MoF: Preference-Aware Mixture Modeling for Black-Box LLM Personalization

Hun Park
Department of Artificial Intelligence
Korea University
Seoul, Republic of Korea
hunbag98@korea.ac.kr

Abstract

Proprietary Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks, yet aligning their outputs with diverse user preferences remains challenging. Existing personalization approaches for black-box LLMs often rely on user-specific scoring heads, causing the number of personalized parameters to grow linearly with the number of users and requiring additional adaptation for unseen users. To address these limitations, we propose **Mixture-of-Facets (MoF)**[1], a scalable personalization framework for black-box LLMs that models user preferences as compositions of shared latent preference facets rather than dedicated user-specific parameters. MoF performs personalization through history-conditioned routing over shared facet heads, enabling personalization for users unseen during training without additional parameter updates. Across diverse personalization tasks, MoF delivers stronger personalization performance while maintaining a more scalable and parameter-efficient design than prior approaches. Additional analysis indicates strong generalization to unseen users.

1 Introduction

Large Language Models (LLMs) have demonstrated strong performance across a wide range of tasks, from general natural language processing to complex reasoning (OpenAI, 2022; OpenAI et al., 2024; Comanici et al., 2025). As LLM-based personal assistants become increasingly widespread (Mok et al., 2025; Li et al., 2025; Anthropic, 2026), there has been growing interest in *LLM personalization*, i.e., tailoring model responses to individual user preferences (Zhang et al., 2025b; Salemi et al., 2024). Such personalization is motivated by the observation that users may prefer different responses to the same query depending on their preferences (Kirk et al., 2024a,b; Chen et al., 2024). In this work, we view user preference as a relative tendency to favor one response over another for a given query and aim to steer LLM outputs toward responses that better align with such preferences.

Despite their strong capabilities, most LLMs are trained using a *one-size-fits-all* paradigm through next-token prediction (Brown et al., 2020; Chung et al., 2024) and are subsequently aligned with general human preferences during post-training (Ouyang et al., 2022). As a result, LLMs tend to generate outputs that reflect dominant or average preferences, which creates a fundamental challenge in aligning LLM outputs with diverse human preferences (Jang et al., 2023; Sorensen et al., 2024; Feng et al., 2024; Zhou et al., 2024).

To address this challenge, prior work has explored parameter-efficient fine-tuning (PEFT) approaches, such as training user-specific LoRA (Hu et al., 2022) modules (Tan et al., 2024b; QI et al., 2024) or incorporating user embeddings into model inputs (Liu et al., 2025; Qiu et al., 2025). While effective, these methods require direct access to model parameters and are therefore not applicable to *black-box LLMs*, which are increasingly deployed through APIs (Sun et al., 2024). To enable personalization for black-box settings, recent work has explored retrieval- and prompting-based approaches (Salemi et al., 2024; Richardson et al., 2023; Shi et al., 2025), which avoid model adaptation but remain vulnerable to irrelevant information. To provide more robust personalization, HYDRA (Zhuang et al., 2024) introduces a **reranker & adapter** framework that personalizes black-box LLM outputs by selecting useful historical items and ranking candidate responses according to user preferences. However, HYDRA relies on user-specific scoring heads, causing the number of personalized parameters to grow linearly with the number of users and requiring additional adaptation for

[1]Code is publicly available at https://github.com/hoonpak/Mixture-of-Facets.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 12290–12309
October 24-29, 2026 ©2026 Association for Computational Linguistics

unseen users. These limitations become increasingly problematic in practical deployments with large user populations.

To overcome these limitations, we propose **Mixture-of-Facets (MoF)**, a preference-aware mixture architecture inspired by the Mixture-of-Experts (MoE) framework (Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022). Rather than modeling user preferences through user-specific parameters, MoF assumes that user preferences can be represented as compositions of shared latent preference facets. Based on this idea, MoF shifts from user-specific scoring heads to shared facet heads, with personalization performed through history-conditioned routing. To encourage the router to focus on preference-related signals, MoF leverages a sparse autoencoder (SAE) (Huben et al., 2024; Qiu et al., 2025) over user histories. This design allows scalable personalization with a fixed number of shared parameters and generalizes to users unseen during training without additional parameter updates. To further promote facet specialization, we introduce a cluster-based sampling (CBS) strategy that exposes facet heads to diverse preference signals during training.

Experimental results across multiple personalization tasks demonstrate that MoF achieves the best overall performance in terms of average rank, while improving scalability and parameter efficiency. Additional analysis shows that MoF maintains competitive performance even without additional adaptation, indicating strong generalization to unseen users. Our key contributions are summarized as follows:

- We propose MoF, a scalable black-box LLM personalization framework that models user preferences as compositions of shared latent preference facets.
- We introduce a history-conditioned routing mechanism that enables personalization for unseen users without requiring user-specific adaptation.
- We analyze the generalization and scalability of MoF, showing competitive performance on unseen users and constant personalization-parameter storage overhead as the number of users increases.

2 Related Work

White-Box LLM Personalization. LLM personalization is defined as the process of adapting a model's responses to individual users based on their profiles, characteristics, preferences, and interaction history (Zhang et al., 2025b). To achieve this, PEFT-based personalization methods commonly utilize techniques such as LoRA (Hu et al., 2022). OPPU (Tan et al., 2024b) assigns each user a personalized LoRA module to capture user-specific preferences. PROPER (Zhang et al., 2025a) further improves personalization by combining parameter composition with MoE-style routing. However, maintaining separate adapters for individual users can introduce substantial parameter and storage overhead. To improve parameter sharing, PER-PCS (Tan et al., 2024a) decomposes personalized PEFT modules into reusable parameter pieces and constructs user-specific adapters by composing components from a shared parameter pool.

Although PROPER and PER-PCS employ compositional parameterization, their components differ from the preference facets considered in MoF. PROPER models group-level user adaptations, while PER-PCS composes reusable PEFT parameter pieces to construct personalized adapters. In contrast, MoF represents individual preferences as weighted compositions of jointly learned shared latent preference facets, with routing weights derived from user history. More importantly, both PROPER and PER-PCS require access to backbone parameters and are thus limited to white-box settings, whereas MoF operates without such access and is applicable to black-box LLMs.

Black-Box LLM Personalization. To address this challenge, prior work has explored prompting-based personalization methods. LaMP (Salemi et al., 2024) introduces retrieval augmentation for personalization by retrieving personal items from user profiles. CFRAG (Shi et al., 2025) extends this direction by incorporating collaborative filtering and retrieving histories from similar users. PAG (Richardson et al., 2023) combines retrieval with task-aware user summaries generated by LLMs to improve personalization. Fermi (Kim and Yang, 2025) learns personalized prompts for each user by iteratively refining prompts using user profiles. Although these methods improve personalization without modifying model parameters, their effectiveness can be affected by irrelevant or noisy contextual signals.

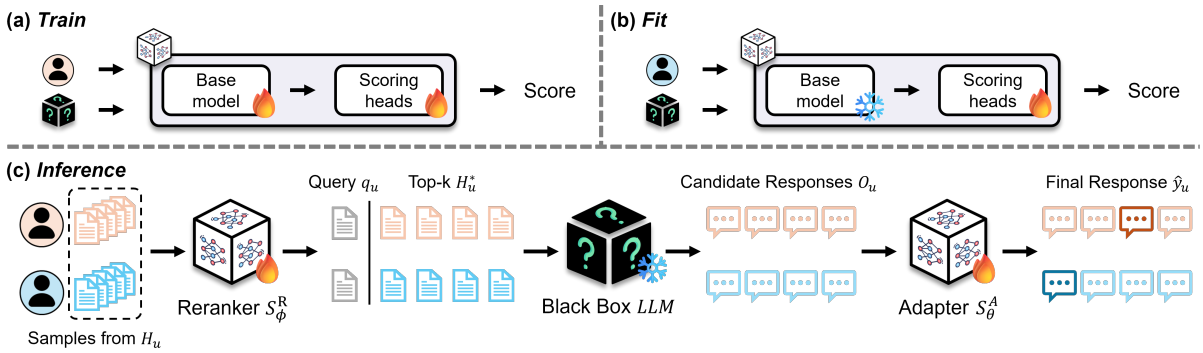

Figure 1: Overview of the training and inference pipeline. **(a)** ***Train stage*:** the shared base model and scoring heads are jointly trained using data collected from multiple users. **(b)** ***Fit stage*:** the base model is frozen while only the scoring heads are updated for unseen users. **(c)** ***Inference stage*:** the reranker selects useful history items, the black-box LLM generates candidate responses, and the adapter scores the candidates to produce the final response.

To further advance black-box LLM personalization, HYDRA (Zhuang et al., 2024) reranks user histories and selects preference-aligned responses via a personalized adapter. However, HYDRA relies on user-specific scoring heads (Fig. 2 (a)), causing personalized parameters to grow linearly with the number of users and requiring an additional adaptation stage for unseen users (Fig. 1 (b)). In contrast, MoF enables scalable personalization through shared facet heads and history-conditioned routing, supporting personalization for unseen users without additional adaptation while requiring no dedicated user-specific parameters.

3 Methodology

3.1 Preliminary

Given a user set $\mathcal{U} = \{u_1, \ldots, u_N\}$, the goal of LLM personalization is to generate a response $\hat{y}_u$ aligned with user preferences for a query q_u. For each user u, the behavior history is defined as $H_u = \{h_u^1, \ldots, h_u^T\}$, where each history item $h_u^t = (q_u^t, y_u^t)$ consists of a past query q_u^t and the corresponding user-preferred response y_u^t. Since different users may prefer different responses to the same query, personalized response generation requires leveraging user-specific preference information extracted from historical interactions.

