Democratizing Large Language Models via Personalized Parameter-Efficient Fine-tuning

Zhaoxuan Tan
ztan3@nd.edu
University of Notre Dame
Notre Dame, IN, USA

Qingkai Zeng
qzeng@nd.edu
University of Notre Dame
Notre Dame, IN, USA

Yijun Tian
yijun.tian@nd.edu
University of Notre Dame
Notre Dame, IN, USA

Zheyuan Liu
zliu29@nd.edu
University of Notre Dame
Notre Dame, IN, USA

Bing Yin
alexbyin@amazon.com
Amazon.com Inc
Palo Alto, CA, USA

Meng Jiang
mjiang2@nd.edu
University of Notre Dame
Notre Dame, IN, USA

ABSTRACT

Personalization in large language models (LLMs) is increasingly important, aiming to align LLM's interactions, content, and recommendations with individual user preferences. Recent advances in LLM personalization have spotlighted effective prompt design, by enriching user queries with non-parametric knowledge through behavior history retrieval and textual profiles. However, these approaches were limited due to a lack of model ownership, resulting in constrained customization and privacy issues. Moreover, they often failed to accurately capture user behavior patterns, especially in cases where user data were complex and dynamic. To address these shortcomings, we introduce **O**ne **P**EFT **P**er **U**ser (OPPU)[1], which employs personalized parameter-efficient fine-tuning (PEFT) modules, to store user-specific behavior patterns and preferences. By plugging in users' personal PEFT parameters, they can own and use their LLMs personally. OPPU integrates parametric user knowledge in the personal PEFT parameters with the non-parametric knowledge acquired through retrieval and profile. This integration adapts individual LLMs to user behavior shifts. Experimental results demonstrate that OPPU significantly outperforms existing prompt-based methods across seven diverse tasks in the LaMP benchmark. Further in-depth studies reveal OPPU's enhanced capabilities in handling user behavior shifts, modeling users at different active levels, maintaining robustness across various user history formats, and displaying versatility with different PEFT methods.

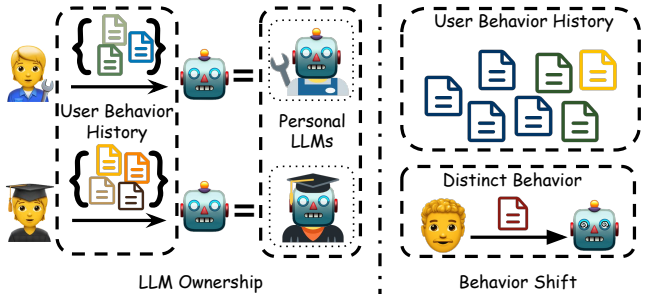

Figure 1: *LLM ownership* and *behavior shift* are two challenges that developing personalized LLMs has to face. Ownership emphasizes that the model needs to be owned by individual user to enhance customization and privacy. Behavior shift adaption refers to the LLMs' ability to effectively generalize and adapt to emerging new patterns in user behaviors.

1 INTRODUCTION

Personalization refers to mining users' behavior history, and therefore tailoring and customizing a system's interactions, content, or recommendations to meet specific needs, preferences, and characteristics of individual users [3, 4, 48]. By adapting to each user's preferences, personalization systems enhance user experience, increasingly getting vital in areas like content recommendation [2, 26, 42, 57], user simulation [9, 59], personalized chatbots [34, 42, 47], user profiling [12, 14], healthcare [13, 22], and education [1, 41].

Large language models (LLMs) display emergent abilities not seen in smaller models [33, 54], as they have billions of parameters and are trained on vast corpora. These abilities include step-by-step reasoning [55], in-context learning [35], and instruction following [53]. However, existing LLMs predominantly follow the "one-size-fits-all" paradigm. They are generally trained on extensive, domain-agnostic datasets, which limits their effectiveness in meeting the specific needs and preferences of individual users [4, 44]. Therefore, the challenge of integrating the strong generative capabilities of LLMs with the tailored requirements of individual users has emerged as a significant area of research [21, 25].

Existing works on personalizing LLMs have predominantly concentrated on the development of prompt templates. These prompt-based personalization methods fall into three categories: vanilla personalized prompt, retrieval-augmented personalized prompt, and profile-augmented prompt. The vanilla personalized prompt approach leverages the in-context learning capability of LLMs, utilizing the user's entire or random sampled behavior history as contextual examples [7, 8, 60]. Considering the growing length of user behavior history and the limited context window of LLMs, some studies have applied retrieval methods to select the most relevant part of user behavior history to enhance LLM personalization [36, 44]. Besides the retrieval, some techniques explicitly generate user preferences and profiles in natural language to augment LLMs' input. For instance, Richardson et al. [43] proposed to employ instruction-tuned LLMs, *e.g.*, ChatGPT, to summarize user preferences and behavior patterns based on their history content.

[1]The code is available at https://github.com/TamSiuhin/OPPU

Despite much research progress has been made in LLM personalization, existing methods face ownership and behavior shift challenges, which are illustrated in Figure 1:

- **Ownership**: Existing methods are processed in a centralized way, where user history is encoded in a personalized prompt and processed by centralized LLMs. This paradigm limits the model's customization and ability to provide deep, personalized experiences tailored to individual users. Moreover, when using a centralized model, users often have to share personal data with the service provider, which raises concerns about how user data are stored, used, and protected.
- **Behavior Pattern Generalization**: As is revealed by Shi et al. [46], LLMs can be easily distracted by irrelevant context information that retrieval can hardly avoid. In the realm of LLM personalization, where the retrieval corpus was confined to a specific user's behaviors, retrieval augmentation might underperform, especially when the user's past behaviors did not closely mirror the patterns needed for the query at hand.

In light of these challenges, we propose **O**ne **P**EFT **P**er **U**ser (OPPU), where each user is equipped with a personalized, parameter-efficient fine-tuning (PEFT) module. Characterized by PEFT's plug-and-play functionality and the minimal weight of updated parameters (typically less than 1% of the base LLM), OPPU facilitates LLM ownership and exhibits superior generalization in scenarios of user behavior shifts. By fine-tuning the PEFT module with the user's personal behavior history, the personalized PEFT parameters encapsulate behavior patterns and preferences. This process, when integrated into base LLMs, allows users to obtain their private LLMs, ensuring LLM ownership and enhancing model customization. Furthermore, as is revealed by Gupta et al. [15], fine-tuning LLMs is more effective than retrieval augmentation when the retrieved instances are not highly relevant to the query. The fine-tuned personal LLMs in OPPU are adept at capturing complex behavior patterns and thus capable of understanding new behaviors with less reliance on highly relevant history data.

Experimental results show that OPPU achieves state-of-the-art performance on all seven public tasks in the Language Model Personalization (LaMP) benchmark [44]. Additional studies highlight the importance of integrating non-parametric user knowledge, sourced from retrieved user history, with parametric user knowledge from personal PEFT parameters. Notably, in scenarios of user behavior shift, where the user history is less relevant to the current user query at hand, OPPU significantly outperforms retrieval-based methods. Moreover, OPPU exhibits strong resilience against varying user history formats and demonstrates versatility across different PEFT methods, among other advantages.

To summarize, the contribution of OPPU lies in its pioneering approach to PEFT-based LLM personalization. Each user (or user cohort) benefits from a personal PEFT module, which not only ensures LLM ownership but also significantly improves the model's ability to adapt to shifts in user behavior. The superiority of OPPU is evidenced by state-of-the-art performance across seven tasks in the LaMP benchmark. By introducing this innovative parametric-based personalization technique, OPPU opens up new opportunities in the realm of democratizing personalized LLMs.

