FAST AND ACCURATE: IMPROVING PERSONALIZED LLM PERFORMANCE BY SEMANTIC CONCENTRATION INSPIRED WORKLOAD SPLIT BETWEEN RAG AND FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

With ever increasing personalized large language model (LLM) services, the next question to answer is how to improve the quality of services/users' experiences, which is indicated by personalized LLM generation accuracy/quality and inference latency. Although state-of-the-art designs, e.g., parameter-efficient fine-tuning (PEFT), retrieval-augmented generation (RAG) or Full PEFT+RAG, can help, they largely ignore the heterogeneous semantic concentrations of users' historical data and cannot fully unleash the potential of personal data for personalized LLM performance enhancement. In this paper, we propose PEFT Union RAG (PUR), a novel "fast and accurate" approach that exploits the semantic concentration of users' personal historical data to split workload between PEFT and RAG, aiming to improve personalized LLM performance in terms of better generation accuracy/quality and lower inference latency. Briefly, PUR partitions user's personal historical data and assigns semantic concentrated "core" information to PEFT for offline parametric fine-tuning, while retaining dispersed information for RAG. During the online LLM inference, PUR directly generates responses for "core" queries covered by the PEFT region and invokes RAG only for sporadic "peripheral" queries. Experimental results show that, by appropriately splitting workload between PEFT and RAG, PUR consistently achieves the best generation accuracy/quality for various users across multiple tasks, while reducing inference latency by 14.4%–46.6% compared with the best performed method.

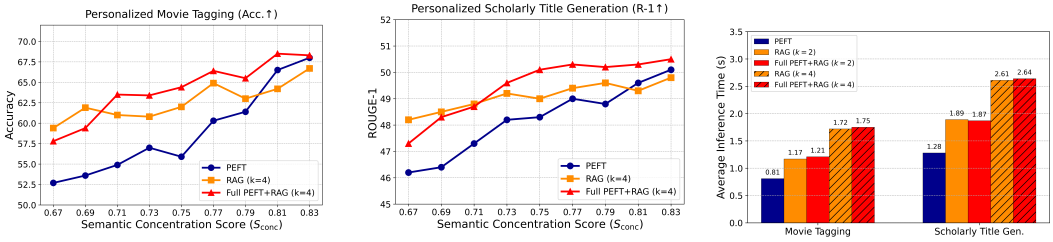
1 INTRODUCTION

While large language models (LLMs) are increasingly employed for user-facing applications such as email drafting (Miura et al., 2025), recommendation (Zhu et al., 2024), and content generation (Tan et al., 2024a), the general responses from base LLMs may not be good enough to meet users' personalized expectations. Nowadays, the concept of personalized LLM is becoming more and more popular, which leverages user's personal historical data to enhance LLMs, so that the generated responses can better align with users' specific needs and preferences (Tan et al., 2024b; Liu et al., 2025). Under such problem settings, the essential question to ask is how to effectively exploit the user's historical data to improve personalized LLM's performance.

To enhance personalized LLM's accuracy, existing approaches can be broadly divided into two categories (Salemi & Zamani, 2025; Tan et al., 2024c): retrieval-augmented generation (RAG) based methods and parameter-efficient fine-tuning (PEFT) based methods. The former treats user historical data as a reference corpus and retrieves content relevant to the current user prompt during inference, using the retrieved content as additional context to guide generation. The latter fine-tunes a small set of trainable parameters based on user historical data, allowing the model to encode user-specific preferences or behavioral patterns into lightweight personalized parameters that can support personalized generation during inference.

To answer the question which one (i.e., RAG or PEFT) can better enhance personalized LLM's performance, prior studies (Salemi & Zamani, 2025; Lakatos et al., 2025; Ovadia et al., 2024) have

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Generation accuracy/quality.

(b) Inference latency.

Figure 1: Preliminary results. (a) The generation accuracy/quality of PEFT, RAG, and Full PEFT+RAG across users with different semantic concentration scores. (b) Average inference latency under different methods, where k denotes the number of retrieved texts used at inference.

extensively compared these two paradigms across multiple personalized LLM tasks involving large numbers of users with personal historical data. Those works show that RAG often serves as a better option and outperforms PEFT in many settings. Motivated by the complementary strengths of retrieval/RAG and parametric adaptation/PEFT, some other studies (Tan et al., 2024c; Ficek et al., 2024; Pingua et al., 2025) further explore to use them both fully (i.e., full PEFT+RAG = 100% PEFT + 100% RAG). Those methods first fully fine-tune the model with user's historical data and then retrieve all relevant user's content as additional contexts during the inference. Experimental results show that full PEFT+RAG can further improve personalized LLM's performance.

However, the state-of-the-art research findings above are based on the averaged performance results over all users, which largely ignores the heterogeneity of users' historical data. Actually, for personalized LLM tasks, users' historical data may substantially differ in semantic concentration. Some users may have more concentrated historical data, reflecting stronger consistency and stability, whereas some other users may have more dispersed historical data that exhibit greater sparsity and stronger long-tail characteristics. As shown in Fig. 1 (a), our empirical studies indicate that prior findings are not applicable, if users are grouped by the semantic concentration levels of their historical data. Specifically, for users with very low semantic concentration, full PEFT+RAG is inferior to RAG alone, while for users with high semantic concentration, PEFT no longer lags behind RAG. Those results motivate us to thoroughly consider the semantic concentration of users' historical data and seek a solution to better integrate PEFT and RAG, to further improve personalized LLM performance.

Besides, as the personalized LLM is being deployed for daily services, the LLM performance should consider not only LLM generation accuracy/quality but also online LLM inference latency for good quality of services. Due to the extra historical corpus retrieval step and the additional computation introduced by long contexts, either RAG or full PEFT+RAG incurs much higher inference latency than directly generating responses from the prompt (Jiang et al., 2025; Shen et al., 2025; Mahapatra et al., 2025). As shown in Fig. 1 (b), for methods involving RAG, inference latency significantly increases as more reference texts are retrieved and incorporated into the input context. It is obvious that the benefits brought by RAG come at the cost of severely delayed LLM response.

To address the observed challenges above, in this paper, we propose PEFT Union RAG (PUR), a novel "fast and accurate" approach that exploits the semantic concentration of users' personal historical data to split workload between PEFT and RAG, aiming to improve personalized LLM performance in terms of better generation accuracy/quality and lower inference latency. Unlike full PEFT+RAG, which directly combines 100% PEFT and 100% RAG, PUR first quantifies the semantic concentration of the user's historical data and then partitions the user's historical data accordingly. Then, PUR allocates semantically concentrated core information to offline PEFT for parametric adaptation, and saves dispersed information as an external corpus for RAG. At the online LLM inference stage, the personalized LLM checks if the knowledge required by the current user query has already been captured by the fine-tuned parameters. If so, the personalized LLM will directly generate the response. Otherwise, the personalized LLM will follow the traditional RAG protocol to generate the response. A sketch of the proposed PUR design is presented in Fig. 2.