Following prior work (Zhuang et al., 2024), we adopt a reranker & adapter personalization framework for black-box LLMs. Given a target query q_u, relevant history items $H'_u = \{h'_{u,1}, \ldots, h'_{u,M}\}$ are retrieved from H_u and then reranked by a **reranker** s_ϕ^{R} to construct a personalized context set H_u^*. Conditioned on H_u^*, the black-box LLM generates a set of candidate responses $O_u = \{o_u^1, \ldots, o_u^B\}$. An **adapter** s_θ^{A} scores the O_u according to their alignment with user preferences, and the highest-scoring response is selected as the final output $\hat{y}_u$. An overview of the framework is illustrated in Figure 1 (c). Detailed formulations of the reranker and adapter objectives are provided in Appendix A.

3.2 Mixture-of-Facets (MoF)

Building on the reranker & adapter framework introduced above, we apply **Mixture-of-Facets (MoF)** to both the reranker and adapter (Fig. 2). MoF consists of two key components: (1) the Facet Heads, which estimate facet-specific scores (§3.2.1), and (2) the Preference-Aware Router, which computes routing weights based on user preferences (§3.2.3). To encourage facet specialization during training, we further introduce a cluster-based sampling (CBS) strategy (§3.3).

3.2.1 Facet Heads

Each facet head is implemented as a two-layer perceptron. We intentionally use lightweight two-layer MLPs to isolate the effect of shared-facet composition while preserving parameter and computational efficiency. Given an input pair $x^R = (q_u, h'_{u,m})$ for the reranker or $x^A = (q_u, o_u^b)$ for the adapter, each facet head produces a preference logit. As illustrated in Figure 2 (b), the input x is first encoded by a BERT (Devlin et al., 2019)-based encoder δ, which is jointly trained with the facet heads. We denote the final hidden representation of the [CLS] token as $\mathbf{z}_x = \delta(x)_{\texttt{[CLS]}}$. Each facet head ψ_f takes $\mathbf{z}_x$ as input and computes the corresponding preference logit:

$$\psi_f(\mathbf{z}_x) = \mathbf{W}_f^{(2)} \tanh\left(\mathbf{W}_f^{(1)}\mathbf{z}_x + \mathbf{b}_f^{(1)}\right) + \mathbf{b}_f^{(2)}. \quad (1)$$

Here, $\mathbf{W}_f^{(1)} \in \mathbb{R}^{d\times d}$ and $\mathbf{W}_f^{(2)} \in \mathbb{R}^{o\times d}$ denote the learnable weight matrices, where d is the hidden representation dimension and o denotes the dimension of the output logits.

The model contains a total of F facet heads, each encouraged to specialize in different latent preference patterns during training. The outputs of all facet heads are aggregated into the following facet logit matrix:

$$\Psi(x) = [\psi_1(\mathbf{z}_x), \ldots, \psi_F(\mathbf{z}_x)] \in \mathbb{R}^{F\times o}. \quad (2)$$

Each row of $\Psi(x)$ represents the output logits generated by a corresponding facet head.

3.2.2 Sparse User Representation

Independently of computing $\Psi(x)$, the router constructs a user-specific preference representation from H_u in order to estimate routing weights for aggregating $\Psi(x)$. We intentionally exclude query representations from the router input so that routing decisions focus on persistent user preference characteristics rather than query-specific semantics. Each history item is encoded using the shared encoder δ, which is kept frozen during router training. For each history item h_u^t, we obtain the corresponding history embedding as $\mathbf{e}_u^t = \delta(h_u^t)_{\texttt{[CLS]}}$. The resulting historical embedding set for user u is defined as $\mathbf{E}_u = \{\mathbf{e}_u^1, \ldots, \mathbf{e}_u^T\}$. To capture patterns within the user history, we apply a Transformer encoder over $\mathbf{E}_u$ and compute the user representation via mean pooling:

$$\mathbf{e}'_u = \text{Mean}\left(\text{Encoder}(\mathbf{E}_u)\right). \quad (3)$$

The dense user representation $\mathbf{e}'_u$ contains general semantic and syntactic information that is not directly relevant to user preferences. To facilitate the router's focus on preference-related signals, we further transform the dense representation into a sparse latent representation using a Sparse Autoencoder (SAE) (Huben et al., 2024). The SAE projects the dense user representation into a higher-dimensional sparse latent space, encouraging preference-related activation patterns to be selectively represented across different latent dimensions. The sparse latent representation is computed as

$$\mathbf{e}_u^{enc} = \text{ReLU}\left(\mathbf{W}_{\text{enc}}\mathbf{e}'_u + \mathbf{b}_{\text{enc}}\right), \quad (4)$$

where $\mathbf{e}_u^{enc} \in \mathbb{R}^{\tau d}$ and $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{\tau d\times d}$. Here, τ denotes the latent expansion factor that controls the dimensionality of the sparse representation. This latent space provides additional capacity for sparse activations, facilitating the selective representation of preference-related signals. The resulting sparse representation $\mathbf{e}_u^{enc}$ is used as the router input representation.

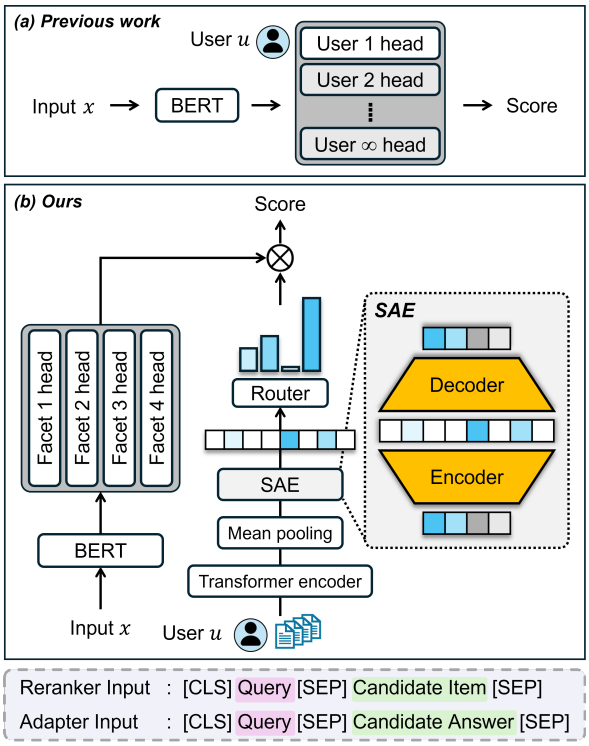

Figure 2: Comparison between (a) user-specific head personalization and (b) the proposed MoF framework. Unlike previous work, MoF combines shared facet heads through preference-aware routing.

The SAE decoder reconstructs the original dense representation from the sparse latent representation:

$$\mathbf{e}_u^{dec} = \mathbf{W}_{\text{dec}}\mathbf{e}_u^{enc} + \mathbf{b}_{\text{dec}}, \quad (5)$$

where $\mathbf{e}_u^{dec} \in \mathbb{R}^d$ and $\mathbf{W}_{\text{dec}} \in \mathbb{R}^{d\times\tau d}$. The SAE is optimized using the following objective:

$$\mathcal{L}_{\text{SAE}} = \left\|\mathbf{e}_u^{dec} - \mathbf{e}'_u\right\|_2^2 + \lambda_1 \left\|\mathbf{e}_u^{enc}\right\|_1. \quad (6)$$

The reconstruction objective preserves preference-relevant information encoded in the user-history representation, while the L_1 regularization promotes sparsity in the latent representation.

3.2.3 User Preference-Aware Router

Using the sparse user representation $\mathbf{e}_u^{enc}$, the router determines which facet heads are more relevant for a particular user. The router is implemented as a linear projection layer that produces routing logits over the F facet heads:

$$\mathbf{g}_u = \mathbf{W}_r\mathbf{e}_u^{enc} + \mathbf{b}_r, \quad (7)$$

where $\mathbf{W}_r \in \mathbb{R}^{F \times \tau d}$ denotes the routing projection matrix and $\mathbf{g}_u \in \mathbb{R}^F$ represents the routing logits over all facet heads. To encourage sparse activation of facet heads, we apply top-k routing over $\mathbf{g}_u$:

$$(\tilde{\mathbf{g}}_u, \mathcal{K}_u) = \text{top-}k(\mathbf{g}_u, K), \tag{8}$$

where $\tilde{\mathbf{g}}_u \in \mathbb{R}^K$ denotes the selected top-k routing logits and $\mathcal{K}_u$ represents the corresponding facet indices. The routing weights are then computed using a softmax over the selected routing logits:

$$\alpha_{u,k} = \frac{\exp(\tilde{g}_{u,k})}{\sum_{j=1}^{K} \exp(\tilde{g}_{u,j})}. \tag{9}$$

The final MoF prediction is obtained as a weighted sum of the selected facet outputs using the routing weights:

$$\text{MoF}(x, H_u) = \sum_{k=1}^{K} \alpha_{u,k}\, \psi_{\mathcal{K}_{u,k}}(x). \tag{10}$$

This sparse routing mechanism enables the model to represent user-specific preference patterns as compositions of shared facet heads, allowing personalization without dedicated user-specific heads.

3.2.4 Objective

The model is trained using a binary classification objective over the MoF prediction and the corresponding supervision label r_i^u. The labeling strategy is described in Appendix A. For a user u and training instance (x_i^u, r_i^u), the prediction probability is computed as

$$p_i^u = \sigma(\text{MoF}(x_i^u, H_u)). \tag{11}$$

The MoF objective is defined as

$$\mathcal{L}_{\text{MoF}} = -r_i^u \log p_i^u - (1 - r_i^u) \log(1 - p_i^u). \tag{12}$$

The final training objective jointly optimizes the MoF objective and the sparse autoencoder objective:

$$\mathcal{L} = \mathcal{L}_{\text{MoF}} + \lambda \mathcal{L}_{\text{SAE}}. \tag{13}$$

Here, λ controls the contribution of the SAE regularization objective during training.

3.3 Cluster-Based Sampling (CBS)

In HYDRA, training instances are constructed by retrieving history items based on relevance to the target query. However, retrieval relevance often emphasizes lexical or semantic similarity, which may fail to reflect the diverse aspects of user preferences. As a result, the sampled training instances may be concentrated on a limited subset of preference patterns. To encourage facet specialization in MoF, we introduce a cluster-based sampling (CBS) strategy over H_u. Specifically, we first partition history items into multiple clusters using k-means clustering over the history embedding set $\mathbf{E}_u$. CBS is applied when constructing training instances for both the reranker and adapter. Rather than sampling history items from a single global retrieval pool, CBS samples them separately from each cluster. This strategy increases the diversity of preference signals observed during training and encourages the facet heads to capture diverse preference patterns rather than focusing on semantically similar history items.

4 Experiment

4.1 Experimental Setup

Datasets. We evaluate MoF on two personalization classification tasks and two personalization generation tasks from the LaMP benchmark (Salemi et al., 2024). Specifically, we use **Movie Tagging Classification (LaMP-2)** and **Product Rating Classification (LaMP-3)** for classification, and **News Headline Generation (LaMP-4)** and **Scholarly Title Generation (LaMP-5)** for generation. Further details on the datasets are provided in Appendix C.

Baselines. We compare MoF against both prompt-based and reranker & adapter-based personalization methods. **Zero-Shot** uses only the input query without any personalization cues. **ICL** augments the input with randomly selected k items from the user history. **RAG** (Salemi et al., 2024) retrieves relevant history items using BM25 (Robertson and Zaragoza, 2009) and augments them into the input. **PAG** (Richardson et al., 2023) further enhances personalization by incorporating LLM-generated user profile summaries together with retrieved history items. **CFRAG** (Shi et al., 2025) incorporates collaborative information by retrieving similar users and relevant items from their histories, followed by personalized reranking for LLM generation. **HYDRA** (Zhuang et al., 2024) is a reranker & adapter-based personalization framework. We additionally report **HYDRA-Reranker**, which uses only the reranker for personalization, and **HYDRA-Adapter**, which applies only the adapter over candidate responses generated from the zero-shot setting.

Table 1: Main results on the LaMP benchmark. Results are averaged over three runs with different sampled user sets. k denotes the number of retrieved user-history items. For HYDRA and MoF, we use $k = 4$. ↑ and ↓ indicate that higher and lower values are better, respectively. **Bold** and underline indicate the best and second-best scores, respectively. Avg. Rank denotes the average rank of each method across all metrics.

Dataset (→)	**LaMP-2**		**LaMP-3**		**LaMP-4**			**LaMP-5**			**Avg. Rank↓**
Method (↓)	Acc ↑	F1 ↑	MAE ↓	RMSE ↓	R-1 ↑	R-L ↑	BLEU ↑	R-1 ↑	R-L ↑	BLEU ↑	
Zero-Shot	0.420	0.257	0.500	0.777	0.147	0.129	0.891	0.446	0.375	5.822	13.5
ICL (k=1)	0.393	0.252	0.727	1.163	0.158	0.140	0.977	0.444	0.373	6.098	13.6
ICL (k=2)	0.467	0.387	0.520	0.879	0.167	0.143	0.837	0.448	0.381	6.032	11.3
ICL (k=4)	0.513	0.403	0.433	0.762	0.161	0.140	1.240	0.447	0.384	5.511	10.0
RAG (k=1)	0.433	0.276	0.613	0.978	0.165	0.145	0.889	0.452	0.385	5.961	12.0
RAG (k=2)	0.493	0.346	0.500	0.812	0.167	0.150	1.273	0.459	0.387	5.388	9.5
RAG (k=4)	0.547	0.409	0.440	0.765	0.176	0.148	1.456	0.457	0.395	5.888	6.9
PAG (k=0)	0.507	0.366	0.333	0.663	0.162	0.140	0.873	0.436	0.361	5.659	10.5
PAG (k=1)	0.473	0.337	0.400	0.739	0.181	0.157	1.040	0.454	0.371	6.063	8.0
CFRAG (k=4)	0.593	0.447	0.440	0.780	0.176	0.151	0.966	0.456	0.392	5.748	7.4
HYDRA-Reranker	0.567	0.383	0.480	0.830	0.168	0.152	1.517	0.465	0.390	6.281	6.8
HYDRA-Adapter	0.447	0.271	0.380	0.735	0.159	0.140	0.852	0.467	0.394	6.059	9.3
HYDRA	0.593	0.417	0.393	0.761	0.183	0.163	**1.849**	0.459	0.396	6.652	3.5
MoF-Reranker (ours)	0.600	**0.451**	0.407	0.724	0.177	0.154	1.502	0.456	0.390	**7.421**	4.2
MoF-Adapter (ours)	0.467	0.308	0.327	0.682	0.161	0.142	1.275	0.477	0.400	6.515	6.5
MoF (ours)	**0.627**	0.438	**0.300**	**0.605**	**0.199**	**0.176**	1.662	**0.481**	**0.413**	6.967	**1.4**

Evaluation Metrics. Following LaMP (Salemi et al., 2024), we use Accuracy (Acc) and F1 score (F1) for LaMP-2, MAE and RMSE for LaMP-3, and ROUGE-1 (R-1), ROUGE-L (R-L), and BLEU for LaMP-4 and LaMP-5. Higher scores indicate better performance for all metrics except MAE and RMSE.

Implementation Details. Following prior work (Zhuang et al., 2024), we conduct training and evaluation using subsets of the original datasets, where we sample 100 users for training and 50 users for testing. For robustness, results are averaged over 3 runs with different randomly sampled user sets. We use `bge-base-en-v1.5` (Xiao et al., 2024) (110M parameters) as the base model for both the reranker and adapter modules. As the black-box LLM, we employ `gpt-3.5-turbo-1106` (OpenAI, 2022). Additional implementation details are provided in Appendix F.

4.2 Main Results

Table 1 presents the main experimental results on the LaMP benchmark. We observe that prompt-based personalization methods do not consistently improve over the zero-shot setting and in some cases even degrade performance. One possible explanation is that prompt-based methods are sensitive to history selection and context coverage because randomly selected or limited histories may fail to represent the user's preferences while introducing distracting context. The extent of this effect also varies across tasks and retrieval sizes. In contrast, reranker & adapter-based approaches generally provide more consistent gains across tasks, suggesting that aligning the output distribution of a black-box LLM with user preferences can be more effective than relying solely on prompt-level personalization.

Compared with HYDRA, which relies on user-specific scoring heads, MoF achieves better overall performance despite eliminating user-specific parameters. For example, MoF improves accuracy from 0.593 to 0.627 on LaMP-2 and reduces MAE from 0.393 to 0.300 on LaMP-3.

MoF achieves the strongest average performance across the tasks with an average rank of 1.4, demonstrating stronger overall performance than existing prompt-based and reranker & adapter-based baselines. Specifically, MoF achieves the best performance on most evaluation metrics across LaMP-2, LaMP-3, LaMP-4, and LaMP-5, with competitive results on the remaining metrics. These results demonstrate the effectiveness of preference-aware

Table 2: Evaluation of unseen user generalization: performance with and without the fit stage. – indicates that the method cannot be applied without a fit stage.

Dataset (→)		LaMP-2	LaMP-3	LaMP-4	
Method (↓)	**Fit**	Acc ↑	MAE ↓	R-1 ↑	R-L ↑
HYDRA	✗	–	–	–	–
HYDRA	✓	0.593	0.393	0.183	0.163
MoF	✗	0.620	0.333	**0.200**	**0.179**
MoF	✓	**0.627**	**0.300**	0.199	0.176
HYDRA-Reranker	✗	–	–	–	–
HYDRA-Reranker	✓	0.567	0.480	0.168	0.152
MoF-Reranker	✗	0.547	0.440	**0.178**	**0.158**
MoF-Reranker	✓	**0.600**	**0.407**	0.177	0.154
HYDRA-Adapter	✗	–	–	–	–
HYDRA-Adapter	✓	0.447	0.380	0.159	0.140
MoF-Adapter	✗	0.440	0.347	0.153	0.137
MoF-Adapter	✓	**0.467**	**0.327**	**0.161**	**0.142**

mixture modeling for black-box LLM personalization.

We further observe that both MoF-Reranker and MoF-Adapter individually achieve competitive performance. This suggests that preference-aware retrieval and personalized candidate selection provide complementary benefits. Additional experimental results on other black-box LLMs are provided in Appendix G, suggesting that the proposed framework generalizes across different proprietary black-box LLMs.

4.3 Generalization to Unseen Users

Unlike HYDRA, which requires an additional fit stage for unseen users because of its user-specific heads, MoF can perform personalization without additional training for new users. Table 2 presents the generalization performance on unseen users with and without the fit stage. MoF maintains competitive performance even without the fit stage, with only minor degradation across datasets. Notably, MoF without the fit stage outperforms HYDRA with the fit stage on all metrics reported in Table 2, demonstrating strong generalization to unseen users. While MoF-Adapter does not outperform its fitted counterpart in every case, it remains competitive without additional training. These findings suggest that MoF can effectively generalize to unseen users without requiring user-specific adaptation.

4.4 Scalability Analysis

Figure 3 (a) and (b) show the effect of the number of training users. Across both datasets, MoF achieves competitive performance with substantially fewer training users than HYDRA. For example, MoF trained with only 30 users in (a) and 50 users in (b) already surpasses HYDRA trained with 100 users, suggesting that MoF can achieve strong personalization performance with fewer training users. Furthermore, performance remains stable as the number of training users increases up to 300, suggesting that the learned shared facets remain effective as the training user population grows.

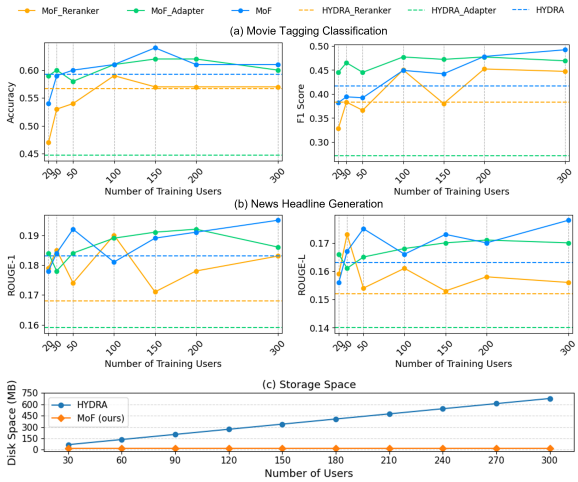

Figure 3: Scalability analysis with increasing user populations. (a), (b) show personalization performance as the number of training users increases, while (c) compares storage growth across methods.

Table 3: Large-scale evaluation on 1,000 unseen users without user-specific fitting.

Dataset (→)	LaMP-2	LaMP-3	LaMP-4	
Method (↓)	Acc ↑	MAE ↓	R-1 ↑	R-L ↑
Zero-Shot	0.391	0.515	0.143	0.124
RAG ($k = 1$)	0.409	0.676	0.155	0.135
RAG ($k = 2$)	0.480	0.521	0.160	0.141
RAG ($k = 4$)	0.508	0.449	0.164	0.146
MoF (ours)	**0.577**	**0.354**	**0.175**	**0.156**

Figure 3 (c) compares storage growth as the number of users increases. While HYDRA requires user-specific scoring heads, causing storage cost to grow linearly with the user population (approximately 2.26 MB per user in our implementation), MoF maintains constant personalization-parameter storage cost through shared facet heads. Appendix B further presents a theoretical analysis showing that MoF exhibits constant personalization complexity with respect to the number of users.

To assess scalability to a larger user population, we evaluate MoF, trained on 100 users, on 1,000 unseen users without additional user-specific fit-

Table 4: Controlled comparison of HYDRA and MoF with and without CBS.

Dataset (→)		LaMP-2	LaMP-3	LaMP-4	
Method (↓)	**CBS**	Acc ↑	MAE ↓	R-1 ↑	R-L ↑
HYDRA	✗	0.593	0.393	0.183	0.163
HYDRA	✓	0.560	0.373	0.170	0.149
MoF	✗	0.587	0.380	0.183	0.168
MoF	✓	**0.627**	**0.300**	**0.199**	**0.176**

Table 5: Effect of applying CBS to different components of MoF. R and A denote the reranker and adapter, respectively.

	CBS		LaMP-2	LaMP-3	LaMP-4	
	R	**A**	Acc ↑	MAE ↓	R-1 ↑	R-L ↑
MoF	✗	✗	0.587	0.380	0.183	0.168
	✓	✗	0.613	0.327	0.179	0.155
	✓	✓	**0.627**	**0.300**	**0.199**	**0.176**

ting. As shown in Table 3, MoF consistently outperforms the zero-shot and RAG baselines on LaMP-2, LaMP-3, and LaMP-4. These results provide further evidence of inference-time scalability and unseen-user generalization to a substantially larger user population.

4.5 Controlled Analysis of CBS

To disentangle the effect of CBS from that of the MoF architecture, we conduct controlled comparisons with and without CBS. As shown in Table 4, applying CBS to HYDRA degrades performance on most reported metrics, whereas combining CBS with MoF improves performance across the reported metrics. Notably, MoF without CBS remains competitive with HYDRA, suggesting that CBS alone does not account for the gains of MoF. Rather, CBS complements the shared-facet architecture by exposing facet heads to diverse preference signals, encouraging specialization across facets.

We further examine where CBS contributes within MoF by selectively applying it to the reranker and adapter. As shown in Table 5, applying CBS to the reranker improves performance on LaMP-2 and LaMP-3 but slightly decreases the ROUGE metrics on LaMP-4. Extending CBS to both the reranker and adapter further improves LaMP-3 and recovers and improves the ROUGE metrics on LaMP-4. These results suggest that extending CBS to both components can provide additional benefits within the MoF framework.

Table 6: Ablation study of MoF.

Dataset (→)	LaMP-2	LaMP-3	LaMP-4	
Method (↓)	Acc ↑	MAE ↓	R-1 ↑	R-L ↑
MoF	**0.627**	**0.300**	**0.199**	**0.176**
w/o SAE	0.607	0.333	0.188	0.165
w/o Router	0.600	0.360	0.184	0.161
w/o Multi-Facet	0.607	0.353	0.183	0.160
w/o CBS	0.587	0.380	0.183	0.168

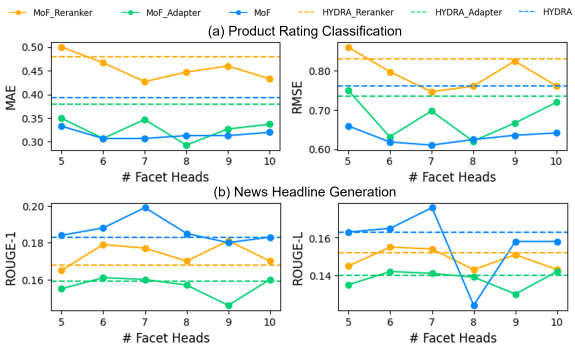

Figure 4: Impact of the number of facet heads under top-3 routing. (a) and (b) show results on LaMP-3 and LaMP-4, respectively. A moderate number of facet heads generally provides the strongest performance.

4.6 Ablation Studies

Table 6 presents an ablation study evaluating the contribution of each component in MoF. Removing CBS results in the largest performance degradation in most settings, indicating the importance of exposing the model to diverse training samples during preference learning. We also observe substantial performance drops when removing the router, demonstrating the benefit of adaptively combining shared preference facets. Replacing multiple shared facet heads with a single head also degrades performance, suggesting that user preferences are better captured by combinations of multiple latent preference patterns. Finally, removing SAE results in smaller but consistent performance degradation across datasets, indicating that sparse representations provide complementary benefits for personalization.

4.7 Impact of Number of Facet Heads

Figure 4 presents the impact of varying the number of facet heads under top-3 routing. We observe that performance generally improves as the number of facet heads increases up to a moderate range, while further increasing the number of heads does not consistently provide additional gains. For LaMP-3 in Figure 4 (a), MoF generally achieves favor-

Table 7: Qualitative analysis of user groups identified from routing patterns. Representative n-grams are obtained using BERTopic on user histories within each group.

ID	Representative n-grams	Interpretation
$G1$	"17th century salem hester prynne"; "years old daughter maddy"; "abuse fanatically pious mother"; "wormhole surpass limitations human space"	Threat-, escape-, and conflict-centered narratives spanning historical, familial, supernatural, and speculative settings
$G2$	"zion defends massive invasion machines"; "young wizards harry discovers trio"; "11th birthday learns powerful wizard"; "zombies perfectly evolved survivors"	Genre-driven action/fantasy narratives emphasizing adventure, large-scale conflict, and survival
$G3$	"actively opposed hitler nazis convictions"; "young prince life threatened evil"; "adventures lion witch wardrobe"; "young writer elizabeth wurtzel earns"; "suddenly single cal adrift"	Character-centered dramatic narratives involving moral conviction, family/romance, youth vulnerability, and emotional struggle

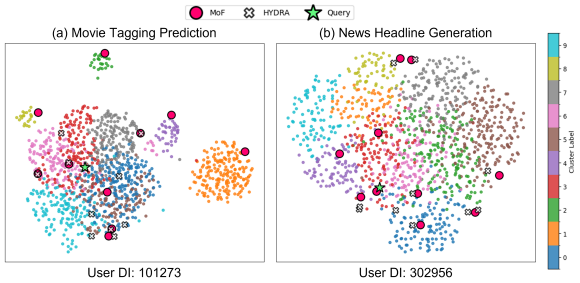

Figure 5: Visualization of CBS for two example users. Colors indicate clusters of user-history embeddings. Red circles and white crosses denote training samples selected using CBS in MoF and BM25 retrieval in HYDRA, respectively, and the green star denotes the query.

able performance when using 6–8 facet heads and performs competitively against the corresponding HYDRA variants across most settings. Similarly, for LaMP-4 in Figure 4 (b), performance tends to improve up to around 6–7 facet heads before showing limited additional gains. These observations suggest that increasing the number of facet heads alone does not necessarily improve personalization performance. Additional results and analyses are presented in Appendix H.

4.8 Visualization

Figure 5 visualizes the history items selected for two example users, illustrating how CBS in MoF retrieves samples from multiple preference clusters compared with the standard retrieval used in HYDRA. The visualization is obtained by projecting k-means-clustered history embeddings (K=10) using UMAP (McInnes et al., 2018). Compared with HYDRA, which tends to retrieve histories concentrated within specific regions, MoF selects items across multiple clusters, exposing the model to more diverse preference signals. We further observe that user histories often span multiple regions in the latent space (Figure 5 (a)) rather than forming a single concentrated pattern. Such heterogeneous preference distributions motivate CBS to provide richer signals for learning facet representations. This observation is consistent with the ablation results, where removing CBS causes the largest performance degradation.