2 RELATED WORK

2.1 Personalization of LLMs

The thrust of existing LLM personalization research is centered on designing prompt that incorporate historical user-generated content and behavior. These approaches help LLMs understand users' preferences, tailoring responses to individual needs [4, 48]. The endeavors towards personalized LLMs mainly fall into three categories: vanilla personalized prompts, retrieval-augmented personalized prompts, and profile-augmented personalized prompts.

In the *vanilla personalized prompt* category, researchers use in-context and few-shot learning to encode either complete or a sample of user behavior history as contextual examples. For instance, Dai et al. [8] and Kang et al. [23] encode the user's personal rating history as few-shot demonstration examples. Liu et al. [30] supply LLMs with user's interaction history related to specific tasks to help LLMs generate personalized content. PerSE [51] design prompt to do personalized story assessment by presenting a few exemplary reviews from the review. BookGPT [60] employs a few-shot prompt strategy to make LLMs understand the correlation between book content and personalized prompting. Moreover, some research works [7, 60] also discovered a long user history would bring better performance. To manage the growing user behavior data and LLMs' limited context windows, the *retrieval-augmented personalized prompt* approach has emerged. For instance, LaMP [44] introduces a retrieval-augmented method to obtain the most relevant content in the user's behavioral history and incorporate it into the prompt. AuthorPred [25] utilizes retrieve relevant past user-written documents for personalized text generation. Pearl [36] proposes a generation-calibrated retriever to select historic user-authored documents for prompt augmentation. Moving beyond simple retrieval, some researchers summarize user preferences and behavior patterns into natural language profiles for input query augmentation, termed *profile-augmented personalized prompts*. Richardson et al. [43] use the instruction-tuned LLMs to generate an abstract summary of user history data, augmenting retrieval-based personalization methods. ONCE [31] employs LLMs to generate users' topics and regions of interest based on their browsing history as user profiles, aiding LLMs in preference capture in downstream tasks. There is also another line of work focusing on designing personalized alignment methods via parameter merging [21] and personalized reward model [5].

Previous works largely hinge on prompt design, integrating retrieval and user profiles. However, existing approaches are limited by model ownership and users' behavior shifts. Our OPPU introduces personalization at the parametric level, via the personal PEFT module, which ensures model ownership and superior generalization, especially in cases with less relevant user history for retrieval.

2.2 Parameter-Efficient Fine-tuning (PEFT)

With the exponentially growing parameters in LLMs, fine-tuning all parameters is expensive [15, 32, 58]. To mitigate this gap, a few lightweight alternatives, known as parameter-efficient fine-tuning (PEFT) have been proposed to update only a small number of extra parameters while keeping the pretrained weights frozen to save the computes [11, 17]. For example, adapter tuning [19] injects learnable parameters in the models' each feedforward layer,

and only the plug-in parameters are updated at the fine-tuning stage. Inspired by the success of discrete textual prompt [45, 52], prefix tuning [27] and prompt tuning [24] are proposed to learn the optimized prompt and prefix for specific use via fine-tuning. LoRA [20] proposes to add low-rank matrices on pretrained weights to approximate parameter updates. $(IA)^3$ [29] plugs in learned vectors to scale the activation in the attention mechanism for efficient fine-tuning. These approaches achieve comparable performance to full parameter fine-tuning by updating less than 1% of the original LLM parameters. Besides parameter savings, PEFT methods are effective in combating catastrophic forgetting [40] and robust to out-of-distribution samples [27].

The small number of updated parameters and plug-and-play nature of PEFT make it an ideal solution for efficient LLM personalization and constitute model ownership. Our work pioneers the concept of storing user history within personal PEFT parameters. Each user is equipped with a unique, easily integrable PEFT module, serving as a key to their own personalized LLMs.

3 PROPOSED APPROACH

3.1 Research Problem Formulation

Generative language models generally take an input token sequence x and output a sequence of tokens y that follows the x. For the personalization of LLMs, happening at time t, the language model's output y_u specifically for user u is conditioned on both input x_u and the user u's behavior history $\mathcal{H}_u$. To elaborate, $\mathcal{H}_u^t = \{h_u^{t_i}\}, t_i < t$, where the user's history data $\mathcal{H}_u$ contains all user behavior h^{t_i} happened before query time t. More specifically, the user behavior $h_u^{t_i}$ may consist of $(x_u^{t_i}, y_u^{t_i})$ pairs, mirroring the task-specific query-answer format (x_u, y_u). Alternatively, $h_u^{t_i}$ could be text sequences that provide context for the user's behavior patterns without necessarily conforming to the task's format. Overall, each data entry contains three components: input sequence x_u given by the user, user's history behavior $\mathcal{H}_u$ that contains the user's behavior that happened before the input query, and a target output y_u specifically for user u that the model is expected to produce.

3.2 Base LLMs Task Adaption

Given that off-the-shelf LLMs are not inherently equipped for personalization tasks, we align with the methods of LaMP [44] and Richardson et al. [43] by fine-tuning LLMs for fair comparison. In this section, we will describe the process of developing base LLMs, a critical step to enhance their general capability in comprehending and executing the specific personalization tasks required, without specific user's preference or bias.

Non-personalized base LLM (LLM_{NP}*)* ignores the user's behavior history and serves as a baseline. It is fine-tuned on $\{(\phi_p(x_u), y_u)\}$ data pairs, where ϕ_p is the prompt construction function. This approach only contains the task-related data and ignores the user behavior history, therefore making it non-personalized.

Retrieval-augmented base LLM (LLM_{RAG}*)* is fine-tuned on input that includes the retrieved top-k relevant user history items, which the input sequence x' can be defined as:

$$\overline{x_u} = \phi_p(x_u, \mathcal{R}(x_u, \mathcal{H}_u, k)), \tag{1}$$

where $\mathcal{R}$ denotes the retriever, ϕ_p denotes the prompt construction function. This approach aims to enhance personalization by integrating the most pertinent user history items directly into the LLM's prompt.

Profile-Augmented Base LLM (LLM_{PAG}*)* enhances its input by appending a natural language user profile that describes the user's preferences and behavior patterns. This profile can be added to the non-personalized and retrieval-augmented prompts. The input x' to LLM could be denoted as:

$$\overline{x_u} = \phi_p(x_u, \mathcal{R}(x_u, \mathcal{H}_u, k), s_u), \tag{2}$$

where s_u is a textual user profile that describes user's preference and behavior patterns that are generated by an instruction-tuned LLM (Vicuna [6] or ChatGPT [37]) based on the user history data.

To make this process more computationally efficient, we adopt the low-rank adaptation (LoRA) [20] for base modeling task adaption that only updates about 0.5% of external parameters compared to the total LLM parameter size. After training the LoRA, we merge the LoRA parameters to the base model to get base LLMs equipped with the capabilities for the corresponding tasks.

3.3 One PEFT Per User (OPPU)

After obtaining the base model that comprehends the responding task, in real-world deployment, users can only assess the parameter of the base model and their personal behavior history data, controlling the privacy risks. In this section, we will introduce how to make personalized LLMs via parameter-efficient fine-tuning (PEFT), and its integration with non-parametric personalization, i.e., retrieval-augmentation and profile-augmentation. To better understand the parametric personal knowledge in PEFT parameters and non-parametric personalized knowledge in the prompt, we decompose OPPU into parametric knowledge only and the integration of both parametric and non-parametric personalized knowledge.

3.3.1 Inject Parametric Personalized Knowledge via PEFT. We first explore injecting the parametric personalized knowledge into base LLMs and exclude the non-parametric augmentation method to focus purely on the impact of parametric modifications. In this scenario, the user history data are all stored in the PEFT module's parameters thus enhancing privacy preservation. For each user u, OPPU would maintain a private PEFT module PEFTu, tailored to capture and adapt to the user's behavioral patterns as reflected in their historical data. More specifically, consider a user u with history $\mathcal{H}_u$, the personalized PEFT module is plugged in base LLM LLM_{NP} and optimized on the user's personal history data. The input sequence x' for LLMs is $x'_u = \phi_p(x_u^{t_i})$ and optimized using the standard cross entropy loss against the corresponding output $y_u^{t_i}$, where $(x_u^{t_i}, y_u^{t_i}) \in \mathcal{H}_u$. This procedure could capture the user behavior pattern presented in all user behavior history more comprehensively store it in plug-and-play personal PEFT parameters, constituting a personalized LLM *owned* by users.

3.3.2 Integrating Non-Parametric & Parametric Knowledge. Inspired by recent research on debating between fine-tuning versus retrieval-augmented generation [15], as well as insights into the integration of non-parametric knowledge through retrieval and parametric knowledge via PEFT, we proceed to investigate the integration

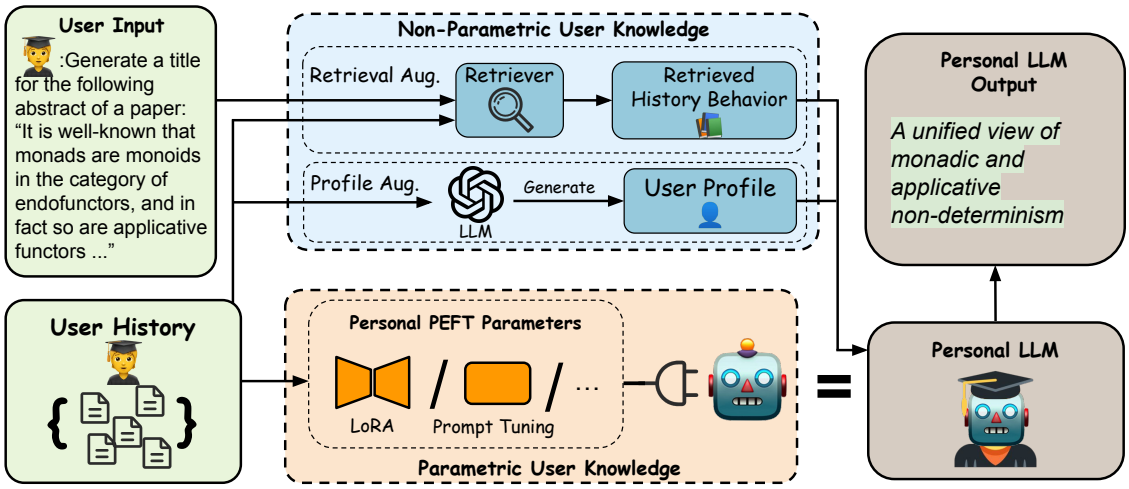

Figure 2: Overview of our proposed OPPU, where each user is equipped with a personal PEFT module and plug-in base LLMs to get their individual LLM. Beyond parametric personalization via PEFT, OPPU is also compatible with the non-parametric user knowledge via retrieval and profile augmentation.

of our personalized PEFT parametric user knowledge with non-parametric user knowledge from both retrieval and profile augmentation methods.

Integrating with Retrieval Augmentation. Our first step involves integrating the personal PEFT module with retrieval augmentation, in which the LLM_{RAG} acts as the base LLM and each user's plug-in PEFT parameters are updated on the following input sequence x':

$$x_u^{t_i\prime} = \phi_p(x_u^{t_i}, \mathcal{R}(x_u^{t_i}, \mathcal{H}_u^{t_i}, k)), \tag{3}$$

where ϕp denotes the prompt construction function and $\mathcal{R}$ represents the retriever. The retriever selects the top k relevant history items from the user's behavior history corpus $\mathcal{H}_u^{t_i}$ in response to the user history query x_u. It's important to note that the retrieval is based on past behavior that occurred before the current query $x_u^{t_i}$. The PEFT parameters are updated using cross entropy loss against the corresponding historical user response $y_u^{t_i}$.

Integrating with User Profile Augmentation. Following Richardson et al. [43], we also incorporate profile augmentation into our method for injecting non-parametric knowledge into prompts. In this scenario, the personal PEFT parameter PEFT_u is plugged in LLM_{PAG} and optimized using the input $x_u^{t_i\prime}$:

$$x_u^{t_i\prime} = \phi_p(x_u^{t_i}, \mathcal{R}(x_u^{t_i}, \mathcal{H}_u^{t_i}, k), s_u), \tag{4}$$

which includes the user's profile s_u on the basis of retrieval augmentation. This profile $s_u = LLM(\mathcal{H}_u)$ is a summary of the user's preferences and behavior patterns, generated by an LLM that can follow human instruction.

By integrating the user knowledge in both private PEFT module and non-parametric retrieval and profile methods, OPPU is better equipped to understand a user's behavior patterns. This integrated approach enhances the model's ability to generalize these patterns to new and emerging user behaviors.

3.3.3 Adapting to Behavior History Beyond Task Conformity. It is worth mentioning that the user behavior history does not always align neatly with the format of a user's query. For instance, in personalized tweet paraphrasing tasks, where to input a sequence of text x_u, the desired output is the corresponding paraphrased text y_u concerning user history $\mathcal{H}_u$. However, $\mathcal{H}_u$ often comprises only the user's historical tweets $\{y_u^{t_i}\} \in \mathcal{H}_u$, not the $(x_u^{t_i}, y_u^{t_i})$ pairs. In scenarios where user history does not directly provide preferences in the specific task format, we find tuning a personal PEFT could also provide LLMs with more context and benefit the personalization task performance. Specifically, we update the personal PEFT parameters using unsupervised pretraining loss to predict next tokens over the user history sequences $h_u^{t_i} \in \mathcal{H}_u$. This approach infuses personal historical knowledge into the LLM via the integrated PEFT parameters.

Overall, we envision the proposed OPPU as a versatile LLM personalization framework, where each user possesses their own PEFT parameters that contain their personal behavior history and preference. By plugging their personal PEFT parameters into the base LLMs, users could get their personalized LLMs, while achieving a better understanding of users' personality from the parametric dimension.

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 Datasets. We adopt the Large Language Model Personalization (LaMP) benchmark [44] for our experiments, which consists of seven public language model personalization tasks, including four classification tasks (personalized citation identification, personalized movie tagging, personalized producing rating, personalized news categorization) and three generation tasks (personalized news

Table 1: Dataset statistics: We report average sequence length in terms of number of tokens. #Q is the number of queries, L_{in} and L_{out} are the average length of input and output sequence respectively, and #History is the number of adopted items. To save space, task names can be found in Table 2.

Task in LaMP	Train			Test			
	#Q	L_{in}	L_{out}	#Q	#History	L_{in}	L_{out}
1	7,919	51.3	1.0	123	317.5	52.0	1.0
2M	3,181	92.1	1.4	3,302	55.6	92.6	2.0
2N	3,662	68.2	1.3	6,033	219.9	63.5	1.1
3	22,388	128.7	1.0	112	959.8	211.9	1.0
4	7,275	33.9	9.2	6,275	270.1	25.2	11.1
5	16,075	162.1	9.7	107	442.9	171.6	10.3
7	14,826	29.7	18.3	109	121.2	29.4	18.0

headline generation, personalized scholarly title generation, personalized tweet paraphrasing).[2] We mainly focus on the most active uses and select 100 users from the time-based dataset version that have the longest history log as the test set, while all other users are used for base LLM training. Statistics of datasets are in Table 1.