Such semantic concentration based workload split in PUR makes the best use of PEFT's and RAG's complementary advantages. Briefly, by restricting PEFT training to highly concentrated core historical data, PUR reduces the noise introduced by dispersed information and thereby improves response

accuracy/quality for frequent "core" queries. For those sporadic "peripheral" queries, PUR still employs RAG to provide additional support to keep personalized LLM response quality. Meanwhile, by allowing "core" queries to bypass unnecessary RAG, PUR helps to effectively reduce the averaged online inference latency. Note that the time consumed by PEFT is offline fine-tuning cost, and should not be counted as part of online LLM inference delay that affects quality of experience for users. Extensive experimental results show that PUR yields the best generation accuracy/quality for users with different semantic concentration levels across various tasks, while reducing inference latency by 14.4%–46.6% compared with the best performed compared method.

2 PRELIMINARIES

The goal of personalized LLM generation is to leverage user historical data to generate outputs that better align with the user's preferences or behavioral patterns. Formally, let M_θ denote an LLM with parameters θ. For a user u, we denote the historical data by $H_u = \{h_1, h_2, \ldots, h_{N_u}\}$, where N_u is the number of historical data instances. Given the current user input x_u, personalized generation can be written as $y_u = M_\theta(x_u \mid H_u)$, where the output y_u depends on both the current input and the user's history. Under this formulation, the key challenge is how to effectively incorporate H_u into personalized generation.

PEFT enhanced personalized LLM. Parameter-Efficient Fine-Tuning (PEFT) is a class of methods for adapting LLMs to downstream tasks with low training cost, with representative approaches including Adapter (Houlsby et al., 2019), LoRA (Hu et al., 2022), and BitFit (Zaken et al., 2022). The core idea of PEFT is to update only a small set of additional trainable parameters or lightweight modules, rather than fine-tuning all parameters.

In PEFT enhanced personalized LLM, the user's historical data H_u is used to train user-specific PEFT parameters ϕ_u during the offline stage. Let M_θ denote the base model with frozen pretrained parameters θ. The user-specific adaptation objective can then be written as $\phi_u^* = \arg\min_{\phi_u} \mathcal{L}\big(M_{\theta,\phi_u}; H_u\big)$, where $\mathcal{L}$ denotes the task-specific loss on H_u, and M_{θ,ϕ_u} is the personalized model obtained by incorporating ϕ_u into the base model. During the online LLM inference, the personalized model directly maps the current input x_u to the personalized output:

$$y_u = M_{\theta,\phi_u^*}(x_u).$$

RAG enhanced personalized LLM. Retrieval-Augmented Generation (RAG) (Lewis et al., 2020; Zamani et al., 2022) is a paradigm that augments LLM generation with external retrieval. Unlike PEFT, which incorporates knowledge into model parameters, RAG stores such knowledge in an external database and retrieves relevant content during inference based on the current input. The retrieved content is then provided to the LLM as additional context to improve the relevance and factual accuracy of the generated output, making RAG particularly effective for knowledge-intensive tasks. In modern RAG systems, retrieval is typically implemented by encoding text into a vector space with a semantic embedding model and returning the most relevant items based on vector similarity (Yu et al., 2024; Shao et al., 2024).

In RAG enhanced personalized LLM, the user historical data H_u is treated as a personalized knowledge source. In the offline stage, each historical record h_i is encoded into a vector representation to build a user-specific retrieval database. During the inference, given the current input x_u, the retriever returns the top-k most relevant historical records from H_u, denoted by $R_k(x_u, H_u)$. The base model then takes x_u together with the retrieved results as context and produces the personalized output:

$$y_u = M_\theta(x_u, R_k(x_u, H_u)).$$

100% PEFT + 100% RAG enhanced personalized LLM. In addition to the two paradigms above, prior works (Tan et al., 2024c; Ficek et al., 2024; Pingua et al., 2025) have also proposed a method that directly combines PEFT and RAG to further improve personalized generation. Specifically, in the offline stage, the full user historical data H_u is used both to train user-specific PEFT parameters ϕ_u and to build a user-specific retrieval database. During the online LLM inference, the system first performs retrieval and then uses the personalized model M_{θ,ϕ_u^*} to generate the personalized output:

$$y_u = M_{\theta,\phi_u^*}(x_u, R_k(x_u, H_u)).$$

3 OBSERVATIONS AND MOTIVATION

To examine how the above methods behave for users with different historical data distributions, we conducted preliminary experiments on the personalized movie tagging (text classification) and the personalized scholarly title generation (text generation) tasks. We grouped users according to the semantic concentration (Sec. 4.1) of their historical data in the embedding space, and measured both generation quality and inference latency under different methods. The results are shown in Fig. 1.

No Strategy Always Wins. As shown in Fig. 1 (a), we observe the following pattern in both tasks. The method that achieves the best task performance varies across users with different levels of semantic concentration, indicating that no single strategy serves as the universal best choice for all users. Moreover, PEFT appears to be particularly sensitive to semantic concentration. For users with lower semantic concentration, PEFT performs noticeably worse than RAG, and even the Full PEFT+RAG strategy fails to outperform RAG alone. In contrast, as semantic concentration increases, PEFT becomes increasingly competitive and no longer lags behind RAG.

Retrieval Slows Down Inference. Retrieval-augmented strategies incur additional inference latency due to the extra retrieval step and longer input contexts. In personalized tasks, because the per-user historical corpus is relatively small, the retrieval step itself is typically not the dominant cost; instead, the main overhead comes from the longer context introduced by retrieved texts. As shown in Fig. 1 (b), when Top-$k = 4$, RAG-related methods increase inference latency by more than 100% compared to the PEFT-only method.

Those results motivate us to thoroughly consider the semantic concentration of users' historical data and seek a solution to better integrate PEFT and RAG, to achieve both better generation accuracy/quality and low inference latency.

4 PEFT UNION RAG DESIGN

Based on the observations above, we propose PEFT Union RAG (PUR), a novel "fast and accurate" approach that exploits the semantic concentration of users' personal historical data to split workload between PEFT and RAG, aiming to improve personalized LLM performance in terms of better generation accuracy/quality and lower inference latency. Figure 2 illustrates the overall workflow of PUR. At the offline stage, unlike directly applying full PEFT+RAG to the entire user's historical data, PUR first quantifies the semantic concentration of the user's historical data and then partitions it accordingly. The more semantically concentrated "core" subset is allocated to PEFT for parametric learning, while the remaining dispersed subset is retained in the reference corpus for RAG. During the online LLM inference, the personalized LLM checks if the knowledge required by the current user query has already been captured by the fine-tuned parameters. If so, the personalized LLM will directly generate the response. Otherwise, the personalized LLM will follow the traditional RAG protocol to generate the response. The following subsections will illustrate the proposed PUR design in detail.

4.1 SEMANTIC CONCENTRATION MODELING

Historical Data Embedding. In PUR, we analyze the semantic concentration of user's historical data in the embedding space. Since local neighborhood structure in the embedding space can reflect semantic similarity between historical records, we first use an embedding model to encode the historical records into vector representations. This mirrors the embedding step commonly used in RAG systems to build a vector database. For user u with historical data $H_u = \{h_1, \ldots, h_{N_u}\}$, we encode each historical record h_i into a vector v_i, and obtain the embedding set $V_u = \{v_1, \ldots, v_{N_u}\}$.

Cluster Structure Analysis. After obtaining the embedding set V_u, we use HDBSCAN (Campello et al., 2015) to identify its cluster structure. This method can detect density-based clusters without requiring the number of clusters to be specified in advance. This step is necessary because our subsequent definition of semantic concentration is based on a single semantic density center. If multiple semantic centers exist, a single global concentration score would mix within-cluster dispersion with between-cluster separation, leading to a distorted estimate under the single-center assumption. Therefore, when multiple clusters are detected, we conduct the subsequent concentration analysis separately within each cluster $\mathcal{C}_i$.

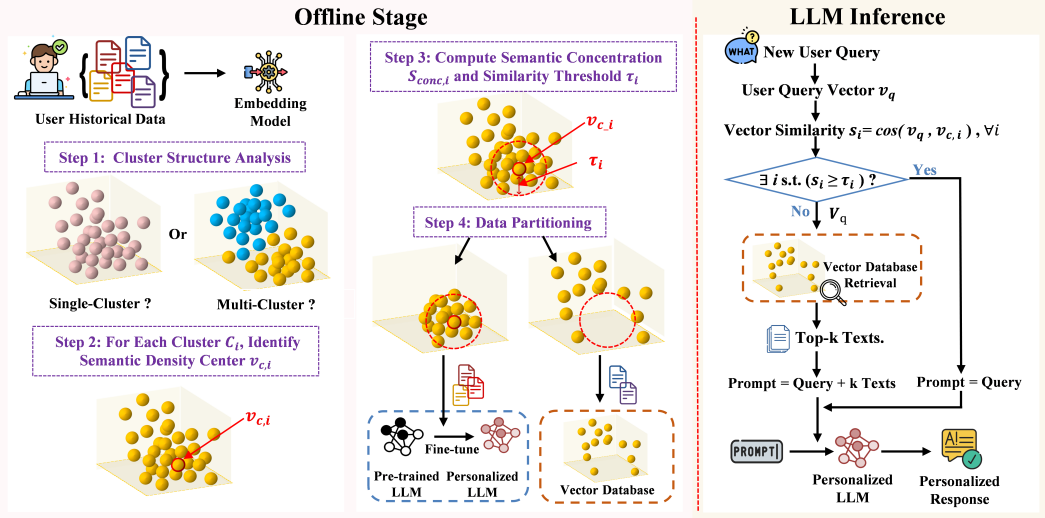


Figure 2: An overview of our proposed PEFT Union RAG (PUR).

Semantic Density Center Identification. Semantic concentration aims to measure how closely samples are distributed around the core semantic region, and therefore depends on an appropriate semantic center. A natural choice is the centroid, but this choice is sensitive to tail points: samples far from the dominant region may substantially bias the centroid and pull it away from the core semantic region. Therefore, to obtain a more robust center estimate, PUR identifies the semantic density center for each cluster based on local density. Specifically, for each sample $v_j \in \mathcal{C}_i$, we compute its k-NN radius (Cover & Hart, 1967), i.e., the distance to its k-th nearest neighbor, denoted by $r_j^{(k)}$. This metric serves as a proxy for local density, where smaller values indicate denser local neighborhoods. We then define the sample with the smallest k-NN radius as the semantic density center of cluster $\mathcal{C}_i$, denoted by $v_{c,i}$. This center is more likely to lie in the densest and most stable semantic region of the cluster.

Semantic Concentration. After estimating the semantic density center $v_{c,i}$, we define a cluster-level semantic concentration score $S_{\text{conc},i}$. For each embedding vector $v_j \in \mathcal{C}_i$, we measure its proximity to the density center using cosine similarity, i.e., $s_j = \cos(v_j, v_{c,i})$. A larger s_j indicates that the sample is semantically closer to the center. Based on these similarities, we first define the average concentration term as $S_{\text{avg},i} = \frac{1}{|\mathcal{C}_i|} \sum_{v_j \in \mathcal{C}_i} s_j$, which measures the average proximity of cluster samples to the density center. A higher value indicates that the cluster is more concentrated around its semantic core.

However, average similarity alone may underestimate clusters that contain a compact dominant core together with a long tail. In such cases, low-similarity tail samples can substantially drag down the average, causing the $S_{\text{avg},i}$ to underestimate the PEFT suitability of the dominant core. To address this issue, we introduce a tail-aware compensation term that fixes the underestimation issue caused by pronounced long-tail structure, rather than rewarding dispersed points.

Specifically, we use the robust z-score (Iglewicz & Hoaglin, 1993) $z_j = (\text{median}(s) - s_j)/(\text{MAD}(s) + \epsilon)$, where $\text{MAD}(s) = \text{median}(|s_j - \text{median}(s)|)$ to identify samples that lie clearly outside the dominant part of the similarity distribution. If a sample has much lower similarity than most others, its robust z-score will be larger. We then get the tail ratio as $\pi_{\text{tail},i} = \Pr(z_j > 3)$, where $\Pr$ denotes the proportion of samples that meet the condition. This term measures how many samples are clearly in the low-similarity tail, and the threshold 3 is a commonly used standard in robust outlier detection (Iglewicz & Hoaglin, 1993). We further define the tail strength as $h_{\text{tail},i} = Q_{50}(s) - Q_5(s)$, where $Q_p(s)$ denotes the p-th quantile of the similarity score s_j. This term measures how far the lower tail is separated from the dominant region. In this way, $\pi_{\text{tail},i}$ captures how many tail samples there are, while $h_{\text{tail},i}$ captures how strong the tail is. Their product therefore naturally reflects the overall influence of the long-tail structure. The final semantic concentration score is then defined as

$$S_{\text{conc},i} = S_{\text{avg},i} + \lambda \, \pi_{\text{tail},i} \, h_{\text{tail},i},$$

where λ controls the strength of the compensation. In this way, the score better preserves the contribution of a compact dominant core in long-tailed clusters, thereby yielding a subsequent data partitioning between PEFT and RAG that is more consistent with the underlying semantic structure.

4.2 Semantic Concentration Inspired Data Partitioning and Workload Split in PUR

After obtaining the concentration score $S_{\mathrm{conc},i}$ for each semantic cluster, PUR determines the PEFT allocation ratio for that cluster based on this score. Intuitively, a higher $S_{\mathrm{conc},i}$ indicates that a larger proportion of samples in the cluster is suitable for PEFT-based parametric learning. However, there is no closed-form mapping from $S_{\mathrm{conc},i}$ to the optimal allocation ratio. Instead, we adopt an empirically calibrated allocation strategy. Specifically, PUR evaluates candidate allocation ratios in several semantic concentration intervals and summarizes appropriate ratios in a lookup table, as shown in Table 2. PUR does not attempt to provide an exact optimal allocation for every $S_{\mathrm{conc},i}$, but provides a robust and practical allocation policy to guide the data allocation. As shown in the subsequent experiments, the calibrated ratios generalize well across multiple tasks and diverse users.

After obtaining the allocation ratio ρ_i for the cluster $\mathcal{C}_i$, PUR employs PEFT to fine-tune the top fraction ρ_i of samples ranked by similarity to the density center. The minimum similarity value among the selected samples is defined as the cluster-level similarity threshold τ_i, which is used later for LLM inference path selection. The remaining data samples are retained in the retrieval corpus to support RAG. After processing all clusters, the user's historical data is partitioned into two disjoint subsets, H_u^{peft} and H_u^{rag}, where $H_u = H_u^{\mathrm{peft}} \cup H_u^{\mathrm{rag}}$. Here, H_u^{peft} is used to fine-tune personalized LLM parameters ϕ_u, while H_u^{rag} is used to build the personalized retrieval vector database.

4.3 Personalized LLM Inference in PUR

At the online LLM inference stage, PUR will choose the "fast and accurate" inference path given a user's new query. For a query x_u, the PUR system first encodes it into a query vector v_q and computes its cosine similarity to each semantic density center $v_{c,i}$: $s_i = \cos(v_q, v_{c,i}), \quad \forall i$. If there exists a cluster such that $s_i \geq \tau_i$, the input is considered to fall within a PEFT-covered core semantic region, and the personalized model M_{θ,ϕ_u^*} directly generates the response without retrieval. Otherwise, the PUR system retrieves relevant historical information from the external corpus H_u^{rag} and performs retrieval-augmented generation. Formally,

$$y_u = \begin{cases} M_{\theta,\phi_u^*}(x_u), & \exists i \text{ such that } s_i \geq \tau_i, \\ M_{\theta,\phi_u^*}(x_u, R_k(x_u, H_u^{\mathrm{rag}})), & \text{otherwise.} \end{cases}$$

5 Experimental Setup

Datasets. We use data from the Large Language Model Personalization (LaMP) benchmark (Salemi et al., 2024), which includes seven personalized LLM tasks. Each user in the benchmark comes with its own historical data, collected from real-world user data.[1] We report accuracy for classification tasks (citation identification and movie tagging), RMSE for product rating task, and ROUGE-1 for text generation tasks (news headline, scholarly title, tweet paraphrasing). Note that all metrics are the higher the better, except for RMSE.

Methods for comparison. We compare PUR with three methods for personalized generation: (1) Retrieval Augmented Generation **(RAG)**, which uses `bge-base-en-v1.5` (Xiao et al., 2024) as the embedding model and retrieves the Top $k = 4$ texts by default; (2) Parameter-Efficient Fine-Tuning **(PEFT)**, implemented using LoRA (Hu et al., 2022) with default rank $r = 16$, applied to the q_proj and v_proj modules, with learning rate 1×10^{-5}, batch size 4, and 3 training epochs; (3) and the 100% PEFT + 100% RAG **(Full PEFT+RAG)**. For PUR and all methods, we adopt `Llama-2-7B` (Touvron et al., 2023) as the base LLM. Note that PUR focuses on workload allocation between parametric adaptation and retrieval augmentation, for which we adopt standard PEFT and

[1]We exclude LaMP-6 as it involves private data that we cannot access. Dataset statistics are presented in Appendix B.

Table 1: Performance under different semantic concentration range and PEFT allocation ratios. For each user group, we compare other methods with PUR instantiated using PEFT allocation ratios from 10% to 90%, to examine how the optimal allocation varies with user semantic concentration. The best score is in **bold** and the second best is underlined.

S_{conc} Range	Compared Methods			PUR (PEFT Allocated Ratio)								
	PEFT	RAG	Full PEFT+RAG	10%	20%	30%	40%	50%	60%	70%	80%	90%
Personalized Movie Tagging (Acc. ↑)												
0.66–0.68	52.7	59.4	57.8	59.4	59.7	**60.9**	59.8	58.1	56.4	56.0	54.9	53.3
0.68–0.70	53.6	61.9	59.4	62.0	63.9	**64.6**	62.5	60.7	57.9	57.1	55.2	53.1
0.70–0.72	54.9	61.0	63.5	60.8	62.5	63.0	**64.4**	63.7	61.8	59.6	57.1	55.8
0.72–0.74	57.0	60.8	63.4	61.4	61.9	63.3	64.1	**65.4**	62.2	60.0	58.9	57.0
0.74–0.76	55.9	62.0	64.4	62.7	62.7	63.8	64.3	65.3	**66.9**	64.4	60.2	58.1
0.76–0.78	60.3	64.9	66.4	64.7	65.4	65.7	64.8	67.2	**69.0**	67.9	66.4	62.8
0.78–0.80	61.4	63.0	65.5	63.6	64.0	64.5	66.4	65.5	68.3	**69.1**	67.3	65.2
0.80–0.82	66.5	64.2	68.5	65.7	66.9	66.7	67.0	67.6	68.8	68.5	**70.4**	69.9
0.82–0.84	68.0	66.7	68.3	66.3	67.1	67.7	68.0	68.3	68.4	68.9	**69.2**	68.4
Personalized Scholarly Title Generation (R-1 ↑)												
0.66–0.68	46.2	48.2	47.3	47.1	48.7	**49.1**	48.7	48.4	48.1	47.8	46.8	46.4
0.68–0.70	46.4	48.5	48.3	48.3	48.6	49.2	**49.4**	48.7	48.3	48.1	47.5	47.1
0.70–0.72	47.3	48.8	48.7	48.6	49.2	49.7	**50.3**	49.4	49.1	48.9	48.3	47.9
0.72–0.74	48.2	49.2	49.6	48.8	48.8	48.9	49.3	**51.2**	50.7	49.8	48.8	48.7
0.74–0.76	48.3	49.0	50.1	49.1	49.0	49.3	49.8	50.3	**51.5**	51.2	49.8	49.0
0.76–0.78	49.0	49.4	50.3	49.4	49.0	49.4	49.7	51.3	51.5	**51.6**	51.3	50.2
0.78–0.80	48.8	49.6	50.2	48.7	49.9	49.8	50.9	50.5	51.2	**51.9**	51.5	51.0
0.80–0.82	49.6	49.3	50.3	49.3	49.7	49.9	50.3	50.2	50.9	51.2	**51.7**	50.9
0.82–0.84	50.1	49.8	50.5	49.7	49.9	49.9	50.0	50.4	50.4	50.7	**51.2**	50.5

RAG implementations. More advanced adaptation or retrieval systems (Li et al., 2023; Jeong et al., 2024) are orthogonal to this allocation mechanism and can be incorporated into PUR.

Implementation. All experiments are conducted on a server equipped with 4 NVIDIA RTX8000 GPUs (48 GB each). For PUR, we use `bge-base-en-v1.5` as the embedding model. Under our HDBSCAN setting with `min_cluster_size` $= 0.3N_u$, the historical data of a single LaMP user is typically identified as a single cluster. Therefore, the main experiments use one S_{conc} score per user. For completeness, we further analyze multi-cluster settings using combined user data in Appendix A.1. We set the KNN parameter k to 5 and $\lambda = 1$. Across all users in the benchmark, the semantic concentration score S_{conc} ranges from 0.66 to 0.84. For grouped analysis, we partition users into intervals of width 0.02 and select 10 users with the largest historical data size from each interval to ensure more stable evaluation. For task evaluation, we use 20% of each user's data as evaluation data and treat the remaining 80% as historical data H_u.

6 Evaluation Results and Analysis

Personalized LLM Generation Accuracy/Quality. To study how the optimal allocation ratio changes across users with different semantic concentrations, we first conduct a probing study over a range of candidate allocation ratios. The results are shown in Table 1. We make the following observations: (1) In every concentration interval, PUR outperforms all other methods when using the best allocation ratio for that interval, indicating that the workload split between PEFT and RAG is consistently beneficial. (2) The optimal allocation ratio varies across users with different semantic concentrations, suggesting that no single ratio is suitable for all users. (3) The optimal ratios do not span all candidate values, and extreme partition choices usually fail to fully realize the advantage of PUR. (4) Even when using ratios near the optimum, PUR often still outperforms all other methods, indicating that it is reasonably robust to the exact ratio choice. (5) The relationship between semantic concentration and the optimal allocation ratio is highly similar across the two tasks, suggesting that this allocation pattern has promising cross-task generalization potential. Based on these observations, we summarize

Table 2: Data allocation rule for PUR.

S_{conc} Range	PEFT Ratio
0.66–0.70	30%
0.70–0.72	40%
0.72–0.74	50%
0.74–0.76	60%
0.76–0.80	70%
0.80–0.84	80%

Table 3: Detailed LLM inference performance analysis on representative users with different semantic concentration levels. Each user's new queries are partitioned into PEFT-Region Queries (PRQ) and RAG-Region Queries (RRQ) according to PUR's inference path selection.

Metrics	PRQ Perf.	RRQ Perf.	Overall Perf.	Avg. Infer. Time (s) ↓	PRQ Perf.	RRQ Perf.	Overall Perf.	Avg. Infer. Time (s) ↓	PRQ Perf.	RRQ Perf.	Overall Perf.	Avg. Infer. Time (s) ↓
Personalized Movie Tagging (Acc. ↑)												
Method	User1 (S_{conc}: 0.689 / PRQ Ratio: 34.5%)				User2 (S_{conc}: 0.752 / PRQ Ratio: 56.5%)				User3 (S_{conc}: 0.811 / PRQ Ratio: 83.6%)			
PEFT	56.7	52.6	54.0	**0.85**	57.7	55.0	56.5	**0.90**	68.3	56.3	66.3	**0.79**
RAG	63.3	**61.4**	62.1	1.75	61.5	62.5	62.0	1.83	67.1	56.3	66.3	1.78
Full PEFT+RAG	63.3	57.9	59.8	1.77	63.5	**65.0**	64.1	1.81	68.3	**68.8**	68.4	1.74
PUR (Ours)	**70.0**	**61.4**	**64.4**	1.41	**67.3**	**65.0**	**66.3**	1.29	**70.7**	**68.8**	**70.4**	0.93
Personalized Scholarly Title Generation (R-1 ↑)												
Method	User1 (S_{conc}: 0.701 / PRQ Ratio: 31.5%)				User2 (S_{conc}: 0.743 / PRQ Ratio: 49.8%)				User3 (S_{conc}: 0.809 / PRQ Ratio: 81.1%)			
PEFT	44.1	47.7	46.5	**1.25**	48.1	48.8	48.4	**1.31**	49.8	49.0	49.6	**1.29**
RAG	47.5	48.4	48.1	2.57	49.1	48.9	49.0	2.64	49.4	49.3	49.4	2.51
Full PEFT+RAG	46.3	47.4	47.1	2.64	50.1	50.3	50.2	2.70	50.3	50.4	50.3	2.66
PUR (Ours)	**50.3**	**48.7**	**49.2**	2.20	**52.3**	**50.8**	**51.6**	1.93	**51.9**	**50.7**	**51.6**	1.53

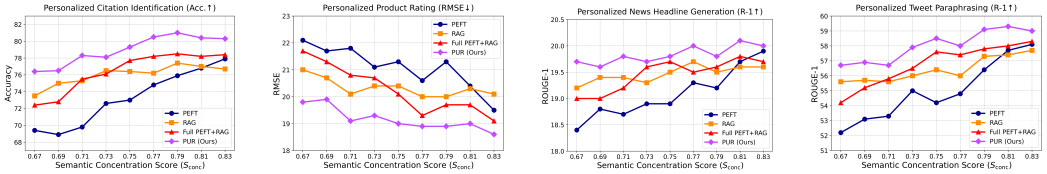


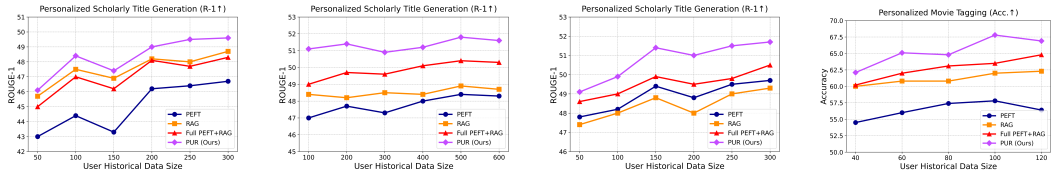
Figure 3: Personalized generation accuracy/quality on the four remaining evaluated LaMP tasks not used in the probing study. PUR applies the allocation ratios obtained from the probing study to evaluate cross-task generalization.

a reference allocation rule for PUR, as shown in Table 2, and use it as the default data partition rule in the subsequent experiments.

To better understand why PUR improves personalized generation accuracy/quality, we select users with different semantic concentrations and conduct a fine-grained analysis of their inference-time evaluation results, as shown in Table 3. Based on PUR's inference-time routing decisions, we partition each user's test queries into PEFT-Region Queries (PRQ) and RAG-Region Queries (RRQ). First, compared with RAG, PUR performs significantly better on PRQ. A key reason is that PUR trains its personalized model on highly concentrated core information, which is consistent with our earlier observation that parametric learning can outperform RAG on highly concentrated data. On RRQ, PUR also relies on retrieval, but its responses are generated by a personalized model that has already learned the core information, allowing it to remain competitive with RAG. Second, compared with PEFT+RAG, PUR benefits from training its personalized model on less dispersed and noisy data. This cleaner training signal improves performance on both PRQ and RRQ, and ultimately leads to better overall performance.

Personalized LLM Inference Latency. Table 3 also reports the average inference latency for users at different semantic concentration levels. The results show that PUR not only achieves the best generation accuracy/quality, but also incurs lower inference latency than the retrieval-based methods. For User 1 with lower semantic concentration, PUR reduces the average inference time by 19.4% and 14.4% on the two tasks, respectively. For User 3 with higher semantic concentration, this latency advantage becomes even more pronounced, reaching 46.6% and 42.5%, respectively. This is because users with higher semantic concentration tend to have a larger fraction of queries routed to the PEFT region in PUR, allowing more responses to be generated without retrieval and thereby reducing average inference latency.

Generalization of the Workload Splitting Strategy. To study how the allocation rule derived from the probing study generalizes beyond the probing setting, we further conduct experiments on the four remaining LaMP tasks that are not used in the probing study. As shown in Fig. 3, PUR consistently achieves the best performance across tasks and user groups with different semantic concentration levels, demonstrating strong cross-task and cross-user generalization. Although this empirical rule cannot guarantee that every user is assigned exactly their optimal allocation ratio, it still consistently outperforms other methods.



(a) S_{conc}: 0.68-0.70 (b) S_{conc}: 0.74-0.76 (c) S_{conc}: 0.80-0.82 (d) S_{conc}: 0.74-0.76

Figure 4: Impact of historical data size on the relative performance of the methods. PUR consistently achieves the best performance across different user historical data sizes, while the relative ranking of the other methods remains largely unchanged within the same semantic concentration interval.

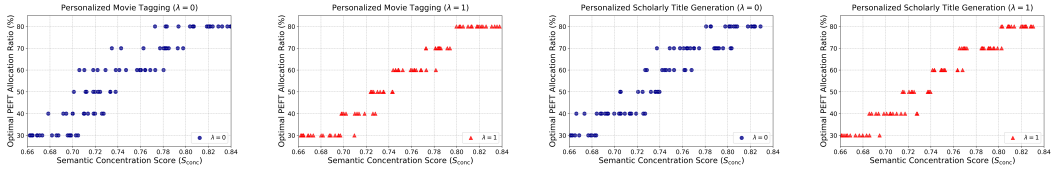


Figure 5: Effect of the long-tail compensation term on the relationship between S_{conc} and the optimal PEFT allocation ratio. Each point represents a sampled user. Adding the compensation term ($\lambda = 1$) yields better alignment in the score space than using average similarity alone ($\lambda = 0$).

Impact of Historical Data Scale. To study the impact of historical data size, we compare the performance of different methods across users with different amounts of historical data within the same semantic concentration interval. Figure 4 shows the results across different tasks and concentration intervals. The results show that PUR consistently achieves the strongest performance across different historical data sizes. More importantly, unlike our earlier observations on semantic concentration, the relative ranking of the other methods remains largely unchanged as historical data size varies. This indicates that, for strategy selection, the distributional structure of historical data is a more important factor than its scale.

Ablation Study of the Long-Tail Compensation Term. To study the effect of the long-tail compensation term in S_{conc}, we analyze how users' S_{conc} values relate to their optimal PEFT allocation ratios under different values of λ. As shown in Fig. 5, when $\lambda = 0$, i.e., without the long-tail compensation term, several users deviate from the overall S_{conc}-to-allocation trend. A closer inspection (Appendix A.4) shows that these deviating users consistently have larger values of the tail-related statistic $\pi_{\text{tail},i}\, h_{\text{tail},i}$, indicating that average similarity alone is insufficient to provide a unified measure across users with different historical data distributions. When $\lambda = 1$, the resulting scores become better aligned with the overall trend, and users sharing the same optimal allocation ratio become more tightly grouped in the score space. This indicates that the long-tail compensation term improves the reliability of S_{conc} as a signal for data allocation selection.

7 Conclusion

In this paper, we presented PEFT Union RAG (PUR), a novel "fast and accurate" personalized LLM approach that allocates user historical data between PEFT and RAG based on semantic concentration. Our empirical results show that such workload split between PEFT and RAG achieves better generation accuracy/quality than PEFT only, RAG only, or full PEFT+RAG, while reducing inference latency compared with the retrieval-based methods. The key reason is that PUR restricts PEFT training to concentrated core historical data, which reduces the impact of dispersed noisy information and leads to better generation quality. Meanwhile, queries whose required knowledge is covered by this core information can bypass retrieval, thereby reducing the averaged online LLM inference latency. More importantly, PUR takes the heterogeneity of users' historical data distributions into account and adaptively determines the data allocation accordingly. Extensive experiments across multiple tasks show that PUR consistently outperforms other methods for users with different semantic concentration, while reducing inference latency by 14.4%–46.6% compared with the best performed compared method. These results collectively show that PUR is a practical and high-performing approach for leveraging historical data in personalized LLMs.

AI USE STATEMENT

In this work, we did not use generative AI tools to formulate or refine research hypotheses, implement the research methods, generate or clean datasets, prove mathematical claims, or interpret experimental results. Tasks related to mathematical proofs and qualitative or thematic data analysis are not applicable to this work.

Additionally, we used generative AI tools to improve the readability of the manuscript and assist with formatting. All AI-assisted manuscript edits were reviewed and validated by the authors. We take full responsibility for the final content of this work, including the text, research claims, experimental design, data, experimental results, and related research artifacts.

ETHICS STATEMENT

This work studies personalized LLM generation using users' historical data. All experiments are conducted on the existing LaMP personalization benchmark, and we did not recruit new human participants or directly collect any new personally identifiable or sensitive user information. The user historical data used in our experiments are obtained through the existing benchmark and are used solely for research and evaluation purposes. We do not attempt to identify, track, or re-identify individual users.

Personalized LLM systems may raise privacy and data-governance concerns when deployed with real user histories. Our work focuses only on the allocation and inference strategies of PEFT and RAG under an existing research benchmark and does not involve the collection or deployment of real-world user data. Any practical deployment should follow applicable data licenses, privacy requirements, and user-consent mechanisms.

REFERENCES

Ricardo JGB Campello, Davoud Moulavi, Arthur Zimek, and Jörg Sander. Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):1–51, 2015.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Aleksander Ficek, Jiaqi Zeng, and Oleksii Kuchaiev. Gpt vs retro: Exploring the intersection of retrieval and parameter-efficient fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 19425–19432, 2024.

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*, pp. 2790–2799. PMLR, 2019.

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

Boris Iglewicz and David C Hoaglin. *Volume 16: how to detect and handle outliers*. Quality Press, 1993.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 7036–7050, 2024.

Wenqi Jiang, Suvinay Subramanian, Cat Graves, Gustavo Alonso, Amir Yazdanbakhsh, and Vidushi Dadu. Rago: Systematic performance optimization for retrieval-augmented generation serving. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, pp. 974–989, 2025.

Robert Lakatos, Peter Pollner, Andras Hajdu, and Tamas Joo. Investigating the performance of retrieval-augmented generation and domain-specific fine-tuning for the development of ai-driven knowledge-based systems. *Machine Learning and Knowledge Extraction*, 7(1):15, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. Llms+ persona-plug= personalized llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9373–9385, 2025.

Rohan Mahapatra, Harsha Santhanam, Christopher Priebe, Hanyang Xu, and Hadi Esmaeilzadeh. In-storage acceleration of retrieval augmented generation as a service. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, pp. 450–466, 2025.

Yusuke Miura, Chi-Lan Yang, Masaki Kuribayashi, Keigo Matsumoto, Hideaki Kuzuoka, and Shigeo Morishima. Understanding and supporting formal email exchange by answering ai-generated questions. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pp. 1–20, 2025.

Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in llms. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 237–250, 2024.

Bhagyajit Pingua, Adyakanta Sahoo, Meenakshi Kandpal, Deepak Murmu, Jyotirmayee Rautaray, Rabindra Kumar Barik, and Manob Jyoti Saikia. Medical llms: Fine-tuning vs. retrieval-augmented generation. *Bioengineering*, 12(7):687, 2025.

Alireza Salemi and Hamed Zamani. Comparing retrieval-augmentation and parameter-efficient fine-tuning for privacy-preserving personalization of large language models. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pp. 286–296, 2025.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, 2024.

Rulin Shao, Jacqueline He, Akari Asai, Weijia Shi, Tim Dettmers, Sewon Min, Luke Zettlemoyer, and Pang W Koh. Scaling retrieval-based language models with a trillion-token datastore. *Advances in Neural Information Processing Systems*, 37:91260–91299, 2024.

Michael Shen, Muhammad Umar, Kiwan Maeng, G Edward Suh, and Udit Gupta. Hermes: Algorithm-system co-design for efficient retrieval-augmented generation at-scale. In *Proceedings of the 52nd Annual International Symposium on Computer Architecture*, pp. 958–973, 2025.

Haochen Tan, Zhijiang Guo, Zhan Shi, Lu Xu, Zhili Liu, Yunlong Feng, Xiaoguang Li, Yasheng Wang, Lifeng Shang, Qun Liu, et al. Proxyqa: An alternative framework for evaluating long-form text generation with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6806–6827, 2024a.

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

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

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.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 641–649, 2024.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. *Advances in Neural Information Processing Systems*, 37:121156–121184, 2024.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Hamed Zamani, Fernando Diaz, Mostafa Dehghani, Donald Metzler, and Michael Bendersky. Retrieval-enhanced machine learning. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 2875–2886, 2022.

Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. Collaborative large language model for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pp. 3162–3172, 2024.

A Further Experimental Results and Discussion

A.1 Applicability to Multi-Cluster Historical Data

Table 4: Performance comparison under multi-cluster historical data. Each team consists of two users whose merged historical data is clearly detected by HDBSCAN as forming two clusters. We report each user's individual evaluation performance and the average performance over all queries from both users (Task: Personalized Scholarly Title Generation).

Team	(C_1 S_{conc}: 0.745, C_2 S_{conc}: 0.752)			(C_1 S_{conc}: 0.811, C_2 S_{conc}: 0.838)			(C_1 S_{conc}: 0.769, C_2 S_{conc}: 0.823)		
Method	User1 Perf.	User2 Perf.	Overall Perf.	User1 Perf.	User2 Perf.	Overall Perf.	User1 Perf.	User2 Perf.	Overall Perf.
PEFT	47.5	48.3	47.9	49.1	49.9	49.5	48.0	49.5	48.9
RAG	49.3	49.6	49.5	49.4	49.2	49.3	47.7	48.8	48.3
Full PEFT+RAG	50.1	50.7	50.4	50.8	50.4	50.6	48.6	50.2	49.5
PUR (Ours)	**51.5**	**52.3**	**51.9**	**51.6**	**52.0**	**51.8**	**50.1**	**52.9**	**51.7**

In the main experiments, a single user's historical data typically exhibits a single-cluster structure under the current HDBSCAN setting. To further examine the applicability of PUR to multi-cluster data, we consider an additional personalized modeling scenario in which multiple team members share one personalized model. Since different members' historical data may reflect different preferences or expression styles, the merged history may contain multiple relatively independent semantic centers.

To simulate this setting, we group users in pairs and run HDBSCAN on the merged historical data. We retain only those user pairs whose merged data are clearly detected as forming two clusters, and evaluate PUR against the other methods on these combinations. For each cluster, we apply the same concentration modeling and data allocation as in the main paper. The PEFT-selected samples from all clusters are then combined to fine-tune a single personalized model, rather than training separate PEFT models for individual clusters.

Table 4 reports the results for three user-pair combinations on the personalized scholarly title generation task. We report both the performance on each user's individual evaluation queries and the average performance over all queries from the two users. The results show that PUR continues to outperform the other methods in these two-cluster settings, suggesting that the cluster-wise PUR design remains effective beyond the single-cluster case considered in the main experiments.

A.2 Effect of the Embedding Model

Table 5: Performance comparison under different embedding models. For the same user groups, we report the resulting S_{conc} range, generation quality, and PUR's PEFT subset overlap under different embedding models (Task: Personalized Scholarly Title Generation).

		User Group 1			User Group 2			User Group 3		
Embedding model	Method	S_{conc} Range	(R-1 ↑)	PEFT Subset Overlap	S_{conc} Range	(R-1 ↑)	PEFT Subset Overlap	S_{conc} Range	(R-1 ↑)	PEFT Subset Overlap
BGE-Base-en-v1.5	PEFT	–	46.4	–	–	48.3	–	–	49.6	–
	RAG	–	48.5	–	–	49.0	–	–	49.3	–
	Full PEFT+RAG	–	48.3	–	–	50.1	–	–	50.3	–
	PUR (Ours)	0.68-0.70	**49.4**	100%	0.74-0.76	**51.5**	100%	0.80-0.82	**51.7**	100%
E5-Base-v2	PEFT	–	46.4	–	–	48.3	–	–	49.6	–
	RAG	–	48.7	–	–	49.3	–	–	49.0	–
	Full PEFT+RAG	–	48.3	–	–	50.0	–	–	49.9	–
	PUR (Ours)	0.79-0.80	**49.1**	77.3%	0.81-0.82	**51.8**	83.6%	0.83-0.84	**51.4**	91.6%
GTE-Base	PEFT	–	46.4	–	–	48.3	–	–	49.6	–
	RAG	–	48.1	–	–	49.0	–	–	49.5	–
	Full PEFT+RAG	–	48.0	–	–	50.1	–	–	50.1	–
	PUR (Ours)	0.77-0.78	**49.1**	82.1%	0.80-0.82	**51.6**	89.5%	0.84-0.85	**51.7**	93.5%

Since PUR's concentration modeling is built on the vector space induced by the embedding model and the corresponding similarity relations, we further evaluate PUR under different embedding models. In addition to the `bge-base-en-v1.5` (Xiao et al., 2024) model used in the main paper, we also consider `e5-base-v2` (Wang et al., 2022) and `gte-base` (Li et al., 2023) models. Specifically, we select users from the personalized scholarly title generation task whose S_{conc} scores under the

BGE embedding model fall into the intervals 0.68–0.70, 0.74–0.76, and 0.80–0.82. We then evaluate the same set of users under different embedding models.

First, as shown by the S_{conc} ranges for the same set of users in Table 5, the overall range of S_{conc} shifts across embedding models, but the relative relationships among users remain broadly consistent. This is expected, since different embedding models induce different representation spaces and similarity scales, making the absolute values of the concentration scores not directly comparable. What matters most is whether the scores preserve the relative concentration differences across users, rather than whether their numerical values are exactly the same.

Based on this observation, for the same set of users under the other two embedding models, we reuse the PEFT ratios derived under the BGE embedding model to partition the data, and then evaluate both generation quality and the overlap between the PEFT-selected subsets across embedding models. As shown in Table 5, PUR continues to outperform the competing methods for the same users under the other two embedding models, and the selected PEFT subsets also exhibit high overlap. These results suggest that the core mechanism of PUR remains stable across embedding models: data partitioning continues to exploit the complementary strengths of PEFT and RAG, and the conclusion that different users should adopt different allocation ratios remains unchanged. What may vary is the precise S_{conc}-to-ratio mapping, which may require slight empirical recalibration under a new embedding model.

A.3 Computational Cost Analysis

Table 6: Computational cost comparison of different methods in the offline and online inference stages. We report the computational cost for the same three users analyzed in Table 3 on the Personalized Movie Tagging task.

Method	User 1 (Low S_{conc})		User 2 (Medium S_{conc})		User 3 (High S_{conc})	
	Offline Comp. (PFLOPs) ↓	Inference Comp. (TFLOPs) ↓	Offline Comp. (PFLOPs) ↓	Inference Comp. (TFLOPs) ↓	Offline Comp. (PFLOPs) ↓	Inference Comp. (TFLOPs) ↓
PEFT	1.096	**30.7**	1.107	**34.7**	1.209	**36.1**
RAG	**0.002**	170.9	**0.002**	208.8	**0.002**	265.4
Full PEFT+RAG	1.098	170.9	1.109	208.8	1.211	265.4
PUR (Ours)	0.329	122.5	0.665	110.4	0.967	73.7

We further compare the computational cost of different methods in both the offline stage and the online inference stage, measured in FLOPs. Table 3 reports the corresponding costs for the three users in Table 3 on the personalized movie tagging task.

The results show that PEFT and RAG exhibit clearly different computational cost profiles. For PEFT, the major cost is concentrated in the offline stage, since parameter training requires substantial computation. However, its online cost remains low since inference only uses a relatively short input context. RAG shows the opposite pattern. Its offline cost is small, but its online cost is much higher because retrieved texts make the input much longer, and the computation grows quickly with context length. Unlike the two extreme cost patterns above, PUR splits historical data between PEFT and RAG, making the computational cost more balanced across the offline and online stages.

A.4 Further Analysis of the Long-Tail Compensation Term

We further analyze the distribution of the S_{avg} scores and the long-tail compensation term $\pi_{\mathrm{tail},i} h_{\mathrm{tail},i}$ for the users involved in Fig. 5. The results in Fig. 6 show that, in real user data, most users have relatively small $\pi_{\mathrm{tail},i} h_{\mathrm{tail},i}$ values, while a smaller subset exhibits substantially larger values, indicating more pronounced long-tail structure in their historical data. These users with larger compensation terms closely align with the points that deviate from the overall trend in Fig. 5 when $\lambda = 0$. This suggests that, when a single S_{conc} score is used to characterize users with different data distribution patterns, relying only on S_{avg} tends to underestimate the appropriate PEFT allocation needs of users with pronounced long-tail structure. Overall, this further supports that the long-tail compensation term improves the reliability of S_{conc} as a signal for data allocation selection.

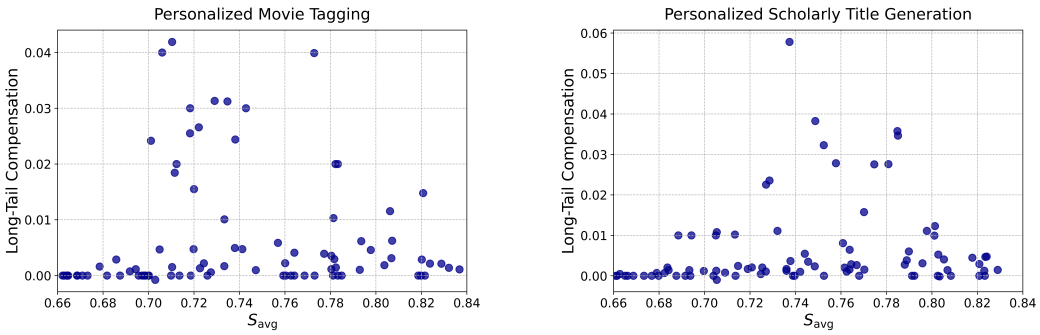


Figure 6: Distribution of S_{avg} and the long-tail compensation term $\pi_{\text{tail},i} h_{\text{tail},i}$ for the users involved in Fig. 5. Most users have relatively small compensation values, while a smaller subset exhibits substantially larger values and corresponds closely to the points that deviate from the overall trend when $\lambda = 0$.

B Dataset Details

B.1 Historical Data Size

Table 7: Average historical data size of the sampled evaluation users across different S_{conc} intervals and LaMP tasks.

Average historical data size of sampled users within each S_{conc} interval						
S_{conc} Range	LaMP-1	LaMP-2	LaMP-3	LaMP-4	LaMP-5	LaMP-7
0.66–0.68	238	181	884	93	341	36
0.68–0.70	335	126	1006	82	450	48
0.70–0.72	403	124	946	119	580	59
0.72–0.74	432	113	923	87	551	85
0.74–0.76	388	137	891	127	528	77
0.76–0.78	296	109	748	60	397	59
0.78–0.80	289	98	410	91	331	99
0.80–0.82	177	152	549	108	263	108
0.82–0.84	210	152	326	151	225	151

Here we present the average historical data size selected for each task in the main experiment across different S-score intervals to illustrate the scale of user data, as shown in Table 7.

B.2 Task Details

We present the task details as follows to help readers gain a better understanding of the task format.

LaMP-1: Personalized Citation Identification is a binary text classification task. Specifically, given user u writes a paper x, the task aims to make the model determine which of the two candidate papers u will cite in paper x based on the user's history data, which contains the publications of user u.

LaMP-2: Personalized Movie Tagging is a 15-way text classification task to make tag assignments aligned with the user's history tagging preference. Specifically, given a movie description x, the model needs to predict one of the tags for the movie x based on the user's historical movie-tag pairs.

LaMP-3: Personalized Product Rating is a 5-way text classification task and can also be understood as a regression task. Given the user u's historical review and rating pairs and the input review x, the model needs to predict the rating corresponding to x selected from 1 to 5 in integer.

LaMP-4: Personalized News Headline Generation is a text generation task to test the model's ability to capture the stylistic patterns in personal data. Given a query x that requests to generate a news headline for an article, as well as the user profile that contains the author's historical article-title pairs, the model is required to generate a news headline specifically for the given user.

LaMP-5: Personalized Scholarly Title Generation is a text generation task to test personalized text generation tasks in different domains. In this task, we require language models to generate titles for an input article x, given a user profile of historical article-title pairs for an author.

LaMP-7: Personalized Tweet Paraphrasing is also a text generation task that tests the model's capabilities in capturing the stylistic patterns of authors. Given a user input text x and the user profile of historical tweets, the model is required to paraphrase x into y that follows the given user's tweet pattern.