4.9 Qualitative Analysis

To investigate whether the learned router captures meaningful preference-related structures, we group users according to their routing-weight distributions and perform BERTopic (Grootendorst, 2022) analysis on their historical interactions. As illustrated in Figure 10, users exhibit distinct routing patterns rather than being assigned uniformly. We analyze the corresponding user groups in Table 7, where users with similar routing patterns also show coherent narrative tendencies. For example, $G1$ is associated with threat- and conflict-centered narratives, whereas $G2$ focuses on action and fantasy themes, while $G3$ emphasizes character-driven and emotional narratives. These findings provide qualitative evidence that routing weights reflect meaningful preference-related structures rather than arbitrary assignment patterns.

5 Conclusion

In this paper, we introduced MoF, a personalization framework for black-box LLMs based on shared preference facets and preference-aware routing. By replacing user-specific heads with a fixed number of shared facet heads, MoF achieves scalable personalization while maintaining constant personalization-parameter storage overhead as the number of users increases. Experimental results further show that preference-aware routing supports effective personalization and generalization to users unseen during training without additional parameter updates.

Limitations

While MoF achieves a substantially stronger average rank across tasks, it does not outperform HYDRA on every individual metric (e.g., BLEU on LaMP-4). Thus, our results support strong overall personalization performance under a more scalable parameterization, rather than uniform superiority over HYDRA on every metric.

Although MoF generalizes to a substantially larger set of unseen users, our primary experiments train the model on a relatively limited number of users. Therefore, the current results demonstrate inference-time scalability to larger user populations but do not fully characterize training behavior under substantially larger and more heterogeneous user populations.

MoF also employs a fixed number of shared facet heads for all users. While this design provides parameter-efficient personalization, a fixed facet capacity may become restrictive as user preferences become increasingly diverse. Future work could explore adaptive facet allocation or routing mechanisms that dynamically adjust model capacity while preserving parameter efficiency.

Finally, our evaluation is limited to the LaMP benchmark and does not include LongLaMP or a direct comparison with FERMI. LongLaMP involves long-form personalization with task-specific profile structures that may require adapting the current MoF formulation, while FERMI performs iterative per-user prompt optimization under a different personalization paradigm. Extending MoF to long-form settings and comparing it with such optimization-based approaches remain important directions for future work.

Ethical Considerations

MoF performs personalization based on user interaction histories, which may contain sensitive information reflecting individual preferences and behavioral patterns. Although MoF eliminates user-specific scoring heads and reduces the need for storing personalized parameters, the preference inference process may still capture sensitive behavioral patterns from historical interactions. In addition, personalization systems may unintentionally reinforce existing user tendencies or reduce exposure to diverse content. Practical deployments should therefore consider privacy-preserving mechanisms such as anonymization, secure handling of user histories, and user-controlled personalization settings.

We use publicly available artifacts according to their respective licenses and terms of use. The LaMP benchmark code and data creation methods are licensed under CC BY-NC-SA 4.0, and we follow the corresponding licenses and usage conditions of the datasets included in the benchmark. We use `bge-base-en-v1.5` (FlagEmbedding) under the MIT License. We additionally use OpenAI models (e.g., `gpt-3.5-turbo-1106` and `gpt-4o-mini-2024-07-18`) according to OpenAI's API terms of use and service policies.

All artifacts were used in a manner consistent with their intended use and access conditions. Public benchmarks, pretrained models, and APIs were used only for research and evaluation purposes and in accordance with their specified usage policies.

References

Anthropic. 2026. Claude code overview. Claude Code documentation.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 1877–1901.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, Kai Zheng, Defu Lian, and Enhong Chen. 2024. When large language models meet personalization: Perspectives of challenges and opportunities. *World wide web*, 27:42.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tai, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research (JMLR)*, 25.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4171–4186.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research (JMLR)*, 23:1–39.

Shangbin Feng, Taylor Sorensen, Yuhan Liu, Jillian Fisher, Chan Young Park, Yejin Choi, and Yulia Tsvetkov. 2024. Modular pluralism: Pluralistic alignment via multi-LLM collaboration. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4151–4171.

Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3:79–87.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the IEEE international conference on data mining (ICDM)*, pages 197–206.

Jaehyung Kim and Yiming Yang. 2025. Few-shot personalization of LLMs with mis-aligned responses. In *Proceedings of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, pages 11943–11974.

Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A. Hale. 2024a. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. *Nature Machine Intelligence*, 6:383–392.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Michael Bean, Katerina Margatina, Rafael Mosquera, Juan Manuel Ciro, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. 2024b. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2025. Hello again! LLM-powered personalized agent for long-term dialogue. In *Proceedings of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, pages 5259–5276.

Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2025. Llms + persona-plug = personalized llms. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 9373–9385.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Jisoo Mok, Ik-hwan Kim, Sangkwon Park, and Sungroh Yoon. 2025. Exploring the potential of LLMs as personalized assistants: Dataset, evaluation, and analysis. In *Findings of the Association for Computational Linguistics: ACL*, pages 10212–10239.

OpenAI. 2022. Introducing chatgpt. OpenAI Blog.

OpenAI. 2024. Gpt-4o mini: Advancing cost-efficient intelligence. OpenAI Blog.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Jiaxing QI, Zhongzhi Luan, Shaohan Huang, Carol Fung, Hailong Yang, and Depei Qian. 2024. FD-LoRA: Personalized federated learning of large language model via dual lora tuning. *arXiv preprint arXiv:2406.07925*.

Yilun Qiu, Tianhao Shi, Xiaoyan Zhao, Fengbin Zhu, Yang Zhang, and Fuli Feng. 2025. Latent inter-user difference modeling for llm personalization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10599–10617.

Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7370–7392.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval augmented generation with collaborative filtering for personalized text generation. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1294–1304.

Taylor Sorensen, Liwei Jiang, Jena D. Hwang, Sydney Levine, Valentina Pyatkin, Peter West, Nouha Dziri, Ximing Lu, Kavel Rao, Chandra Bhagavatula, Maarten Sap, John Tasioulas, and Yejin Choi. 2024. Value kaleidoscope: Engaging ai with pluralistic human values, rights, and duties. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 38:19937–19947.

Haotian Sun, Yuchen Zhuang, Wei Wei, Chao Zhang, and Bo Dai. 2024. Bbox-adapter: Lightweight adapting for black-box large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 47280–47304.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024a. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6459–6475.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024b. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6476–6491.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 641–649.

Linhai Zhang, Jialong Wu, Deyu Zhou, and Yulan He. 2025a. PROPER: A progressive learning framework for personalized large language models with group-level adaptation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 16399–16411.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, and 2 others. 2025b. Personalization of large language models: A survey. *Transactions on Machine Learning Research (TMLR)*.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew Dai, Zhifeng Chen, Quoc Le, and James Laudon. 2022. Mixture-of-experts with expert choice routing. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 7103–7114.

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

Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. 2024. Hydra: Model factorization framework for black-box llm personalization. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

A Detailed Preliminary Framework

A.1 Reranker

The reranker s_ϕ^{R} takes the current query q_u and retrieved history candidates as input, and outputs a score indicating the usefulness of each history item for generating a user-preferred response. For each user u, we first retrieve a set of M history candidates $H'_u = \{h'_{u,1}, h'_{u,2}, \ldots, h'_{u,M}\}$ from the full user history H_u based on their relevance to the target query q_u. To construct the reranker training dataset D^{R}, we randomly sample M history items

from H_u and treat the query in each sampled history item as a target query for retrieval-based training instance construction. We create a binary supervision label $r^R_{u,m} \in \{0, 1\}$ indicating whether the history item $h'_{u,m}$ helps generate a user-preferred response. For classification tasks, a positive label is assigned when the model output matches the ground-truth label. For generation tasks, a positive label is assigned when the evaluation score exceeds the average score across candidate responses. The reranker is optimized using the following binary classification objective:

$$\mathcal{L}_{\mathrm{R}}(\phi) = \sum_{u \in \mathcal{U}} \sum_{m=1}^{M} \ell_{\mathrm{BCE}}\Big(s^{\mathrm{R}}_{\phi}(q_u, h'_{u,m}), r^R_{u,m}\Big). \tag{14}$$

Based on the reranker scores, the top-k history items are selected to construct the personalized context set:

$$H^*_u = \arg\text{top-}k_{h'_{u,m} \in H'_u} s^{\mathrm{R}}_{\phi}(q_u, h'_{u,m}) \tag{15}$$

A.2 Adapter

Given the query q_u and the personalized context set H^*_u, we sample a set of B candidate responses $O_u = \{o^1_u, \ldots, o^B_u\}$ from the black-box LLM. The adapter s^{A}_{θ} acts as a lightweight selection module that scores candidate responses without modifying the black-box LLM parameters, thereby aligning the output distribution with user preferences. The adapter assigns a preference score to each candidate response in O_u based on its alignment with user preferences. To build the adapter training dataset D^{A}, we generate candidate response sets O_u for all history items in H_u. We assign a binary supervision label $r^A_{u,b} \in \{0, 1\}$ to each candidate response o^b_u, where $r^A_{u,b} = 1$ indicates that the response aligns with user preferences. Following the labeling strategy used for reranker training, positive labels are assigned based on exact matching for classification tasks and relative evaluation scores for generation tasks. The adapter is optimized using the following binary classification objective:

$$\mathcal{L}_{\mathrm{A}}(\theta) = \sum_{u \in \mathcal{U}} \sum_{b=1}^{B} \ell_{\mathrm{BCE}}\Big(s^{\mathrm{A}}_{\theta}(q_u, o^b_u), r^A_{u,b}\Big). \tag{16}$$

The final personalized response is selected as the candidate with the highest adapter score:

$$\hat{y}_u = \arg\max_{b} s^{\mathrm{A}}_{\theta}(q_u, o^b_u). \tag{17}$$

A.3 Training & Inference

As illustrated in Figure 1, both the reranker s^{R}_{ϕ} and the adapter s^{A}_{θ} are trained in two stages: a train stage and a fit stage. During the train stage, the base model and scoring heads are jointly trained using data collected from a large number of users. This enables the model to capture shared preference signals across users while learning user-specific preference patterns through separate heads. During the fit stage, only the heads are updated for unseen users while the shared base model remains frozen. This stage adapts the personalization modules to the preference patterns of new users without modifying the shared parameters.

During inference, retrieved history items are first reranked to select the more useful historical items H^*_u for the current query, as defined in Eq. (15). Conditioned on H^*_u, the black-box LLM generates a set of candidate responses O_u. The adapter then assigns preference scores to the candidate responses in O_u, and the response with the highest score is selected as the final personalized response $\hat{y}_u$, as defined in Eq. (17).

B Parameter Scalability Analysis

We analyze how the number of trainable personalization parameters scales with the number of users. Let N denote the number of users, F the number of facet heads. For user-specific personalization approaches such as HYDRA, each user maintains an independent scoring head. The trainable personalization parameters can therefore be expressed as

$$P_{\mathrm{HYDRA}}(N) = N P_{\mathrm{head}}, \tag{18}$$

where P_{head} denotes the parameter count of a single user-specific head. Consequently, introducing one additional user increases the parameter count by

$$P_{\mathrm{HYDRA}}(N+1) - P_{\mathrm{HYDRA}}(N) = P_{\mathrm{head}}. \tag{19}$$

Thus, the personalization parameter complexity grows linearly with the number of users:

$$P_{\mathrm{HYDRA}}(N) = O(N). \tag{20}$$

In contrast, MoF employs a fixed set of shared facet heads coupled with a shared routing module. The trainable personalization parameters become

$$P_{\mathrm{MoF}} = F P_{\psi} + P_{\mathrm{router}} + P_{\mathrm{SAE}}, \tag{21}$$

where P_ψ denotes the parameter count of a single facet head, and P_{router} and P_{SAE} denote the parameters of the shared routing and sparse autoencoder modules, respectively. Since all terms in P_{MoF} are independent of the number of users, introducing a new user does not increase the parameter count:

$$P_{\text{MoF}}(N+1) - P_{\text{MoF}}(N) = 0. \tag{22}$$

Therefore, MoF exhibits constant parameter complexity with respect to the number of users:

$$P_{\text{MoF}} = O(1). \tag{23}$$

This result indicates that personalization for unseen users can be achieved solely through history-derived routing weights without introducing additional user-specific parameters. Consequently, MoF can scale to large and dynamically evolving user populations without increasing the number of trainable personalization parameters.

C Task Details

We evaluate LLM personalization using the LaMP benchmark (Salemi et al., 2024). The benchmark consists of seven personalized NLP tasks spanning both classification and generation settings. We exclude LaMP-1 and LaMP-7 because these tasks involve task-specific input structures that are not directly compatible with the reranker & adapter formulation adopted in our framework and prior black-box personalization methods. This exclusion reflects a limitation of the current reranker & adapter instantiation rather than an inherent restriction of the MoF framework. Since our goal is to evaluate personalized preference modeling under a unified black-box personalization setting, we focus on tasks that share comparable interaction structures and evaluation protocols. In addition, LaMP-6 is not publicly available and therefore cannot be included in our experiments. This evaluation setting is consistent with the evaluation setting used in prior black-box personalization work (Zhuang et al., 2024).

LaMP-2: Movie Tagging Classification. LaMP-2 evaluates whether a model can predict a user's tagging behavior for movies. Each user's history consists of a sequence of movie description–tag pairs, where the descriptions correspond to previously viewed movies and the tags represent the user's assigned labels. The task requires the model to capture personalized preference patterns from the sequential interaction history and predict the appropriate tag for a target movie description. Following prior work, we evaluate performance using Accuracy (Acc) and F1 score (F1).

Table 8: Size of training data used in the train and fit stages of MoF across different LaMP datasets.

Method	Module	Stage	LaMP-2	LaMP-3	LaMP-4	LaMP-5
MoF	Reranker	Train	10,228	10,852	10,816	10,998
		Fit	4,468	4,926	4,841	4,999
	Adapter	Train	167,979	140,195	289,368	71,226
		Fit	45,401	76,336	128,871	39,294

Table 9: Training and fitting time of MoF across different LaMP datasets.

Method	Module	Stage	LaMP-2	LaMP-3	LaMP-4	LaMP-5
MoF	Reranker	Train	00:04:40	00:09:37	00:03:36	00:09:52
		Fit	00:00:56	00:01:14	00:00:58	00:01:19
	Adapter	Train	00:57:53	01:53:06	01:24:06	00:50:08
		Fit	00:31:42	00:21:19	00:25:03	00:08:21

LaMP-3: Product Rating Classification. LaMP-3 evaluates personalized rating prediction based on users' product reviews. User history is represented as a sequence of review–rating pairs generated by the same user. Given a target review, the model predicts the rating score that the user would assign. Although LaMP-3 is formulated as an ordinal classification task in the original benchmark, following prior work we report MAE and RMSE as evaluation metrics.

LaMP-4: News Headline Generation. LaMP-4 evaluates personalized text generation by requiring the model to generate headlines for news articles. User history consists of article–headline pairs written by the same journalist. The task requires extracting the journalist's writing characteristics and stylistic patterns from historical interactions and generating a personalized headline for a target article. We evaluate generation quality using ROUGE-1 (R-1), ROUGE-L (R-L), and BLEU.

LaMP-5: Scholarly Title Generation. LaMP-5 focuses on generating titles for academic papers based on their abstracts. Similar to LaMP-4, each user's history consists of abstract–title pairs written by the same author. The model is expected to infer the author's writing style and title construction patterns from historical data and generate a personalized title for a target abstract. We evaluate performance using ROUGE-1 (R-1), ROUGE-L (R-L), and BLEU.

Table 10: Performance on the LaMP benchmark using `gpt-4o-mini-2024-07-18` as the underlying proprietary black-box LLM. Experimental settings follow Table 1, while replacing the black-box LLM used in the main experiments. Results indicate that MoF maintains strong overall performance, demonstrating robustness beyond a specific black-box model.

Dataset (→)	**LaMP-2**		**LaMP-3**		**LaMP-4**			**LaMP-5**			**Avg. Rank↓**
Method (↓)	Acc ↑	F1 ↑	MAE ↓	RMSE ↓	R-1 ↑	R-L ↑	BLEU ↑	R-1 ↑	R-L ↑	BLEU ↑	
Zero-Shot	0.467	0.281	0.320	0.639	0.150	0.134	0.849	0.422	0.343	3.113	12.0
ICL (k=1)	0.513	0.344	0.413	0.757	0.150	0.129	0.908	0.434	0.364	4.525	12.0
ICL (k=2)	0.493	0.365	0.393	0.716	0.152	0.135	1.250	0.433	0.364	4.043	11.2
ICL (k=4)	0.520	0.388	0.327	0.648	0.156	0.140	**1.681**	0.440	0.375	5.230	7.9
RAG (k=1)	0.520	0.349	0.340	0.647	0.157	0.141	0.806	0.432	0.359	4.204	10.4
RAG (k=2)	0.527	0.361	0.367	0.651	0.165	0.145	1.425	0.461	0.385	5.934	7.6
RAG (k=4)	0.573	0.405	0.327	0.594	0.163	0.147	1.304	0.466	**0.410**	**6.078**	4.4
PAG (k=0)	0.507	0.370	0.287	0.572	0.127	0.109	0.183	0.402	0.329	2.631	12.0
PAG (k=1)	0.567	0.470	**0.280**	**0.566**	0.142	0.123	0.354	0.399	0.340	2.890	10.2
CFRAG (k=4)	**0.673**	**0.533**	0.347	0.642	0.171	0.148	1.441	0.445	0.383	5.137	5.3
HYDRA-Reranker	0.613	0.417	0.347	0.635	0.160	0.140	1.331	0.464	0.401	5.362	6.4
HYDRA-Adapter	0.527	0.343	0.307	0.610	0.143	0.126	0.728	0.410	0.328	3.127	11.5
HYDRA	0.653	0.448	0.347	0.632	0.168	0.147	1.230	0.468	0.405	5.603	5.0
MoF-Reranker (ours)	0.587	0.452	0.320	0.611	0.169	0.148	1.459	0.462	0.403	5.252	4.2
MoF-Adapter (ours)	0.507	0.336	0.300	0.604	0.152	0.132	1.116	0.419	0.345	2.985	10.5
MoF (ours)	0.653	0.504	0.327	0.627	**0.180**	**0.157**	1.109	**0.472**	0.405	5.643	**3.6**

D Data Statistics

Table 8 reports the number of training instances used in the train and fit stages for the reranker and adapter modules across different LaMP datasets.

E Computational Cost

Table 9 reports the training and fitting runtime of the reranker and adapter modules across different LaMP datasets.

F Implementation Details

All experiments are conducted on a machine equipped with an AMD Ryzen Threadripper 1950X CPU and a single NVIDIA GeForce RTX 3090 GPU. For black-box LLM generation, we set the temperature to 1.0. For training, we employ AdamW (Loshchilov and Hutter, 2019) with a learning rate of 5.0×10^{-5}, a batch size of 64, and a weight decay of 0.01. All models are trained for 2 epochs. We set $\lambda = 0.5$, $\lambda_1 = 8.6 \times 10^{-4}$ and $\tau = 2$. For the reranker, we partition each user's history into 10 clusters and retrieve one item from each cluster using BM25, yielding $M = 10$ sampled candidates. For the adapter, we set the number of candidate responses B to 8. Based on validation performance, we set the number of facet heads to $F = 7$ and use top-3 sparse routing across all experiments. To construct user embeddings, we use up to the most recent 128 history items for each user. Additional sensitivity analyses are provided in Appendix H.

G Generalization Across LLMs

Table 10 presents additional experimental results using `gpt-4o-mini-2024-07-18` (OpenAI, 2024) as the underlying proprietary black-box LLM while keeping all other experimental settings identical to those in the main experiments. We observe that the overall trends remain largely consistent with those in Table 1. Similar to the main results, prompt-based personalization methods do not consistently improve performance across tasks, whereas reranker & adapter-based approaches generally provide stronger and more stable gains. MoF maintains the strongest overall performance with the best average rank of 3.6, demonstrating strong performance across multiple tasks under a different black-box model setting. Specifically, MoF achieves the best performance on several evaluation metrics while remaining competitive on the remaining settings. For example, MoF improves F1 from 0.448 to 0.504 on LaMP-2 compared with HYDRA and also achieves stronger performance on

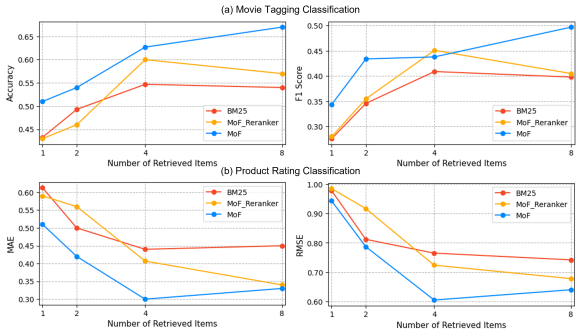

Figure 6: Impact of the number of retrieved user-history items on personalization performance. (a) shows results on LaMP-2 and (b) shows results on LaMP-3. MoF generally benefits from additional retrieved information up to moderate retrieval sizes and maintains strong performance across different retrieval sizes.

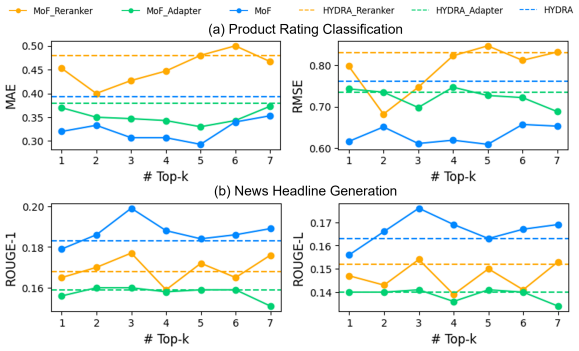

Figure 7: Impact of top-k routing on personalization performance while fixing the total number of facet heads to 7. (a) and (b) present results on LaMP-3 and LaMP-4, respectively. Performance varies with the number of activated facet heads, indicating the importance of routing selection in preference aggregation.

LaMP-4, improving R-1 from 0.168 to 0.180 and R-L from 0.147 to 0.157. Although absolute performance differs from the main experimental setting, the relative advantage of MoF remains largely preserved. These results suggest that the proposed framework can generalize across different proprietary black-box LLMs.

H Sensitivity Analysis

H.1 Impact of Number of Retrieved Items

Figure 6 presents the impact of varying the number of retrieved items on personalization performance. We observe that increasing the number of retrieved items generally improves performance in both LaMP-2 and LaMP-3 up to a moderate retrieval size, while larger retrieval sizes may provide diminishing returns. For LaMP-2 in Figure 6 (a), MoF shows consistent performance improvements as the number of retrieved items increases, whereas BM25 and MoF-Reranker exhibit weaker gains or performance saturation. This trend suggests that the proposed framework benefits more from additional retrieved information. A similar trend can be observed for LaMP-3 in Figure 6 (b). Although performance slightly degrades at larger retrieval sizes, MoF maintains lower MAE and RMSE values than the baselines across different retrieval settings. These observations suggest that the proposed framework remains relatively robust under larger retrieval sizes while benefiting from richer user-history information.

H.2 Impact of Top-k Routing

Figure 7 presents the impact of varying the number of selected facet heads under top-k routing while fixing the total number of facet heads to 7. We observe that performance generally improves up to a moderate routing size, while larger routing sizes provide limited additional benefits. For LaMP-3 in Figure 7 (a), MoF achieves its best performance around k=2, obtaining lower MAE and RMSE values than the corresponding HYDRA variants across all settings. Similarly, for LaMP-4 in Figure 7 (b), performance reaches its highest values around k=3 for both R-1 and R-L metrics. In addition, MoF achieves stronger performance than HYDRA across most routing settings. These observations suggest that selecting a moderate routing size provides a better balance between utilizing diverse preference signals and maintaining focused preference representations, which is consistent with prior MoE studies emphasizing sparse expert activation and effective expert specialization (Zhou et al., 2022). While activating additional facet heads may incorporate richer preference information, excessively large routing sizes may reduce the effectiveness of preference aggregation by introducing redundant or less relevant preference signals.

H.3 Impact of User Embedding History Length

Figure 8 presents the impact of varying the number of recent user interactions used for constructing user embeddings. We observe that increasing the history length does not consistently improve personalization performance. For LaMP-3 in Figure 8

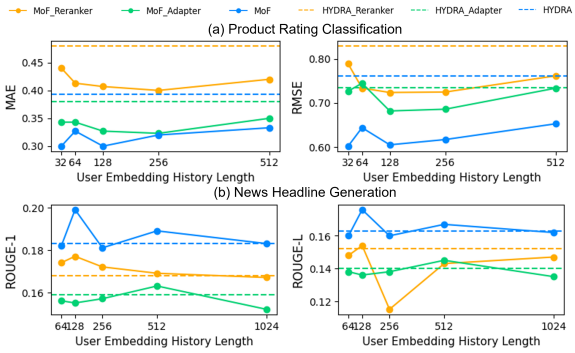

Figure 8: Impact of the number of recent user interactions used for user embedding construction on personalization performance. (a) and (b) present results on LaMP-3 and LaMP-4, respectively. Moderate history lengths generally provide stronger performance across tasks.

(a), MoF achieves its strongest performance at relatively short-to-moderate history lengths, with the best results observed around 128 recent interactions. Similarly, for LaMP-4 in Figure 8 (b), performance also peaks around moderate history lengths before showing limited additional gains. Across both tasks, MoF generally maintains stronger performance than the corresponding HYDRA variants. These observations suggest that incorporating additional historical interactions does not necessarily provide proportional benefits. Prior work on sequential recommendation has shown that user representations benefit from focusing on behavior signals that are more relevant to current preferences rather than uniformly aggregating long interaction histories (Kang and McAuley, 2018). Our results exhibit a similar trend, suggesting that moderate history lengths may provide a favorable balance between preserving relevant preference information and limiting the influence of older interactions.

I Case Study

Figure 9 presents qualitative examples from LaMP-2 to illustrate how MoF performs preference-aware personalization under the same input query. In this task, the model is asked to predict the movie tag for the same target movie from a candidate tag set, while each user provides a different interaction history. Although the target query is identical, MoF retrieves different user-history samples and assigns different preference alignment scores depending on the user's past tagging behavior.

For User `101291`, MoF selects historical movies that are mostly associated with the tag "based on a book." The selected examples are distributed across multiple regions in the history embedding space, indicating that MoF does not simply rely on local semantic similarity, but instead gathers preference-relevant evidence from diverse historical clusters. As a result, the model assigns the highest scores to the "based on a book" candidates, suggesting that the user's previous tagging patterns provide strong evidence for this preference.

In contrast, for User `10497`, MoF retrieves a different set of historical examples under the same target query. The selected histories include multiple examples associated with "comedy," and the resulting preference scores are highest for the "comedy" candidates. This shows that MoF can adapt its prediction behavior to a different user's preference pattern even when the input movie is unchanged. The comparison demonstrates that MoF personalizes the decision process by conditioning on user-specific historical preferences rather than producing a single globally preferred prediction.

Overall, this case study highlights two important properties of MoF. First, the cluster-based history selection encourages the model to use diverse evidence from the user's interaction history, which helps capture heterogeneous preference signals. Second, the preference-aware routing mechanism enables different users to emphasize different latent facets, leading to user-specific score distributions and final predictions. These qualitative results support the main experimental findings that MoF can effectively model personalized preferences while avoiding user-specific scoring heads.

J Prompt Templates

Following prior work (Zhuang et al., 2024; Salemi et al., 2024), we employ prompt templates throughout the framework, including black-box LLM generation and the construction of training instances for both the reranker and adapter modules. The prompts combine task-specific instructions with personalized contextual information derived from user histories. The complete prompt templates used in our experiments are illustrated in Figures 11, 12, 13, and 14.

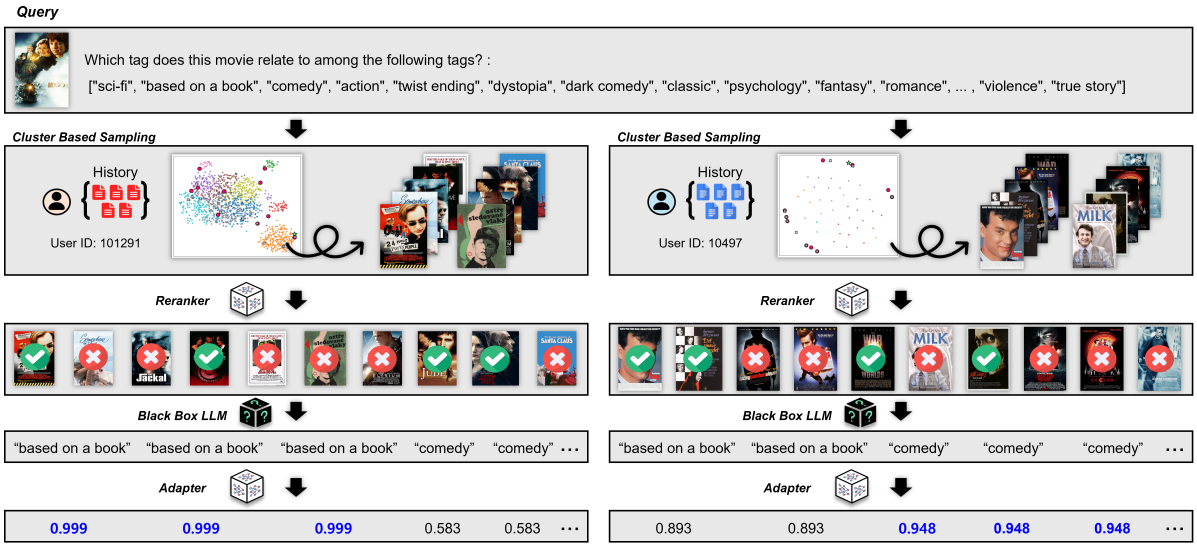

Figure 9: Case study on LaMP-2 illustrating how MoF personalizes prediction behavior for the same query. MoF selects different user-history examples and assigns distinct preference alignment scores according to each user's tagging patterns, resulting in user-specific evidence selection and predictions.

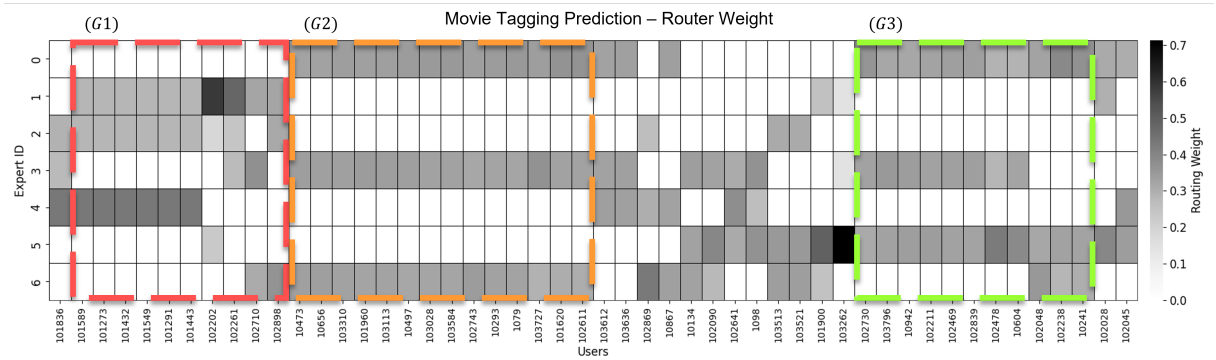

Figure 10: Heatmap of routing weight distributions across users. Users with similar routing patterns are grouped together.

Figure 11: Prompt template used for Movie Tagging Classification (LaMP-2). The prompt incorporates retrieved user history examples together with task instructions and the target movie description.

Movie Tagging Classification (LaMP-2)

The tag for the movie: "in the town of blithe hollow, norman babcock can speak to the dead, but no one other than his eccentric new friend believes his ability is real. one day, norman's eccentric uncle tells him of a ritual he must perform to protect the town from a curse cast by a witch centuries ago." **is** "fantasy" ,
and The tag for the movie: "in the futuristic action thriller looper, time travel will be invented but it will be illegal and only available on the black market. when the mob wants to get rid of someone, they will send their target 30 years into the past where a looper, a hired gun, like joe is waiting to mop up. joe is getting rich and life is good until the day the mob decides to close the loop, sending back joe's future self for assassination." **is** "sci-fi" ,
and The tag for the movie: "wallace rents out gromit's former bedroom to a penguin, who takes up an interest in the techno pants created by wallace. however, gromit later learns that the penguin is a wanted criminal." **is** "comedy" ,
and The tag for the movie: "a scientist in a surrealist society kidnaps children to steal their dreams, hoping that they slow his aging process." **is** "dystopia" .
Which tag does this movie relate to among the following tags? Just answer with the tag name without further explanation.
tags:
[sci-fi, based on a book, comedy, action, twist ending, dystopia, dark comedy, classic, psychology, fantasy, romance, thought-provoking, social commentary, violence, true story]
description:
Kuzco is a self-centered emperor who summons Pacha from a village and to tell him that his home will be destroyed to make room for Kuzco's new summer home. Kuzco's advisor, Yzma, tries to poison Kuzco and accidentally turns him into a llama, who accidentally ends up in Pacha's village. Pacha offers to help Kuzco if he doesn't destroy his house, and so they form an unlikely partnership.

Figure 12: Prompt template used for Product Rating Classification (LaMP-3). The prompt incorporates retrieved user history examples together with task instructions and the target review.

Product Rating Classification (LaMP-3)

5 **is the score for** "this is a thrillers! loved the characters and the way god used each person to complete the story line in gods special plan" ,
and 5 **is the score for** "they worked together in rugged conditions with joy. loved the tensions and surprises. i want to read all of her books" ,
and 5 **is the score for** "what a great lesson in love and forgiveness! this is a happy story despite tragedy. all dogs can be loving." ,
and 5 is the score for "waiting for gods plan in our lives is exciting. he makes all things work together for our good, even the things we would not ask for" .
What is the score of the following review on a scale of 1 to 5? just answer with 1, 2, 3, 4, or 5 without further explanation.
review:
This story is alive visually and with great emotion. God worked all things together to bring them to acknowledge the forgiveness and restoring power of love.

Figure 13: Prompt template used for News Headline Generation (LaMP-4). The prompt incorporates retrieved user history examples together with task instructions and the target news article.

News Headline Generation (LaMP-4)

"'Black Panther' Expected To Have Enormous $165 Million-Plus Opening Weekend" **is the title for** "the film, which comes out feb. 16, has already been breaking records." ,
and "Rosé Chocolate Exists, And We've Officially Gone Too Far" **is the title for** "we wonder if it tastes different if you eat it in the hamptons..." ,
and "The Weather Channel Is Not Holding Back When It Comes To Climate Change" **is the title for** "and if you don't care about it, they'll convince you otherwise." ,
and "24 Times J.K. Rowling Wrote Or Said Something That Hit All The Feels" **is the title for** "warning : this will make you want to curl up with all her books and disappear." .
Generate a headline for the following article:
But if you want to make one, Twitter suggests breaking out the Manischewitz.

Figure 14: Prompt template used for Scholarly Title Generation (LaMP-5). The prompt incorporates retrieved user history examples together with task instructions and the target paper abstract.

Scholarly Title Generation (LaMP-5)

"Lectures on Jacques Herbrand as a Logician" **is a title for** "we give some lectures on the work on formal logic of jacques herbrand, and sketch his life and his influence on automated theorem proving. the intended audience ranges from students interested in logic over historians to logicians. besides the well - known correction of herbrand's false lemma by goedel and dreben, we also present the hardly known unpublished correction of heijenoort and its consequences on herbrand's modus ponens elimination. besides herbrand's fundamental theorem and its relation to the loewenheim - skolem - theorem, we carefully investigate herbrand's notion of intuitionism in connection with his notion of falsehood in an infinite domain. we sketch herbrand's two proofs of the consistency of arithmetic and his notion of a recursive function, and last but not" ,
and "THF0 --- The Core of the TPTP Language for Higher-Order Logic" **is a title for** "one of the keys to the success of the thousands of problems for theorem provers (tptp) problem library and related infrastructure is the consistent use of the tptp language. this paper introduces the core of the tptp language for higher - order logic - - - thf0, based on church's simple type theory. thf0 is a syntactically conservative extension of the untyped first - order tptp language." ,
and "DiaWOz-II: a tool for wizard-of-Oz experiments in mathematics" **is a title for** "we present diawoz - ii, a configurable software environment for wizard - of - oz studies in mathematics and engineering. its interface is based on a structural wysiwyg editor which allows the input of complex mathematical formulae. this allows the collection of dialog corpora consisting of natural language interleaved with non - trivial mathematical expressions, which is not offered by other wizard - of - oz tools in the field. we illustrate the application of diawoz - ii in an empirical study on tutorial dialogs about mathematical proofs, summarize our experience with diawoz - ii and briefly present some preliminary observations on the collected dialogs." ,
and "Agent based mathematical reasoning" **is a title for** "in this contribution we propose an agent architecture for theorem proving which we intend to investigate in depth in the future. the work reported in this paper is in an early state, and by no means nished. we present and discuss our proposal in order to get feedback from the calculemus community.".
Following the given patterns Generate a title for the following abstract of a paper:
Abstract We present a straightforward embedding, of quantified multim odal logic in simple type theory