# LLMs + Persona-Plug = Personalized LLMs

**Jiongnan Liu**[1], **Yutao Zhu**[1], **Shuting Wang**[1], **Xiaochi Wei**[3]
**Erxue Min**[3], **Yu Lu**[3], **Shuaiqiang Wang**[3], **Dawei Yin**[3], **Zhicheng Dou**[1,2]
[1]Gaoling School of Artificial Intelligence, Renmin University of China
[2]Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE
[3]Baidu Inc.
liujn@ruc.edu.cn, yutaozhu94@gmail.com, dou@ruc.edu.cn

## Abstract

Personalization plays a critical role in numerous language tasks and applications, since users with the same requirements may prefer diverse outputs based on their interests. This has led to the development of various personalized approaches aimed at adapting large language models (LLMs) to generate customized outputs aligned with user preferences. Some of them involve fine-tuning a unique personalized LLM for each user, which is too expensive for widespread application. Alternative approaches introduce personalization information in a plug-and-play manner by retrieving the user's relevant historical texts as demonstrations. However, this retrieval-based strategy may break the continuity of the user history and fail to capture the user's overall styles and patterns, hence leading to sub-optimal performance. To address these challenges, we propose a novel personalized LLM model, PPlug. It constructs a user-specific embedding for each individual by modeling all her historical contexts through a lightweight plug-in user embedder module. By attaching this embedding to the task input, LLMs can better understand and capture user habits and preferences, thereby producing more personalized outputs without tuning their parameters. Extensive experiments on various tasks in the language model personalization (LaMP) benchmark demonstrate that the proposed model significantly outperforms existing personalized LLM approaches.

## 1 Introduction

Large language models (LLMs) have demonstrated extraordinary capabilities in natural language understanding, generation, and reasoning (Zhao et al., 2023; Brown et al., 2020; Zhu et al., 2023; Liu et al., 2023; Zhu et al., 2024), becoming increasingly essential tools for assisting with everyday tasks. However, the dominant usage pattern of LLMs follows a *one-size-fits-all* approach, where similar responses are provided to different users given the same input. While sampling-based decoding strategies can introduce some diversity, this approach fails to account for individual user preferences, reducing engagement in human-machine interactions. This problem is even severe in scenarios requiring tailored responses to align with subjective user profiles, such as drafting personalized speeches. Consequently, personalized LLMs have attracted significant interest in both industry and academic research (Salemi et al., 2024b; Zhuang et al., 2024; Kumar et al., 2024; Tan et al., 2024b; Chen et al., 2023; Zhang et al., 2024b).

A straightforward strategy for building personalized LLMs is to fine-tune a specific LLM on individual user data, allowing the model to learn the specific patterns and preferences of each user (Tan et al., 2024b,a; Zhang et al., 2024a). While effective, this method requires extensive computing resources for both training and inference, making it challenging for deployment in real applications. These approaches also suffer when users only have limited training data (Salemi and Zamani, 2024). Another way to achieve personalization is directly feeding all user histories into the LLM, and then generating tailored results according to the current user requests (Christakopoulou et al., 2023). This strategy avoids the need for additional model training but is often constrained by the maximum input length of the LLM, resulting in unsatisfying performance. To tackle this problem, recent studies have proposed leveraging retrieval models to select relevant behaviors from user histories based on the user input (Kumar et al., 2024; Salemi et al., 2024b,a; Richardson et al., 2023). These retrieved behaviors are then used as in-context demonstrations to guide the LLM in generating personalized outputs. While this strategy can introduce some degree of personalization, it is not always reliable. For producing personalized results, it is more important for LLMs to understand the users' overall styles than to refer to specific histories. Unfortunately, the



*Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9373–9385

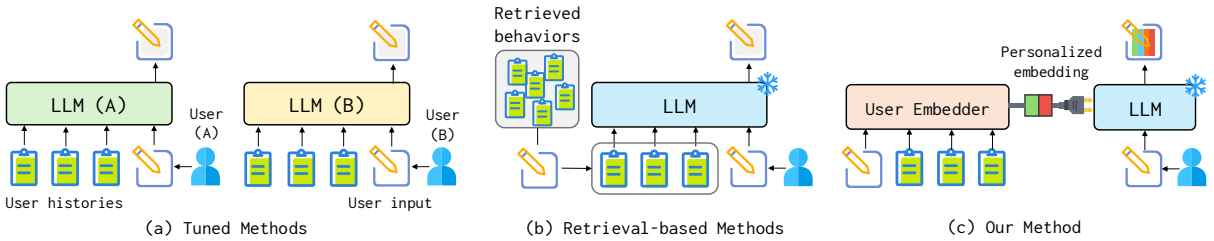


Figure 1: The comparison of our proposed personalized LLM and previous personalized LLM approaches

retrieval process typically focuses on relevance to the current input rather than identifying deeper user preferences embedded in all historical data. As a result, the selective utilization of user histories in retrieval-based personalized LLMs may disturb the model to capture user comprehensive manners and lead to sub-optimal performance.

Therefore, a better strategy for personalizing LLMs is to plug the user's holistic styles into the LLMs without modification of their structures and parameters. To achieve this, we propose a personal-plug (PPlug) model. It involves a lightweight plug-in user embedder module that embeds user historical patterns into a single user-specific embedding in input for LLMs to refer to. In the user embedder module, we first develop a user behavior encoder to represent a user's each historical behavior into a dense vector. Then, an input-aware personal aggregator synthesizes all these vectors into a user-specific personal embedding according to their relevance to current task inputs. This unique personal embedding from all histories is supposed to capture the user's general patterns in language tasks. After obtaining this personalized embedding, we directly attach it to the current input to guide fixed LLMs in tailoring their outputs according to user preferences. In this way, our PPlug model can better perform personalized tasks relying on the extracted user's comprehensive personal patterns in a plug-and-play strategy, shown in Figure 1. Furthermore, PPlug model can also be optimized in an end-to-end manner by all users' data, which is more efficient and effective compared with resorting to the limited data of each user to fine-tune personalized LLMs.

Experiments on six tasks in the public language model personalization (LaMP) benchmark (Salemi et al., 2024b) show that our proposed PPlug model achieves significant improvements over existing personalized LLM models from 1.4% to 35.8%. The main contribution of our work is three-fold:

(1) To better guide LLMs in personalized language generation, we propose a novel personalization framework that only attaches one user's personal embedding for LLMs to refer to.

(2) Compared with tuning a specific LLM for each user, the proposed PPlug model follows the plug-and-play paradigm and brings no additional parameters to LLMs.

(3) Compared with retrieval-based LLMs, PPlug can capture user holistic patterns and preferences, leading to better personalization performance.

## 2 Related Work

With the rising development of large language model techniques in many NLP applications and tasks (Zhao et al., 2023; Zhu et al., 2023), personalization in LLMs has attracted attention, and many approaches have been recently proposed (Salemi et al., 2024b,a; Zhuang et al., 2024; Kumar et al., 2024; Tan et al., 2024b,a; Richardson et al., 2023; Wozniak et al., 2024). These approaches facilitate LLMs with the personal content of users to generate customized outputs. Most of them can be categorized into the following two kinds:

**Fine-tuned Personalized LLMs.** The simple strategy for personalized language generation is to tune a unique LLM for each user based on their own data. However, fine-tuning all parameters in LLMs is too expensive; approaches in this category mainly devise the parameter-efficient fine-tuning (PEFT) technique to tune LLMs. Specifically, OPPU (Tan et al., 2024b) adopts the LoRA methods (Hu et al., 2022) to tune the Llama model (Touvron et al., 2023a) for each user. Tan et al. (2024a) further improve it by clustering users into different groups and tuning a model for each group. Zhuang et al. (2024) optimize a distinct language head for each user to tailor LLMs output. Zhang et al. (2024a) modify the model by searching for the best configuration of PEFT methods for each user.

**Retrieval-based Personalized LLMs.** The fine-tuned personalized LLMs need to train the LLMs for each user separately, which introduces huge computation costs and is difficult to devise in real

applications. Retrieval-based personalized LLMs leverage personalized information from another perspective without tuning LLMs. Inspired by the success of the retrieval-augmented generation (RAG) strategy in question-answering tasks (Zhu et al., 2025), these approaches retrieve relevant documents from user histories as in-context demonstrations for LLMs to produce personalized texts. Salemi et al. (2024b) explore these methods by applying different retrieval models. Salemi et al. (2024a) further improve it by optimizing the retrieval model through rewards calculated based on the LLM-generated outputs. They also explore the selection of different retrieval methods while facing different inputs. There also exist models that directly utilize all user histories to prompt LLMs or apply language models to generate text-based summaries as prompts (Richardson et al., 2023; Christakopoulou et al., 2023; Tang et al., 2024). However, these approaches cannot handle user histories that are extremely long due to the input length limits.

Some approaches (Doddapaneni et al., 2024; Ning et al., 2024; Hebert et al., 2024) in recommendation areas also aim at personalizing language models. However, they only adopt small language models such as T5-base model and the private PaLM 2-XXS (Anil et al., 2023) model and mainly focus on recommendations instead of language tasks requiring extensive world knowledge.

## 3 Methodology

Personalized large language models (LLMs) aim to satisfy users' specific demands and preferences by tailoring responses based on users' historical behaviors.[1] Following existing studies (Salemi et al., 2024a,b), the personalization task can be defined as: for a certain user $u$, generating a personalized response $y^u$ to a given user input $x^u$, utilizing the user's historical behaviors $H^u = [h_1^u, \cdots, h_n^u]$. Each user behavior $h_i^u$ corresponds to historical interactions similar in nature to the current input $x^u$. For example, if a user requests assistance with generating a title for a research paper, their historical behaviors may include titles and abstracts they have previously created for papers.

In this work, we introduce a method for LLM personalization called the persona-plug (PPlug) with a plug-in user embedder module. It encodes each historical behavior of a user into a dense vector and aggregates these embeddings into a single personal embedding considering the current input $x^u$. This personal embedding is then incorporated into the input to guide a fixed LLM in generating personalized responses. PPlug is a lightweight, plug-and-play approach, where each user has a distinct personal embedding calculated by the shared user embedder. The LLM uses these embeddings as input without requiring any additional modification to its own parameters. An overview of our proposed PPlug method is shown in Figure 2.

### 3.1 User Behavior Encoder

User behaviors often reflect how a user deals with a specific task, which contains valuable personal preferences and linguistic patterns. Therefore, effectively representing user behaviors is a critical step for personalization. Inspired by recent studies on sentence embedding and dense retrieval (Gao et al., 2021; Izacard et al., 2021), we employ a user behavior encoder to obtain user behavior representations. Specifically, for each user historical behavior $h_i^u$, we leverage an encoder-based model $\text{Enc}^{\text{his}}(\cdot)$ to encode $h_i^u$ as a vector $\mathbf{h}_i^u$:

$$\mathbf{h}_i^u = \text{Enc}^{\text{his}}(h_i^u). \quad (1)$$

Similarly, the representation of the current user input $x^u$ is computed as:

$$\mathbf{x}^u = \text{Enc}^{\text{input}}(x^u), \quad (2)$$

where the $\text{Enc}^{\text{input}}(\cdot)$ denotes the encoder specific to the user's current input, such as personalized product review.[2] All tasks are introduced in Section 4.1. To ensure efficient training of our proposed model, we freeze the parameters of $\text{Enc}^{\text{his}}$ and only fine-tune the input encoder $\text{Enc}^{\text{input}}$.

We choose small-sized encoder-based models for two primary reasons: (1) Bi-directional attention can effectively capture interactions across all tokens in user behaviors. Previous studies in information retrieval have demonstrated that encoder models can effectively condense document information into compact and dense representations (Morris et al., 2023). (2) A lightweight encoder improves the efficiency of both optimization

[1] Other personal information, such as user attributes, can also be used for personalization. However, due to the absence of such data in the current dataset, we follow existing studies (Salemi et al., 2024a,b) and focus solely on users' histories.

[2] In our implementation, we use the BGE-base-en-v1.5 model (Zhang et al., 2023; Xiao et al., 2023) as the encoder, https://huggingface.co/BAAI/bge-base-en-v1.5.

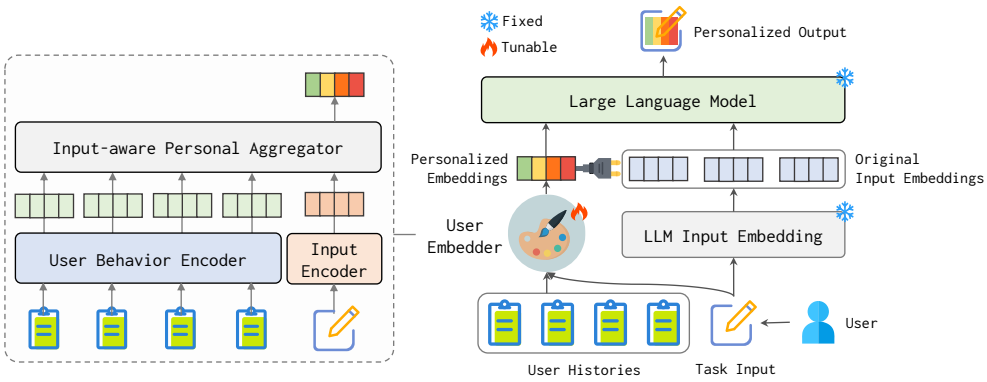


Figure 2: The overall framework of the proposed PPlug model.

and inference in our PPlug model. In our implementation, the encoder model introduces approximately 220M parameters, accounting for only 3.1% of the total parameters in a 7B LLM.

### 3.2 Input-aware Personal Aggregator

After obtaining representations of a user's historical behaviors and input, the next step is to aggregate them into a comprehensive personal embedding. A common approach is to treat each historical behavior as equally important and simply average them to represent the user profile. However, previous studies on personalized search and recommendation (Ge et al., 2018; Kang and McAuley, 2018; Wang et al., 2023) show that the importance of historical behaviors for the ongoing task should consider their relevance to the current input. For example, in generating academic titles based on abstracts, the model would benefit from prioritizing historical titles and abstracts that align more closely with the topics of the current abstract. Therefore, to improve task performance, historical behaviors that are more relevant to the current input should be assigned higher weights. To this end, we devise an attention mechanism that dynamically assigns weights to each historical behavior based on its relevance to the current user input. The personal embedding is calculated as follows:

$$w_i = \frac{\exp(\mathbf{x}^{u\top}\mathbf{h}_i^u)}{\sum_k \exp(\mathbf{x}^{u\top}\mathbf{h}_k^u)}, \quad (3)$$

$$\mathbf{P}^u = \sum_i w_i \cdot \text{Proj}(\mathbf{h}_i^u), \quad (4)$$

where $\text{Proj}(\cdot)$ projects the user embeddings from the encoder space to the LLM representation space by a 2-layer MLP, and $\mathbf{P}^u$ denotes the calculated personal embedding. In this manner, PPlug can mimic the retrieval manipulation in the retrieval-based strategy to make LLMs pay more attention to historically relevant behaviors. However, different from retrieval-based personalization methods that focus only on the most relevant histories, our approach integrates all user behaviors. This enables the personal embedding $\mathbf{P}^u$ to capture a more holistic representation of the user's preferences and patterns, enhancing the PPlug's ability to tailor personalized outputs.

### 3.3 PPlug for LLM Personalization

Once the personal embedding $\mathbf{P}^u$ is obtained, it is attached to the input to guide a fixed LLM in generating personalized outputs. Specifically, given the user's current input $x^u$ and previously generated personalized content $y^u_{<i}$, the next token prediction loss is defined as:

$$\mathbf{X}_i^u = [\mathbf{I}; \mathbf{P}^u; \text{Emb}_{\text{LLM}}(x^u); \text{Emb}_{\text{LLM}}(y^u_{<i})], \quad (5)$$

$$\mathcal{L} = -\sum_u \sum_i \log p_{\text{LLM}}(y_i^u | \mathbf{X}_i^u), \quad (6)$$

where $\text{Emb}_{\text{LLM}}(\cdot)$ denotes the LLM's embedding layer, $p_{\text{LLM}}$ is the predicted token distributions. Note that in addition to incorporating a personal embedding, we introduce a trainable instruction embedding $\mathbf{I}$ into the input. This is inspired by several recent studies on instruction tuning (Su et al., 2023; Zhang et al., 2023), which have shown that including an instruction embedding helps the LLM better understand and perform the task. Particularly, the LLM used in the PPlug model is fixed and only the instruction embedding $\mathbf{I}$, the input encoder $\text{Enc}^{\text{input}}(\cdot)$, and the projector $\text{Proj}(\cdot)$ are tuned, which is efficient for application.

### 3.4 Comparison with Previous Models

To highlight the advantages of the PPlug model, we provide a comparison with existing methods.

**PPlug vs. Fine-tuned Methods** Both the fine-tuned personalization methods (Zhang et al., 2024a; Tan et al., 2024b) and our PPlug model train the personalized framework to capture user general interests to guide personalized language generation, leading to promising performances. Besides, PPlug model has two additional advantages: (1) In training, unlike fine-tuned methods that require training a separate LLM for each user relying on their own limited data, PPlug trains a shared encoder to capture personalized user information using all data more efficiently and effectively. (2) In inference, PPlug operates in a plug-and-play manner, where a single LLM is used for all users, with user-specific personalized embeddings provided as input. This is highly advantageous for LLM service providers, as it enables the deployment of a single model to deliver effective personalization across users, streamlining infrastructure and maintenance.

**PPlug vs. Retrieval-based Methods** Retrieval-based methods achieve personalization by selecting relevant user historical behaviors. Similarly, PPlug incorporates an input-aware attention mechanism to evaluate the relevance of each behavior. However, unlike retrieval-based approaches that focus only on the most relevant behaviors, PPlug assigns dynamic weights to all behaviors. This allows it to capture a more comprehensive view of the user's general preferences across their entire history, leading to improved personalization outputs. Furthermore, while retrieval-based LLMs need to record the whole user histories for retrieval, PPlug only utilizes user embeddings, which can be produced by users themselves through the lightweight user embedder, which can better protect user privacy.

## 4 Experiments

### 4.1 Datasets and Metrics

**Datasets** We conduct experiments using the public Language Model Personalization (LaMP) benchmark (Salemi et al., 2024b), which consists of seven different personalization tasks. Consistent with previous studies (Tan et al., 2024b; Zhuang et al., 2024; Richardson et al., 2023; Tan et al., 2024a), we evaluate model performance on six tasks, excluding the Personalized Email Subject Generation task (LaMP-6), as it is not publicly available. Concretely, the six tasks include three personalized text classification tasks: (1) LaMP-1 Personalized Citation Identification; (2) LaMP-2 Personalized Movie Tagging; (3) LaMP-3 Personalized Product Rating, and three personalized text generation tasks: (4) LaMP-4 Personalized News Headline Generation; (5) LaMP-5 Personalized Scholarly Title Generation; and (6) LaMP-7 Personalized Tweet Paraphrasing. We use the time-based datasets provided by the LaMP benchmark, in which the data for each user is split into train, validation, and test sets in chronological order. Detailed information can be found in Appendix A.

**Evaluation Metrics** We use the default metrics in LaMP benchmark to evaluate the performance of each task: accuracy for LaMP-1, accuracy and F1-measure for LaMP-2, mean absolute error (MAE) and root mean squared error (RMSE) for LaMP-3, and ROUGE-1 and ROUGE-L (Lin, 2004) for LaMP-4, LaMP-5, and LaMP-7. For MAE and RMSE, lower values indicate better performance, as these metrics measure the discrepancy between predictions and ground-truth. For all other metrics, higher values correspond to better performance.

### 4.2 Implementaion Details

We use FlanT5-XXL (11B) (Chung et al., 2022) as the default LLM, which is consistent with previous studies (Salemi et al., 2024b,a). We use BGE-base-en-v1.5 (Xiao et al., 2023) as our default history and input encoder. Experimental results on different LLMs and encoder models are provided in Section 4.5. The maximum input lengths are set to 256 tokens for the LLMs and 512 tokens for the encoder. We employ beam search (Freitag and Al-Onaizan, 2017) with a beam size of 4 during generation. We train our PPlug model for 2 epochs across all tasks, except for LaMP-3, where 1 epoch is sufficient due to the larger dataset size. The batch size in all experiments is set to 64. The codes are available in https://github.com/rucliujn/PPlug.

### 4.3 Baselines

We compare our PPlug model with the following baselines covering four kinds of approaches:

(1) **Ad-hoc methods**: We use FlanT5-XXL to generate outputs solely based on the original task inputs. It serves as a non-personalized baseline.

(2) **Fine-tuned Personalization methods (FTP)**: Fine-tuning a specific LLM for each user requires extensive computational resources for training and inference (10,000 hours of A100 GPU com-

Table 1: Performance of all models on six LaMP tasks. "Valid" and "Test" refer to the results on the validation and test sets, respectively. The best results are in **bold**.

| Dataset | Metric | Ad-hoc | FTP | Naive RBP | | | Optimized RBP | | | | PPlug |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | FlanT5-XXL | | BM25 | Recency | Contriever | ROPG-RL | ROPG-KD | RSPG-Pre | RSPG-Post | |
| LaMP-1 | Valid Accuracy ↑ | 0.498 | - | 0.629 | 0.639 | 0.641 | **0.682** | 0.676 | 0.672 | 0.670 | 0.680 |
| | Test Accuracy ↑ | 0.502 | 0.506 | 0.626 | 0.622 | 0.636 | 0.655 | 0.668 | 0.663 | 0.672 | **0.700** |
| LaMP-2 | Valid Accuracy ↑ | 0.326 | - | 0.345 | 0.361 | 0.362 | 0.365 | 0.365 | 0.391 | 0.416 | **0.565** |
| | Valid F1 ↑ | 0.255 | - | 0.282 | 0.291 | 0.282 | 0.292 | 0.291 | 0.312 | 0.337 | **0.501** |
| | Test Accuracy ↑ | 0.359 | 0.360 | 0.387 | 0.377 | 0.396 | 0.391 | 0.396 | 0.405 | 0.430 | **0.559** |
| | Test F1 ↑ | 0.276 | 0.278 | 0.306 | 0.295 | 0.304 | 0.300 | 0.306 | 0.314 | 0.339 | **0.495** |
| LaMP-3 | Valid MAE ↓ | 0.335 | - | 0.293 | 0.305 | 0.297 | 0.273 | 0.274 | 0.266 | 0.246 | **0.231** |
| | Valid RMSE ↓ | 0.639 | - | 0.585 | 0.596 | 0.592 | 0.561 | 0.566 | 0.560 | 0.539 | **0.534** |
| | Test MAE ↓ | 0.308 | 0.301 | 0.298 | 0.296 | 0.299 | 0.286 | 0.290 | 0.282 | 0.264 | **0.242** |
| | Test RMSE ↓ | 0.611 | 0.600 | 0.611 | 0.605 | 0.616 | 0.591 | 0.604 | 0.585 | 0.568 | **0.557** |
| LaMP-4 | Valid ROUGE-1 ↑ | 0.173 | - | 0.192 | 0.194 | 0.190 | 0.190 | 0.193 | 0.195 | 0.207 | **0.216** |
| | Valid ROUGE-L ↑ | 0.157 | - | 0.175 | 0.177 | 0.174 | 0.174 | 0.176 | 0.179 | 0.188 | **0.197** |
| | Test ROUGE-1 ↑ | 0.176 | 0.178 | 0.186 | 0.189 | 0.183 | 0.191 | 0.187 | 0.190 | 0.203 | **0.211** |
| | Test ROUGE-L ↑ | 0.160 | 0.163 | 0.171 | 0.173 | 0.169 | 0.177 | 0.172 | 0.176 | 0.186 | **0.193** |
| LaMP-5 | Valid ROUGE-1 ↑ | 0.472 | - | 0.467 | 0.469 | 0.471 | 0.473 | 0.472 | 0.479 | 0.480 | **0.487** |
| | Valid ROUGE-L ↑ | 0.419 | - | 0.419 | 0.422 | 0.421 | 0.425 | 0.423 | 0.429 | 0.429 | **0.435** |
| | Test ROUGE-1 ↑ | 0.478 | 0.478 | 0.477 | 0.475 | 0.483 | 0.475 | 0.477 | 0.483 | 0.480 | **0.487** |
| | Test ROUGE-L ↑ | 0.428 | 0.429 | 0.427 | 0.426 | 0.433 | 0.427 | 0.428 | 0.431 | 0.429 | **0.439** |
| LaMP-7 | Valid ROUGE-1 ↑ | 0.454 | - | 0.451 | 0.452 | 0.440 | 0.458 | 0.451 | 0.460 | 0.468 | **0.536** |
| | Valid ROUGE-L ↑ | 0.401 | - | 0.401 | 0.402 | 0.391 | 0.407 | 0.402 | 0.409 | 0.416 | **0.484** |
| | Test ROUGE-1 ↑ | 0.449 | 0.449 | 0.446 | 0.444 | 0.440 | 0.448 | 0.441 | 0.450 | 0.461 | **0.537** |
| | Test ROUGE-L ↑ | 0.396 | 0.397 | 0.394 | 0.393 | 0.390 | 0.397 | 0.391 | 0.400 | 0.409 | **0.484** |

putation and 18 TB checkpoint storage are needed for LaMP benchmark as reported in Salemi and Zamani (2024), while PPlug model only requires about 100 hours and 150GB storage), making it difficult to reproduce. Therefore, we directly copy the results of PEFT personalization reported in (Salemi and Zamani, 2024) which only contains the test results without validation ones under the same evaluation setup as PPlug. They applied LoRA on FlanT5-XXL to tune a specific LLM for each user, similar to OPPU (Tan et al., 2024b).

(3) **Naive retrieval-based personalization methods (Naive RBP)**: We employ BM25 (Robertson and Zaragoza, 2009), Recency, and Contriever (Izacard et al., 2021) methods to retrieve the top-4 user historical behaviors as demonstrations for FlanT5-XXL to produce personalized outputs. These methods are not tuned for personalization tasks and thus are referred to as naive RBP.

(4) **Optimized Retrieval-based Personalization (Optimized RBP)**: ROPG-RL, ROPG-KD, RSPG-Pre, and RSPG-Post are four baseline methods designed by Salemi et al. (2024a). ROPG-RL and ROPG-KD optimize the Contriever-based retrieval model by reinforcement learning and knowledge distillation strategies according to the evaluation metrics. RSPG-Pre and RSPG-Post introduce a retrieval selection module that selects the optimal retrieval model from multiple candidates based on task inputs and model outputs, respectively.

### 4.4 Experimental Results

The results on the validation and test sets are shown in Table 1. Generally, PPlug achieves the best performance, demonstrating its superiority on personalization tasks. Furthermore, we observe that:

(1) Fine-tuned Personalization methods (FTP) only achieve subtle improvements over non-personalized methods (ad-hoc). The reason may be that fine-tuning personalized LLMs requires sufficient user behavior data, but most users in the LaMP benchmark only have limited histories (Salemi and Zamani, 2024). (2) Both retrieval-based methods (RBP) and PPlug can achieve better performance. This indicates that incorporating user historical behaviors is an effective way to capture user personal preferences. (3) Compared to naive RBP, optimized RBP can perform better. This is consistent with our speculation, as the retrievers in naive RBP are not optimized for personalized generation tasks, and tuning the retrievers with the feedback from LLMs' output is beneficial for personalization tasks. (4) Our PPlug outperforms all baselines in almost all tasks. Specifically, the relative improvements of PPlug over the best baseline (RSPG-Post) are from 1.4% to 35.8%. These improvements confirm that our idea of compris-

Table 2: Overall performance of models with different LLMs and encoders on the validation set. We use Acc to abbreviate Accuracy and R to abbreviate ROUGE respectively.

| LLM | Encoder | LaMP-1 | LaMP-2 | | LaMP-3 | | LaMP-4 | | LaMP-5 | | LaMP-7 | | # Best |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Acc ↑ | Acc ↑ | F1 ↑ | MAE ↓ | RMSE ↓ | R-1 ↑ | R-L ↑ | R-1 ↑ | R-L ↑ | R-1 ↑ | R-L ↑ | |
| FlanT5-XL | BGE | 0.636 | 0.463 | 0.375 | 0.242 | 0.537 | 0.193 | 0.174 | 0.478 | 0.424 | 0.509 | 0.456 | 0 |
| FlanT5-XXL | BGE | 0.680 | 0.565 | 0.501 | **0.231** | 0.534 | **0.216** | **0.197** | **0.487** | **0.436** | **0.536** | **0.484** | 7 |
| FlanT5-XXL | Contriver | **0.687** | 0.553 | 0.501 | 0.236 | **0.527** | **0.216** | **0.197** | 0.485 | **0.436** | 0.535 | 0.482 | 5 |
| Llama 2 7B | BGE | 0.663 | 0.585 | 0.540 | 0.259 | 0.581 | 0.212 | 0.194 | 0.467 | 0.418 | 0.503 | 0.450 | 0 |
| Llama 2 7B | Contriver | 0.611 | **0.589** | **0.547** | 0.261 | 0.582 | **0.216** | 0.196 | 0.466 | 0.417 | 0.504 | 0.450 | 3 |

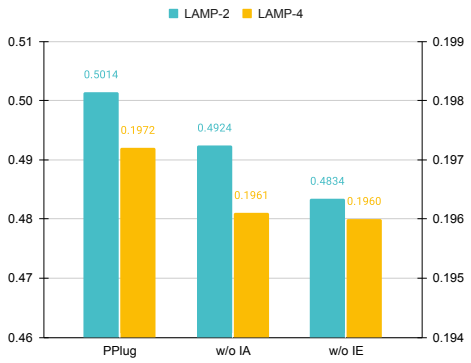


Figure 3: Overall performance of ablation models on the validation set.

ing user historical behaviors into a single personal representation and facilitating LLMs to perform personalized tasks is very effective. (5) Compared with FTP methods, the PPlug model can utilize data from all users instead of each user's own limited data, which is more effective. Compared with ROPG and RSPG that leverage reinforcement learning and knowledge distillation techniques to optimize models, our PPlug can be directly optimized in an end-to-end manner, which is much more efficient.

### 4.5 Further Analysis

We further conduct a series of experiments to analyze our PPlug method. Since the test data are held out by the benchmark organizers, we mainly report the results on the validation set in this section.

**LLM and Encoder Analysis** By default, we use FlanT5-XXL and BGE-base as the LLM and encoder. To investigate their impact on the final performance, we conduct more experiments by replacing them with other models. Specifically, we replace the LLM by FlanT5-XL (Chung et al., 2022) and Llama 2 7B Chat (Touvron et al., 2023b) and the encoder model by Contriever (Izacard et al., 2021) and test the performance of these variants. The results are shown in Table 2. We can find: (1) When using FlanT5-XXL as the backbone LLM, PPlug with both BGE-base and Contriever can achieve comparable performance, and both of them can outperform previous personalization methods significantly. This result clearly validates the robustness of our method. (2) When using the BGE-base encoder, PPlug's performance is positively correlated with the size of the LLM (FlanT5-XXL 11B > Llama 2 7B > FlanT5-XL 3B). This result is consistent with the scaling law (Kaplan et al., 2020), where larger models have stronger capabilities and perform better on NLP tasks.

**Ablation Study** In our proposed PPlug, we design an input-aware personal aggregator that dynamically constructs personal embeddings based on the current task input (Section 3.2). Additionally, we employ an instruction embedding to capture global patterns relevant to specific tasks (Section 3.3). To investigate the effect of these components, we perform an ablation study. Due to limited space, we report the results on two representative tasks: LaMP-2 and LaMP-4 tasks. The performance observed in these tasks is consistent with trends across other tasks. Complete results can be found in Table 3 in Appendix B.1.

(1) **Impact of input-aware attention**: We first remove the input-aware personal aggregator and summarize the personal embedding by averaging the representation of each historical behavior. As shown in Figure 3, this model performs worse than the full PPlug, indicating that the input-aware aggregator can better capture user patterns according to the current input. Nevertheless, it is worth noting that even without this component, PPlug still achieves strong results compared to baselines, suggesting that the user's overall behavior patterns are crucial for personalized language generation.

(2) **Impact of instruction embedding**: Next, we remove the instruction embedding [**I**] from the LLM input in Equation (5). Intriguingly, this vari-

Table 3: Performance of PPlug integrated with retrieval on the validation set.

| Dataset | Metric | PPlug | PPlug + Retrieval |
|---|---|---|---|
| LaMP-1 | Accuracy ↑ | 0.680 | **0.687** |
| LaMP-2 | Accuracy ↑ | **0.565** | 0.545 |
| | F1 ↑ | **0.501** | 0.485 |
| LaMP-3 | MAE ↓ | 0.231 | **0.215** |
| | RMSE ↓ | 0.534 | **0.506** |
| LaMP-4 | ROUGE-1 ↑ | 0.216 | **0.220** |
| | ROUGE-L ↑ | 0.197 | **0.203** |
| LaMP-5 | ROUGE-1 ↑ | 0.487 | **0.498** |
| | ROUGE-L ↑ | 0.436 | **0.448** |
| LaMP-7 | ROUGE-1 ↑ | 0.534 | **0.547** |
| | ROUGE-L ↑ | 0.484 | **0.495** |

Table 4: Performance of PPlug with history selection on the validation set.

| Dataset | Metric | PPlug | PPlug with selection |
|---|---|---|---|
| LaMP-1 | Accuracy ↑ | **0.680** | 0.675 |
| LaMP-2 | Accuracy ↑ | **0.565** | 0.492 |
| | F1 ↑ | **0.501** | 0.441 |
| LaMP-3 | MAE ↓ | **0.231** | 0.239 |
| | RMSE ↓ | **0.534** | 0.542 |
| LaMP-4 | ROUGE-1 ↑ | **0.216** | 0.205 |
| | ROUGE-L ↑ | **0.197** | 0.188 |
| LaMP-5 | ROUGE-1 ↑ | **0.487** | 0.485 |
| | ROUGE-L ↑ | **0.436** | **0.436** |
| LaMP-7 | ROUGE-1 ↑ | **0.534** | 0.530 |
| | ROUGE-L ↑ | **0.484** | 0.477 |

ant also outperforms baselines, indicating that the primary improvements of the PPlug stem from the personal embedding from user histories. However, the performance decline highlights that the instruction embedding helps the model disentangle global task-related knowledge from user-specific patterns, thereby enhancing personalization performance.

**Integration with Retrieval-based Strategy** In our experiments, we observe that the retrieval-based personalization methods can yield improvements over non-personalization methods. Therefore, we investigate whether integrating our PPlug method with retrieval-based strategies can further enhance performance. Specifically, we first use the BGE-base-en-v1.5 model (Zhang et al., 2023; Xiao et al., 2023) to retrieve the most relevant historical behavior from the user's history based on the current input. Then, the retrieved content $h_k^u$ is appended to the input as demonstrations for the LLM to reference for producing personalized outputs. In this manner, the inputs can be formatted as $\mathbf{X}_i^u = [\text{Emb}_{\text{LLM}}(h_k^u); \mathbf{I}; \mathbf{P}^u; \text{Emb}_{\text{LLM}}(x^u)]$. We train the model using the same training data and refer to this model as "PPlug + Retrieval". The results are shown in Table 3. Overall, the integration of retrieval-based strategies with the PPlug model leads to further performance gains over the original PPlug method. Indeed, PPlug provides a coarse-grained user style embedding, capturing general user habits and preferences. In contrast, retrieval-based methods offer fine-grained, task-specific historical contexts that help retrieve knowledge relevant to the current task. Therefore, combining these approaches allows for more effective personalized generation. This raises a new research question of how to optimize the use of coarse-grained user embeddings versus fine-grained retrieved references, suggesting a direction for future research.

**History Selection Study** As discussed in Section 1, retrieval-based personalization approaches only select historical content most relevant to the current task input to serve as demonstrations for the LLM, which may hinder the model from capturing the user's broader interests. To explore this, we modify our input-aware personal aggregator module to select only the top-4 history embeddings, $\mathbf{h}_i^u$, based on their associated weights, $w_i$, to construct the personal embedding $\mathbf{P}^u$. This setting is consistent with retrieval-based personalized LLMs, which rely on a small set of top historical behaviors. We refer to this variant as "PPlug with Selection". The results are presented in Table 4. We can observe that the performance of PPlug model decreases when using only the top-4 history embeddings to build the user's personal embedding. This suggests that the selective usage of histories can impair the model's ability to capture general user patterns, leading to sub-optimal outputs. In contrast, aggregating all histories, as in the original PPlug model, provides a more comprehensive representation of the user's preferences, resulting in improved performance. We further conduct experiments to analyze the impact of the number of selected histories, which can be found in Appendix B.2.

**History Length Impact Study** In this part, we investigate the impact of the amount of user historical data on model performance. Specifically, we group users in LaMP-3 Personalized Product Rating Task (a classification task) and LaMP-5 Personalized Scholarly Title Generation Task (a gener-

Table 5: Performance of PPlug with users with different history length.

| User Group | LaMP-3 RMSE ↓ | LaMP-5 ROUGE-1 ↑ |
|---|---|---|
| Short | 0.548 | 0.474 |
| Medium | 0.531 | 0.482 |
| Long | **0.522** | **0.499** |
| Overall | 0.534 | 0.487 |

ation task) into three groups (short/medium/long) according to their history length by 1:1:1 ratio. We show the performances on each group on the validation set in Table 5. We can observe that the performance of our PPlug model increases with the length of user histories because longer histories can provide more information about user interests. Besides, the performance of the model on users with different histories is generally robust, suggesting that PPlug is also effective even when users have limited histories.

## 5 Conclusion

In this work, we propose a persona-plug (PPlug) model for personalized language generation. In PPlug model, we devise a lightweight and plug-and-play user embedder module to encode a user's all historical behaviors to dense vectors and then aggregate them into one single user personal embedding in an input-aware manner. We believe this distinct personal embedding for each user can represent their general linguistic styles and habits in all histories and guide LLMs to personalize their outputs. Experimental results on the LaMP benchmark show that the proposed model can significantly outperform existing retrieval-based LLM models.

## Acknowledgement

Yutao Zhu and Zhicheng Dou are the corresponding authors. This work was supported by Beijing Municipal Science and Technology Project No. Z231100010323009, National Natural Science Foundation of China No. 62272467, Beijing Natural Science Foundation No. L233008, and the fund for building world-class universities (disciplines) of Renmin University of China. The work was partially done at the Beijing Key Laboratory of Research on Large Models and Intelligent Governance.

## Limitations

In this study, we propose a novel personalized LLM model that encodes a specific user's all history into user-specific personal embeddings and attaches it to inputs for LLMs to perform personalization. We admire several limitations in this work for further exploration and investigation.

First, in our PPlug model, we only represent histories at the behavior level. However, some terms and phrases that users frequently use in their histories can also help us to capture general user patterns and styles. A potential future work is to augment the personal embedding with fine-grained term-level information. Second, as we experimented and discussed in Section 4.5, PPlug can be integrated with retrieval-based methods to improve performance. In the future, we can study when to utilize the user embedding and when to use the in-context retrieved references for personalizing LLM-generated outputs.

## Ethical Considerations

The LaMP benchmark used in our experiments is publicly available on the Web and does not have privacy concerns. For the applications of personalized language generation, they usually require the collection of user historical data, which may cause privacy leakage problems. Although there may exist risks of abusing and leaking user data in personalization tasks, our proposed PPlug model indeed alleviates or even solves the problems. LLM service providers only need to release the tuned user embedder model to users, and users can build and upload their specific personal embeddings by themselves to guide LLMs in providing personalized results. During this process, users do not need to upload their own historical text data. In contrast, previous personalized LLM approaches need to obtain user data for retrieval or tuning.

## References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin

Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, and et al. 2023. Palm 2 technical report. *CoRR*, abs/2305.10403.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, Defu Lian, and Enhong Chen. 2023. When large language models meet personalization: Perspectives of challenges and opportunities. *CoRR*, abs/2307.16376.

Konstantina Christakopoulou, Alberto Lalama, Cj Adams, Iris Qu, Yifat Amir, Samer Chucri, Pierce Vollucci, Fabio Soldo, Dina Bseiso, Sarah Scodel, Lucas Dixon, Ed H. Chi, and Minmin Chen. 2023. Large language models for user interest journeys. *CoRR*, abs/2305.15498.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416.

Sumanth Doddapaneni, Krishna Sayana, Ambarish Jash, Sukhdeep S. Sodhi, and Dima Kuzmin. 2024. User embedding model for personalized language prompting. *CoRR*, abs/2401.04858.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *NMT@ACL*, pages 56–60. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP (1)*, pages 6894–6910. Association for Computational Linguistics.

Songwei Ge, Zhicheng Dou, Zhengbao Jiang, Jian-Yun Nie, and Ji-Rong Wen. 2018. Personalizing search results using hierarchical RNN with query-aware attention. In *CIKM*, pages 347–356. ACM.

Liam Hebert, Krishna Sayana, Ambarish Jash, Alexandros Karatzoglou, Sukhdeep S. Sodhi, Sumanth Doddapaneni, Yanli Cai, and Dima Kuzmin. 2024. PERSOMA: personalized soft prompt adapter architecture for personalized language prompting. *CoRR*, abs/2408.00960.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*. OpenReview.net.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *CoRR*, abs/2112.09118.

Wang-Cheng Kang and Julian J. McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*, pages 197–206. IEEE Computer Society.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, and Hamed Zamani. 2024. Longlamp: A benchmark for personalized long-form text generation. *CoRR*, abs/2407.11016.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiongnan Liu, Jiajie Jin, Zihan Wang, Jiehan Cheng, Zhicheng Dou, and Ji-Rong Wen. 2023. RETA-LLM: A retrieval-augmented large language model toolkit. *CoRR*, abs/2306.05212.

John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. 2023. Text embeddings reveal (almost) as much as text. In *EMNLP*, pages 12448–12460. Association for Computational Linguistics.

Lin Ning, Luyang Liu, Jiaxing Wu, Neo Wu, Devora Berlowitz, Sushant Prakash, Bradley Green, Shawn O'Banion, and Jun Xie. 2024. User-llm: Efficient LLM contextualization with user embeddings. *CoRR*, abs/2402.13598.

Christopher 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. *CoRR*, abs/2310.20081.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024a. Optimization methods for personalizing large language models through retrieval augmentation. In *SIGIR*, pages 752–762. ACM.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024b. Lamp: When large language models meet personalization. In *ACL (1)*, pages 7370–7392. Association for Computational Linguistics.

Alireza Salemi and Hamed Zamani. 2024. Comparing retrieval-augmentation and parameter-efficient fine-tuning for privacy-preserving personalization of large language models. *CoRR*, abs/2409.09510.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *ACL (Findings)*, pages 1102–1121. Association for Computational Linguistics.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024a. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *EMNLP*, pages 6459–6475. Association for Computational Linguistics.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024b. Democratizing large language models via personalized parameter-efficient fine-tuning. In *EMNLP*, pages 6476–6491. Association for Computational Linguistics.

Xiangru Tang, Xingyao Zhang, Yanjun Shao, Jie Wu, Yilun Zhao, Arman Cohan, Ming Gong, Dongmei Zhang, and Mark Gerstein. 2024. Step-back profiling: Distilling user history for personalized scientific writing. *CoRR*, abs/2406.14275.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Shuting Wang, Zhicheng Dou, Jing Yao, Yujia Zhou, and Ji-Rong Wen. 2023. Incorporating explicit subtopics in personalized search. In *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, pages 3364–3374. ACM.

Stanislaw Wozniak, Bartlomiej Koptyra, Arkadiusz Janz, Przemyslaw Kazienko, and Jan Kocon. 2024. Personalized large language models. *CoRR*, abs/2402.09269.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *CoRR*, abs/2309.07597.

Kai Zhang, Lizhi Qing, Yangyang Kang, and Xiaozhong Liu. 2024a. Personalized LLM response generation with parameterized memory injection. *CoRR*, abs/2404.03565.

Peitian Zhang, Shitao Xiao, Zheng Liu, Zhicheng Dou, and Jian-Yun Nie. 2023. Retrieve anything to augment large language models. *CoRR*, abs/2310.07554.

Zhehao Zhang, Ryan A Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, et al. 2024b. Personalization of large language models: A survey. *arXiv preprint arXiv:2411.00027*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Yutao Zhu, Zhaoheng Huang, Zhicheng Dou, and Ji-Rong Wen. 2025. One token can help! learning scalable and pluggable virtual tokens for retrieval-augmented large language models. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 26166–26174. AAAI Press.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and

Ji-Rong Wen. 2023. Large language models for information retrieval: A survey. *CoRR*, abs/2308.07107.

Yutao Zhu, Peitian Zhang, Chenghao Zhang, Yifei Chen, Binyu Xie, Zheng Liu, Ji-Rong Wen, and Zhicheng Dou. 2024. INTERS: unlocking the power of large language models in search with instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 2782–2809. Association for Computational Linguistics.

Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. 2024. HYDRA: model factorization framework for black-box LLM personalization. *CoRR*, abs/2406.02888.

## A Dataset Details

Detailed statistics for the six tasks are provided in Table 6. The formats of input, output, and user histories of the six tasks are shown in Table 7.

## B More Experimental Results

### B.1 Complete Ablation Study

We show the complete ablation results on the LaMP benchmark in Table 8. Our PPlug model generally outperforms all ablation models. The results are consistent with the results in Section 4.5.

### B.2 Further History Selection Study

In this section, we further analyze the impact of the number of histories used to build the personal embedding in Section 4.5. Specifically, we modify our input-aware personal aggregator module to utilize only the top-$K$ history embeddings for constructing personal embedding, where $K$ ranges from 2 to 8. For convenient comparison, we normalize the results $R$ on each task by:

$$R_{\text{normalize}} = \frac{R - R_{K=2}}{R_{\text{all}} - R_{K=2}} + \epsilon \qquad (7)$$

where $R_{\text{all}}$ denotes the result of PPlug model using all histories,$R_{K=2}$ denotes the result of using only top-2 histories. We set $\epsilon = 0.1$. The results are shown in Figure 4.

We can observe that with the number of utilized histories increasing, the performance of the PPlug model keeps rising. However, the performance is consistently lower than PPlug model using all histories except the LaMP-7 Personalized Tweet Paraphrasing task. The reason may be that the user history length is shorter compared with other tasks, thus the selection manipulation may not break the overall user patterns severely but function as a denoising operation.

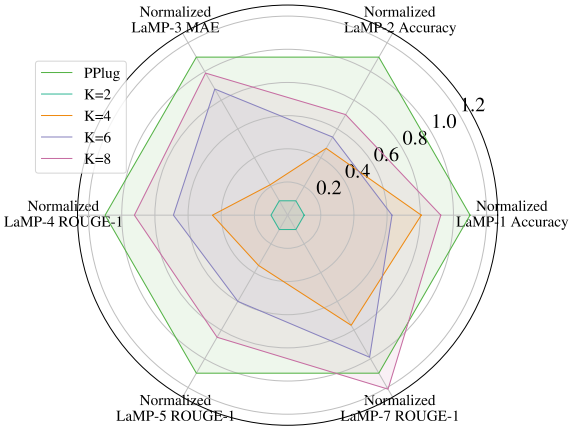


Figure 4: Performance of PPlug selecting only top-$K$ user histories on the validation set.

Table 6: Data statistics of the six experimented tasks in the LaMP benchmark.

| Task | Task Type | #Train | #Validation | #Test | Input Length | Output Length | History Length | #Classes |
|---|---|---|---|---|---|---|---|---|
| LaMP-1 | Binary classification | 6,542 | 1,500 | 1,500 | 51.43 ± 5.70 | - | 84.15 ± 47.54 | 2 |
| LaMP-2 | Categorical classification | 5,073 | 1,410 | 1,557 | 92.39 ± 21.95 | - | 86.76 ± 189.52 | 15 |
| LaMP-3 | Ordinal classification | 20,000 | 2,500 | 2,500 | 128.18 ± 146.25 | - | 185.40 ± 129.30 | 5 |
| LaMP-4 | Text generation | 12,500 | 1,500 | 1,800 | 29.97 ± 12.09 | 10.07 ± 3.10 | 204.59 ± 250.75 | - |
| LaMP-5 | Text generation | 14,682 | 1,500 | 1,500 | 162.34 ± 65.62 | 9.71 ± 3.21 | 87.88 ± 53.63 | - |
| LaMP-7 | Text generation | 13,437 | 1,498 | 1,500 | 29.72 ± 7.01 | 16.96 ± 5.67 | 15.71 ± 14.85 | - |

Table 7: Format of the input, output, and user histories of six tasks in the LaMP benchmark. *Italic text* will be replaced with realistic data for each task during training and inference.

| Task | Input | Output | User History |
|---|---|---|---|
| LaMP-1 | For an author who has written the paper with the title "*{title}*", which reference is related? Just answer with [1] or [2] without explanation. [1]: "*{reference1}*" [2]: "*{reference2}*" | [1] | title: *{title}* abstract: *{abstract}* |
| LaMP-2 | 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, ...] description: *{movie}* | sci-fi | description: *{movie}* tag: *{tag}* |
| LaMP-3 | 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: *{review}* | 3 | text: *{review}* score: *{score}* |
| LaMP-4 | Generate a headline for the following article: *{article}* | How I Got 'Rich' | title: *{title}* text: *{article}* |
| LaMP-5 | Generate a title for the following abstract of a paper: *{abstract}* | Distributed Partial Clustering | title: *{title}* text: *{abstract}* |
| LaMP-7 | Paraphrase the following tweet without any explanation before or after it: *{tweet}* | gotta make the most of my last full day in ktown | text: *{tweet}* |

Table 8: Overall performance of ablation models on the validation set. We use "Acc" to denote Accuracy and "R" to denote ROUGE, respectively.

| Model | LaMP-1 | LaMP-2 | | LaMP-3 | | LaMP-4 | | LaMP-5 | | LaMP-7 | | # Best |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc ↑ | Acc ↑ | F1 ↑ | MAE ↓ | RMSE ↓ | R-1 ↑ | R-L ↑ | R-1 ↑ | R-L ↑ | R-1 ↑ | R-L ↑ | |
| PPlug | **0.6800** | **0.5652** | **0.5014** | 0.2312 | 0.5337 | **0.2162** | **0.1972** | **0.4869** | 0.4359 | 0.5338 | **0.4836** | 7 |
| *w/o.* IE | 0.6786 | 0.5510 | 0.4834 | **0.2304** | **0.5238** | 0.2142 | 0.1960 | 0.4852 | 0.4350 | 0.5301 | 0.4781 | 2 |
| *w/o.* IA | 0.6786 | 0.5644 | 0.4924 | 0.2320 | 0.5333 | 0.2160 | 0.1961 | 0.4852 | **0.4363** | **0.5351** | 0.4818 | 2 |