4.1.2 Baselines. We compare our proposed OPPU with the non-personalized baseline and the retrieval-augmented and profile-augmented LLM personalization methods.

- **Non-Personalized Baseline**: We present two approaches under the non-personalized setting: non-retrieval and random history. *Non-retrieval method* refers to only feeding the user's query without revealing the user's behavior history to the LLMs. *Random history* baseline means augmenting the user's query with random history behavior from all user history corpus.
- **Retrieval-Augmented Personalization (RAG)**: We follow the retrieval-augmented personalization method presented in LaMP [44], where the user's query is augmented with top k retrieved items from the corresponding user's history corpus. We take k=1, 2, 4 in this work.
- **Profile-Augmented Personalization (PAG)**: This method is taken from Richardson et al. [43], in which the user's input sequence would concatenate the user's profile summarizing the user's preference and behavior patterns. In our experiments, we generate user profiles using the `vicuna-7B` [6] model. Moreover, the profile-augmented method could be combined with the retrieval augmentation. In this case, we take the number of retrieval items k=1 following the setting of Richardson et al. [43].

For all baselines, we choose one of the most widely adopted open-source large language model `Llama-2-7B` [49] as our base LLM and take BM25 [50] for all retrieval operations to ensure efficient and fair comparison.

4.1.3 Evaluation Metrics. Following LaMP [44], we use accuracy and F1-score for classification tasks (LaMP-1: personalized citation identification, LaMP-2N: personalized news categorization, and LaMP-2M: personalized movie tagging), MAE and RMSE for LaMP-3: personalized product rating, and adopt ROUGE-1 and ROUGE-L [28] for text generation tasks (LaMP-4: personalized news headline generation, LaMP-5: personalized scholarly title generation, LaMP-7: personalized tweet paraphrasing). Note that all metrics are the higher the better, except for RMSE and MAE used for the LaMP-3: personalized product rating task.

[2]We exclude the LaMP-6: personalized Email subject generation task since it involves private data that we don't have access to.

4.2 Main Results

Table 2 shows the performance on the test set of all seven public tasks in the LaMP benchmark. From the experimental results, we observe that

- **Impact of OPPU.** Models equipped with OPPU outperform all corresponding personalization baseline methods across all seven tasks. Notably, in personalized classification tasks, OPPU achieves an average relative improvement of 17.38% and 8.89% on MAE and RMSE metrics on personalized product rating prediction task, as well as 11.87% accuracy and 7.56% F1-score performance gain on personalized movie tagging task. For personalized text generation tasks, we observe a 3.42% and 3.87% enhancement in ROUGE-1 and ROUGE-L scores, respectively, for personalized scholarly title generation.
- **Integrating non-parametric & parametric knowledge.** Combining OPPU's parametric knowledge stored in PEFT parameters and the non-parametric in retrieved items and user profiles, results in notable performance gains. For instance, averaging across all seven tasks, combining retrieval in OPPU will bring 1.93% and 2.48% relative improvement compared with the non-retrieval and non-OPPU yet retrieval version model, respectively. Moreover, integrating OPPU with user profiles would also bring 4.56% and 7.18% performance gain against non-profile and non-OPPU versions, respectively. Overall, integrating non-parametric knowledge from retrieval and user profile and parametric knowledge from personalized PEFT parameter in OPPU generally presents the best performance.
- **Impact of task & history format difference.** In tasks like personalized citation identification, there's a notable discrepancy between the format of user history and the task itself. Here, the user history comprises the user's publication history, while the task involves a binary classification to identify the correct citation paper. This disparity is similarly observed in the personalized tweet paraphrasing task. In such cases, our proposed OPPU can robustly enhance performance. Specifically, in the personalized citation identification task, OPPU contributes to a notable increase of 3.48% in accuracy and 3.52% in the F1-score. This improvement is attributed to the provision of personalized context knowledge in a parametric manner via private PEFT.
- **Number of retrieved items.** Our experimental results generally indicate that an increase in the number of retrieved items correlates with improved performance. However, we also observe that some data points don't fit this trend, and we hypothesize that this inconsistency may arise from the retrieved items introducing noise and irrelevant behavior patterns, potentially complicating the model's process of understanding user preferences.

4.3 Performance under User Behavior Shift

Recent studies have shown that retrieval-augmented generation methods tend to underperform when the retrieved corpus does not

Table 2: Main experiment results on the LaMP benchmark. R-1 and R-L denote ROUGE-1 and ROUGE-L, respectively. k refers to the number of retrieved items, with $k = 0$ indicating no retrieval. ↑ indicates that higher values are better, and ↓ implies lower values are preferable. For each task, the best score is in bold and the second best is underlined.

Task	Metric	Non-Personalized		RAG			PAG		RAG+OPPU (Ours)				PAG+OPPU (Ours)	
		k=0	Random	k=1	k=2	k=4	k=0	k=1	k=0	k=1	k=2	k=4	k=0	k=1
LaMP-1: Personalized Citation Identification	Accy ↑	0.659	0.650	0.659	0.691	0.691	0.756	0.755	0.683	0.675	0.707	0.723	0.772	**0.797**
	F1 ↑	0.657	0.647	0.657	0.689	0.690	0.755	0.755	0.682	0.674	0.705	0.723	0.772	**0.794**
LaMP-2N: Personalized News Categorization	Acc ↑	0.787	0.785	0.820	0.832	0.832	0.817	0.817	0.810	0.823	0.834	**0.838**	0.827	0.831
	F1 ↑	0.538	0.527	0.598	0.632	0.647	0.623	0.621	0.589	0.615	0.635	**0.661**	0.648	0.638
LaMP-2M: Personalized Movie Tagging	Acc ↑	0.478	0.499	0.587	0.598	0.622	0.534	0.587	0.600	0.626	0.634	0.645	0.636	**0.648**
	F1 ↑	0.425	0.441	0.512	0.514	0.542	0.476	0.506	0.493	0.531	0.535	**0.553**	0.536	0.540
LaMP-3: Personalized Product Rating	MAE ↓	0.223	0.259	0.214	0.214	0.232	0.321	0.223	0.179	0.196	0.214	0.223	0.205	**0.143**
	RMSE ↓	0.491	0.590	0.535	0.463	0.535	0.582	0.473	0.443	0.518	0.463	0.526	0.473	**0.378**
LaMP-4: Personalized News Headline Gen.	R-1 ↑	0.186	0.187	0.191	0.196	0.198	0.187	0.193	0.904	0.194	0.196	**0.199**	0.189	0.194
	R-L ↑	0.167	0.168	0.172	0.176	0.178	0.168	0.173	0.171	0.175	0.177	**0.180**	0.170	0.175
LaMP-5: Personalized Scholarly Title Gen.	R-1 ↑	0.476	0.478	0.505	0.510	0.499	0.486	0.516	0.519	0.522	0.511	**0.526**	0.490	0.525
	R-L ↑	0.415	0.418	0.445	0.444	0.434	0.429	0.440	0.442	0.457	0.440	0.467	0.428	**0.473**
LaMP-7: Personalized Tweet Paraphrasing	R-1 ↑	0.527	0.524	0.568	0.577	0.562	0.542	0.568	0.539	0.579	0.575	**0.581**	0.542	0.577
	R-L ↑	0.474	0.474	0.521	0.527	0.514	0.501	0.518	0.483	**0.533**	0.531	0.528	0.492	**0.533**

Table 3: Performance under user behavior shift, where we remove the user behavior history highly similar to the query at hand. k denotes the number of retrieved history items, and $k = 0$ means non-retrieval. Armed with irrelevant user history, the retrieval-only method falls short and performs close to the non-personalized baseline, while OPPU shows stronger generalizability in the user behavior shift scenario.

