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

Anonymous ACL submission

Abstract

Personalization in large language models (LLMs) is increasingly important, aiming to align the LLMs' interactions, content, and recommendations with individual user preferences. Recent advances have highlighted effective prompt design by enriching user queries with non-parametric knowledge through behavior history retrieval and textual profiles. However, these methods faced limitations due to a lack of model ownership, resulting in constrained customization and privacy issues, and often failed to capture complex, dynamic user behavior patterns. To address these shortcomings, we introduce **O**ne **P**EFT **P**er **U**ser (OPPU), employing personalized parameter-efficient fine-tuning (PEFT) modules to store user-specific behavior patterns and preferences. By plugging in personal PEFT parameters, users can own and use their LLMs individually. OPPU integrates parametric user knowledge in the personal PEFT parameters with non-parametric knowledge from retrieval and profiles, adapting 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 studies reveal OPPU's enhanced capabilities in handling user behavior shifts, modeling users at different activity levels, maintaining robustness across various user history formats, and displaying versatility with different PEFT methods.

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 (Tan and Jiang, 2023; Chen, 2023). By adapting to each user's preferences, personalization systems enhance user experience, increasingly getting vital in areas like content recommendation (Qian et al., 2013; Wu et al., 2023; Baek et al., 2023), user simulation (Dejescu et al., 2023), personalized chatbots (Srivastava et al., 2020; Ma et al., 2021), user profiling (Gu et al., 2020; Gao et al., 2023), healthcare (Goldenberg et al., 2021), and education (Pratama et al., 2023).

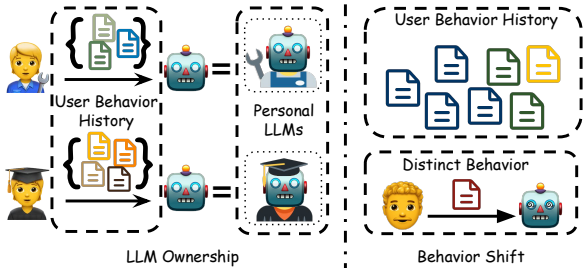

Figure 1: *LLM ownership* and *behavior shift* are two challenges that developing personalized LLMs has to face. Ownership emphasizes that the model needs to be owned by individual user to enhance customization and privacy. Behavior shift adaption refers to the LLMs' ability to effectively generalize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

Large language models (LLMs) display emergent abilities not seen in smaller models (Wei et al., 2022; Lu et al., 2023), as they have billions of parameters and are trained on vast corpora. 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 (Chen et al., 2023). 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 (Li et al., 2023).

Existing works on personalizing LLMs have predominantly concentrated on developing prompt templates, which fall into three categories: vanilla, retrieval-augmented, and profile-augmented personalized prompts. The vanilla personalized prompt approach leverages the in-context learning capability of LLMs, utilizing the user's entire or

randomly sampled history as contextual examples (Dai et al., 2023; Zhiyuli et al., 2023). Considering the growing length of user behavior history and the limited LLM context length, some studies applied retrieval methods to select the most relevant part of user behavior history to enhance LLM personalization (Mysore et al., 2023). Besides the retrieval, some techniques explicitly generate user preferences and profiles in natural language to augment LLMs' input (Richardson et al., 2023).

Despite much research progress has been made in LLM personalization, existing methods face ownership and behavior shift challenges (Fig. 1):

- **Ownership**: Existing methods are processed centralized, 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. (2023), LLMs can be easily distracted by irrelevant context information that retrieval can hardly avoid. In LLM personalization, where the retrieval corpus is confined to a specific user's behaviors, retrieval augmentation might underperform, especially when the user's past behaviors do 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), equipping each user 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 enhances 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. (2024), 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 outperforms all baselines on seven public tasks in the Language Model Personalization (LaMP) benchmark (Salemi et al., 2023). Additional studies emphasize the importance of integrating non-parametric user knowledge from retrieved history with parametric knowledge from personal PEFT parameters. In scenarios of user behavior shifts, where history is less relevant, OPPU significantly outperforms retrieval-based methods. Moreover, OPPU is resilient to 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 democratizing personalized LLMs.

2 Preliminaries

2.1 Research Problem Formulation

For personalizing LLMs at time t, the output r_u for user u is conditioned on both input q_u and the user's behavior history $\mathcal{H}_u$. Specifically, $\mathcal{H}_u = \{h_u\}$, includes all user behaviors h_u before time t. User behavior h_u may consist of (x_u, y_u) pairs, aligning with the task-specific query-answer format (q_u, r_u), or plain text sequences x_u providing context for behavior patterns. We aim to obtain personalized parameters Θ_u for each user u.

2.2 Base LLMs Task Adaption

Given that off-the-shelf LLMs are not inherently equipped for personalization tasks, we follow the methods of LaMP (Salemi et al., 2023) by fine-tuning LLMs for fair comparison. This section outlines the development of base LLMs with a set of held-out users to enhance their general capabilities for personalization tasks without involving target user preferences. Specifically, we provide three

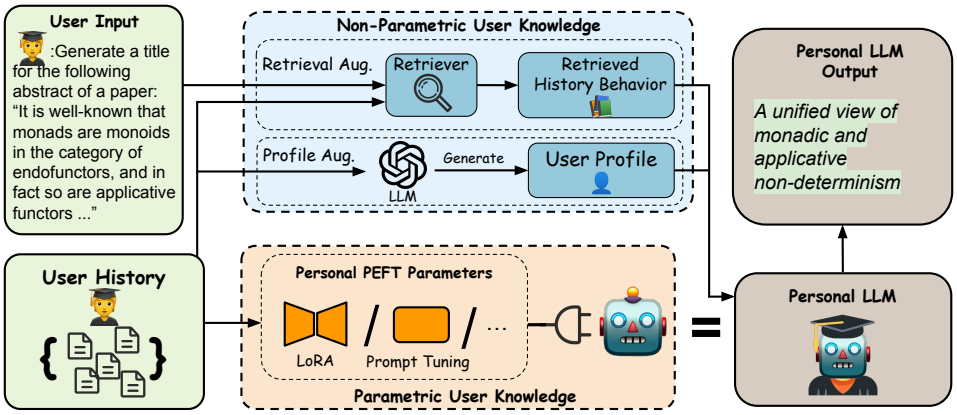

Figure 2: Overview of our proposed OPPU, where each user is equipped with a personal PEFT module and plug-in base LLMs to get their individual LLM. Beyond parametric personalization via PEFT, OPPU is also compatible with the non-parametric user knowledge via retrieval and profile augmentation.

alternatives: base LLM $\Theta^{(\mathrm{B})}$ that only involves task related data, retrieval-augmented base LLM $\Theta^{(\mathrm{R})}$ that augment input with top-k relevant user history, and profile-augmented base LLM $\Theta^{(\mathrm{P})}$ that involves textual user profiles as input. Note that introducing RAG and PAG means users would expose their historical data or profiles to a centralized LLM, potentially affect the model ownership. For users prioritizing privacy and ownership, OPPU without retrieval avoids revealing user data to service providers. Conversely, those seeking optimal performance and consent to reveal data to centralized LLMs should opt for RAG or PAG. The fine-tuning objectives of three base models are:

$$\begin{cases} \mathcal{L}_{\mathrm{B}} = \mathrm{CE}[\Theta^{(\mathrm{B})}(\phi_t(q_u)), r_u] \\ \mathcal{L}_{\mathrm{R}} = \mathrm{CE}[\Theta^{(\mathrm{R})}(\phi_r(q_u, \mathcal{D}_u)), r_u], \\ \mathcal{L}_{\mathrm{P}} = \mathrm{CE}[\Theta^{(\mathrm{P})}(\phi_p(q_u, \mathcal{D}_u, s_u)), r_u], \end{cases}$$

where CE denotes the cross entropy loss function, ϕ_t, ϕ_r, and ϕ_p denote prompt construction function for base, retrieval-augmented, and profile-augmented LLM. The retrieved user history $\mathcal{D}_u = \mathcal{R}(q_u, \mathcal{H}_u, k)$ denotes the top-k user history from retriever $\mathcal{R}$. $s_u = \mathrm{LLM}(\mathcal{H}_u)$ is a textual user profile generated by an instruction-tuned LLM, *e.g.*, `Vicuna` (Chiang et al., 2023), based on user history.

To make this process more computationally efficient, we adopt the low-rank adaptation (LoRA) (Hu et al., 2021) for base LLM task adaption that only updates about 0.5% external parameters compared to the total LLM parameter size. After training, LoRA parameters are merged into the base model, equipping LLMs with task capabilities.

3 One PEFT Per User (OPPU)

Once the base model for task adaption is obtained, users can only access the base model parameters and their personal behavior history data, controlling privacy risks. This section introduces personalized LLMs for target users through parametric PEFT and integrates non-parametric knowledge such as retrieval and profile augmentation. For each user, we plug a personal trainable PEFT module (LoRA by default) $\Delta\Theta_u^{(\mathrm{B})}$, $\Delta\Theta_u^{(\mathrm{R})}$, $\Delta\Theta_u^{(\mathrm{P})}$ to corresponding base LLM under three settings to obtain personalized LLM $\Theta_u^{(\mathrm{B})}$, $\Theta_u^{(\mathrm{R})}$, and $\Theta_u^{(\mathrm{P})}$, while base LLM parameters $\Theta^{(\mathrm{B})}$, $\Theta^{(\mathrm{R})}$, $\Theta^{(\mathrm{P})}$ are frozen.

$$\begin{cases} \Theta_u^{(\mathrm{B})} = \Theta^{(\mathrm{B})} \oplus \Delta\Theta_u^{(\mathrm{B})}, \\ \Theta_u^{(\mathrm{R})} = \Theta^{(\mathrm{R})} \oplus \Delta\Theta_u^{(B)}, \\ \Theta_u^{(\mathrm{P})} = \Theta^{(\mathrm{P})} \oplus \Delta\Theta_u^{(\mathrm{B})}. \end{cases}$$

We then use the user data $\mathcal{H}_u$ for LLM fine-tuning to learn the personalized PEFT parameters. The training objectives for user u under base, retrieval-augmented, and profile-augmented settings are:

$$\begin{cases} \mathcal{L}_u^{(\mathrm{B})} = \mathrm{CE}[\Theta_u^{(\mathrm{B})}(\phi_t(x_u)), y_u], \\ \mathcal{L}_u^{(\mathrm{R})} = \mathrm{CE}[\Theta_u^{(\mathrm{R})}(\phi_r(x_u, \mathcal{D}_u^{<t(x_u)})), y_u], \\ \mathcal{L}_u^{(\mathrm{P})} = \mathrm{CE}[\Theta_u^{(\mathrm{P})}(\phi_p(x_u, \mathcal{D}_u^{<t(x_u)}), s_u), y_u], \end{cases}$$

where $\mathcal{D}_u^{<t(x_u)} = \mathcal{R}(\phi_t(x_u), \mathcal{H}_u^{<t(x_u)}, k)$, $\mathcal{H}_u^{<t(x_u)}$ is restricted to user u's past behavior history that occurred before x_u.

User behavior history often does not align neatly with the query format. For example, in personalized tweet paraphrasing tasks, where the input is a text sequence q_u and the output is the paraphrased

tweet r_u, the history $\mathcal{H}_u$ only includes historical tweets. In scenarios where user history does not directly aligned with the specific task format, denoted as $\mathcal{H}_u = \{x_u\}$, we replace the user history output y_u in personal PEFT training objectives $\mathcal{L}_u^{(\mathrm{B})}$, $\mathcal{L}_u^{(\mathrm{R})}$, $\mathcal{L}_u^{(\mathrm{P})}$ with right-shifted history x'_u for unsupervised next token prediction.

By optimizing personal PEFT parameters with the objectives mentioned above, OPPU comprehensively capture the user behavior patterns in PEFT parameters $\Delta\Theta_u^{(\mathrm{B})}$, $\Delta\Theta_u^{(\mathrm{R})}$, $\Delta\Theta_u^{(\mathrm{P})}$, creating personalized LLMs owned by users. We envision the proposed OPPU as a versatile LLM personalization framework, where each user possesses their own PEFT parameters that contain personal behavior history and preferences. By plugging their personal PEFT parameters into the base LLMs, users can get their personalized LLMs, while achieving a better understanding and generalization of users' preferences from the parametric dimension.

4 Experimental Settings

Datasets We use data from the Large Language Model Personalization (LaMP) benchmark (Salemi et al., 2023), which includes seven public language model personalization tasks: four classification tasks and three generation tasks.[1] To promote LLM ownership, we emphasize the need for users to contribute extensive historical data for personalizing their model. Therefore, we focus on the most active users, selecting 100 users with the longest history logs from the time-based dataset version as the test set, while using all other users for base LLM training. Dataset statistics are in Appendix B.

Baselines We compare our proposed OPPU with the non-personalized baseline and the retrieval-augmented (RAG) and profile-augmented (PAG) LLM personalization methods. For all baselines and OPPU, we choose one of the most widely adopted open-source LLM `Llama-2-7B` (Touvron et al., 2023) as our base LLM and take BM25 (Trotman et al., 2014) for all retrieval operations to ensure efficient and fair comparison.[2]

Evaluation Metrics Following LaMP (Salemi et al., 2023), we use accuracy and F1-score for classification tasks (LaMP-1, LaMP-2N, and LaMP-2M), MAE and RMSE for LaMP-3, and adopt ROUGE-1 and ROUGE-L (Lin, 2004) for text generation tasks (LaMP-4, LaMP-5, LaMP-7). Note that all metrics are the higher the better, except for RMSE and MAE used for the LaMP-3.

[1] We exclude LaMP-6 as it involves private data that we cannot access.

[2] Baselines and hyperparameter details are presented in Appendix D and A to facilitate further research.

5 Results

Table 1 shows the performance on the test set for all seven public tasks in the LaMP benchmark, we have observations as follows.

OPPU brings universal improvement. Models equipped with OPPU outperform all baseline personalization methods across all seven tasks. Notably, in personalized classification tasks, OPPU achieves an average relative improvement of 17.38% in MAE and 8.89% in RMSE for personalized product rating prediction. Additionally, it shows an 11.87% improvement in accuracy and 7.56% in F1-score for personalized movie tagging. For personalized text generation tasks, OPPU enhances ROUGE-1 and ROUGE-L scores by 3.42% and 3.87%, respectively, in personalized scholarly title generation.

Integrating non-parametric and parametric knowledge performs the best. 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, combining non-parametric retrieval and profile knowledge with parametric PEFT knowledge in OPPU delivers the best performance.

Performance *w.r.t.* difference between task and history format. In tasks like personalized citation identification, there is a notable discrepancy between the user history format and the task itself. Here, the user history comprises the user's publication history, while the task involves binary classification to identify the correct citation paper. This disparity is also seen in the personalized tweet paraphrasing task. In these cases, OPPU significantly enhances performance. Specifically, for personalized citation identification, OPPU increases accu-

Table 1: 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.'*' indicates significant improvement against counterparts without OPPU.

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	Acc ↑	.659	.650	.659	.691	.691	.756	.755	.683*	.675*	.707*	.723*	.772*	**.797***
	F1 ↑	.657	.647	.657	.689	.690	.755	.755	.682*	.674*	.705*	.723*	.772*	**.794***
LaMP-2N: Personalized News Categorization	Acc ↑	.787	.785	.820	.832	.832	.817	.817	.810*	.823	.834	**.838***	.827*	.831*
	F1 ↑	.538	.527	.598	.632	.647	.623	.621	.589*	.615*	.635	**.661***	.648*	.638*
LaMP-2M: Personalized Movie Tagging	Acc ↑	.478	.499	.587	.598	.622	.534	.587	.600*	.626*	.634*	.645*	.636*	**.648***
	F1 ↑	.425	.441	.512	.514	.542	.476	.506	.493*	.531*	.535*	**.553***	.536*	.540*
LaMP-3: Personalized Product Rating	MAE ↓	.223	.259	.214	.214	.232	.321	.223	.179*	.196*	.214	.223*	.205*	**.143***
	RMSE ↓	.491	.590	.535	.463	.535	.582	.473	.443*	.518*	.463	.526*	.473*	**.378***
LaMP-4: Personalized News Headline Gen.	R-1 ↑	.186	.187	.191	.196	.198	.187	.193	.191*	.194*	.196	**.199**	.189*	.194
	R-L ↑	.167	.168	.172	.176	.178	.168	.173	.171*	.175	.177	**.180***	.170*	.175
LaMP-5: Personalized Scholarly Title Gen.	R-1 ↑	.476	.478	.505	.510	.499	.486	.516	.519*	.522*	.511	**.526***	.490*	.525*
	R-L ↑	.415	.418	.445	.444	.434	.429	.440	.442*	.457*	.440	.467*	.428*	**.473***
LaMP-7: Personalized Tweet Paraphrasing	R-1 ↑	.527	.524	.568	.577	.562	.542	.568	.539*	.579*	.575*	**.581***	.542	.577*
	R-L ↑	.474	.474	.521	.527	.514	.501	.518	.483*	**.533***	.531*	.528*	.492	**.533***

Table 2: 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

racy by 3.48% and F1-score by 3.52%, thanks to personalized context knowledge provided through personal PEFT.

The more retrieved items, the better performance. 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.

6 Analysis

Performance under User Behavior Shift Recent studies have shown that retrieval-augmented generation methods tend to underperform when the retrieved corpus does not contain highly relevant documents (Shi et al., 2023; Gupta et al., 2024). This problem is common in personalization contexts where the user's behavior history does not closely match their current queries. To simulate this scenario, we use `DeBERTa-v3` (He et al., 2022) to extract features from the user's historical behaviors and current query, computing cosine similarity to assess relevance. We then rank the historical behaviors and select the top 100 items with the lowest relevance scores as irrelevant user history.

Table 2 shows that limiting user history to less relevant items significantly reduces the performance of retrieval-based methods, often aligning with non-personalized approaches. In contrast, OPPU demonstrates stronger robustness and generalization to less relevant history, even outperforming models trained with all user history items. Additionally, the combination of parametric and non-parametric knowledge (OPPU, k=1) enhances robustness in personalized text generation tasks, while models using only parametric knowledge (OPPU, k=0) perform better in personalized text classification tasks.

Modeling Users with Different Active Levels In our main experiment, we focus on highly active users. However, many users exhibit lower activity levels, resulting in shorter behavior histories. To examine the impact of user activity levels

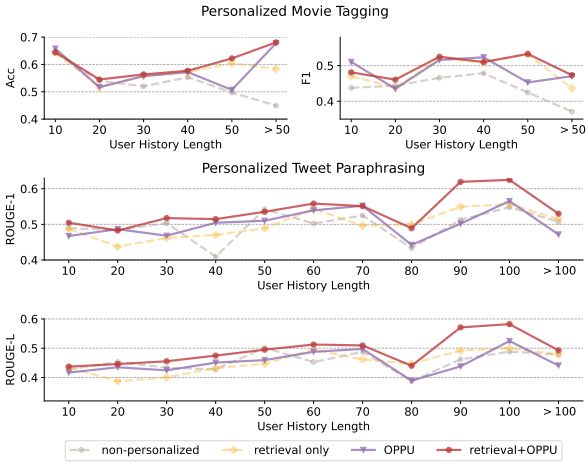

Figure 3: Model performance on personalized movie tagging and personalized tweet paraphrasing for users with different numbers of behavior history.

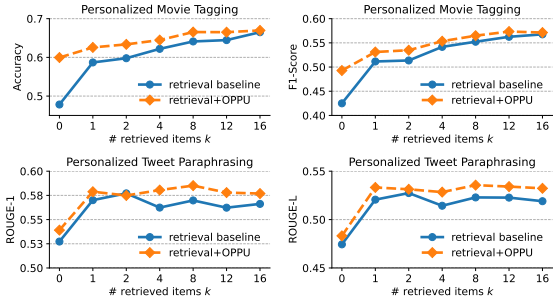

Figure 4: Performance of OPPU and retrieved-only baseline when the number of retrieved items k increases.

on model performance, we randomly selected 20 users from each activity range. Figure 3 shows that LLMs equipped with OPPU consistently outperform baseline methods across various activity levels. Key observations include: *1)* The longer the user history, the more pronounced the superiority of retrieval + OPPU over baselines. *2)* Including non-parametric user knowledge via retrieval improves performance compared to methods without retrieval. *3)* Integrating parametric knowledge in OPPU with non-parametric knowledge from retrieval yields the strongest performance across different user activity levels.

Performance *w.r.t.* Retrieved History Items *k* 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 4 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 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.

Table 3: 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	✓		.530	.488	.486	.437	.624	.539
		✓	.567	.514	.499	.440	**.634**	**.548**
	✓	✓	.587	.512	.600	.493	.626	.531
	w/ abs.	w/ title	R-1	R-L	R-1	R-L	R-1	R-L
5	✓		.493	.422	.497	.434	.495	.449
		✓	.475	.425	.489	.430	.492	.429
	✓	✓	.505	.445	.519	.442	**.522**	**.457**

Robustness against Task Formats Our main results demonstrate that OPPU significantly improves performance even when the user history corpus does not strictly follow the task format. We tested this robustness by ablating the history format in personalized movie tagging (LaMP-2M) and personalized scholarly title generation (LaMP-5) tasks, covering both text classification and generation categories. In both tasks, each user history item consists of input and output aligned with the user query x_u and output y_u. We ablated history behavior items from the input and output sides, comparing them with the retrieval baseline to test OPPU's robustness against mismatched formats.

Shown in Table 3, OPPU achieves performance close to that with full history in the text generation task, even with incomplete user behavior history. In news categorization, LLMs struggle with only parametric knowledge, but integrating retrieval augmentation, OPPU shows robust performance, outperforming models tuned on complete user history data. Overall, results reveal that combining non-parametric and parametric knowledge makes OPPU robust to different user history formats.

On PEFT Method Choices We propose OPPU as a versatile PEFT-based LLM personalization framework compatible with various PEFT meth-

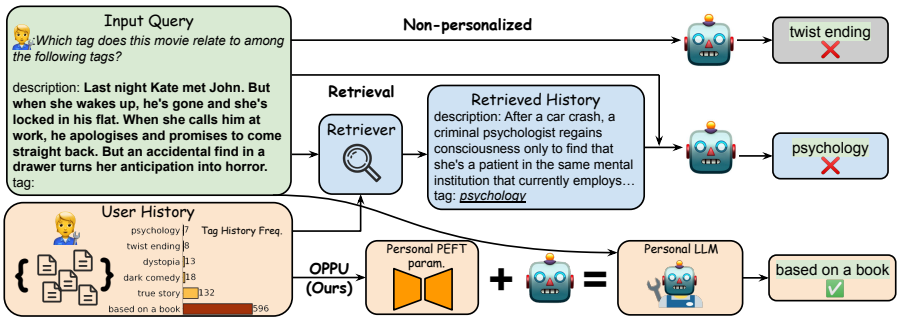

Figure 5: Case study in the personalized movie tagging task. It is shown that the retrieval-augmented personalization method can be easily distracted by less relevant user behavior history. In contrast, our OPPU demonstrates a more effective and comprehensive ability to capture the user's behavior patterns.

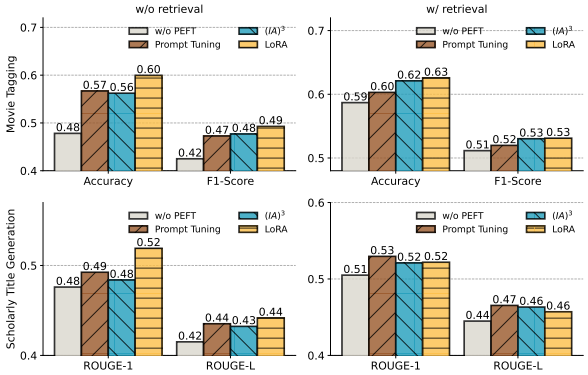

Figure 6: Performance of OPPU on personalized movie tagging and personalized scholarly title generation tasks when equipped with different PEFT methods. We find that a larger proportion of trainable parameters generally results in better personalization performance.

ods. This study evaluates OPPU's performance across different PEFT approaches, including LoRA, prompt tuning, and $(IA)^3$, which plug in external learnable parameters in the embedding space and scale the attention factor, respectively. As shown in Figure 6, OPPU enhances performance with all three PEFT types, demonstrating its effectiveness and versatility. 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.

Case Study To illustrate the effectiveness of OPPU, we conduct a case study on personalized movie tagging task for an individual user. Figure 5 shows that the non-personalized method, relying solely on query input, ignores user behavior history and yields incorrect answers. The retrieval-based method, though incorporating user history, fails to retrieve closely matched behaviors to the query, also resulting in errors. We argue that retrieval augmentation with a few user history examples cannot fully capture user preferences. In contrast, OPPU uses a personalized PEFT module to effectively understand the user's behavior patterns across the entire user history. In this case, OPPU successfully recognizes the user's frequent tagging of "based on a book" and provides the correct response.

Similarities Between Personalized PEFTs To understand how user behavior patterns are reflected in their private PEFT parameters, we analyze the cosine similarities between these parameters across different users, as shown in Figure 9. We select two representative tasks from text classification and generation categories and compute the cosine similarities for 100 users' PEFT parameters in the test set. The private PEFT similarities generally range from 0.4 to 0.7, with the highest average similarities observed in the scholarly title generation task, likely due to its task-specific nature. Relative differences among users offer additional insights: in personalized text classification tasks, similarities vary more, indicating that some users have higher similarities than others. Conversely, in personalized text generation tasks, the similarities are relatively uniform, suggesting that personal preferences in these tasks are harder to categorize.

7 Related Work

7.1 Personalization of LLMs

The thrust of existing LLM personalization research is centered on designing prompts that incorporate historical user-generated content and behavior. These approaches help LLMs understand

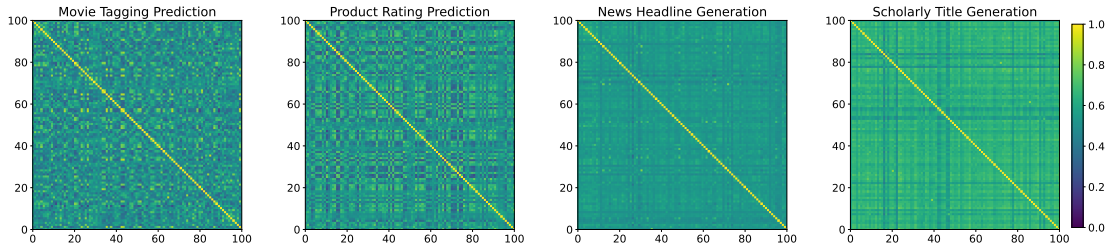

Figure 7: Similarities between personal PEFT parameters under personalized text classification and generation.

users' preferences, tailoring responses to individual needs (Tan and Jiang, 2023; Chen et al., 2023). The endeavors towards personalized LLMs mainly fall into three categories: vanilla, retrieval-augmented, 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 (Liu et al., 2023a; Wang et al., 2023). For instance, Dai et al. (2023) and Kang et al. (2023) encode the user's personal rating history as few-shot demonstration examples. Moreover, some research works (Christakopoulou et al., 2023; Zhiyuli et al., 2023) 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 (Salemi et al., 2023; Li et al., 2023). For instance, Pearl (Mysore et al., 2023) proposes a generation-calibrated retriever to select historic user-authored documents for prompt augmentation. Beyond simple retrieval, some researchers summarize user preferences and behavior patterns into natural language profiles for input query augmentation, termed *profile-augmented personalized prompts* (Liu et al., 2023b; Sun et al., 2024). Richardson et al. (2023) use the instruction-tuned LLMs to generate an abstract summary of user history data, augmenting retrieval-based personalization methods. There is also another line of work focusing on personalized alignment methods via parameter merging (Jang et al., 2023) and personalized reward model (Cheng et al., 2023).

7.2 Parameter-Efficient Fine-tuning (PEFT)

With the exponential growth in LLM parameters, fine-tuning all parameters is expensive (Liu et al., 2022b; Xu et al., 2023; Gupta et al., 2024). To address this, parameter-efficient fine-tuning (PEFT) methods update only a small number of extra parameters while keeping pretrained weights frozen (He et al., 2021; Fu et al., 2023). For example, adapter tuning (Houlsby et al., 2019) injects learnable parameters into each feedforward layer, updating only these during fine-tuning. Inspired by discrete textual prompts (Sanh et al., 2022; Wang et al., 2022), prefix tuning (Li and Liang, 2021) and prompt tuning (Lester et al., 2021) optimize prompts and prefixes for specific tasks. LoRA (Hu et al., 2021) adds low-rank matrices to approximate parameter updates, and $(\mathrm{IA})^3$ (Liu et al., 2022a) scales activation in the attention mechanism. These methods achieve performance comparable to full fine-tuning by updating less than 1% of the original parameters, are effective against catastrophic forgetting (Pfeiffer et al., 2021), and are robust to out-of-distribution samples (Li and Liang, 2021).

Previous works focused on prompt design, limited by model ownership and user behavior shifts. PEFT's small number of updated parameters and plug-and-play nature make it ideal for efficient LLM personalization and model ownership. OPPU introduces personalization at the parametric level via a personal PEFT module, pioneers storing user history within personal PEFT parameters, equipping each user with a unique, easily integrable PEFT module for model ownership.

8 Conclusion

We introduced OPPU, equipping each user with a personal PEFT module that facilitates model ownership and generalization under behavior shifts. By tuning these parameters with a user's history, OPPU captured personalized behavioral patterns. It integrated non-parametric user knowledge via retrieval and user profiles, showing superior performance across all seven LaMP benchmark tasks. Additional experiments demonstrated OPPU's versatility, robustness, and effectiveness for users with varying activity levels. Our framework paved the way for new opportunities in PEFT-based LLM personalization, enhancing LLM modularity for effective and democratized personalization.

9 Limitations

We identify three 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.

10 Ethical Considerations

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.

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.

References

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*.

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*.

Junyi Chen. 2023. A survey on large language models for personalized and explainable recommendations. *arXiv preprint arXiv:2311.12338*.

Pengyu Cheng, Jiawen Xie, Ke Bai, Yong Dai, and Nan Du. 2023. Everyone deserves a reward: Learning customized human preferences. *Preprint*, arXiv:2309.03126.

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)*.

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*.

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*.

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):3781.

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*.

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*, volume 37, pages 12799–12807.

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*.

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*, pages 1123–1126.

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*, pages 223–231.

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*.

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):357–362.

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*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

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*, pages 2790–2799. PMLR.

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*.

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

Wang-Cheng Kang, 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. *Preprint*, arXiv:2305.06474.

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*, pages 3045–3059.

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*.

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)*, pages 4582–4597, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

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

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

Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2023b. Once: Boosting content-based recommendation with both open- and closed-source large language models. *Preprint*, arXiv:2305.06566.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. 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)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

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*.

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*, pages 555–564.

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*.

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.

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*, pages 487–503.

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):350–357.

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):1763–1777.

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*.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. Lamp: When large language models meet personalization. *arXiv preprint arXiv:2304.11406*.

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*.

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*, pages 31210–31227. PMLR.

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

Chenkai Sun, Ke Yang, Revanth Gangi Reddy, Yi R Fung, Hou Pong Chan, ChengXiang Zhai, and Heng Ji. 2024. Persona-db: Efficient large language model personalization for response prediction with collaborative data refinement. *arXiv preprint arXiv:2402.11060*.

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*.

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*.

Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to bm25 and language models examined. In *Proceedings of the 19th Australasian Document Computing Symposium*, pages 58–65.

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*.

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*, pages 5085–5109.

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*.

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*, pages 38–45.

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

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*.

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*.

Table 4: 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

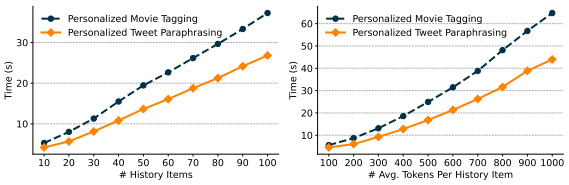

Figure 8: Efficiency analysis of OPPU, in which we alter the number of history items and average token per history item and record the training time.

A Hyperparameters

The hyperparameters of OPPU are presented in Table 4 to facilitate further research. For LoRA, we add trainable low-rank matrice in the W_q and W_v

B Dataset Statistics

The dataset statistics are presented in Table 5.

B.1 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 9. 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 9, 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

Table 5: 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 1.

Task in LaMP	Base LLM Training			Personal PEFT Training			
	#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

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.

C 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

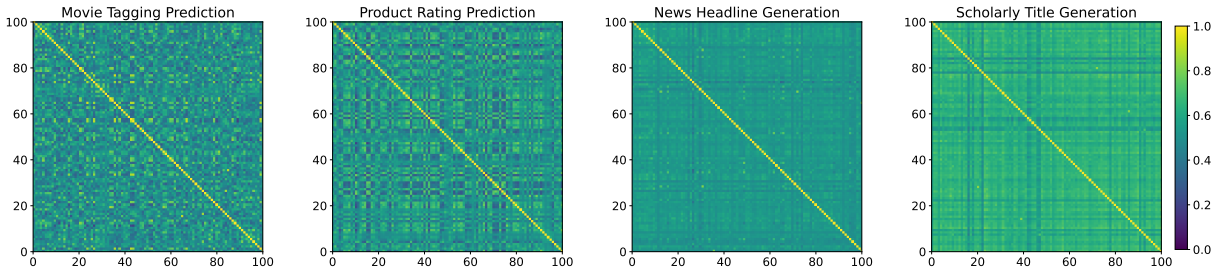

Figure 9: Cosine similarities between personal PEFT parameters under personalized text classification and generation tasks.

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.

D Baseline Details

The baseline details are presented as follows:

- **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 (Salemi et al., 2023), 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. (2023), 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` (Chiang et al., 2023) 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. (2023).

E Scientific Artifacts

OPPU is built with the help of many existing scientific artifacts, including PyTorch (Paszke et al., 2019), Numpy (Harris et al., 2020), huggingface transformers (Wolf et al., 2020), and bitsandbytes (Dettmers et al., 2022). 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 for Personalization Tasks

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:

J Prompt for User Profile Generation

For user profile generation, we follow the prompt template in Richardson et al. (2023).

J.1 Personalized Citation Identification

Write a summary, in English, of the research interests and topics of a researcher who has published the following papers. Only generate the summary, no other text. User History: {USER HISTORY} Answer:

J.2 Personalized News Categorization

Look at the following past articles this journalist has written and determine the most popular category they write in. Answer in the following form: most popular category: <category>. User History: {USER HISTORY} Answer:

J.3 Personalized Movie Tagging

Look at the following past movies this user has watched and determine the most popular tag they labeled. Answer in the following form: most popular tag: <tag>. User History: {USER HISTORY} Answer:

J.4 Personalized Product Rating

Based on this user's past reviews, what are the most common scores they give for positive and negative reviews? Answer in the following form: most common positive score: <most common positive score>, most common negative score: <most common negative score>. User History: Answer:Look at the following past movies this user has watched and determine the most popular tag they labeled. Answer in the following form: most popular tag: <tag>. User History: {USER HISTORY} Answer:

J.5 Personalized News Headline Generation

Given this author's previous articles, try to describe a template for their headlines. I want to be able to accurately predict the headline gives one of their articles. Be specific about their style and wording, don't tell me anything generic. User History: {USER HISTORY} Answer:

J.6 Personalized Scholarly Title Generation

Given this author's previous publications, try to describe a template for their titles. I want to be able to accurately predict the title of one of the papers from the abstract. Only generate the template description, nothing else. User History: {USER HISTORY} Answer:

J.7 Personalized Tweet Paraphrasing

Given this person's previous tweets, try to describe a template for their tweets. I want to take a generic sentence and rephrase it to sound like one of their tweets, with the same style/punctuation/capitalization/wording/tone/etc. as them. Only give me the template description, nothing else. User History: {USER HISTORY} Answer: