# PerX: User-Adaptive Personalized Text Generation for Large Language Models

Anonymous ACL submission


## Abstract

Personalized Large Language Models (LLMs) have attracted significant attention for customized recommendation and generation, yet important challenges remain. Existing *one-for-one* approaches that fine-tune a separate model for each user are inefficient and impractical in real-world applications, whereas *one-for-all* models trained on aggregated user data lack sufficient granularity. To address these issues, we propose a novel personalization framework that, in a single forward pass, customizes user-adaptive module weights and integrates them into LLMs to produce user-specific content. Specifically, we introduce **Per**sonalization e**X**pert (PerX) that directly predicts user-specific LoRA modules, which enables efficient and fine-grained personalization without per-user fine-tuning. Specifically, we design a lightweight, stable prediction network, use distinct user representations as inputs, and employ output-wise diversity for data augmentation. PerX combines the strong performance of LoRA with the flexibility of one-for-all models, offering an effective and scalable solution for LLM personalization. Extensive experiments on personalization benchmarks demonstrate that PerX achieves strong performance across diverse tasks, surpassing existing personalized LLM approaches in most cases.


## 1 Introduction

Personalized text generation (Zhang et al., 2025c) aims to create content tailored to individual users based on their preferences or historical behavior. This task has become increasingly important in natural language processing, with broad applications across social media and online platforms, including posts, comments, and recommendations. With the growing power of Large Language Models (LLMs) (OpenAI, 2023; AI, 2024; DeepSeek-AI, 2024), personalized text generation has advanced substantially. Typically, user queries are directly fed into an LLM, optionally augmented with retrieved historical information via retrieval-augmented generation (RAG) (Lewis et al., 2020; Borgeaud et al., 2021; Izacard et al., 2023), yielding roughly personalized outputs. However, without customizing the LLM itself, these methods often lack fine-grained personalization capability.

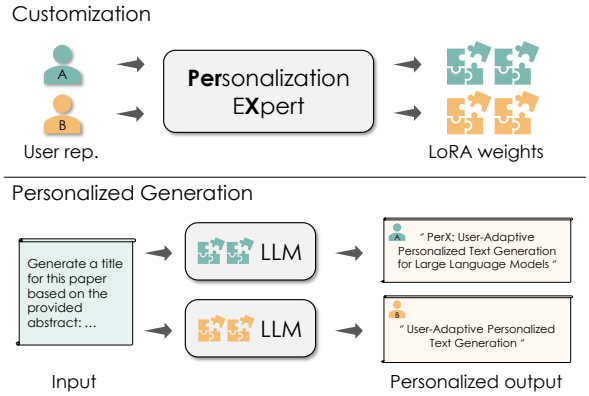


Figure 1: **Customization followed by personalized text generation.** For any arbitrary user, PerX takes an identical user representation as input to adaptively customize a set of user-specific LoRA weights. By effectively integrating these weights into LLMs, the model produces content that mirrors the user's persona.

A straightforward solution for effective personalization is to fine-tune the LLM accordingly in a *one-for-one* manner, where a separate model is fine-tuned for each individual user. Although recent works (Salemi et al., 2024a; Tan et al., 2024a; Liu et al., 2025; Huber et al., 2025) have explored this paradigm, they struggle to simultaneously achieve two goals : i) *friendly efficiency and flexible deployment*, and ii) *fine-grained control over personalization*. Specifically, training dedicated models for individual users (e.g., via full fine-tuning or parameter-efficient fine-tuning (PEFT) (Houlsby et al., 2019; Hu et al., 2022; Liu et al., 2022b)), while effective in capturing user-specific characteristics, becomes extremely inefficient as the number of users grows, thus failing to meet goal (i). Recent

*one-for-all* approaches (Liu et al., 2025; Qiu et al., 2025a) instead inject user embeddings into frozen LLMs. While more efficient, embedding-based control is often insufficient for fine-grained personalization, leaving goal (ii) inadequately addressed. These limitations motivate us to develop a more effective and scalable personalization framework.

As shown in Fig. 1, we propose an innovative personalization framework that predicts user-specific LoRA (Hu et al., 2022) weights from user representations in a single forward pass and integrates them into an LLM for personalized generation. To achieve goal (i), we introduce a *one-for-all* model, termed personalization expert (PerX), which predicts user-specific LoRA weights similarly to HyperNetworks (Ha et al., 2017), instead of optimizing separate modules for each user. To achieve goal (ii), we adopt LoRA as the personalization module due to its superior performance over alternatives such as prefix-tuning (Li and Liang, 2021) and P-tuning (Liu et al., 2022b) (Tan et al., 2024b,a). To avoid parameter explosion when predicting LoRA weights for large models, PerX is designed to be lightweight and equipped with tailored initialization strategies for stable training. Furthermore, instead of conventional input-aware embeddings (Liu et al., 2025), we employ user-aware embeddings learned through contrastive learning, enabling PerX to generate more user-specific LoRA weights and stronger personalization.

Our framework is trained in two stages. First, a user profile encoder is optimized with contrastive learning to produce distinct user representations. Then, PerX is trained with standard autoregressive supervision with output-wise diversity augmentation. We conduct extensive experiments on two representative personalization benchmarks, LaMP (Salemi et al., 2024b) and LongLaMP (Kumar et al., 2024), which target short- and long-text generation, respectively. PerX achieves state-of-the-art performance on most tasks, surpassing the strongest baseline by 0.4% to 45.2% across the LaMP and LongLaMP benchmarks. We further demonstrate the scalability and hot-update capability of PerX, highlighting its adaptability to dynamic and evolving user data.

Our main contributions are as follows:

- We propose a *one-for-all* personalization expert that predicts user-specific LoRA weights conditioned on user representations, enabling efficient and fine-grained personalization without per-user fine-tuning.
- We show that PerX possesses strong scalability and hot-update capability, allowing efficient adaptation to dynamic and evolving user data in real-world applications.
- Extensive experiments on both short- and long-text personalization benchmarks demonstrate that PerX achieves state-of-the-art performance on most tasks, validating the effectiveness and generalizability of our approach.

## 2 Related Work

**Personalized Text Generation.** Recent personalized text generation methods can be broadly categorized into training-free and training-based approaches. Training-free methods adapt LLMs without updating model parameters, mainly through prompt augmentation. Early approaches directly prepend long user histories or few-shot examples into prompts (Christakopoulou et al., 2023; Zhao et al., 2025a). To improve scalability, retrieval-based methods (Salemi et al., 2024b; Lewis et al., 2020; Mysore et al., 2024; Salemi et al., 2024a; Shi et al., 2025) select the most relevant historical information, while profile-based methods (Sun et al., 2025; Richardson et al., 2023; Qiu et al., 2025b) summarize user histories into compact profiles for more efficient conditioning. Beyond explicit prompt augmentation, several works (Zhang et al., 2025b; Ma et al., 2025) exploit intrinsic properties of pretrained LLMs, such as steering vectors, to enhance personalization through lightweight control signals. Nevertheless, without additional training, these approaches remain limited in modeling fine-grained personalization.

Training-based approaches instead adapt model parameters for personalization. Some works (Tan et al., 2024b) train a dedicated model for each user, while others (Tan et al., 2024a) build a small set of personalized anchor models and generalize to unseen users through weighted model merging. More recently, one-for-all paradigms (Liu et al., 2025; Qiu et al., 2025a; Zhao et al., 2025b; Huber et al., 2025; Salemi et al., 2025) have attracted growing attention. These methods typically derive user embeddings from historical data and inject them into LLMs for personalized generation. For example, PPlug (Liu et al., 2025) constructs input-aware persona embeddings conditioned on both user history and input queries. Although efficient, these methods are still constrained by embedding injection mechanisms largely inspired by prefix tuning (Li

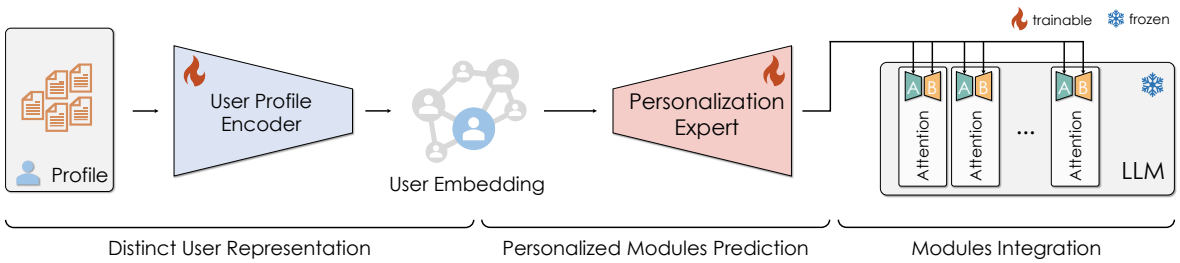


Figure 2: **Method overview.** Given a profile, we first encode it into a distinct user embedding that captures the user's personalized characteristics. Then we pass it to the personalization expert (PerX), which predicts user-specific module weights as customized LoRAs. These predicted LoRA weights are finally integrated into the LLM backbone, enabling personalized generation.

and Liang, 2021) and P-tuning (Liu et al., 2022b).

**Large Language Models Fine-tuning.** Full fine-tuning is often the most effective strategy for customizing LLMs. However, its computational and storage costs become prohibitive as model size increases. This challenge has motivated extensive research on PEFT methods (Hu et al., 2022; Liu et al., 2022b; Li and Liang, 2021; Ben Zaken et al., 2022; Liu et al., 2022a; Mahabadi et al., 2021). Among these, LoRA (Hu et al., 2022) has become one of the most widely adopted approaches due to its simplicity and effectiveness. Inspired by the success of Mixture-of-Experts (MoE) (Jacobs et al., 1991; Mu and Lin, 2025), several recent works (Zhao et al., 2024; Wang et al., 2025; Feng et al., 2024; Dou et al., 2024; Liu et al., 2024) further extend LoRA by introducing multiple LoRA experts together with dynamic routing mechanisms. Beyond LoRA-based methods, alternative PEFT strategies have also been explored. Prefix tuning (Li and Liang, 2021) and P-tuning (Liu et al., 2022b) inject trainable prompt representations into pretrained transformers for efficient adaptation, while BitFit (Ben Zaken et al., 2022), IA3 (Liu et al., 2022a), and adapter-based methods (Hu et al., 2023; Mahabadi et al., 2021) adapt only a small subset of parameters or lightweight modules. Although PEFT methods substantially reduce training costs, they often face trade-offs between parameter efficiency and adaptation capability.

# 3 Preliminaries

## 3.1 Problem Formulation

Given an input query $q_u$ for a user $u$ and the user history $\mathcal{H}_u = \{h_u^1, h_u^2, ..., h_u^n\}$, personalized text generation aims to generate a personalized response $r_u$ conditioned on both the query and historical behaviors. Each behavior $h_u^i$ consists of a query-response pair $(q_u^i, r_u^i)$ with timestamp $t_u^i$. Our goal is to develop a *one-for-all* model that enables efficient and fine-grained personalization.

## 3.2 LoRA Adaptation and HyperNetworks

**LoRA Adaptation.** LoRA (Hu et al., 2022) fine-tunes a pretrained model by introducing a low-rank decomposition, where the original weight matrix $W \in \mathbb{R}^{d_{out} \times d_{in}}$ is augmented with two low-rank matrices $A \in \mathbb{R}^{r \times d_{in}}$ and $B \in \mathbb{R}^{d_{out} \times r}$, where rank $r \ll \min(d_{in}, d_{out})$. During training, the origin weights remain frozen, and only $A$ and $B$ are optimized. For a feature projection $h = Wx$, the forward pass with LoRA becomes:

$$h = Wx + \Delta Wx = Wx + BAx. \quad (1)$$

LoRA's effectiveness in enabling efficient adaptation makes it an ideal choice for building a *one-for-all* personalization model.

**HyperNetworks.** HyperNetworks (Ha et al., 2017) are auxiliary networks that generate weight offsets for a primary model conditioned on a signal $c$. Given a primary model $M$ with parameters $\Theta$, a HyperNetwork $H$ predicts parameter adjustments $\Delta\Theta(c)$ as:

$$y = M(x; \Theta + \Delta\Theta(c)) = M(x; \Theta + H(c)), \quad (2)$$

where $x$ and $y$ denote the input and output of the primary network, respectively. By dynamically generating weights conditioned on inputs, HyperNetworks enable more flexible model adaptation.

# 4 PerX: Personalization Expert

## 4.1 Architecture of PerX

Aiming at the two goals in Sec. 1, PerX represents each user as an input and predicts user-specific modules, which are then integrated into the LLM for personalized generation. As shown in Fig. 2, we employ LoRA as the core personalization module.

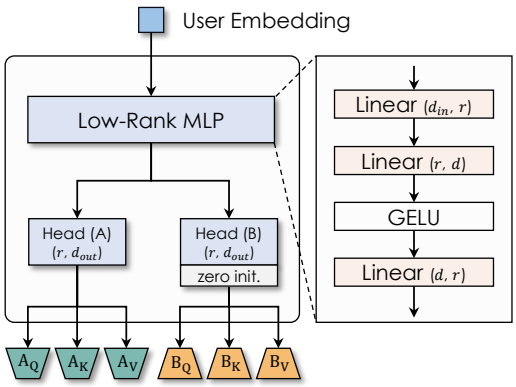

Figure 3: Architecture of our personalization expert.

In standard LoRA-based fine-tuning, LoRA modules are typically applied to QKV projections across all attention layers. However, either training separate LoRA modules per user or sharing a single global LoRA across users fails to satisfy our objectives. To address this, we generate LoRA modules via a HyperNetwork-like mechanism (Ha et al., 2017), enabling user-conditioned adaptation.

**Model Architecture.** To generate user-specific LoRA modules, we design PerX, an MLP-based network as illustrated in Fig. 3. Given a user embedding, PerX predicts LoRA parameters through a shared MLP followed by two linear heads, Head (A) and Head (B), which output the low-rank matrices $A$ and $B$, respectively. Following common practice, we generate LoRA matrices for QKV projections in each attention block. For block $l$, the expert produces $\mathcal{A}^l = \{A_Q^l, A_K^l, A_V^l\}$ and $\mathcal{B}^l = \{B_Q^l, B_K^l, B_V^l\}$, forming the personalized adaptation modules. We will present how to obtain the user embeddings in the next subsection.

**Lightweighting and Initialization.** In PerX, the shared MLP and prediction heads are implemented as linear layers. Let $d$ be the LLM hidden size, $r$ the LoRA rank, and $L$ the number of attention blocks. A naive design leads to $O(Lrd^2)$ space complexity, which becomes prohibitive for large models.

To improve efficiency, we introduce a LoRA-style low-rank factorization of the MLP weights. Specifically, a linear layer $W \in \mathbb{R}^{d\times d}$ is decomposed into $W_A \in \mathbb{R}^{r\times d}$ and $W_B \in \mathbb{R}^{d\times r}$, reducing the complexity to $O(Ldr^2)$ with $r \ll d$. The resulting lightweight architecture and layer specifications are presented in Fig. 3. An in-depth analysis of the rank selection is presented in Sec. 5.3.

For stable training, we follow a standard LoRA initialization strategy, where Head (B) is initialized to zero while Head (A) is randomly initialized. We further apply a $\tanh$ activation with an empirical factor $s = 0.3$ on Head (B) to prevent gradient instability. The final computation is expressed as:

$$\{A_Q, A_K, A_V\} = \text{Head}_{(A)}(\text{MLP}(e)),$$
$$\{B_Q, B_K, B_V\} = s \cdot \tanh(\text{Head}_{(B)}(\text{MLP}(e))), \tag{3}$$

where $e$ denotes the user embedding.

### 4.2 Distinct User Representation

The user embedding input to PerX is crucial for personalization. Recently, PPlug (Liu et al., 2025) constructed input-aware user embeddings conditioned on queries, such representations can vary across different inputs, which is undesirable for generating a consistent user-specific LoRA module. Inspired by (Shi et al., 2025), we instead learn a query-independent user representation via contrastive learning using a user profile encoder $E_{usr}$.

Fig. 4(a) present the details of our user profile encoder. We first encode each user's historical behaviors $\mathcal{H}_u$ into embeddings $\mathcal{E}_u = \{e_u^1, e_u^2, ..., e_u^n\}$ using a pretrained text encoder (e.g., BGE (Xiao et al., 2024)). A QFormer (Li et al., 2023) is then used as the user profile encoder, where a learnable token serves as Query and the historical embeddings are projected to serve as Keys and Values. We also add learnable positional encodings to inject temporal information, enabling the model to handle variable-length histories.

As illustrated in Fig. 4(b), we train the encoder using InfoNCE (van den Oord et al., 2018). Given a user $U_a$, we construct two augmented views of its history via cropping and masking (Shi et al., 2025), forming an anchor-positive pair $(e^{\text{anc}}, e^{\text{pos}})$. A negative embedding $e^{\text{neg}}$ is sampled from another user $U_b$. We optimize the cosine similarity-based contrastive objective:

$$\mathcal{L}_{\text{CL}} = -\log \frac{\exp(\text{sim}(e^{\text{anc}}, e^{\text{pos}})/\tau)}{\exp(\text{sim}(e^{\text{anc}}, e^{\text{pos}})/\tau) + C^{\text{neg}}}, \tag{4}$$

where $C^{\text{neg}} = \sum_{k=1}^{K} \exp(\text{sim}(e^{\text{anc}}, e_k^{\text{neg}})/\tau)$, $\text{sim}(\cdot,\cdot)$ is cosine similarity, $\tau$ is the temperature, and $\{e_k^{\text{neg}}\}_{k=1}^{K}$ are negatives from other users.

### 4.3 Data Augmentation and Training Process

**Data Augmentation.** Existing benchmarks (Salemi et al., 2024b; Kumar et al., 2024) typically provide only one interaction per user, limiting data diversity for training PerX. While PPlug (Liu et al.,

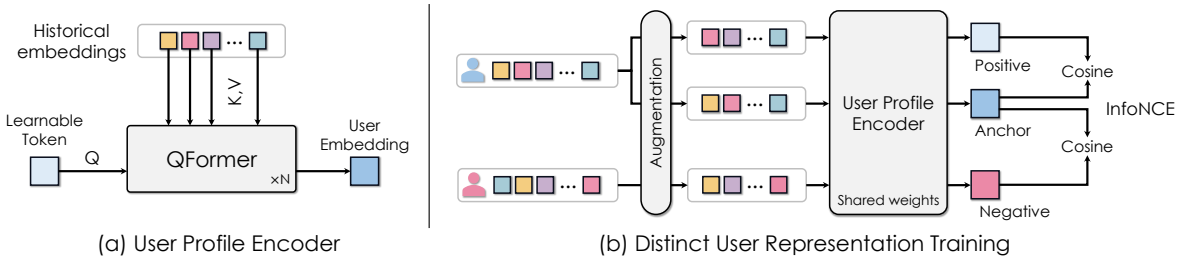


Figure 4: **Model architecture and training illustration of the user profile encoder.** (a) The encoder is built upon a multilayer Q-Former, which takes user historical embeddings as input and produces a distinct user embedding. To promote discriminability across users, the encoder is trained with a contrastive learning objective to obtain distinct user representations.

2025) improves diversity by sampling different subsets of user histories, it still reuses the same input-output pairs, resulting in limited supervision signals. To address this issue, we design an output-wise diversity augmentation tailored to our formulation. Specifically, for each user, we construct training samples by selecting one interaction $(q_u^i, r_u^i)$ as the input-output pair, while using the remaining interactions $\{h_u^{i+1}, h_u^{i+2}, \ldots, h_u^n\}$ as historical context. This yields multiple distinct samples per user, improving output diversity and providing richer supervision for PerX.

**Model Training.** We train our framework in two stages. First, we train the user profile encoder using the contrastive loss in Eq. 4. Then, we optimize PerX while freezing both the encoder and the backbone LLM. The objective is:

$$\arg\min_{\Theta} \ \mathrm{CE}(\mathrm{LLM}(q_u; \Theta_{\mathrm{LLM}}, \mathrm{PerX}(e_u; \Theta)), r_u), \tag{5}$$

where $\mathrm{CE}(\cdot, \cdot)$ is the cross-entropy loss, $\Theta_{\mathrm{LLM}}$ and $\Theta_{\mathrm{PerX}}$ are the parameters of LLM and PerX, respectively, and $e_u$ is the user embedding. $(q_u, r_u)$ is the input-output training pair.

### 4.4 Generation Pipeline

When deployed in production, our framework consists of a text-embedding model (e.g., BGE (Xiao et al., 2024)), the user profile encoder described in Sec. 4.2, the personalization expert (PerX) from Sec. 4.1, and a pretrained LLM. For a user $u$, during customization, we first encode the historical behaviors $\mathcal{H}_u$ into embeddings $\mathcal{E}_u = E(\mathcal{H}_u)$ using a text embedding model, and obtain the user representation via the user profile encoder $e_u = E_{usr}(\mathcal{E}_u)$. PerX then generates user-specific LoRA modules $\{A_Q, A_K, A_V\}$ and $\{B_Q, B_K, B_V\}$ conditioned on $e_u$. During inference, the LoRA weights are merged into the LLM as:

$$W'_Q = W_Q + B_Q A_Q, \quad W'_K = W_K + B_K A_K, \quad W'_V = W_V + B_V A_V. \tag{6}$$

To recover the original LLM, one can simply subtract the corresponding LoRA weights $B_* A_*$.

Our framework supports two key properties: i) scalability to users with varying amounts of historical behaviors, and ii) hot updates, where updated user embeddings and LoRA weights can be generated efficiently when new interactions arrive. We will demonstrate these capabilities in Sec. 5.4.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets and Metrics.** We evaluate PerX on the public LaMP benchmark (Salemi et al., 2024b), which contains six available personalization tasks: three classification tasks (LaMP-1/2/3) and three generation tasks (LaMP-4/5/7). We follow the official time-based train/validation splits for all experiments. Following prior work (Liu et al., 2025), we report Accuracy for LaMP-1, Accuracy and F1 for LaMP-2, MAE and RMSE for LaMP-3, and Rouge-1/Rouge-L (Lin, 2004) for the generation tasks. Additional experiments on LongLaMP (Kumar et al., 2024) are provided in the Appendix.

**Baselines.** We compare PerX against both non-retrieval and retrieval-based methods. The non-retrieval baselines include a naive LLM without personalization, PEFT-based methods (e.g., LoRA (Hu et al., 2022)), and PPlug (Liu et al., 2025). The retrieval-based baselines include RAG (Lewis et al., 2020), ROPG (Salemi et al., 2024a), PAG (Richardson et al., 2023), as well as combined approaches such as RAG+PEFT (LoRA) and RAG+PPlug. To ensure reproducibility, we report all implementation details of PerX as well as all baseline methods

Table 1: **Comparison of PerX with other methods on the LaMP benchmark.** The best and second-best methods are highlighted in bold and underlined, respectively. Statistical significance tests against PEFT (LoRA) and PPlug indicate that PerX achieves statistically significant improvements ($p < 0.05$) across all tasks. *: For LaMP-1, PerX was trained without data augmentation.

| | Methods (Flan-T5-XXL) | LaMP-1 | LaMP-2 | | LaMP-3 | | LaMP-4 | | LaMP-5 | | LaMP-7 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Accuracy ↑ | Accuracy ↑ | F1 ↑ | MAE ↓ | RMSE ↓ | Rouge-1 ↑ | Rouge-L ↑ | Rouge-1 ↑ | Rouge-L ↑ | Rouge-1 ↑ | Rouge-L ↑ |
| Non Retrieval | No Personalization | 0.496 | 0.326 | 0.258 | 0.317 | 0.604 | 0.176 | 0.161 | 0.482 | 0.431 | 0.456 | 0.403 |
| | PEFT (LoRA) (Hu et al., 2022) | 0.658 | 0.549 | 0.485 | 0.226 | 0.523 | 0.206 | 0.186 | 0.493 | 0.441 | 0.532 | 0.479 |
| | PPlug (Liu et al., 2025) | **0.680** | 0.565 | 0.501 | 0.231 | 0.534 | **0.216** | **0.197** | 0.487 | 0.435 | 0.536 | 0.484 |
| | **PerX (Ours)** | 0.645* | **0.679** | **0.640** | **0.222** | **0.508** | 0.199 | 0.183 | **0.501** | **0.450** | **0.538** | **0.486** |
| Retrieval based | RAG (Lewis et al., 2020) | 0.641 | 0.363 | 0.284 | 0.294 | 0.587 | 0.190 | 0.174 | 0.472 | 0.423 | 0.442 | 0.393 |
| | ROPG (Salemi et al., 2024a) | 0.668 | 0.396 | 0.306 | 0.290 | 0.604 | 0.187 | 0.172 | 0.477 | 0.428 | 0.441 | 0.391 |
| | PAG (Richardson et al., 2023) | 0.728 | 0.543 | 0.480 | 0.277 | 0.594 | 0.173 | 0.159 | 0.447 | 0.408 | 0.512 | 0.459 |
| | RAG+PEFT (LoRA) (Hu et al., 2022) | **0.769** | 0.593 | 0.544 | 0.214 | 0.496 | **0.226** | **0.208** | 0.506 | 0.453 | 0.551 | 0.501 |
| | RAG+PPlug (Liu et al., 2025) | 0.687 | 0.545 | 0.485 | 0.215 | 0.506 | 0.220 | 0.203 | 0.498 | 0.448 | 0.547 | 0.495 |
| | **RAG+PerX (Ours)** | 0.755* | **0.686** | **0.654** | **0.205** | **0.484** | 0.222 | 0.206 | **0.511** | **0.460** | **0.554** | **0.505** |

in Appendix C.

### 5.2 Experimental Results

**Main Results.** Table 1 presents the comparison of PerX with existing approaches on the LaMP benchmark. We summarize the key findings as follows:

1) Focusing on non-retrieval methods, **PerX achieves state-of-the-art performance on most tasks**, with relative improvements over the strongest baselines ranging from 0.4% to 27.7%. This highlights the effectiveness of our method in capturing user-specific preferences for personalized text generation.

2) **LoRA shows strong potential over embedding injection.** The results of PEFT (LoRA) in Table 1 show that training a generic LoRA without leveraging histories remains competitive with PPlug. Considering that PPlug extracts user embeddings from historical data and injects them into the LLM, while PEFT (LoRA) is trained solely on input-output pairs, this comparison suggests that adopting LoRA as a personalization module holds significant potential over embedding injection.

3) **PerX enables finer-grained personalization.** PerX substantially surpasses both PEFT (LoRA) and PPlug on LaMP-2/3/5/7, demonstrating that: i) a generic LoRA struggles to adapt to diverse user needs, whereas our design of predicting LoRA enables more accurate and adaptive personalization; and ii) PerX provides finer-grained controllability over personalization than PPlug, as it relies on embedding injection with limited granularity in capturing user characteristics.

4) **Limitations on LaMP-1 and LaMP-4.** PerX underperforms on LaMP-1 and LaMP-4. For LaMP-1, the task format is not well suited to our augmentation strategy, preventing effective data augmentation, and highlighting the dependence of PerX on high-quality supervision. For LaMP-4, the news headline generation task suffers from noisy and incomplete inputs, as the news content provided in the dataset is always truncated. This makes the evaluation less reliable. Despite these limitations, PerX consistently outperforms the baselines on the remaining tasks, demonstrating strong robustness and generalization.

**Results Integrated with RAG.** Considering RAG (Lewis et al., 2020) as a general and effective retrieval-based personalization strategy, we integrate it into PerX and compare against recent retrieval-based methods. As reported in the bottom part of Table 1, incorporating RAG into PerX brings consistent gains, achieving relative improvements over the strongest retrieval-based baselines ranging from 0.5% to 20.2%. We further summarize the following observations:

1) **LoRA-based personalization outperforms embedding injection.** With RAG integration, PEFT (LoRA) consistently surpasses RAG+PPlug, indicating the stronger personalization capability of LoRA-based adaptation.

2) **User-adaptive LoRA customization remains superior.** Even with RAG augmentation, PerX consistently outperforms PEFT (LoRA) on LaMP-2/3/5/7, further validating the advantage of predicting user-adaptive LoRA modules over learning shared ones.

3) **Training enhances personalization beyond retrieval only.** Compared to training-free RAG, approaches that combine retrieval with trainable modules (RAG+PPlug, RAG+PEFT (LoRA), and RAG+PerX), achieve significantly better

Table 2: **Ablation study.** The importance of user embeddings and data augmentation is well validated through our ablation analysis.

| | Methods (Flan-T5-XXL) | LaMP-2 | | LaMP-3 | | LaMP-5 | | LaMP-7 | |
|---|---|---|---|---|---|---|---|---|---|
| | | Accuracy ↑ | F1 ↑ | MAE ↓ | RMSE ↓ | Rouge-1 ↑ | Rouge-L ↑ | Rouge-1 ↑ | Rouge-L ↑ |
| (i) | PerX (full) | **0.679** | **0.640** | **0.222** | **0.508** | **0.501** | **0.450** | **0.538** | **0.486** |
| (ii) | w/ user embedding from PPlug (Liu et al., 2025) | 0.669 | 0.636 | 0.223 | 0.510 | 0.497 | 0.448 | 0.537 | 0.485 |
| (iii) | w/o data aug. | 0.669 | 0.639 | 0.241 | 0.522 | 0.486 | 0.435 | 0.534 | 0.480 |
| (iv) | w/ aug. on profile diversity | 0.662 | 0.632 | 0.258 | 0.546 | 0.485 | 0.435 | 0.521 | 0.468 |
| (v) | case (ii) + case (iii) | 0.653 | 0.632 | 0.238 | 0.524 | 0.490 | 0.440 | 0.528 | 0.476 |

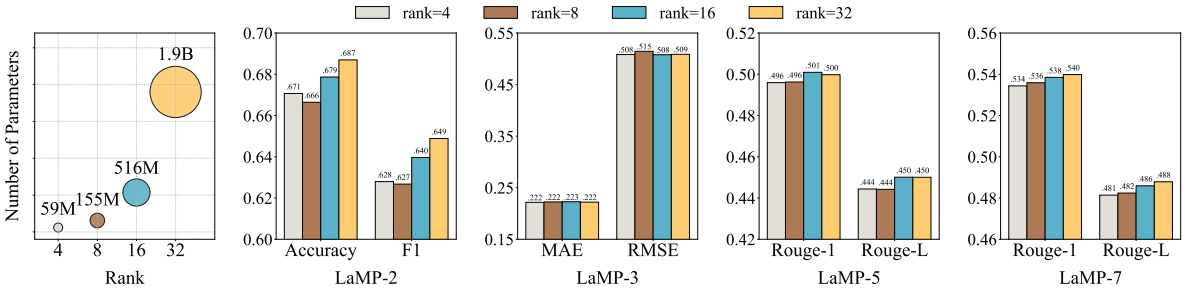


Figure 5: **Impact of rank in the personalization expert.** A larger rank generally leads to better performance, but at the cost of quadratically increased parameter size.

results, highlighting the importance of model adaptation for fine-grained personalization.

**Results on LongLaMP.** We further evaluate our method on the LongLaMP benchmark (Kumar et al., 2024), where consistent improvements of 1.4%-45.2% over the PEFT (LoRA) baseline are observed across tasks. Detailed results are provided in the Appendix.

### 5.3 Ablation Analysis

This section presents the ablation analysis, where we mainly evaluate the contributions of two key components in our framework: the user embeddings used as input to PerX and the data augmentation strategy. We replace each component with a baseline alternative to assess its impact on overall performance. Table 2 lists the full ablation results on LaMP-2/3/5/7.

**Effectiveness of Distinct User Representation.** We first validated the effectiveness of our distinct user representation, by replacing it with the input-aware embeddings from PPlug (Liu et al., 2025). As shown in case (ii) of Table 2, our user-aware embeddings consistently achieve better performance, indicating that user-aware representations are more effective for generating LoRA modules tailored to individual users than query-aware embeddings.

**Effectiveness of Output-Wise Diversity Augmentation.** We also validated the effectiveness of our output-wise diversity augmentation with two alternatives: no augmentation and profile-diversity augmentation. Results in cases (iii) and (iv) of Table 2 show that our strategy consistently improves PerX training, while profile-based diversity augmentation can even underperform the no augmentation setting. This suggests that shuffled historical data for the same input-output pair is less suitable for user-specific customization and instead align more naturally with input-specific scenarios, as in PPlug.

Moreover, replacing both components with the baseline alternatives (i.e., case (v)), the overall performance degrades even further. In summary, these two components play a crucial role in improving the effectiveness of PerX, thereby advancing the state of personalization methods.

**Impact of Rank in the Personalization Expert.** As the LoRA rank and the low-rank MLP in PerX jointly affect both performance and parameter size, we conducted a systematic analysis to balance these two factors. For simplicity, we assign the same rank to LoRA and the low-rank MLP. Fig. 5 presents the results on LaMP-2/3/57. As the rank increases, the parameter size grows quadratically, which is consistent with the space complexity of PerX, $O(Ldr^2)$. In terms of performance, PerX exhibits a clear upward trend across most tasks (i.e., LaMP-2/5/7) as rank increases. For LaMP-3, although such an increasing trend is not observed, the performance remains largely stable across different ranks. Con-

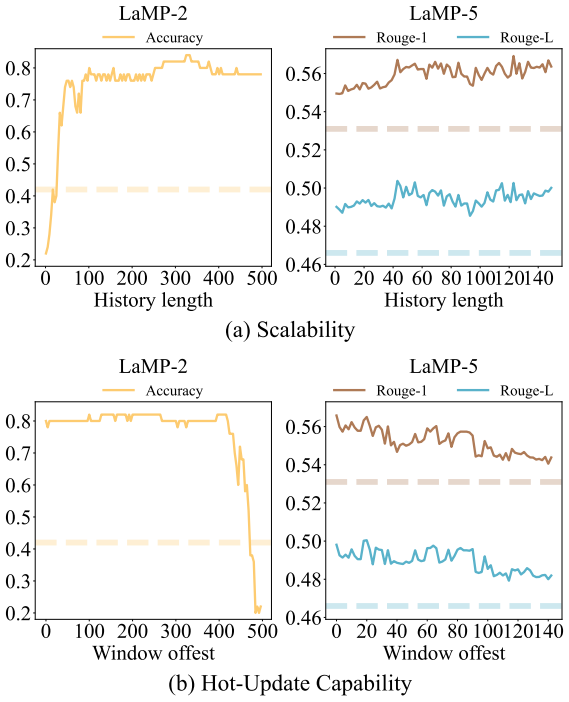


Figure 6: **Scalability and hot-update capability of PerX.** We select 50 samples with the longest historical records to verify these properties. The solid lines represent results from PerX, while the dashed lines correspond to the baseline PEFT (LoRA).

sidering the trade-off between parameter size and model performance, we finally set the rank to 16 in PerX, which only introduces 4.63% additional parameters relative to the backbone LLM (Flan-T5-XXL, 11B) (Chung et al., 2022) in our experiments.

### 5.4 Scalability and Hot-Update Capability

To further investigate the properties of our proposed method, we conduct additional experiments focusing on two aspects: scalability and hot-update capability, as introduced in Sec. 4.4. Specifically, we evaluate PerX on two representative tasks: LaMP-2 for classification and LaMP-5 for generation. For each task, We collect 50 samples from the validation set with the longest historical data, where each sample contains more than 774 and 208 historical entries for LaMP-2 and LaMP-5, respectively. These experiments are designed to systematically examine how PerX leverages user histories of varying lengths and adapts to newly arriving data.

**Scalability.** To assess scalability, we vary the number of the most recent user histories provided as input to the user profile encoder $E_{usr}$. The resulting user embedding is then fed into PerX to predict LoRA module weights for personalization. For each selected sample, we incrementally increase the number of histories from 1 up to 512, which is the maximum input length supported by $E_{usr}$ in our setting, and measure the corresponding evaluation metrics. As shown in Fig. 6(a), model performance improves steadily as more historical data are incorporated. This demonstrates that PerX effectively leverages variable-length user histories, producing more accurate and stable personalized outputs as additional data are included, thereby confirming its scalability. Moreover, across most histories input settings, PerX outperforms the baseline PEFT (LoRA), further underscoring its superiority.

**Hot-Update Capability.** As the historical records accumulate, they may exceed the maximum input length of $E_{usr}$, which is set to 512 in our setting. Moreover, since user personas evolve over time, it is crucial to accurately capture their most recent preferences. To this end, we investigate the hot-update capability of PerX. Specifically, we employ a fixed-size window of histories and progressively shift the starting point away from the most recent record. We set the window size to 256 for LaMP-2 and 64 for LaMP-5, and use the histories within each window as the input to $E_{usr}$ for personalization. Fig. 6(b) reports the results, where the window offset denotes how far the window start is from the most recent history. As the offset increases, performance gradually declines, indicating that PerX is sensitive to the latest user behavior. These results highlight that PerX can rapidly adapt to evolving user preferences by generating updated LoRA weights from newly available histories in a single forward pass, thereby validating the hot-update capability discussed in Sec. 4.4.

## 6 Conclusion

We present PerX, an efficient framework for personalized text generation. By predicting user-adaptive LoRA modules, PerX captures fine-grained personalization while remaining lightweight and efficient. Extensive experiments on the personalization benchmarks demonstrate clear and consistent advantages over strong baselines, while ablation studies confirm the importance of each component. With its scalability and hot-update capability, PerX holds strong potential as an effective and practical solution for personalized text generation.

## 7 Limitations

While the extensive experiments presented above demonstrate the effectiveness of PerX on the LaMP (Salemi et al., 2024b) and LongLaMP (Kumar et al., 2024) benchmark, along with its scalability and hot-update capability, our method still faces several limitations.

First, the performance of PerX relies on the informativeness and quality of user histories. In cases where user data is sparse or noisy, as in LaMP-4 of LaMP, the predicted LoRA modules may fail to capture stable personalization patterns, leading to suboptimal performance compared with some existing methods.

Further, although PerX remains lightweight compared to full fine-tuning, predicting and injecting user-specific LoRA modules introduces minor latency and additional computational cost during inference.

Despite these limitations, our experiments clearly demonstrate the effectiveness and superiority of PerX for short- and long-text personalization. It provides a strong step toward efficient and scalable personalization in LLMs. As future work, we will explore broader application domains to further validate the generality of PerX.

## References


Meta AI. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, T. W. Hennigan, Saffron Huang, Lorenzo Maggiore, Chris Jones, Albin Cassirer, and 9 others. 2021. Improving language models by retrieving from trillions of tokens. In *Proc. of IEEE Int. Conf. on Machine Learning*.

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. *arXiv preprint arXiv:2305.15498*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, and 16 others. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

DeepSeek-AI. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Shihan Dou, Enyu Zhou, Yan Liu, Songyang Gao, Wei Shen, Limao Xiong, Yuhao Zhou, Xiao Wang, Zhiheng Xi, Xiaoran Fan, Shiliang Pu, Jiang Zhu, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. LoRAMoE: Alleviating world knowledge forgetting in large language models via MoE-style plugin. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 1932–1945, Bangkok, Thailand. Association for Computational Linguistics.

Wenfeng Feng, Chuzhan Hao, Yuewei Zhang, Yu Han, and Hao Wang. 2024. Mixture-of-loras: An efficient multitask tuning for large language models. *arXiv preprint arXiv:2403.03432*.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proc. of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver. Association for Computational Linguistics.

David Ha, Andrew M. Dai, and Quoc V. Le. 2017. Hypernetworks. In *Proc. of Int. Conf. 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 *Proc. of Int. Conf. on Learning Representations*.

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 *Proc. of Int. Conf. on Learning Representations*.

Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. 2023. LLM-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proc. of Conf. on Empirical Methods in Natural Language Processing*, pages 5254–5276, Singapore. Association for Computational Linguistics.

Bernd Huber, Gahazal Fazelnia, Andreas Damianou, Sebastian Peleato, Max Lefarov, Praveen Ravichandran, Marco De Nadai, Mounia Lalmas-Roellke, and Paul N. Bennett. 2025. Embedding-to-prefix: Parameter-efficient personalization for pre-trained large language models. *arXiv preprint arXiv:2505.17051*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. on Machine Learning Research*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: few-shot learning with retrieval augmented language models. *The Journal of Machine Learning Research*, 24(1).

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proc. of IEEE Int. Conf. on Machine Learning*, volume 202, pages 19730–19742.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, 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 Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

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

Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Derong Xu, Feng Tian, and Yefeng Zheng. 2024. When moe meets llms: Parameter efficient fine-tuning for multi-task medical applications. In *Proc. of Conf. on Research and Development in Information Retrieval*, page 1104–1114, New York, NY, USA. Association for Computing Machinery.

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 *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proc. of Int. Conf. on Learning Representations*.

Xinyu Ma, Yifeng Xu, Yang Lin, Tianlong Wang, Xu Chu, Xin Gao, Junfeng Zhao, and Yasha Wang. 2025. DRESSing up LLM: Efficient stylized question-answering via style subspace editing. In *Proc. of Int. Conf. on Learning Representations*.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

Siyuan Mu and Sen Lin. 2025. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. *arXiv preprint arXiv:2503.07137*.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. 2024. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. In *Proc. of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 198–219, Miami, Florida, USA. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Yilun Qiu, Tianhao Shi, Xiaoyan Zhao, Fengbin Zhu, Yang Zhang, and Fuli Feng. 2025a. Latent inter-user difference modeling for llm personalization. *arXiv preprint arXiv:2507.20849*.

Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yimeng Bai, Wenjie Wang, Hong Cheng, Fuli Feng, and Tat-Seng Chua. 2025b. Measuring what makes you unique: Difference-aware user modeling for enhancing LLM personalization. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 21258–21277, Vienna, Austria. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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, Surya Kallumadi, and Hamed Zamani. 2024a. Optimization methods for personalizing large language models through retrieval augmentation. In *Proc. of Conf. on Research and Development in Information Retrieval*, page 752–762, New York, NY, USA. Association for Computing Machinery.

Alireza Salemi, Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, Tao Chen, Zhuowan Li, Michael Bendersky, and Hamed Zamani. 2025. Reasoning-enhanced self-training for long-form personalized text generation. *arXiv preprint arXiv:2501.04167*.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024b. LaMP: When large language models meet personalization. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval augmented generation with collaborative filtering for personalized text generation. In *Proc. of Conf. on Research and Development in Information Retrieval*, New York, NY, USA. Association for Computing Machinery.

Chenkai Sun, Ke Yang, Revanth Gangi Reddy, Yi Fung, Hou Pong Chan, Kevin Small, ChengXiang Zhai, and Heng Ji. 2025. Persona-DB: Efficient large language model personalization for response prediction with collaborative data refinement. In *Proc. of Int. Conf. on Computational Linguistics*, pages 281–296, Abu Dhabi, UAE. Association for Computational Linguistics.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024a. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Proc. of Conf. on Empirical Methods in Natural Language Processing*, pages 6459–6475, Miami, Florida, USA. 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 *Proc. of Conf. on Empirical Methods in Natural Language Processing*, pages 6476–6491, Miami, Florida, USA. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 2025. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. In *Findings of the Association for Computational Linguistics: NAACL*, pages 4823–4836, Albuquerque, New Mexico. Association for Computational Linguistics.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. In *Proc. of Conf. on Research and Development in Information Retrieval*, page 641–649, New York, NY, USA. Association for Computing Machinery.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025a. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Yijing Zhang, Dyah Adila, Changho Shin, and Frederic Sala. 2025b. Personalize your LLM: Fake it then align it. In *Findings of the Association for Computational Linguistics: NAACL*, pages 7287–7301, Albuquerque, New Mexico. Association for Computational Linguistics.

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

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. 2025a. Do LLMs recognize your preferences? evaluating personalized preference following in LLMs. In *Proc. of Int. Conf. on Learning Representations*.

Xiaoyan Zhao, Juntao You, Yang Zhang, Wenjie Wang, Hong Cheng, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2025b. Nextquill: Causal preference modeling for enhancing llm personalization. *arXiv preprint arXiv:2506.02368*.

Ziyu Zhao, Leilei Gan, Guoyin Wang, Wangchunshu Zhou, Hongxia Yang, Kun Kuang, and Fei Wu. 2024. LoraRetriever: Input-aware LoRA retrieval and composition for mixed tasks in the wild. In *Proc. of Annual Meeting of the Association for Computational Linguistics*, pages 4447–4462, Bangkok, Thailand. Association for Computational Linguistics.

## A Ethical Considerations

In our experiments, we use the LaMP (Salemi et al., 2024b) and LongLaMP (Kumar et al., 2024) benchmark, which is publicly available and does not raise direct privacy concerns. Nevertheless, personalized text generation inherently depends on user-specific historical data, which may involve sensitive information. To mitigate privacy risks, PerX is designed so that personalization can be achieved through compact LoRA modules derived from user embeddings, without requiring direct sharing of raw user data. Another concern lies in potential biases embedded in user histories, which could lead to unfair outputs. While PerX provides a flexible mechanism to adapt to individual preferences, future work should consider explicit bias detection and debiasing strategies to ensure equitable personalization. Finally, as with other personalization techniques, misuse for generating harmful or manipulative content remains a risk, underscoring the importance of responsible deployment and ongoing monitoring.

## B Baselines

This section provides a brief overview of the non-retrieval and retrieval-based baselines included in our comparisons.

**Non-Retrieval baselines** include:

- **No Personalization** directly feeds the input query to the LLM in a zero-shot manner, with the outputs serving as naive results.
- **PEFT (LoRA)** (Hu et al., 2022) trains a single task-specific LoRA using only input-output pair, without leveraging user history.
- **PPlug** (Liu et al., 2025) learns an encoder to aggregate historical data into a persona embedding, which is then injected into the LLM.

**Retrieval-based baselines** include:

- **RAG** (Lewis et al., 2020) retrieves the most relevant historical data to augment the input prompt, which is then processed by the LLM.
- **ROPG** (Salemi et al., 2024a) optimizes the retrieval model to improve retrieval quality, followed by the same strategy as **RAG**.
- **PAG** (Richardson et al., 2023) summarizes the user profile into a textual description, then augments the input prompt for generation.
- **RAG+PEFT (LoRA)** (Hu et al., 2022) augments the input prompt via RAG and trains LoRA as in **PEFT (LoRA)**.
- **RAG+PPlug** (Liu et al., 2025) augments the input prompt via RAG and performs training and inference following **PPlug**.

Implementation details of these baselines, along with our approach, are provided in the next section.

## C Implementation Details

To ensure reproducibility, we provide detailed implementation settings for both our proposed PerX and the baseline methods in this section.

For the proposed approach, we implemented it on both encoder-decoder and decoder-only architectures, using Flan-T5-XXL (11B) (Chung et al., 2022) and Qwen2.5-7B-Instruct (Qwen et al., 2024) as the default backbone LLM, respectively. In the manuscript, all main experiments are conducted using the encoder-decoder model (Flan-T5), while additional results for the decoder-only model (Qwen2.5) are reported in Appendix E. For encoding user histories, we use BGE-base-en-v1.5 (Xiao et al., 2024) as the text embedding model by default, consistent with previous studies (Liu et al., 2025). The maximum input length is set to 512 tokens for both the LLM and the text embedding model. For the user profile encoder, we use a 3-layer QFormer with an embedding dimension of 768. The temperature $\tau$ in the InfoNCE loss is fixed at 0.1. We cap the maximum history length at 512, meaning that at most 512 historical records are considered when constructing a user embedding. The rank of both LoRA and the low-rank MLP is set to 16 by default. We train our user profile encoder for 1000 steps, and the PerX model for 2 epochs across all tasks. The batch size is set to 1024 for the user profile encoder and 64 for PerX. Both models are optimized using AdamW (Loshchilov and Hutter, 2019). The learning rates are set to $5e$-4 and $5e$-5, respectively. During generation, we employ beam search (Freitag and Al-Onaizan, 2017) with a beam size of 4. All experiments are conducted on a machine equipped with 8 NVIDIA H800 GPUs. In addition, we also extend PerX with RAG to augment the input prompt, and then fed into the LLM for training and inference.

For the PEFT (LoRA) baseline, we align the LoRA rank with PerX, setting to 16. In retrieval-based approaches, we employ Contriever (Izacard et al., 2022) as the retriever and augment the inputs with the top-4 most relevant histories. All remaining baselines are evaluated using their respective official configurations.

Table 3: Overview of prompt details, including input structure, output format, and user history, for the six LaMP benchmark tasks. Text shown in blue represents placeholders that will be replaced with actual data during training and inference.

| Task | Input | RAG-Enhanced Context | 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] {reference 1} [2] {reference 2} | {title} and {title} and ... are written by the author before. | [2] | abstract: {abstract}<br>title: {title} |
| 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, dark comedy, classic, psychology, fantasy, romance, thought-provoking, social commentary, violence, true story] description: {movie} | The tag for the movie: {movie} is {tag}, and the tag for the movie: {movie} is {tag}, ... | sci-fi | description: {movie}<br>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} | {score} is the score for {review}, and {score} is the score for {review}, ... | 5 | text: {review}<br>score: {score} |
| LaMP-4 | Generate a headline for the following article: {article} | {title} is the title of article: {article}, and {title} is the title for {article}, ... | Finding Happiness After Divorce – It Can Happen | text: {article}<br>title: {title} |
| LaMP-5 | Generate a title for the following abstract of a paper: {abstract} | {title} is the title for abstract {abstract}, and {title} is the title for abstract, ... | Link-Reliability Based Two-Hop Routing for Wireless Sensor Networks. | text: {abstract}<br>title: {title} |
| LaMP-7 | Paraphrase the following tweet without any explanation before or after it: {tweet} | {tweet} and {tweet} and ... are written by a person. Following the given patterns, | @Blitzscream Aaawww... *hugges* If he doesn't bring Silverbolt back, he shall suffer my righteous wrath! >:-[ ... er somthin'! :p | text: {tweet} |
| LaMP-7 (Aug.) | Generate a tweet: | {tweet} and {tweet} and ... are written by a person. Following the given patterns, | @Blitzscream Aaawww... *hugges* If he doesn't bring Silverbolt back, he shall suffer my righteous wrath! >:-[ ... er somthin'! :p | text: {tweet} |

## D Prompt Details

To further enhance the reproducibility, we provide the detailed prompt templates used for LLMs in our experiments.

**LaMP.** Table 3 presents an overview of the prompts, which largely follow the design of the LaMP (Salemi et al., 2024b) benchmark. Table 3 outlines the structure of input prompts, output for-

Table 4: Overview of prompt details, including input structure, output format, and user history, for the two selected tasks from LongLaMP benchmark. Text shown in blue represents placeholders that will be replaced with actual data during training and inference.

| Task | Input | Output | User History |
|---|---|---|---|
| Personalized Review Writing | Generate the review text written by a reviewer who has a given an overall rating of "{score}" for a product with description "{description}". The summary of the review text is "{summary}". | Some days you simply need a feel good story. A story that will restore your faith in the universe. And that is the key The First Phone Call From Heaven - Faith. The first phone call is received by Tess Rafferty who happens to live in the small town of Coldwater, Michigan. It is from her deceased mother; and the call's message brings much comfort to Tess. Shortly thereafter, a select few other residents of the town receive calls of their own from a deceased relative, friend, or business associate. Are these calls real? Can they be real? Is there truly a hot-line to heaven? As news of the calls spreads the local police as well as all denominations of the local clergy find themselves thrust into the center of attention. On the other side of this reality is Sullivan Harding. His story is one of despair and he tries to shield his young son from believing that his deceased mother will call him on a toy phone. Sullivan sets out to prove that this entire scenario is a hoax. But with the town becoming overrun with believers, news reporters, and protesters he has a tough time. This story is enchanting, humorous, serious, and believable. The characters are finely drawn and soon become old friends. Overall an easy read and an uplifting one. The story itself will have you second guessing the verity of the phone calls but the ending will have you believing that those on the 'other side' are listening, if not communicating. | review: {review}<br>score: {score}<br>description: {description}<br>summary: {summary} |
| Personalized Topic Writing | Generate the content for a reddit post "{post}". | I'd definitely say the defense in general. Look at the Patriots- their three Super Bowls were won when they had a good defense at the beginning of the 2000s. Or the Packers- their defense played their best football in the 2010 playoffs, and has been pretty bad since then. Not coincidentally, that is the lone Super Bowl that Aaron Rodgers won, and Rodgers only has one other playoff win (and that came when Joe Webb was helping out their defense). | content: {content}<br>post: {post} |

mats, and user history representations. In addition, we include the construction of prompts used for RAG-enhanced inputs, where the retrieved content is prepended to the original input prompt.

In practice, the placeholders highlighted in blue are replaced with real data items during both training and inference. While our method employs data augmentation during training, it is not uniformly applicable across all LaMP tasks due to format inconsistencies between the input-output pairs and the user histories. Specifically, LaMP-1 cannot be augmented under our problem formulation, so no data augmentation is applied to it in any experiment. For LaMP-7, although the history and input-output formats are misaligned, we convert the original paraphrasing task into a generation task to enable augmentation. The last row in Table 3 illustrates the prompt used in this setting. Notably, during inference, we revert to the original paraphrasing prompt to maintain consistency with the benchmark.

**LongLaMP.** Table 4 presents prompt examples that follow the same design as the LongLaMP (Kumar et al., 2024) benchmark. The placeholders highlighted in blue are replaced following the same settings as those used in the LaMP benchmark above.

## E Additional Results

To further assess the generality of PerX, we extend our experiments to decoder-only LLMs and alternative text embedding models.

We first evaluate PerX on the decoder-only model, Qwen2.5-7B-Instruct (Qwen et al., 2024), following the same experimental configurations as in the encoder-decoder setting, except that the maximum input length is set to 8192 tokens during training. Table 5 reports the results on the LongLaMP (Kumar et al., 2024) benchmark across two selected tasks. PerX consistently outperforms the baselines across both LongLaMP tasks under most evaluation metrics, achieving relative improvements of 1.4%-45.2% over the PEFT (LoRA) baseline, which demonstrates its robust and metric-agnostic advantages.

In addition, we apply PerX with the decoder-only architecture to the LaMP (Salemi et al., 2024b) benchmark, with results reported in Table 6. The results show that PerX consistently outperforms most baselines and achieves performance comparable to that obtained with the larger encoder-decoder back-

Table 5: **Comparison of PerX with other methods on the LongLaMP benchmark.** The best method is highlighted in bold.

| Methods (Qwen2.5-7B-Instruct) | Personalized Review Writing | | | | Personalized Topic Writing | | | |
|---|---|---|---|---|---|---|---|---|
| | BLEU ↑ | Rouge-1 ↑ | Rouge-L ↑ | Meteor ↑ | BLEU ↑ | Rouge-1 ↑ | Rouge-L ↑ | Meteor ↑ |
| No Personalization | 2.449 | 0.302 | 0.135 | 0.185 | 1.566 | 0.261 | 0.117 | 0.171 |
| PEFT (LoRA) (Hu et al., 2022) | 2.457 | 0.335 | 0.139 | 0.187 | 1.136 | 0.263 | 0.122 | **0.172** |
| **PerX (Ours)** | **2.492** | **0.335** | **0.152** | **0.199** | **1.650** | **0.274** | **0.134** | 0.167 |

Table 6: **Summarized results of PerX and baseline methods on the LaMP benchmark across different backbone LLMs.** ↑ indicates that a higher value for the corresponding metric is better, while ↓ indicates that lower is better. The best and the second-best methods are highlighted in bold and underlined fonts, respectively. *: For LaMP-1, PerX was trained without data augmentation.

| LLM | Methods | LaMP-1 | LaMP-2 | | LaMP-3 | | LaMP-4 | | LaMP-5 | | LaMP-7 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Accuracy ↑ | Accuracy ↑ | F1 ↑ | MAE ↓ | RMSE ↓ | Rouge-1 ↑ | Rouge-L ↑ | Rouge-1 ↑ | Rouge-L ↑ | Rouge-1 ↑ | Rouge-L ↑ |
| Flan-T5 XXL (11B) | No Personalization | 0.496 | 0.326 | 0.258 | 0.317 | 0.604 | 0.176 | 0.161 | 0.482 | 0.431 | 0.456 | 0.403 |
| | RAG (Lewis et al., 2020) | 0.641 | 0.363 | 0.284 | 0.294 | 0.587 | 0.190 | 0.174 | 0.472 | 0.423 | 0.442 | 0.393 |
| | PEFT (LoRA) (Hu et al., 2022) | 0.658 | 0.549 | 0.485 | 0.226 | 0.523 | 0.206 | 0.186 | 0.493 | 0.441 | 0.532 | 0.479 |
| | **PerX (Ours)** | 0.645* | 0.679 | 0.640 | 0.222 | 0.508 | 0.199 | 0.183 | 0.501 | 0.450 | 0.538 | 0.486 |
| | **RAG+PerX (Ours)** | **0.755*** | **0.686** | **0.654** | **0.205** | **0.484** | **0.222** | **0.206** | **0.511** | **0.460** | **0.554** | **0.505** |
| Qwen2.5 (7B) | No Personalization | 0.484 | 0.299 | 0.256 | 0.650 | 1.072 | 0.138 | 0.123 | 0.429 | 0.360 | 0.371 | 0.323 |
| | RAG (Lewis et al., 2020) | 0.682 | 0.413 | 0.348 | 0.496 | 0.967 | 0.165 | 0.149 | 0.450 | 0.386 | 0.411 | 0.358 |
| | PEFT (LoRA) (Hu et al., 2022) | 0.604 | 0.480 | 0.441 | 0.270 | 0.580 | 0.176 | 0.158 | 0.461 | 0.408 | 0.480 | 0.425 |
| | **PerX (Ours)** | 0.648* | 0.680 | 0.646 | 0.227 | 0.531 | 0.174 | 0.157 | 0.497 | 0.449 | 0.541 | 0.487 |
| | **RAG+PerX (Ours)** | **0.782*** | **0.689** | **0.655** | **0.206** | **0.498** | **0.200** | **0.183** | **0.509** | **0.460** | **0.551** | **0.498** |

Table 7: **Performance of PerX with different text embedding models on the LaMP benchmark across different backbone LLMs.** ↑ indicates that a higher value for the corresponding metric is better, while ↓ indicates that lower is better. The best method is highlighted in bold. *: For LaMP-1, PerX was trained without data augmentation.

| LLM | Embedding model | LaMP-1 | LaMP-2 | | LaMP-3 | | LaMP-4 | | LaMP-5 | | LaMP-7 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Accuracy ↑ | Accuracy ↑ | F1 ↑ | MAE ↓ | RMSE ↓ | Rouge-1 ↑ | Rouge-L ↑ | Rouge-1 ↑ | Rouge-L ↑ | Rouge-1 ↑ | Rouge-L ↑ |
| Flan-T5 XXL (11B) | BGE | 0.645* | **0.679** | **0.640** | 0.222 | 0.508 | 0.199 | 0.183 | **0.501** | **0.450** | **0.538** | 0.486 |
| | Contriever | 0.641* | 0.673 | 0.637 | 0.224 | 0.510 | 0.202 | 0.185 | 0.499 | 0.447 | 0.537 | 0.484 |
| | Qwen3-Embedding | **0.649*** | 0.663 | 0.625 | **0.219** | **0.507** | **0.205** | **0.187** | 0.496 | 0.445 | 0.537 | **0.487** |
| Qwen2.5 7B | BGE | 0.648* | **0.680** | **0.646** | **0.227** | 0.531 | 0.174 | 0.158 | **0.497** | **0.449** | **0.541** | **0.487** |
| | Contriever | 0.639* | 0.665 | 0.632 | 0.229 | **0.528** | **0.176** | **0.161** | 0.496 | 0.448 | 0.535 | 0.481 |
| | Qwen3-Embedding | **0.649*** | 0.679 | 0.643 | 0.232 | 0.536 | 0.173 | 0.156 | 0.496 | 0.448 | 0.537 | 0.484 |

bone (Flan-T5-XXL (Chung et al., 2022), 11B). The results also confirm that the key findings in the manuscript hold across both encoder-decoder and decoder-only architectures, demonstrating the robustness and generality of PerX.

Furthermore, we explore the impact of different text embedding models for encoding user histories. In addition to BGE (Xiao et al., 2024) used in our main setting, we experiment with Contriever (Izacard et al., 2022) and Qwen3-Embedding (Zhang et al., 2025a) (0.6B). The comparative results across the two LLM backbones on the LaMP benchmark are summarized in Table 7, illustrating that PerX maintains strong performance regardless of the underlying embedding model.

## F Case Study

To further qualitatively evaluate the performance of different methods, we conduct a case study on two generation tasks from the LaMP benchmark: LaMP-5 and LaMP-7. Table 8 presents several representative examples, showing the input, ground truth, and outputs of three baselines, our PerX, and RAG+PerX. Due to space constraints, we only include three cases and omit the user profiles. As shown in Table 8, PerX generates outputs that closely align with the user-specific style, maintaining consistency in word choice, tone, and overall writing style compared to the baselines. By integrating RAG, RAG+PerX further enhances personalization, producing responses that are even more faithful to the user's linguistic style and context. For instance, in LaMP-5, our methods generate titles beginning with the method name "PBMarsNet," closely following the style of the ground truth, whereas baseline outputs often omit it. Similarly, in LaMP-7, PerX and RAG+PerX produce paraphrases with word choice and tone more consistent with the original tweets, demonstrating stronger preservation of the user-specific style.

Table 8: Case study on LaMP-5 and LaMP-7. We show the input, ground truth, and generated results from five methods. The user profile is omitted due to space constraints.

| | | | |
|---|---|---|---|
| LaMP-5 (case i) | Input | Generate a title for the following abstract of a paper: Gene Regulatory Network (GRN) is a directed graph which describes the regulations between genes. The problem of reconstructing GRNs has been studied for decades. Most of existing methods infer the GRNs from gene expression data. Previous studies use random forest, partial least squares or other feature selection techniques to solve it. In this paper, we propose a Multivariate Adaptive Regression Splines (MARS) based method to estimate the feature importance and reconstruct the GRNs. MARS can catch the nonlinear relationships between genes. To avoid the overfitting and make the estimation robust, we apply an ensemble model of MARS based on bootstrap and weighted features by PMI (Part mutual information), called PBMarsNet. The results show that PBMarsNet performs better than the state-of-the-art methods. | |
| | Ground Truth | PBMarsNet: A Multivariate Adaptive Regression Splines Based Method to Reconstruct Gene Regulatory Networks. | |
| | Method | No Personalization | Multivariate Adaptive Regression Splines Based Gene Regulatory Network Reconstruction |
| | | PEFT (LoRA) | Robust Gene Regulatory Network Reconstruction Using Multivariate Adaptive Regression Splines |
| | | PPlug | Multivariate Adaptive Regression Splines-based Gene Regulatory Network Reconstruction. |
| | | **PerX (Ours)** | PBMarsNet: A Multivariate Adaptive Regression Splines Based Method for Reconstructing Gene Regulatory Networks. |
| | | **RAG+PerX (Ours)** | PBMarsNet: a robust multivariate adaptive regression splines based method for gene regulatory network reconstruction. |
| LaMP-7 (case i) | Input | Paraphrase the following tweet without any explanation before or after it: Congratulations @JaviLovesPizza :/ Perhaps you tweet more frequently than I do… | |
| | Ground Truth | @JaviLovesPizza congrats :/ Maybe you do tweet more than me... | |
| | Method | No Personalization | Congratulations @JaviLovesPizza :/ Perhaps you tweet more frequently than I do... |
| | | PEFT (LoRA) | @JaviLovesPizza Congratulations :/ Maybe you tweet more than I do... |
| | | PPlug | Congratulations @JaviLovesPizza :/ Maybe you tweet more than I do... |
| | | **PerX (Ours)** | @JaviLovesPizza congrats :/ maybe you tweet more than me... |
| | | **RAG+PerX (Ours)** | @JaviLovesPizza congrats :/ Maybe you tweet more than I do... |
| LaMP-7 (case ii) | Input | Paraphrase the following tweet without any explanation before or after it: Nice to meet you! My username is FoxySmile and I'm in Danny's IDF thread. Hello @Jessie0309. | |
| | Ground Truth | @Jessie0309 Hi, I'm FoxySmile in Danny's IDF thread Nice to meet you! | |
| | Method | No Personalization | Nice to meet you! My username is FoxySmile and I'm in Danny's IDF thread. Hello @Jessie0309 |
| | | PEFT (LoRA) | @Jessie0309 Hi! I'm FoxySmile and I'm in Danny's IDF thread. Nice to meet you! |
| | | PPlug | @Jessie0309 Hi! I'm FoxySmile. I'm in Danny's IDF thread. Nice to meet you! |
| | | **PerX (Ours)** | @Jessie0309 Hi! I'm FoxySmile and I'm in Danny's IDF thread. Nice to meet you! |
| | | **RAG+PerX (Ours)** | @Jessie0309 Hi, I'm FoxySmile in Danny's IDF thread. Nice to meet you! |