LaMP Task	History Type	Non-Personalized		Retrieval k=1		OPPU k=0		OPPU k=1	
1		Acc	F1	Acc	F1	Acc	F1	Acc	F1
	full	.659	.657	.659	.657	.683	.682	.675	.674
	irrelevant			.626	.626	.683	.683	.699	.697
3		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
	full	.223	.491	.214	.535	.179	.443	.196	.518
	irrelevant			.268	.583	.196	.463	.241	.559
5		R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L
	full	.476	.415	.505	.445	.519	.442	.522	.457
	irrelevant			.475	.417	.493	.437	.490	.417
7		R-1	R-L	R-1	R-L	R-1	R-L	R-1	R-L
	full	.527	.474	.571	.521	.539	.483	.579	.533
	irrelevant			.543	.495	.528	.482	.563	.523

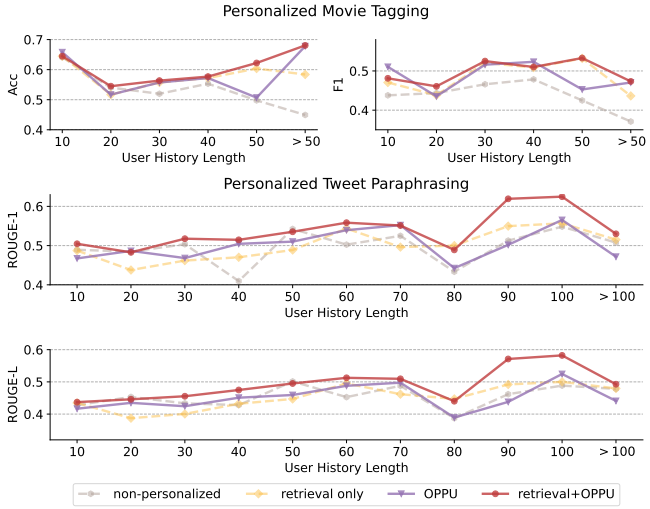

Figure 3: Model performance on personalized movie tagging and personalized tweet paraphrasing for users with different numbers of behavior history.

contain highly relevant documents [15, 39, 46]. This issue often emerges in personalization contexts where the retrieval corpus is confined to a specific user's behavior history and the user does not have a history of behavior closely matching their current queries. In our experiment, we simulate such a scenario where there is little or no overlap between the user's behavior history corpus and their current query. Specifically, we employ the encoder-only language model DeBERTa-v3 [18] to extract features for the user's historical behaviors and the query, then compute the cosine similarity between the query and all historical items to assess their relevance. By ranking the historical behaviors, we select the top 100 items with the lowest relevance scores as irrelevant user history.

As is demonstrated in Table 3, limiting user history to less relevant results in a marked decline in the performance of retrieval-based methods, often close to the performance of non-personalized approaches. Our proposed OPPU shows stronger robustness and generalization to these less relevant history behaviors and would even outperform using all user history items for private PEFT training. Moreover, the integration of both parametric and non-parametric knowledge (OPPU, k=1) exhibits enhanced robustness in personalized text generation tasks. In contrast, models utilizing only parametric user knowledge (OPPU, k=0) perform better in personalized text classification tasks.

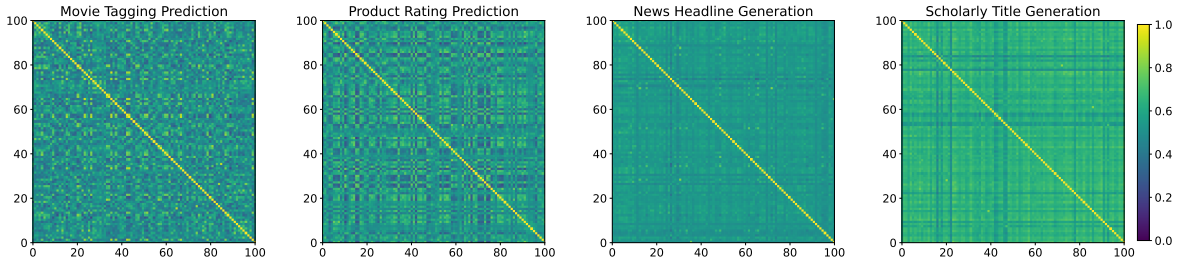

Figure 4: Cosine similarities between personal PEFT parameters under personalized text classification and generation tasks.

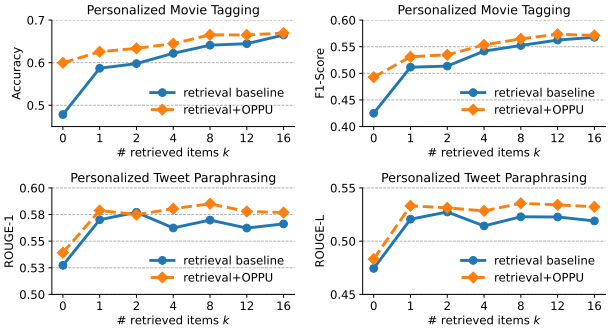

Figure 5: Performance of OPPU and retrieved-only baseline when the number of retrieved items k increases.

Table 4: Performance of OPPU with different ablated versions of user history configurations. k refers to the number of retrieved items, and $k = 0$ denotes non-retrieval. The best score is in bold and the second best is underlined.

Task in LaMP	History		Retrieval k=1		OPPU k=0		OPPU k=1	
	w/ desc.	w/ tag	Acc	F1	Acc	F1	Acc	F1
2M	✓		0.530	0.488	0.486	0.437	0.624	0.539
		✓	0.567	0.514	0.499	0.44	**0.634**	**0.548**
	✓	✓	0.587	0.512	0.600	0.493	0.626	0.531
	w/ abs.	w/ title	R-1	R-L	R-1	R-L	R-1	R-L
5	✓		0.493	0.422	0.497	0.434	0.495	0.449
		✓	0.475	0.425	0.489	0.430	0.492	0.429
	✓	✓	0.505	0.445	0.519	0.442	**0.522**	**0.457**

4.4 Modeling Users with Different Active Levels

In the main experiment, we concentrate on modeling highly active users. However, it is worth noting that a significant number of users exhibit lower levels of activity, resulting in a comparatively short behavior history. To investigate the impact of user activity levels quantified by the number of historical behavioral items on model performance, we randomly chose 20 users from each range of active levels. As illustrated in Figure 3, equipped with OPPU, LLMs generally outperform the baseline methods across different user activity levels. Specifically, 1) the longer the user history length, the superiority of retrieval+OPPU over the baseline is generally larger. 2) The inclusion of non-parametric user knowledge via retrieval results in performance improvements compared to methods without retrieval. 3) Integrating the parametric knowledge in OPPU and non-parametric knowledge in retrieval generally shows the strongest performance over users across different active levels.

4.5 Impact of Retrieved History Items

In this study, we alter the number of retrieved items of both retrieval-only baseline and retrieval+OPPU to gain a better understanding of the integration of non-parametric and parametric user knowledge. Figure 5 illustrates that as we increase the number of retrieved historical behavior items, both the retrieval-only baselines and the retrieval+OPPU approaches show improved performance. Interestingly, we observe that as the number of retrieved items k becomes larger, the performance difference between the retrieval-only baseline and retrieval+OPPU narrows. This trend could be attributed to the longer logs of user behavior history in non-parametric prompts, which reduce the gap between the comprehensive user behavior history encapsulated in personalized PEFT parameters and the non-parametric user knowledge included in the prompts.

4.6 Similarities Between Personalized PEFTs

To gain a better understanding of how users' behavior biases are encapsulated within their private PEFT parameters, we analyze the cosine similarities between these parameters across different users, as illustrated in Figure 4. Specifically, we select two representative tasks from text classification and generation categories respectively, then we compute the cosine similarities on the 100 users' PEFT parameters in the test set. As shown in Figure 4, we observe that the private PEFT similarities generally range from 0.4 to 0.7. Interestingly, the personalized scholarly title generation task exhibits the highest average similarities, likely due to task-specific characteristics that entail less personal bias. Besides the absolute values of these similarities, the relative differences among various users provide additional insights. In personalized text classification tasks, the similarities tend to exhibit more variance, suggesting that some users have higher similarities compared to others. In contrast, the similarities in personalized text generation tasks remain relatively uniform. This pattern leads us to speculate that personal preferences in text generation tasks are more challenging to categorize, making it harder to distinctly group users based on their preferences. On the other hand, preferences in text classification tasks appear to be more identifiable and classifiable.

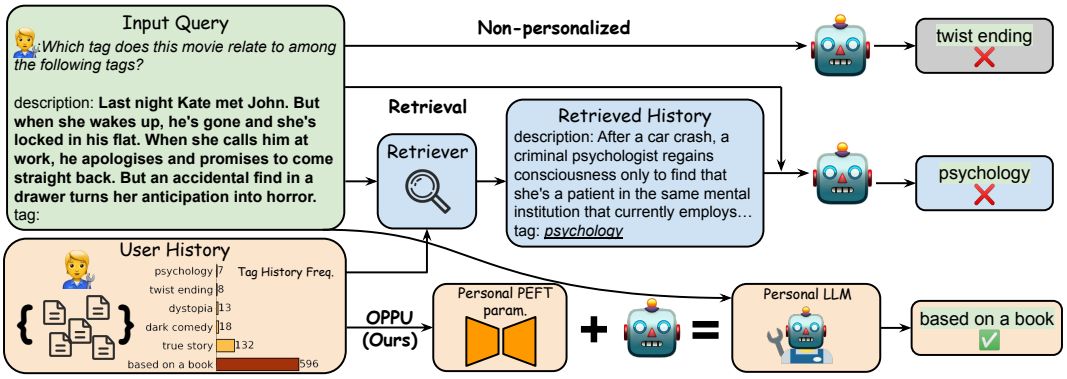

Figure 6: We present a case study on a specific query from a user and analyze the responses generated by a non-personalized model in the personalized movie tagging task, a retrieval-augmented personalization model, and our proposed OPPU. It is shown that the retrieval-augmented personalization method can be easily distracted by less relevant user behavior history. In contrast, our OPPU demonstrates a more effective and comprehensive ability to capture the user's behavior patterns.

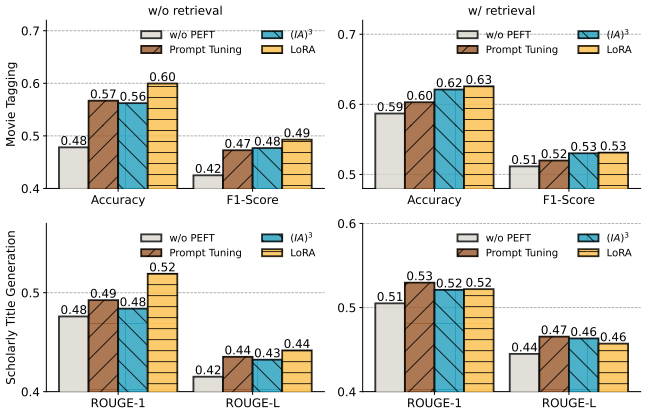

Figure 7: Performance of OPPU on personalized movie tagging and personalized scholarly title generation tasks when equipped with different PEFT methods. We find that a larger proportion of trainable parameters generally results in better personalization performance.

4.7 Robustness against Task Formats

Our main results demonstrate that even with a history corpus that does not strictly follow the task format, our plug-in OPPU would bring significant performance improvement. In this study, we ablate the history format to test the robustness against user history format on personalized movie tagging (LaMP-2M) and personalized scholarly title generation (LaMP-5), from text classification and generation category respectively. Specifically, in both tasks, each user history item consists of both input and output aligned with the user query x_u and output y_u at hand. We ablate the history behavior items from the input and output side respectively and compare them with the retrieval baseline to test OPPU's robustness against the history in mismatched format.

As is shown in Table 4, even with incomplete user behavior history that is not aligned with task format, OPPU still achieves relatively close performance with full history in text generation task. In the news categorization, LLM struggles to correctly classify with only parametric knowledge. However, integrating with retrieval augmentation, OPPU shows strong and robust performance, which can even outperform the full model tuned on the complete pairs of user history data. Overall, experimental results reveal that the integration of both non-parametric and parametric knowledge can make a robust model with different formats of user history.

4.8 On PEFT Method Choices

We envision the proposed OPPU as a versatile, PEFT-based LLM personalization framework compatible with various PEFT methods. This study demonstrates OPPU's performance across different PEFT approaches. Beyond the most commonly adopted LoRA, we explore prompt tuning and $(IA)^3$, which plug in external learnable parameters in the embedding space and scale the attention factor, respectively. As Figure 7 illustrates, the OPPU framework enhances performance with all three PEFT types, demonstrating the effectiveness and versatility of OPPU across various PEFT methods. Notably, LoRA typically delivers the highest performance, followed by $(IA)^3$, and then prompt tuning. This hierarchy aligns with the proportion of trainable parameters in each method: LoRA at 0.01%, $(IA)^3$ at 0.06%, and prompt tuning at 0.001%. These results suggest that a greater number of trainable parameters in a personalized PEFT method generally leads to improved personalization performance.

4.9 Case Study

To provide a qualitative understanding of OPPU, we conduct a case study focusing on an individual user in the personalized movie tagging task. As shown in Figure 6, the non-personalized method overlooks the user's behavior history and bases its prediction solely on the user's query input, leading to an evidently incorrect answer. The retrieval-based method incorporates user behavior history by retrieving the most relevant user history items, but it falls short of identifying the user's most pertinent historical behaviors that mirror the query, leading to incorrect LLM output. We argue that retrieval augmentation only involves user history data via several retrieval examples, limiting the comprehensive understanding of

user preferences. In contrast, our proposed OPPU employs a personalized PEFT module to effectively capture the user's behavior preferences and patterns across the entire user history corpus. In this case, based on the user's most frequently tags movies "based on a book" in behavior history, OPPU successfully identifies this pattern and consequently provides the correct response.

5 CONCLUSION

LLMs personalization has emerged as a rapidly evolving research area, aiming to tailor LLMs' emergent abilities to users' unique needs. In this work, we introduce OPPU, which utilizes personalized PEFT parameters as a proxy for LLM personalization, demonstrating the advantage of model ownership and enhanced generalization under user behavior shift. By tuning these parameters with a user's behavioral history, OPPU encapsulates the user's behavioral history and patterns in private PEFT parameters. Furthermore, OPPU can be integrated with non-parametric user knowledge via history retrieval and user profile augmentation, exhibiting state-of-the-art performance across all seven public tasks in the LaMP benchmark. Additional experiments highlight OPPU's versatility, robustness, and superiority in modeling users across different active levels. Our proposed OPPU framework paves the way for new opportunities in PEFT-based LLM personalization, enhancing the modularity of LLMs for more effective and democratizing personalization.

REFERENCES

[1] Hamdan A Alamri, Sunnie Watson, and William Watson. 2021. Learning technology models that support personalization within blended learning environments in higher education. *TechTrends* 65 (2021), 62–78.

[2] Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Sujay Kumar Jauhar, et al. 2023. Knowledge-Augmented Large Language Models for Personalized Contextual Query Suggestion. *arXiv preprint arXiv:2311.06318* (2023).

[3] Junyi Chen. 2023. A Survey on Large Language Models for Personalized and Explainable Recommendations. *arXiv preprint arXiv:2311.12338* (2023).

[4] Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, et al. 2023. When large language models meet personalization: Perspectives of challenges and opportunities. *arXiv preprint arXiv:2307.16376* (2023).

[5] Pengyu Cheng, Jiawen Xie, Ke Bai, Yong Dai, and Nan Du. 2023. Everyone Deserves A Reward: Learning Customized Human Preferences. arXiv:2309.03126 [cs.CL]

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* (2023).

[7] Konstantina Christakopoulou, Alberto Lalama, Cj Adams, Iris Qu, Yifat Amir, Samer Chucri, Pierce Vollucci, Fabio Soldo, Dina Bseiso, Sarah Scodel, et al. 2023. Large Language Models for User Interest Journeys. *arXiv preprint arXiv:2305.15498* (2023).

[8] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering ChatGPT's Capabilities in Recommender Systems. *arXiv preprint arXiv:2305.02182* (2023).

[9] Cosmina Andreea Dejescu, Lucia V Bel, Iulia Melega, Stefana Maria Cristina Muresan, and Liviu Ioan Oana. 2023. Approaches to Laparoscopic Training in Veterinary Medicine: A Review of Personalized Simulators. *Animals* 13, 24 (2023), 3781.

[10] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. *arXiv preprint arXiv:2208.07339* (2022).

[11] Zihao Fu, Haoran Yang, Anthony Man-Cho So, Wai Lam, Lidong Bing, and Nigel Collier. 2023. On the effectiveness of parameter-efficient fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 12799–12807.

[12] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524* (2023).

[13] Dmitri Goldenberg, Kostia Kofman, Javier Albert, Sarai Mizrachi, Adam Horowitz, and Irene Teinemaa. 2021. Personalization in practice: Methods and applications. In *Proceedings of the 14th ACM international conference on web search and data mining*. 1123–1126.

[14] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical user profiling for e-commerce recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 223–231.

[15] Aman Gupta, Anup Shirgaonkar, Angels de Luis Balaguer, Bruno Silva, Daniel Holstein, Dawei Li, Jennifer Marsman, Leonardo O Nunes, Mahsa Rouzbahman, Morris Sharp, et al. 2024. RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture. *arXiv preprint arXiv:2401.08406* (2024).

[16] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. 2020. Array programming with NumPy. *Nature* 585, 7825 (2020), 357–362.

[17] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2021. Towards a Unified View of Parameter-Efficient Transfer Learning. In *International Conference on Learning Representations*.

[18] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations*.

[19] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*. PMLR, 2790–2799.

[20] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

[21] 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* (2023).

[22] Kevin B Johnson, Wei-Qi Wei, Dilhan Weeraratne, Mark E Frisse, Karl Misulis, Kyu Rhee, Juan Zhao, and Jane L Snowdon. 2021. Precision medicine, AI, and the future of personalized health care. *Clinical and translational science* 14, 1 (2021), 86–93.

[23] Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction. arXiv:2305.06474 [cs.IR]

[24] Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 3045–3059.

[25] Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023. Teach LLMs to Personalize–An Approach inspired by Writing Education. *arXiv preprint arXiv:2308.07968* (2023).

[26] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. *arXiv preprint arXiv:2305.13731* (2023).

[27] Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 4582–4597. https://doi.org/10.18653/v1/2021.acl-long.353

[28] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[29] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems* 35 (2022), 1950–1965.

[30] Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149* (2023).

[31] Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2023. ONCE: Boosting Content-based Recommendation with Both Open- and Closed-source Large Language Models. arXiv:2305.06566 [cs.IR]

[32] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 61–68. https://doi.org/10.18653/v1/2022.acl-short.8

[33] Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2023. Are Emergent Abilities in Large Language Models just In-Context Learning? *arXiv preprint arXiv:2309.01809* (2023).

[34] Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. 2021. One chatbot per person: Creating personalized chatbots based on implicit user profiles. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 555–564.

[35] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 11048–11064.

[36] Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. 2023. PEARL: Personalizing Large Language Model Writing Assistants with Generation-Calibrated Retrievers. *arXiv preprint arXiv:2311.09180* (2023).

[37] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[39] Fabio Petroni, Patrick Lewis, Aleksandra Piktus, Tim Rocktäschel, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2020. How context affects language models' factual predictions. *arXiv preprint arXiv:2005.04611* (2020).

[40] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. 487–503.

[41] Muh Putra Pratama, Rigel Sampelolo, and Hans Lura. 2023. Revolutionizing education: harnessing the power of artificial intelligence for personalized learning. *Klasikal: Journal of Education, Language Teaching and Science* 5, 2 (2023), 350–357.

[42] Xueming Qian, He Feng, Guoshuai Zhao, and Tao Mei. 2013. Personalized recommendation combining user interest and social circle. *IEEE transactions on knowledge and data engineering* 26, 7 (2013), 1763–1777.

[43] 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* (2023).

[44] Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. LaMP: When Large Language Models Meet Personalization. *arXiv preprint arXiv:2304.11406* (2023).

[45] Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask Prompted Training Enables Zero-Shot Task Generalization. In *ICLR 2022-Tenth International Conference on Learning Representations*.

[46] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*. PMLR, 31210–31227.

[47] Biplav Srivastava, Francesca Rossi, Sheema Usmani, and Mariana Bernagozzi. 2020. Personalized chatbot trustworthiness ratings. *IEEE Transactions on Technology and Society* 1, 4 (2020), 184–192.

[48] Zhaoxuan Tan and Meng Jiang. 2023. User Modeling in the Era of Large Language Models: Current Research and Future Directions. *arXiv preprint arXiv:2312.11518* (2023).

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

[50] Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to BM25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*. 58–65.

[51] Danqing Wang, Kevin Yang, Hanlin Zhu, Xiaomeng Yang, Andrew Cohen, Lei Li, and Yuandong Tian. 2023. Learning personalized story evaluation. *arXiv preprint arXiv:2310.03304* (2023).

[52] Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 5085–5109.

[53] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).

[54] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research* (2022).

[55] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[56] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*. 38–45.

[57] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2023. Personalized news recommendation: Methods and challenges. *ACM Transactions on Information Systems* 41, 1 (2023), 1–50.

[58] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment. *arXiv preprint arXiv:2312.12148* (2023).

[59] Shuo Zhang and Krisztian Balog. 2020. Evaluating conversational recommender systems via user simulation. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining*. 1512–1520.

[60] Aakas Zhiyuli, Yanfang Chen, Xuan Zhang, and Xun Liang. 2023. BookGPT: A General Framework for Book Recommendation Empowered by Large Language Model. *arXiv preprint arXiv:2305.15673* (2023).

Table 5: Hyperparameter settings of OPPU accross various tasks on LaMP benchmark. We find our hyperparameter settings robust across all 7 tasks.

Tasks	rank	#epoch	lr	R2 reg.	batch size
LaMP-1: Personalized Citation Identification	8	3	$1e^{-5}$	$1e^{-2}$	16
LaMP-2: Personalized News Categorization	8	3	$1e^{-5}$	$1e^{-2}$	16
LaMP-2: Personalized Movie Tagging	8	3	$1e^{-5}$	$1e^{-2}$	4
LaMP-3: Personalized Product Rating	8	3	$1e^{-5}$	$1e^{-2}$	3
LaMP-4: Personalized News Headline Generation	8	2	$1e^{-5}$	$1e^{-1}$	8
LaMP-5: Personalized Scholarly Title Generation	8	2	$1e^{-5}$	$1e^{-1}$	4
LaMP-7: Personalized Tweet Paraphrasing	8	2	$1e^{-5}$	$1e^{-1}$	8

A LIMITATIONS

We identify two key limitations in OPPU. Firstly, limited by the dataset, we mainly focus on one specific task per user rather than examining user behaviors across multiple tasks and domains. For example, in the movie tagging task, users are solely engaged in that specific activity, without the inclusion of behaviors from other areas. Despite this, the OPPU framework is inherently adaptable to any text sequence generation task and is capable of conducting diverse user instructions across different tasks and domains. The exploration of LLM personalization across a broader range of tasks and domains remains an area for future investigation. Secondly, OPPU serves as a general framework that incorporates the entirety of a user's behavior history into their private PEFT module. However, user interests are dynamic and may display inconsistencies or conflicts over time. Future research directions include examining methodologies for selecting the most relevant or valuable items from a user's history and devising strategies to effectively manage any discrepancies or conflicts within this historical data.

B ETHICS STATEMENT

Privacy. Personalization in LLMs involves tailoring responses based on user-specific data, which may include sensitive or private information. The capacity of an LLM to adapt its outputs to individual users raises privacy concerns, as it might inadvertently reveal personal details. This underscores the importance of implementing robust privacy safeguards in LLM personalization, ensuring that personal data is handled respectfully and securely to prevent any unintended disclosures.

Data Bias. Personalizing LLMs heavily relies on the personal data fed into the system. If this personal data is biased or unrepresentative, the model's outputs could potentially perpetuate these biases, leading to unfair or prejudiced responses. It is crucial to monitor and mitigate such biases in the personal data and the personalized model we obtain to ensure that personalized LLMs are fair and harmless in their responses.

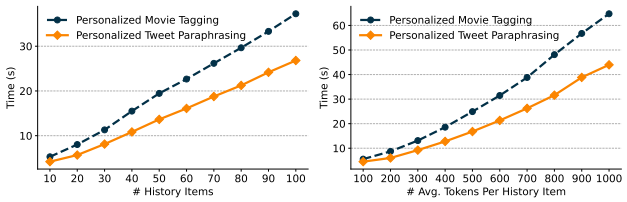

Figure 8: Efficiency analysis of OPPU, in which we alter the number of history items and average token per history item and record the training time.

Accessibility. By advancing the field of LLM personalization, we aim to enrich user interactions with AI systems. However, the complexity and resource-intensive nature of LLMs might pose accessibility challenges. Smaller entities or individual researchers with limited computational power and budgetary constraints might find it difficult to engage with advanced personalized LLMs, potentially widening the gap in AI research and application. It is essential to develop strategies that make personalized LLM technologies more accessible to a broader range of users and researchers, ensuring equitable progress in this domain.

C HYPERPARAMETERS

The hyperparameters of OPPU are presented in Table 5 to facilitate further research.

D EFFICIENCY ANALYSIS

Personalization is a technique that aims at universally benefiting everyone, where scalability and efficiency are crucial factors in large-scale deployment. In this experiment, we study the training efficiency of our proposed OPPU. We specifically examine two critical factors: the number of user history items and the average token numbers per history item across classification and generation tasks. Given that the training of each user's private PEFT can occur simultaneously or in a distributed manner, we choose not to consider the user count factor in this scenario, concentrating instead on the efficiency of training for an individual user. Initially, we set a consistent count of 100 whitespace-separated tokens for each history entry and vary the number of history items from 10 to 100. We then fix the history item count at 10 and adjust the token count from 10 to 100. The training time for each configuration, necessary for users to develop their personal PEFT modules. Presented in Figure 8, the results suggest that training time increases linearly with the number of user history items. Theoretically, training time grows quadratically with the increase in average tokens per history entry, yet our observations indicate a trend more akin to linear growth. It's noteworthy that the longer training durations for personalized movie tagging tasks, as opposed to personalized tweet paraphrasing, are attributed to different training epochs.

E SCIENTIFIC ARTIFACTS

OPPU is built with the help of many existing scientific artifacts, including PyTorch [38], Numpy [16], huggingface transformers [56],

and bitsandbytes [10]. We will make the OPPU implementation publicly available to facilitate further research.

F COMPUTATION RESOURCES DETAILS

All experiments are implemented on a server with 3 NVIDIA A6000 GPU and Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz with 20 CPU cores. Training 100 personal PEFT sequentially took around 12 minutes to 12 hours depending on the size of the behavior history corpus and the sequence length per history item.

G PEFT COSINE SIMILARITY DETAILS

Each user's private PEFT parameters contain multiple learnable tensors, we first flatten the tensors and calculate the cosine similarities between corresponding private PEFT parameters, then average cosine similarities for each pair of PEFT modules. A pseudo-code using PyTorch is as follows:

```
def cosine_similarity(PEFT_1, PEFT_2):
    similarity_sum = 0
    count = 0
    for key in PEFT_1:
        if key in PEFT_2:
            v1 = PEFT_1[key].flatten()
            v2 = PEFT_2[key].flatten()

            dot = torch.dot(v1, v2)
            norm_1 = torch.linalg.norm(v1)
            norm_2 = torch.linalg.norm(v2)

            similarity = dot / (norm_1 * norm_2)
            similarity_sum += similarity
            count += 1

    return similarity_sum / count
```

H TASK DETAILS

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

- **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.
- **Personalized News Categorization** is a 15-way text classification task to classify news articles written by a user u. Formally, given a news article x written by user u, the language model is required to predict its category from the set of categories based on the user's history data, which contains the user's past article and corresponding category.
- **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.
- **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.
- **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.
- **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.
- **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.

I PROMPT DETAILS

We present the prompt used in our experiments in this section, where the text in {BRACES} can be replaced with content specific to different users and queries.

I.1 Personalized Citation Identification

{USER PROFILE}
{RETRIEVED HISTORY}
Identify the most relevant reference for the listed publication by the researcher. Select the reference paper that is most closely related to the researcherś work. Please respond with only the number that corresponds to the reference.
paper title: {QUERY PAPER TITLE}
reference: [1] - {OPTION1} [2] - {OPTION2}
answer:

I.2 Personalized News Categorization

{USER PROFILE}
{RETRIEVED HISTORY}
Which category does this article relate to among the following categories? Just answer with the category name without further explanation. categories: [travel, education, parents, style & beauty, entertainment, food & drink, science & technology, business, sports, healthy living, women, politics, crime, culture & arts, religion]
article: {QUERY ARTICLE} category:

I.3 Personalized Movie Tagging

{USER PROFILE}
{RETRIEVED HISTORY}
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: {QUERY DESCRIPTION} tag:

I.4 Personalized Product Rating

{USER PROFILE}
{RETRIEVED HISTORY}
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: {QUERY REVIEW} score:

I.5 Personalized News Headline Generation

{USER PROFILE}
{RETRIEVED HISTORY}
Generate a headline for the following article.
article: {QUERY ARTICLE} headline:

I.6 Personalized Scholarly Title Generation

{USER PROFILE}
{RETRIEVED HISTORY}
Generate a title for the following abstract of a paper.
abstract: {QUERY ABSTRACT} title:

I.7 Personalized Tweet Paraphrasing

{USER PROFILE}
{RETRIEVED HISTORY}
Following the given pattern, paraphrase the following text into tweet without any explanation before or after it.
text: {QUERY TEXT} tweet: