (# Wins / # Ties)	
		Gemma-2B	LLaMA-2-7B
NA	Raw Model	67.27% (4408 / 28)	76.79% (5032 / 31)
	FedAvg	28.66% (1878 / 10)	28.23% (1850 / 10)
	FedDPO	49.03% (3213 / 23)	77.02% (5047 / 30)
Qwen-2 (0.5B)	FedBis	86.69% (5681 / 26)	89.63% (5874 / 38)
	FedBiscuit	78.45% (5141 / 29)	83.29% (5458 / 48)
	FedPrism	87.82% (5755 / 29)	91.79% (6015 / 45)
Gemma (2B)	FedBis	77.34% (5068 / 39)	86.22% (5650 / 39)
	FedBiscuit	83.81% (5492 / 43)	91.32% (5984 / 28)
	FedPrism	**89.29% (5851 / 40)**	93.13% (6103 / 42)
LLaMA-2 (7B)	FedBis	82.56% (5410 / 28)	91.87% (6020 / 40)
	FedBiscuit	81.90% (5367 / 41)	90.84% (5951 / 37)
	FedPrism	86.54% (5671 / 33)	**94.34% (6182 / 43)**

Table 1: Performance comparison on the summarization task evaluated by Auto-J. Comparison is made across different backbones and various expert architectures. **Bold** denotes the best result under the specific backbone, while underline indicates the best result within a specific expert group.

FedPrism maintains consistent improvements. This indicates that our dynamic routing strategy is fundamentally superior to static clustering in handling the nuances of preference heterogeneity, as it allows for flexible, per-instance expert selection rather than rigid client grouping.

4.3 Quantitative Evaluation on QA Task

Table 2 presents the length-controlled (LC) win rates against GPT-4-Turbo on AlpacaEval 2.0. The results highlight the severe challenge of negative transfer in non-IID QA tasks, where standard FedAvg degrades performance significantly below the raw model. FedPrism effectively overcomes this hurdle, establishing a new state-of-the-art across all settings. On the LLaMA-2-7B backbone, FedPrism achieves a peak win rate of 5.92% under SafeRLHF prompts, surpassing the strongest clustered baseline by 0.29% and the raw model by over 1%. More critically, on the Gemma-2B backbone, we observe that the clustered baseline FedBiscuit underperforms the simpler FedBis. In contrast, FedPrism maintains its dominance with 4.73%. This specific case validates that our instance-wise routing is more stable and adaptable than static clustering, particularly when the backbone model capability is limited. FedPrism further demonstrates superior robustness regarding the quality of unlabeled prompts used in Phase III. As shown in Table 2, while using high-quality SafeRLHF prompts yields the best results, FedPrism remains the top performer even with noisier Reddit prompts. Notably, in the Reddit (4 completions) setting for LLaMA-2, FedPrism widens the gap against FedBiscuit, proving that its MoE router effectively filters noise and extracts valuable preference signals from diverse data streams.

4.4 Ablation Study

To verify the effectiveness of FedPrism's core components, we conduct comprehensive ablation studies on the summarization task using the Gemma-2B backbone. As detailed in Table 3, we compare the full framework against three specific variants to isolate the contribution of each module. First, we investigate the necessity of our routing mechanism. Replac-

Unlabeled Prompts	Methods	LC Win-rate (%)	
		Gemma-2B	LLaMA-2-7B
NA	Raw Model	2.40 ± 0.16	4.80 ± 0.41
	FedAvg	1.88 ± 0.12	3.93 ± 0.23
	FedDPO	3.28 ± 0.23	4.98 ± 0.30
Reddit Posts (2 completions)	FedBis	3.85 ± 0.25	5.06 ± 0.30
	FedBiscuit	3.98 ± 0.24	5.42 ± 0.32
	FedPrism	4.34 ± 0.28	5.83 ± 0.35
Reddit Posts (4 completions)	FedBis	3.66 ± 0.25	5.08 ± 0.31
	FedBiscuit	4.14 ± 0.27	5.34 ± 0.32
	FedPrism	4.60 ± 0.30	5.68 ± 0.31
SafeRLHF	FedBis	4.58 ± 0.30	5.25 ± 0.32
	FedBiscuit	4.03 ± 0.28	5.63 ± 0.34
	FedPrism	**4.73 ± 0.31**	**5.92 ± 0.34**

Table 2: Performance comparison on the QA task evaluated by AlpacaEval 2.0 (LC Win-rate). Results are reported across different unlabeled prompt sources used in Phase III. **Bold** indicates the best performance, and underline denotes the best result within a specific prompt setting.

Method / Variant	Win Rate (%)	Δ vs. Full (%)
FedPrism (Ours)	89.29	-
Core Mechanism Analysis		
1) w/o **Semantic Routing (Random)**	75.97	↓ **13.32**
2) w/o **PGPD (Fixed Router)**	83.19	↓ **6.10**
Training Strategy Analysis		
3) w/o **Warm-up Strategy**	86.46	↓ **2.83**

Table 3: Ablation analysis of FedPrism on the Summarization task. All experiments utilize Gemma-2B as both the backbone and expert model, trained for 400 rounds.

ing the semantic router with a random assignment strategy results in a precipitous 13.32% drop in win rate. This substantial gap confirms that the performance gains of FedPrism stem from the precise, instance-wise matching between prompts and experts, rather than merely from the increased parameter capacity of the expert ensemble. Second, we evaluate the PGPD mechanism. When the router is frozen after initialization, the win rate decreases by 6.10% to 83.19%. This indicates that a static router is insufficient; the router must actively evolve via posterior guidance to effectively track and adapt to the changing capabilities of local experts during the federated training process. Finally, we analyze the impact of the warm-up strategy. Removing the warm-up phase leads to a 2.83% performance decline compared to the full method. This degradation validates our hypothesis that expert representations in the early stages are noisy and unstable. A warm-up period is therefore essential to prevent the router from overfitting to these immature signals, ensuring robust convergence and long-term stability.

4.5 Convergence Analysis and Expert Utilization

We analyze the training dynamics and expert utilization on the QA task, as illustrated in Fig. 3. The convergence curves reveal distinct characteristics across methods. FedBis exhibits the sharpest initial rise, as it trains a single global model without a warm-up constraint; however, its performance eventually plateaus and lags behind, confirming the limitations of a monolithic model in handling non-IID preferences. In contrast, FedBiscuit shows the slowest convergence. We attribute this to its specific warm-up mechanism, where multiple selectors are activated and trained sequentially rather than simultaneously, leading to a delayed accumulation of effective knowledge. For FedPrism, we observe a significant trade-off governed by the expert count M. Increasing M from 3 to 7 slows down the initial convergence. This phenomenon stems from local data dilution: partitioning a client's limited dataset among more experts results in fewer training samples per expert per round, causing early-stage under-fitting. However, as training progresses, models with higher M surpass those with lower M, as the increased capacity allows for finer-grained modeling of data heterogeneity. Notably, $M = 5$ strikes an optimal balance, achieving near-peak performance with superior efficiency compared to $M = 7$. To verify the router's stability, we visualize the activation distribution of the 5 experts on 55,327 test samples. The stacked bar chart demonstrates that expert utilization remains highly balanced from round 100 to 400. Unlike typical MoE failures where the router collapses to selecting a single expert, our semantic router consistently distributes diverse prompts across all available experts. This stability confirms that the router has successfully learned to disentangle prompt semantics and dispatch them to the most appropriate specialists throughout the training process.

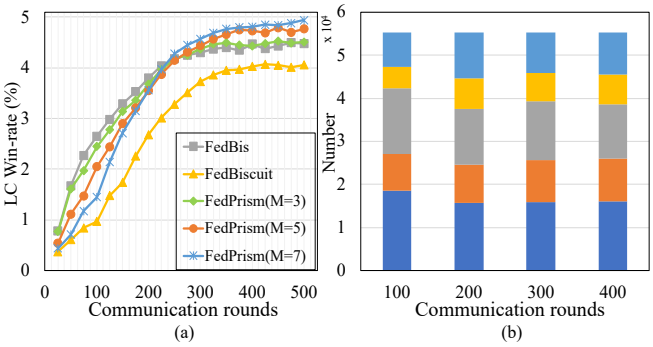

Figure 3: (a) Convergence curves of LC Win-rate on the QA task (Gemma-2B) across different methods and expert counts. (b) Evolution of expert activation counts for FedPrism ($M = 5$) evaluated on the test set across communication rounds.

5 Conclusion

In this paper, we propose FedPrism, a framework that addresses the challenge of preference heterogeneity in federated RLHF by transforming it from a liability into an asset via a MoE architecture. By employing an instance-wise semantic router optimized through PGPD, FedPrism effectively disentangles conflicting preferences into specialized expert capabilities without requiring explicit labels. Our extensive evaluations confirm that this approach consistently establishes a new state-of-the-art across diverse benchmarks, successfully mitigating the negative transfer issues prevalent in standard federated methods. Ultimately, our findings demonstrate that dynamic, instance-wise routing offers a significantly more robust and effective paradigm than static clustering for collaborative alignment under non-IID conditions.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No.62472431, No.U25A20429) and the Hunan Provincial Science Fund for Distinguished Young Scholars (No.2025JJ20066).

References

[Achiam *et al.*, 2023] 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.

[Christiano *et al.*, 2017] 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.

[Dai *et al.*, 2023] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

[Dubois *et al.*, 2024] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

[Ethayarajh *et al.*, 2022] Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning*, pages 5988–6008. PMLR, 2022.

[Fan *et al.*, 2023] Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. Fate-llm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*, 2023.

[Gao *et al.*, 2025] Liang Gao, Li Li, Yingwen Chen, Shaojing Fu, Dongsheng Wang, Siwei Wang, Cheng-Zhong Xu, and Ming Xu. Noise-robust federated learning via inter-client co-distillation. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[Ghosh *et al.*, 2020] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. *Advances in neural information processing systems*, 33:19586–19597, 2020.

[Houlsby *et al.*, 2019] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[Hu *et al.*, 2022] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[Jiang *et al.*, 2024] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[Jiang *et al.*, 2025] Wenhao Jiang, Yuchuan Luo, Guilin Deng, Silong Chen, Xu Yang, Shihong Wu, Xinwen Gao, Lin Liu, and Shaojing Fu. Federated large language models: Feasibility, robustness, security and future directions. *arXiv preprint arXiv:2505.08830*, 2025.

[Kuang *et al.*, 2024] 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*, pages 5260–5271, 2024.

[Li *et al.*, 2023] Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*, 2023.

[Lin *et al.*, 2023] 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.

[McMahan *et al.*, 2017] 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*, pages 1273–1282. PMLR, 2017.

[Ouyang *et al.*, 2022] 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.

[Rafailov *et al.*, 2023] 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.

[Santurkar *et al.*, 2023] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In *International Conference on Machine Learning*, pages 29971–30004. PMLR, 2023.

[Sattler *et al.*, 2020] Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Prox-

imal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Shazeer *et al.*, 2017] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[Sun *et al.*, 2023] Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yiran Chen, and Holger R Roth. Fedbpt: Efficient federated black-box prompt tuning for large language models. *arXiv preprint arXiv:2310.01467*, 2023.

[Sun *et al.*, 2024] Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving lora in privacy-preserving federated learning. *arXiv preprint arXiv:2403.12313*, 2024.

[Team *et al.*, 2024] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[Touvron *et al.*, 2023] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[Voigt and Von dem Bussche, 2017] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A practical guide, 1st ed., Cham: Springer International Publishing*, 10(3152676):10–5555, 2017.

[Völske *et al.*, 2017] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, 2017.

[Wu *et al.*, 2025a] Feijie Wu, Xiaoze Liu, Haoyu Wang, Xingchen Wang, Lu Su, and Jing Gao. Towards federated rlhf with aggregated client preference for llms. In *ICLR*, 2025.

[Wu *et al.*, 2025b] Yebo Wu, Chunlin Tian, Jingguang Li, He Sun, Kahou Tam, Zhanting Zhou, Haicheng Liao, Zhijiang Guo, Li Li, and Chengzhong Xu. A survey on federated fine-tuning of large language models. *arXiv preprint arXiv:2503.12016*, 2025.

[Ye *et al.*, 2024] 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*, pages 6137–6147, 2024.

[Yi *et al.*, 2023] 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.

[Yi *et al.*, 2024] Liping Yi, Han Yu, Chao Ren, Heng Zhang, Gang Wang, Xiaoguang Liu, and Xiaoxiao Li. pfedmoe: Data-level personalization with mixture of experts for model-heterogeneous personalized federated learning. *arXiv preprint arXiv:2402.01350*, 2024.

[Zec *et al.*, 2020] Edvin Listo Zec, Olof Mogren, John Martinsson, Leon René Sütfeld, and Daniel Gillblad. Specialized federated learning using a mixture of experts. *arXiv preprint arXiv:2010.02056*, 2020.

[Zhang *et al.*, 2023] Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Annual Meeting of the Association of Computational Linguistics 2023*, pages 9963–9977. Association for Computational Linguistics (ACL), 2023.

[Zhang *et al.*, 2024] 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 *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6915–6919. IEEE, 2024.

[Zhao *et al.*, 2023] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.