Personalized Language Models via Privacy-Preserving Evolutionary Model Merging

Kyuyoung Kim[1], **Jinwoo Shin**[1], **Jaehyung Kim**[2]
[1]KAIST AI, [2]Yonsei University
{kykim,jinwoos}@kaist.ac.kr, jaehyungk@yonsei.ac.kr

Abstract

Personalization in language models aims to tailor model behavior to individual users or user groups. Prompt-based methods incorporate user preferences into queries, while training-based methods encode them into model parameters. Model merging has also been explored for personalization under limited data. However, existing methods often fail to directly optimize task-specific utility and lack explicit mechanisms for privacy preservation. To address the limitations, we propose Privacy-Preserving Model Merging via Evolutionary Algorithms (PriME), a novel personalization approach that employs gradient-free methods to directly optimize utility while reducing privacy risks. By integrating privacy preservation into the optimization objective, PriME creates personalized modules that effectively capture target user preferences while minimizing privacy risks for data-sharing users. Experiments on the LaMP benchmark show that PriME consistently outperforms a range of baselines, achieving up to a 45% improvement in task performance. Further analysis demonstrates that PriME achieves a superior privacy-utility trade-off compared to a prior state-of-the-art, with enhanced robustness to membership inference attacks and greater utility in capturing user preferences.

1 Introduction

Pre-trained on web-scale data, large language models (LLMs) have emerged as powerful tools (Radford et al., 2018; Brown et al., 2020), capable of performing a wide range of natural language processing (NLP) tasks, from conversational agents, translation, and question-answering to code generation (Achiam et al., 2023; Team et al., 2024). Notably, LLMs have demonstrated remarkable effectiveness as generalist models, able to perform complex tasks with little to no task-specific data (Radford et al., 2019). To extend their utility beyond general-purpose applications, LLM personalization seeks to tailor model responses to the preferences of individual users or user groups (Salemi et al., 2024). Personalized LLMs are particularly important in domains such as education (Kasneci et al., 2023), healthcare (Liu et al., 2023), and content recommendation (Baek et al., 2024) to enhance performance and user experience.

Previous approaches to LLM personalization can be categorized as methods based on prompt augmentation, training, and, more recently, model merging. Prompt-based methods incorporate user preferences or relevant historical data into input prompts, enabling models to better understand and respond to user-specific queries (Mysore et al., 2024; Richardson et al., 2023; Kim and Yang, 2025). While these methods are relatively simple to implement and avoid additional training, the performance is inherently constrained by the model's context length and may result in significantly higher inference costs due to enlarged input prompts.

In contrast, training-based approaches aim to capture user preferences directly in model parameters. One straightforward method is to train a parameter-efficient fine-tuning (PEFT) module for each user on the user's private data (Tan et al., 2024b). However, when data is limited, these methods may struggle to achieve effective personalization. Moreover, individually owned modules limits its opportunities for collaborative benefits through shared adaptation. To address this, several methods leverage data from multiple users or groups to model group-level preferences (Li et al., 2024; Zhao et al., 2024). However, direct access to raw data from multiple users or groups is rare in practice and raises substantial privacy concerns, especially in the absence of explicit safeguards. Model merging has also been explored for personalization under limited data (Tan et al., 2024a), but existing approaches do not directly optimize for task-specific utility or privacy.

To address the challenges, we propose **Pri**vacy-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 34442–34457
November 4-9, 2025

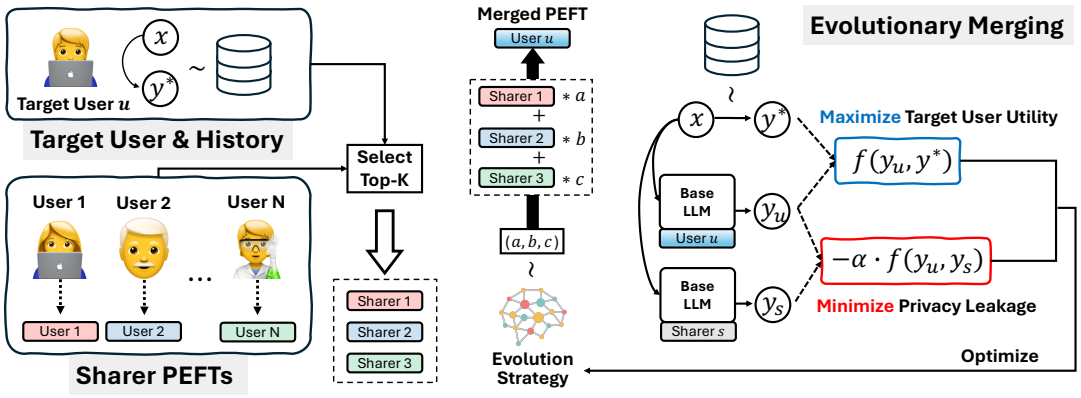


Figure 1: **Overview of PriME.** PriME is an evolutionary method for LLM personalization that optimizes model merging to create a personalized, parameter-efficient fine-tuning module. Using gradient-free optimization, PriME directly optimizes utility metrics with a privacy-preserving mechanism to achieve a superior privacy-utility trade-off.

Preserving **M**odel Merging via **E**volutionary Algorithms (PriME), a novel LLM personalization approach that leverages evolutionary methods to utilize knowledge from a community of data-sharing users while minimizing the risk of privacy leakage. Specifically, given a set of users who have consented to share their PEFT modules, PriME employs evolutionary algorithms to find optimal weights for model merging (Wortsman et al., 2022), creating a personalized module that captures the preferences of a target user. As each shared module is trained on private data, the merged module poses a potential risk of privacy leakage for sharing users, e.g., through membership inference attacks (MIA), a widely used approach for assessing practical privacy leakage (Shokri et al., 2017). To this end, we incorporate privacy preservation into the merging process by optimizing similarity with the target user while minimizing the average similarity with the sharing users. The gradient-free nature of evolutionary algorithms enables direct optimization of non-differentiable metrics, such as ROUGE, used to measure the prediction similarity. This allows us to achieve a superior privacy-utility trade-off between capturing the target user's preferences and minimizing privacy risks for sharing users.

Experimental results on LaMP (Salemi et al., 2024), a widely used personalization benchmark consisting of various text classification and generation tasks, demonstrate that PriME consistently outperforms both prompt- and training-based baselines, achieving up to a 45% improvement in performance. Our findings show that PriME produces personalized modules that not only better capture user preferences but also exhibit greater robustness to MIA risks compared to a prior state-of-the-art merging method. Qualitatively, we observe that with an appropriate choice of the parameter controlling the privacy-utility trade-off, the modules produced by PriME yield predictions that both better align with the target user's behavior patterns and are more distinct from those of the sharing users. Notably, this is achieved without introducing additional complexity to the model architecture or training, unlike prior approaches (Tan et al., 2024a). These results highlight the potential of evolutionary methods for achieving effective privacy-preserving LLM personalization.

Our main contributions are as follows:

- We introduce PriME, an evolutionary approach that automates the discovery of optimal model merging recipes for privacy-preserving LLM personalization.
- To our knowledge, we are the first to examine privacy leakage via MIA in the context of merging-based LLM personalization, and to demonstrate that gradient-free approaches can effectively mitigate these risks.
- We demonstrate that PriME outperforms both prompt- and training-based baselines on the LaMP benchmark, achieving superior privacy-utility trade-offs by producing personalized modules that better capture user preferences and exhibit enhanced robustness to MIA.

2 Preliminaries

2.1 LLM personalization

Personalizing LLMs involves tailoring model responses to individual users or groups based on their historical behavior data. Specifically, given an LLM $\mathcal{M}$ and history data $\mathcal{H}_u = \{h_u\}$ for user

u, we aim to generate an output y that is closely aligned with the user's preferences conditioned on both the input x and $\mathcal{H}_u$:

$$\max_{y \sim \mathcal{M}(\cdot|x,\mathcal{H}_u)} f(y, y^*), \tag{1}$$

where f is a target utility metric, such as ROUGE, that measures the similarity between the generated response y and the user-preferred ground-truth response y^* to x. Each history item $h_u \in \mathcal{H}_u$ may consist of either a (x_u, y_u) pair in a task-specific query-response format, capturing how the user has historically responded to similar queries, or a text sequence that provides contextual information about the user's behavior patterns.

2.2 Model merging for personalization

Given a group of sharing users $\mathcal{S}$, each with a personalized PEFT ΔW_s trained on their private data, our goal is to create a module ΔW_u for a target user u that aligns with the user's historical data $\mathcal{H}_u$ by leveraging the shared modules. To achieve this, we (1) identify a subset of users $\bar{\mathcal{S}} \subset \mathcal{S}$ that are most relevant to the target user, along with the corresponding modules $\{\Delta W_s \mid s \in \bar{\mathcal{S}}\}$, and (2) determine the optimal weights $\{w_s \mid s \in \bar{\mathcal{S}}\}$ to linearly interpolate the selected modules. Using the interpolation weights, we construct ΔW_u as

$$\Delta W_u = \sum_{s \in \bar{\mathcal{S}}} w_s \Delta W_s. \tag{2}$$

If the weights are scalars, merging is performed at the module level, whereas if they are vectors, it occurs at a more granular level, such as per-layer LoRA weights. Once we obtain a module ΔW_u for user u, we condition the LLM $\mathcal{M}$ on ΔW_u to generate personalized responses:

$$\mathcal{M}_u(\cdot) = \mathcal{M}(\cdot \mid \Delta W_u), \tag{3}$$

where $\mathcal{M}_u$ denotes the LLM with the personalized PEFT module ΔW_u integrated into $\mathcal{M}$. For the remainder, we present our method using LoRA (Hu et al., 2022) given its widespread adoption.

2.3 Privacy risks in LLMs

LLMs show strong performance across a wide range of tasks, but their reliance on large-scale data in various stages of training and inference introduces significant privacy risks. For example, LLMs have been shown to memorize portions of their training data, often reproducing the data verbatim when prompted appropriately (Carlini et al., 2023). Prior work has studied such privacy risks via methods such as MIAs in the contexts of pre-training (Duan et al., 2024b), fine-tuning (Mireshghallah et al., 2022), and prompt-based adaptation (Duan et al., 2024a). As personalization involves sensitive user data, evaluating and mitigating such risks becomes even more crucial. To our knowledge, we are the first to provide empirical assessments of privacy risks in merging-based personalization and to propose a method that explicitly addresses the challenges.

3 Privacy-Preserving Evolutionary LLM Personalization

In this section, we present PriME, a novel approach for achieving privacy-preserving LLM personalization via evolutionary model merging.

3.1 Personalization via model merging

User embedding and selection. Selecting only the most relevant LoRA modules for the target user is important for privacy, as it limits the number of modules trained on private data involved in merging. To measure user similarity, we adopt a method from prior work (Tan et al., 2024a), where an encoder-only language model $\mathcal{E}$ encodes each history item $h_u \in \mathcal{H}_u$, and the user embedding is given by the average of the encoded items:

$$\mathbf{e}_u = \sum_{h_u \in \mathcal{H}_u} \mathcal{E}(h_u)/|\mathcal{H}_u|. \tag{4}$$

Given the user embedding $\mathbf{e}_u$ for the target user u and $\mathbf{e}_s$ for each sharing user $s \in \mathcal{S}$, computed as in Eq. 4, we compute the cosine similarity to select the top-k most relevant sharing users:

$$\bar{\mathcal{S}} = \left\{ s \in \mathcal{S} \mid \cos(\mathbf{e}_u, \mathbf{e}_s) \geq \text{top-}k_{s \in \mathcal{S}}(\cos(\mathbf{e}_u, \mathbf{e}_s)) \right\}. \tag{5}$$

We empirically find that selecting only a few LoRA modules via this approach often suffices for effective personalization (see Section 4.2).

Evolutionary merging for personalization. Inspired by the success of evolutionary approaches in developing strong foundation models via model merging (Akiba et al., 2025), we extend the idea to automatically discover merging recipes for creating personalized LoRA modules. Specifically, we employ an evolutionary algorithm, such as CMA-ES (Hansen, 2006), to find the optimal weights for linearly interpolating a subset of LoRAs $\{\Delta W_s \mid$

$s \in \bar{\mathcal{S}}\}$, selected according to Eq. 5. The algorithm searches for optimal interpolation weights over a fixed number of iterations based on how well the resulting merged module aligns with the preferences of the target user u:

$$\underset{\{w_s|s\in\bar{\mathcal{S}}\}}{\arg\max}\ \mathbb{E}_{y\sim\mathcal{M}_u}\left[f(y, y^*)\right], \tag{6}$$

where $\mathcal{M}$ is an LLM, and f is a target utility metric that measures the similarity between the response y, generated by the LLM conditioned on ΔW_u, and the user-preferred response y^*. As evolutionary algorithms are gradient-free, one can directly optimize f, which is often non-differentiable. Notably, f can also be a suitable combination of multiple utility metrics relevant to the task.

3.2 Threat model and membership inference

Constructing a personalized module from a set of LoRAs raises concerns for the sharing users due to potential privacy leakage, e.g., via membership inference attacks (MIA) (Shokri et al., 2017; Yeom et al., 2018). Specifically, we consider an adversary with black-box access to the personalized LLM $\mathcal{M}_u$, conditioned on the module ΔW_u, aiming to infer whether a particular data point is a member of the training data for a sharing user involved in the merging. The adversary can query $\mathcal{M}_u$ with a set of task-specific input-response pairs (x_u, y_u) and obtain the model's loss on each y_u. Following prior MIA approaches, we use the loss as a score, which is thresholded to infer the membership of a data point (Yeom et al., 2018; Duan et al., 2024b).

3.3 Optimizing privacy preservation

Privacy measure. To mitigate the MIA risks, we propose not only maximizing alignment with the target user's preferences but also minimizing similarity with the sharing users selected for merging. Specifically, for a sharing user s, we measure this similarity using the utility metric f, where predictions from the sharing user's LoRA ΔW_s serve as reference labels and those from the merged LoRA ΔW_u as regular predictions. Assuming access only to the target user's data, we compute the similarity over the target user's history $\mathcal{H}_u$ as follows:

$$P_u(s) = \sum_{h_u\in\mathcal{H}_u} f(\mathcal{M}_u(h_u), \mathcal{M}_s(h_u))/|\mathcal{H}_u|. \tag{7}$$

Intuitively, this evaluates the extent to which information about the sharing users can potentially be inferred from the merged LoRA ΔW_u.

Algorithm 1 PriME algorithm

Input: LLM $\mathcal{M}$, target user u, target user history $\mathcal{H}_u$, target user embedding $\mathbf{e}_u$, sharing users $\mathcal{S}$, sharing user embeddings $\{\mathbf{e}_s\}$, shared LoRAs $\{\Delta W_s\}$, target utility metric f, privacy coefficient α, optimization budget b

```
/* Select similar sharing users */
S̄ ← {s ∈ S | cos(e_u, e_s) ≥ top-k_{s∈S}(cos(e_u, e_s))}
/* Evolve interpolation weights */
solver ← EvolutionStrategy()
for t = 0 to b − 1 do
  ŵ ← solver.ask()
  /* Adapt with proposed weights */
  ΔŴ_u ← Σ_{s∈S̄} ŵ(s) ΔW_s
  M̂_u(·) ← M(· | ΔŴ_u)
  /* Evaluate candidate M̂_u */
  U_u ← Σ_{h_u∈H_u} f(M̂_u(h_u), y*(h_u))/|H_u|
  P_u ← {Σ_{h_u∈H_u} f(M̂_u(h_u), M_s(h_u))/|H_u|}
  r ← U_u − α · Σ_{s∈S̄} P_u(s)/|S̄|
  /* Update optimal weights */
  solver.tell(r)
  w ← solver.results()
end for
/* Create M_u with optimal weights */
ΔW_u ← Σ_{s∈S̄} w(s) ΔW_s
M_u(·) ← M(· | ΔW_u)
return personalized LLM M_u for user u
```

Privacy-aware optimization. LLM personalization can be viewed as a multi-objective optimization, where we maximize alignment with the target user's preferences while minimizing the privacy risks associated with the shared components. To this end, we optimize the following objective, leveraging the privacy measure defined in Eq. 7:

$$\underset{\{w_s|s\in\bar{\mathcal{S}}\}}{\arg\max}\ \mathbb{E}_{y\sim\mathcal{M}_u}\left[f(y, y^*)\right] - \alpha\cdot\sum_{s\in\bar{\mathcal{S}}}\frac{P_u(s)}{|\bar{\mathcal{S}}|}, \tag{8}$$

where α controls the trade-off between capturing the target user's preferences and minimizing the similarity with the sharing users. The use of f in the privacy measure allows us to account for task-specific prediction similarities and to naturally define an objective that balances privacy and utility. We empirically find that, for a suitable α, the resulting module yields predictions that are more distinct from those of sharing users, with greater robustness to MIA compared to competing methods. Notably, this is achieved while significantly better capturing the target user's preferences.

4 Experiments

We evaluate PriME in terms of (a) its effectiveness in capturing target user preferences and (b) its ability to mitigate privacy risks for sharing users.

Task	Metric	NP	RAG	PAG		PER-PCS			PRIME (Ours)			OPPU		
		k=0	k=1	k=0	k=1	Base	+RAG	+PAG	Base	+RAG	+PAG	Base	+RAG	+PAG
LAMP-1: PERSONAL. CITATION IDENTIF.	Acc ↑	0.608	0.616	0.640	0.656	0.584	0.608	0.648	0.600	0.608	**0.680**	0.560	0.584	0.664
	F1 ↑	0.605	0.615	0.634	0.653	0.579	0.608	0.644	0.598	0.608	**0.675**	0.553	0.567	0.658
LAMP-2: PERSONAL. MOVIE TAGGING	Acc ↑	0.348	0.428	0.470	0.490	0.363	0.442	0.495	0.523	0.537	**0.551**	0.463	0.467	0.507
	F1 ↑	0.286	0.359	0.361	0.402	0.296	0.363	0.408	0.388	0.424	**0.442**	0.320	0.349	0.385
LAMP-3: PERSONAL. PRODUCT RATING	MAE ↓	0.291	0.252	0.274	0.270	0.503	0.408	0.408	0.272	**0.244**	0.256	0.262	0.272	0.266
	RMSE ↓	0.575	0.528	0.547	0.544	0.727	0.648	0.648	0.543	**0.510**	0.522	0.558	0.580	0.561
LAMP-4: PERSONAL. NEWS HEADLINE GEN.	R-1 ↑	0.186	0.202	0.194	0.207	0.179	0.192	0.195	0.198	0.208	**0.210**	0.193	0.205	0.209
	R-L ↑	0.168	0.183	0.176	0.188	0.162	0.174	0.177	0.179	0.189	**0.191**	0.173	0.185	0.190
LAMP-5: PERSONAL. SCHOLARLY TITLE GEN.	R-1 ↑	0.489	0.508	0.495	0.511	0.494	0.500	0.509	0.492	0.508	**0.512**	0.490	0.509	0.512
	R-L ↑	0.440	0.453	0.442	0.458	0.448	0.453	**0.460**	0.443	0.454	0.458	0.439	0.457	0.459
LAMP-7: PERSONAL. TWEET PARAPHRASING	R-1 ↑	0.527	0.553	0.536	0.559	0.507	0.557	0.554	0.520	**0.563**	0.554	0.529	0.559	0.561
	R-L ↑	0.480	0.511	0.485	0.516	0.460	0.511	0.511	0.469	**0.521**	0.512	0.480	0.515	0.519

Table 1: **Main results on LaMP.** R-1 and R-L denote ROUGE-1 and ROUGE-L, respectively. k is the number of retrieved items. For both Per-Pcs and PriME, k=1 is used for RAG. NP (Non-Personalized) refers to the performance of the task-adapted model. OPPU, training personalized PEFTs directly on user data, can be regarded as an upper bound in case of sufficient data. The highest score is shown in **bold**, and the second-highest is underlined.

4.1 Setup

Benchmark and tasks. For our main evaluation, we use LaMP (Salemi et al., 2024), a widely used benchmark of personalized text classification and generation tasks. Specifically, we focus on three classification tasks (LaMP-1, LaMP-2, and LaMP-3) and three generation tasks (LaMP-4, LaMP-5, and LaMP-7). We follow the approach from prior work for data splitting and selecting sharing users (Tan et al., 2024a). See Appendices A and B for details on the benchmark and setup.[1]

Baselines. We evaluate PriME along with several state-of-the-art baselines. For prompt-based methods, we consider retrieval-augmented generation (RAG) (Salemi et al., 2024), which augments queries with retrieved history items, and profile-augmented generation (PAG) (Richardson et al., 2023), which adds a textual summary of the user's profile to the prompt. We also assess Per-Pcs (Tan et al., 2024a), a competing merging-based method that creates personalized modules by autoregressively merging relevant sharing modules, with selection guided by *gating* vectors. We use Llama-2-7b (Touvron et al., 2023) as the base model and BM25 (Robertson et al., 2009) for retrieval unless stated otherwise.

Evaluation. For utility evaluation, we follow the standard protocol for the benchmark (Salemi et al., 2024), using accuracy and F1 score for text classification tasks (LaMP-1 and LaMP-2), mean absolute error (MAE) and root mean squared error (RMSE) for ordinal multi-class classification tasks (LaMP-3), and ROUGE-1 and ROUGE-L for generation tasks (LaMP-4, LaMP-5, and LaMP-7). Privacy risks are evaluated via MIA on the personalized modules produced by PriME and Per-Pcs, using AUC as the metric, with higher values indicating greater risk of membership inference.

[1]Code and implementation details are available at https://github.com/kykim0/PriME

4.2 Main results

Overall performance. Table 1 summarizes the results across six LaMP tasks, where PriME optimizes solely for alignment with target user preferences, i.e., α set to 0. In classification tasks, PriME achieves relative improvements in base performance over Per-Pcs of 2.7% in accuracy and 3.3% in F1 score on LaMP-1, and 44.1% and 31.1% on LaMP-2, respectively. For LaMP-3, despite extensive tuning, we observe significantly worse performance for Per-Pcs than the NP baseline, suggesting sensitivity to hyperparameters or limitations of its autoregressive merging. In contrast, PriME improves MAE and RMSE by 45.9% and 25.3%, respectively, surpassing also the originally reported Per-Pcs results (Tan et al., 2024a). In generation tasks, PriME achieves relative improvements of 10.5% in ROUGE-1 and 11.1% in ROUGE-L on LaMP-4, and 2.6% and 2.0% on LaMP-7.

PriME performs competitively with OPPU (Tan et al., 2024b), a simple approach that directly trains on target user data, often surpassing its performance. Note that, due to limited reproducibility and missing details, we include in Table 1 the results as originally reported. On LaMP-3, LaMP-4, and LaMP-5, PriME slightly outperforms OPPU in terms of base performance, while on LaMP-7, it marginally lags behind. More substantial im-

	LaMP-2			LaMP-4			LaMP-7		
	Acc ↑	F1 ↑	AUC ↓	R-1 ↑	R-L ↑	AUC ↓	R-1 ↑	R-L ↑	AUC ↓
Per-Pcs	0.369	0.310	0.523	0.182	0.165	0.514	0.518	0.467	0.502
α=0.0	0.523	0.388	0.539	**0.198**	**0.179**	0.462	0.520	0.469	0.512
α=0.2	**0.547**	**0.412**	0.531	0.190	0.173	0.441	**0.527**	0.472	0.510
α=0.6	0.527	0.388	0.529	0.186	0.168	0.412	0.524	**0.476**	0.501
α=1.0	0.524	0.385	0.522	0.185	0.168	**0.408**	0.522	0.472	0.499
α=2.0	0.503	0.368	**0.507**	0.182	0.166	**0.408**	0.519	0.469	**0.498**

Table 2: **Privacy-utility trade-off with varying α.** Overall, as α increases, task utility metrics decrease, and MIA effectiveness also drops. α values of 1.0 and 2.0 achieve a strictly better privacy-utility trade-off than Per-Pcs.

Prompt: Generate a headline for the following article: But it got us thinking about the concept of a "healthier" option – is it always, in fact, better for you? In some instances
s_1**:** Healthier Food Swaps That Aren't Really Healthier (PHOTOS)
α**=0.0:** Healthier Foods That Aren't Actually Healthier (PHOTOS)
α**=5.0:** 10 'Healthy' Foods That Are Actually Bad For You
Prompt: Generate a headline for the following article: "If you leave the house without sunscreen, you might as well be naked." – Meghan McCain "We talk about skin health, ..."
s_1**:** Meghan McCain, Kate Upton And More Celebrities Reveal Their Beauty Secrets (PHOTOS)
α**=0.0:** Meghan McCain 's Skin Care Routine: 'I'm Not Afraid To Try New Things' (VIDEO)
α**=5.0:** The Best Beauty Advice From The Women Of The View

Table 3: **Response comparison with varying α.** On LaMP-4, at $\alpha = 0.0$, responses are more aligned with the top sharing user s_1 than at $\alpha = 5.0$, where greater lexical and phrasal differences appear.

provements are observed on LaMP-1 and LaMP-2, with relative gains of 7.1% in accuracy and 8.1% in F1 score on LaMP-1, and 13.8% in accuracy and 34.4% in F1 score on LaMP-2. These results suggest that strong personalization can be achieved without directly training on target user data, but instead by merging shared LoRA modules.

We observe that PAG, particularly when combined with RAG, performs well on several tasks, including LaMP-1 and LaMP-4. This likely arises from the users' consistent behavior patterns, which can be captured succinctly in text. However, for users with more complex or evolving behaviors, text-based encoding may be less effective. In such cases, combining prompt- and training-based methods could yield better results, though conflicts between parametric and non-parametric knowledge are often observed in our experiments.

Privacy-utility trade-off analysis. We evaluate PriME in terms of the privacy-utility trade-off it achieves compared to Per-Pcs under varying privacy levels across three benchmark tasks: LaMP-2, LaMP-4, and LaMP-7. Utility is measured using two task-specific metrics, and privacy risk is assessed by MIA performance, measured in terms of the AUC score, where lower values indicate stronger privacy. As shown in Table 2, PriME consistently outperforms Per-Pcs in utility, achieving comparable or better results at every α value considered. As α increases, utility generally declines, with a similar reduction in MIA risk. Notably, at α values of 1.0 and 2.0, PriME achieves a strictly better privacy-utility trade-off compared to Per-Pcs, with a 20.6% reduction in MIA risk while maintaining comparable utility on LaMP-4 at α of 2.0. These results demonstrate that our method enables a controllable privacy-utility trade-off, outperforming the baseline for appropriate values of α. Note that while AUC values near 0.5 generally suggest limited privacy leakage, the key result is that PriME consistently lowers MIA risks compared to Per-Pcs while achieving superior utility. Extending the benchmark to enable more comprehensive and realistic MIA evaluations remains an important direction for future work.

Table 3 presents additional qualitative examples comparing responses generated with LoRA modules merged at different α values to those of the most similar sharing user. At α of 0.0, the responses closely resemble those of the top sharing user, with similar sentence structure and word

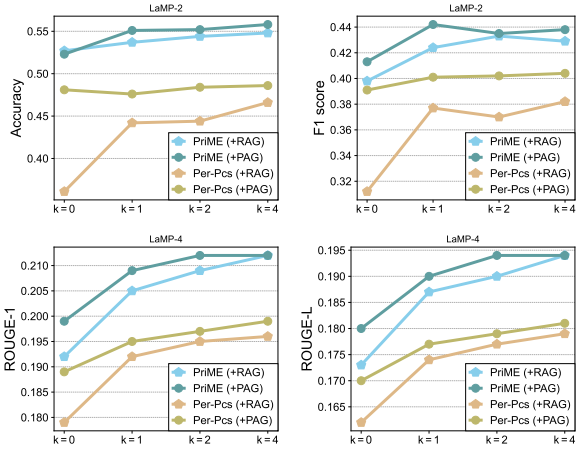


Figure 2: **Performance comparison with varying k in RAG, with and without PAG**. Overall performance improves with increasing k and profile augmentation for both PriME and Per-Pcs. Per-Pcs benefits more from RAG and PAG, showing greater reliance on the augmentation methods to achieve competitive results.

	LaMP-2		LaMP-4		LaMP-7	
	Acc ↑	F1 ↑	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑
Per-Pcs	0.363	0.296	0.179	0.162	0.507	0.460
$n = 1$	0.501	0.372	0.193	0.175	**0.523**	**0.470**
$n = 3$	**0.523**	**0.388**	**0.198**	**0.179**	0.520	0.469
$n = 5$	**0.523**	**0.388**	0.195	0.177	0.517	0.465

Table 4: **Performance with varying number of LoRAs.** Performance often improves with more LoRAs, with a single LoRA in PriME often outperforming Per-Pcs.

choices. In contrast, at α of 5.0, the responses become more distinct. These examples illustrate how varying α values control the degree of response similarity with the sharing user.

4.3 Ablations and analysis

Use of RAG and PAG. We evaluate the impact of varying the number of retrieved items (k) in RAG on the effectiveness of PriME and Per-Pcs, and assess how PAG further affects performance. As shown in Figure 2, performance improves with k for both methods, though the gains plateau beyond k at 2, with additional benefits from profile augmentation. Per-Pcs shows greater dependence on RAG and PAG for competitive performance, while PriME sees more moderate improvements, especially on LaMP-2. Notably, PriME without RAG outperforms Per-Pcs with RAG at k of 4 on LaMP-2, indicating that a well-personalized module can reduce the need for prompt augmentation.

LoRA count and merging granularity. We evaluate PriME on LaMP-2, LaMP-4, and LaMP-7

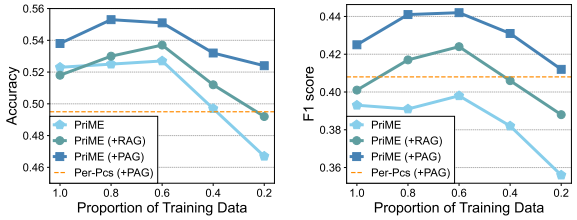


Figure 3: **Performance with varying proportions of training data used for optimization.** The dotted line represents the best performance achieved by Per-Pcs using the full training data.

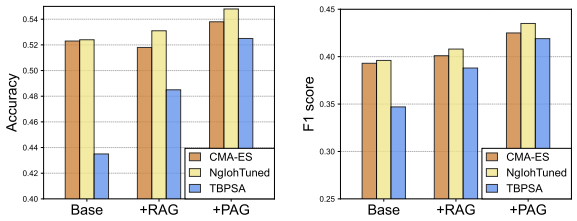


Figure 4: **Comparison of different gradient-free optimization algorithms in PriME.** General-purpose optimizers like CMA-ES provide competitive results on the benchmark, while specialized optimizers such as TBPSA may be considered in specific scenarios, such as when strong noise is present in evaluating f.

with different numbers of LoRAs used for merging. As shown in Table 4, performance often improves with more LoRAs. Notably, using only a single LoRA, which produces a target user module simply by scaling the selected LoRA, still outperforms Per-Pcs, which performs merging at the layer level. This shows that effective personalization is possible with coarser LoRA-level merging, provided interpolation weights are appropriately optimized.

Data sampling for efficiency. While gradient-free methods enable direct optimization of arbitrary objectives, computing metrics such as ROUGE requires full language model decoding, resulting in substantial computational cost at scale. To improve efficiency, we evaluate PriME with varying fractions of the data used in merging. As illustrated in Figure 3, on LaMP-2, training with only 60% of the data achieves performance comparable to that of using the full dataset. Moreover, with only 20% of the data, we surpass the best-performing variant of Per-Pcs, which is with PAG. These results indicate that significant reductions in training data are possible for PriME without compromising effectiveness, enhancing the scalability of the method.

Optimizer comparison. Although PriME is agnostic to the choice of gradient-free optimzer, we

Task	Metric	Per-Pcs			PriME (Ours)		
		Base	+RAG	+PAG	Base	+RAG	+PAG
LaMP-2	Acc ↑	0.381	0.436	0.473	0.514	0.534	**0.545**
	F1 ↑	0.338	0.374	0.405	0.408	0.425	**0.434**
LaMP-4	R-1 ↑	0.196	0.205	0.209	0.203	0.213	**0.218**
	R-L ↑	0.176	0.185	0.188	0.184	0.194	**0.198**

(a) Llama3.1-8B

Task	Metric	Per-Pcs			PriME (Ours)		
		Base	+RAG	+PAG	Base	+RAG	+PAG
LaMP-2	Acc ↑	0.327	0.402	0.462	0.496	0.515	**0.527**
	F1 ↑	0.278	0.349	0.391	0.366	0.411	**0.413**
LaMP-4	R-1 ↑	0.180	0.191	0.190	0.184	0.197	**0.202**
	R-L ↑	0.162	0.173	0.172	0.165	0.180	**0.183**

(b) Llama3.2-3B

Table 5: **LaMP results for Llama 3 models.** For both the 8B and 3B models, PriME outperforms Per-Pcs on LaMP-2 (multi-class classification) and LaMP-4 (generation) tasks.

	LaMP-2			LaMP-4		
	Acc ↑	F1 ↑	AUC ↓	R-1 ↑	R-L ↑	AUC ↓
Per-Pcs	0.381	0.338	0.526	0.196	0.176	0.526
α=0.0	0.514	0.408	0.523	**0.203**	**0.184**	0.467
α=0.2	0.520	**0.414**	0.517	0.201	0.182	0.452
α=0.6	0.510	0.401	0.514	0.195	0.175	0.421
α=1.0	**0.526**	**0.414**	0.513	0.192	0.173	0.416
α=2.0	0.513	0.396	**0.503**	0.193	0.174	**0.413**

(a) Llama3.1-8B

	LaMP-2			LaMP-4		
	Acc ↑	F1 ↑	AUC ↓	R-1 ↑	R-L ↑	AUC ↓
Per-Pcs	0.327	0.278	0.530	0.180	0.162	0.497
α=0.0	**0.496**	**0.366**	0.524	**0.184**	**0.165**	0.451
α=0.2	0.493	0.360	0.517	0.177	0.159	0.433
α=0.6	0.484	0.359	0.518	0.171	0.153	0.405
α=1.0	0.479	0.349	0.503	0.168	0.152	0.402
α=2.0	0.458	0.330	**0.499**	0.168	0.151	**0.400**

(b) Llama3.2-3B

Table 6: **Privacy-utility trade-offs for Llama 3 models.** PriME achieves strictly better trade-offs for multiple values of α on LaMP-2 and LaMP-4 for both the 8B and 3B models.

evaluate several popular optimizers and their impact on performance. Specifically, we consider three optimizers: vanilla CMA-ES, a widely used evolutionary strategy; NgIohTuned (Bennet et al., 2021), a meta-optimizer that dynamically selects among strategies (including CMA-ES variants) depending on the problem setting; and TBPSA (Hellwig and Beyer, 2016), which adapts population size to better handle potentially strong noise in the fitness function. As shown in Figure 4, general-purpose optimizers such as CMA-ES and NgIohTuned perform well on LaMP-2, while TBPSA yields comparatively worse results. Although language model decoding introduces some stochasticity, the evaluation function f is deterministic in our setting. Since evolutionary algorithms are generally robust to moderate noise in the objective function (Arnold, 2012), highly specialized optimizers like TBPSA may not be the most suitable in this context. Nevertheless, using gradient-free methods allows more flexibility across different conditions, improving broad applicability.

Alternative base models. To assess how PriME performs across model architectures and sizes, we conduct additional experiments with the Llama 3 family of models (Grattafiori et al., 2024) on LaMP-2 and LaMP-4, specifically using the Llama3.1-8B and Llama3.2-3B models. Table 5 summarizes the results for the two models with PriME optimizing solely for alignment with target user preferences, i.e., α set to 0. Similar to the results for Llama-2-7b, PriME demonstrates superior performance on both the multi-class classification (LaMP-2) and generation (LaMP-4) tasks compared to Per-Pcs. On LaMP-2, PriME achieves relative improvements over Per-Pcs in base performance with gains of 35.0% in accuracy and 20.7% in F1 score for the 8B model, and 51.7% in accuracy and 31.7% in F1 score for the 3B model. On LaMP-4, PriME achieves improvements of 3.3% in ROUGE-1 and 4.5% in ROUGE-L for the 8B model, and 2.2% in ROUGE-1 and 1.9% in ROUGE-L for the 3B model. Notably, Llama3.2-3B with PriME outperforms Llama3.1-8B with Per-Pcs on LaMP-2.

Regarding privacy-utility trade-offs, Table 6 shows the trade-offs achieved by Per-Pcs and PriME for varying values of α. On LaMP-2, PriME consistently achieves strictly better trade-offs across all α values, with AUC improving as α increases. On LaMP-4, PriME results in better trade-offs for $\alpha = 0.0$ and $\alpha = 0.2$ for the 8B model, and for $\alpha = 0.0$ for the 3B model.

5 Related Work

LLM personalization. Personalization aims to adapt general-purpose models to individual user preferences. Prompt-based methods incorporate user preferences into input prompts, enabling personalization without additional training (Mysore et al., 2024; Richardson et al., 2023; Kim and Yang, 2025). Approaches range from in-context learn-

ing with few-shot user examples (Christakopoulou et al., 2023) to retrieval-augmented generation (RAG) (Salemi et al., 2024), which integrates relevant user records into prompts, and profile-augmented generation (PAG) (Richardson et al., 2023), which leverages abstract user profile summaries to enhance query relevance. While simple to implement, these methods are constrained by the model's context length and incur higher inference costs due to enlarged prompts. Moreover, including potentially sensitive user information as raw text in a prompt can introduce privacy risks.

Training-based methods, on the other hand, directly encode user preferences into model parameters. OPPU (Tan et al., 2024b) is a simple method that fine-tunes a LoRA (Hu et al., 2022) module per user, but performance may degrade when only limited user data is available for training. To address this, methods such as group preference optimization (Zhao et al., 2024) and soft prompting (Li et al., 2024) leverage data from multiple users or groups. However, accessing such collective data is rarely available in practice and can raise privacy concerns without explicit safeguards.

Model merging offers alternative methods. PerPcs (Tan et al., 2024a) merges PEFT modules each trained on different user data to create a personalized module for a target user, enabling more effective personalization when target user data is limited. However, it introduces additional model complexity, does not optimize for task-specific utility, and fails to explicitly address the privacy risks for sharing users. In contrast, PriME employs evolutionary methods to optimize both capturing user preferences and reducing privacy risks for sharing users in merging-based personalization.

Model merging. Model merging is a surprisingly effective approach for developing models by integrating diverse models with complementary capabilities into a unified architecture. Simple techniques, such as averaging the weights of fine-tuned models, have shown strong performance on tasks such as image classification (Wortsman et al., 2022). In language models, methods such as task arithmetic (Ilharco et al., 2023) construct task vectors that represent model weights encoding task-specific abilities, which are then combined to create a model with the desired capabilities. Despite its effectiveness, model merging often relies on human intuition and expertise for selecting models and designing merging recipes. Recently, evolutionary algorithms have been applied to automatically discover effective merging strategies in developing foundation models with diverse capabilities (Akiba et al., 2025). In this work, we build on this approach by applying evolutionary methods to personalize language models, with an explicit optimization objective for reducing privacy risks associated with shared user data.

6 Conclusion

In this work, we introduce PriME, a novel approach to LLM personalization that leverages evolutionary algorithms to directly optimize task utility while mitigating privacy risks for users sharing their data. Experiments on LaMP demonstrate that PriME outperforms competing methods, achieving substantial improvements in capturing user preferences and enhanced robustness to membership inference attacks. These findings highlight the potential of evolutionary model merging as a promising framework for privacy-aware LLM personalization.

Future work. Further exploration of advanced model merging techniques—such as random weight dropping (Yu et al., 2024) and the removal of conflicting weights (Yadav et al., 2024)—and their impact on the resulting personalized modules and privacy risks associated with sharing users would be a valuable direction for future work. Additionally, addressing conflicts between parametric and non-parametric knowledge when combining prompt-based and training-based methods, or between two sources of non-parametric knowledge (e.g., RAG and PAG), as observed in our empirical evaluation, could provide deeper insights into further enhancing LLM personalization.

Limitations

While our proposed PriME demonstrates promising results in LLM personalization, further investigation is needed to assess how well it scales with larger models and datasets. Our experimental results suggest that PriME remains effective even with substantially reduced training data. However, developing more sophisticated strategies for selecting the most useful subset of data for merging could further improve its scalability. Moreover, while our evaluation shows that the approach empirically mitigates MIA risks, it does not offer formal privacy guarantees. Future work should explore integrating rigorous mechanisms such as differential privacy (Dwork, 2006).

Ethics Statement

Personalization in LLMs aims to adapt general-purpose models to capture the preferences of individual users or groups. This often involves access to personal data, which may contain sensitive information, making privacy protection crucial. While our evaluation shows that PriME achieves effective personalization and reduces the risk of membership inference attacks, emerging threats—such as LLMs inferring personal attributes from seemingly innocuous text (Staab et al., 2024; Kim et al., 2025)—remain a concern. Future research on personalization should carefully consider such emerging privacy risks.

We used an AI assistant (ChatGPT) to refine the writing during the preparation of this work. All models and datasets are publicly available and used in accordance with their intended purpose; details are provided in Appendix B.

Acknowledgments

This research was supported in part by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2019-II190075, Artificial Intelligence Graduate School Support Program (KAIST); No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University); No. RS-2021-II212068, Artificial Intelligence Innovation Hub; No. RS-2025-02215344, Development of AI Technology with Robust and Flexible Resilience Against Risk Factors) and the Mobile eXperience (MX) Business, Samsung Electronics Co., Ltd.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Takuya Akiba, Makoto Shing, Yujin Tang, Qi Sun, and David Ha. 2025. Evolutionary optimization of model merging recipes. *Nature Machine Intelligence*, 7(2):195–204.

Dirk V Arnold. 2012. *Noisy optimization with evolution strategies*, volume 8. Springer Science & Business Media.

Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen Herring, and Sujay Kumar Jauhar. 2024. Knowledge-augmented large language models for personalized contextual query suggestion. In *Proceedings of the ACM on Web Conference 2024*, pages 3355–3366.

Pauline Bennet, Carola Doerr, Antoine Moreau, Jeremy Rapin, Fabien Teytaud, and Olivier Teytaud. 2021. Nevergrad: black-box optimization platform. *ACM SIGEVOlution*, 14(1):8–15.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*.

Konstantina Christakopoulou, Alberto Lalama, Cj Adams, Iris Qu, Yifat Amir, Samer Chucri, Pierce Vollucci, Fabio Soldo, Dina Bseiso, Sarah Scodel, and 1 others. 2023. Large language models for user interest journeys. *arXiv preprint arXiv:2305.15498*.

Haonan Duan, Adam Dziedzic, Mohammad Yaghini, Nicolas Papernot, and Franziska Boenisch. 2024a. On the privacy risk of in-context learning. *arXiv preprint arXiv:2411.10512*.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024b. Do membership inference attacks work on large language models? In *First Conference on Language Modeling*.

Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nikolaus Hansen. 2006. The cma evolution strategy: a comparing review. *Towards a new evolutionary computation: Advances in the estimation of distribution algorithms*, pages 75–102.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In *International Conference on Learning Representations (ICLR)*.

Michael Hellwig and Hans-Georg Beyer. 2016. Evolution under strong noise: A self-adaptive evolution strategy can reach the lower performance bound - the pccmsa-es. In *International Conference on Parallel Problem Solving from Nature*, pages 26–36. Springer.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, and 1 others. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, 103:102274.

Jaehyung Kim and Yiming Yang. 2025. Few-shot personalization of llms with mis-aligned responses. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Kyuyoung Kim, Hyunjun Jeon, and Jinwoo Shin. 2025. Self-refining language model anonymizers via adversarial distillation. *arXiv preprint arXiv:2506.01420*.

Junyi Li, Ninareh Mehrabi, Charith Peris, Palash Goyal, Kai-Wei Chang, Aram Galstyan, Richard Zemel, and Rahul Gupta. 2024. On the steerability of large language models toward data-driven personas. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Xin Liu, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak Patel. 2023. Large language models are few-shot health learners. *arXiv preprint arXiv:2305.15525*.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David K Evans, and Taylor Berg-Kirkpatrick. 2022. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

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*.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. Lamp: When large language models meet personalization. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE.

Robin Staab, Mark Vero, Mislav Balunovic, and Martin Vechev. 2024. Beyond memorization: Violating privacy via inference with large language models. In *International Conference on Learning Representations (ICLR)*.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024a. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

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 *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

CodeGemma Team, Heri Zhao, Jeffrey Hui, Joshua Howland, Nam Nguyen, Siqi Zuo, Andrea Hu, Christopher A Choquette-Choo, Jingyue Shen, Joe Kelley, and 1 others. 2024. Codegemma: Open code models based on gemma. *arXiv preprint arXiv:2406.11409*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems (NeurIPS)*.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Siyan Zhao, John Dang, and Aditya Grover. 2024. Group preference optimization: Few-shot alignment of large language models. In *International Conference on Learning Representations (ICLR)*.

A Benchmark and Task Details

In this section, we provide additional details on the individual tasks in the benchmark.

LaMP-1: Personalized citation identification. This is a binary classification task in which a language model identifies which of two candidate papers a user would cite, given the user's history of previous publications. User's profile contains a set of titles, abstracts, and citations of previous publications. During task adaptation, the model is trained to generate a citation based on the title of a paper.

LaMP-2: Personalized movie tagging. In this multi-class classification task, a language model needs to classify a given movie description into one of 15 possible categories, e.g., 'comedy' or 'action'. Given a user's historical tagging assignments, the model is adapted to generate appropriate tags for movie descriptions.

LaMP-3: Personalized product rating. In this ordinal multi-class classification task, a language model predicts product ratings based on a user's review history. Given pairs of a review and the corresponding historical rating, the model needs to generate an integer rating from 1 to 5.

LaMP-4: Personalized news headline generation. In this task, a language model generates a news headline for an article based on a user's stylistic patterns as an author. Based on the user's article history and titles, the model is adapted to generate headlines that align with the identified style.

LaMP-5: Personalized scholarly title generation. Similar to LaMP-4, in this task, a language model generates research article titles based on a user's profile of historical article-title pairs. In LaMP-5, only the abstracts of articles are provided.

LaMP-7: Personalized tweet paraphrasing. This is another personalized generation task in which a language model needs to paraphrase a tweet in the style of a user based on the user's historical tweets.

B Experimental Details

B.1 Setup details

Data splits. Following a similar approach to that from prior work (Tan et al., 2024a), we split the data by using 25% of the users for adapting the base model, 100 users for testing, and the rest as candidates for sharing their LoRA modules.

Sharing user selection. To select sharing users, we first compute user embeddings with the DeBERTa V3 large model (He et al., 2023) (see Section 3.1) and apply k-means clustering to group them into up to 50 clusters. From each cluster, we select the user with the largest history as the sharing user. For each selected user, we train a LoRA module using the task-adapted model as the base.

User profile generation. User profiles are generated by summarizing a random sample of the user's history using Mistral-7B-Instruct v0.2 (Jiang et al., 2023). These summaries capture user preferences or behavior patterns, e.g., the user's most frequently explored research topics, in text.

Baselines. As prior work (Tan et al., 2024a) does not report task adaptation details, we perform additional hyperparameter tuning to achieve comparable results on LaMP-2, LaMP-4, LaMP-5, and LaMP-7, and improved results on LaMP-1 and LaMP-3. For Per-Pcs, we follow the reported settings and also conduct additional hyperparameter tuning, reporting the best-performing results.

PriME. For PriME, we optimize the sum of the two utility metrics for each task. Note that for LaMP-1, the training data consists of textual citations paired with titles and abstracts, differing from the evaluation task, which is a binary classification between two citation options. Hence, for LaMP-1, we optimize the sum of ROUGE-1 and ROUGE-L scores with the target citations.

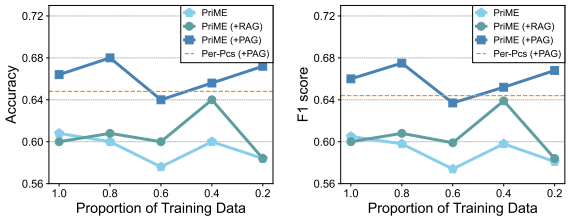
MIA. We adopt standard MIA setups, assuming an adversary with black-box access to the personalized LLM who queries it with input-response pairs and uses the resulting losses as scores (Yeom et al., 2018; Duan et al., 2024b). The adversary then predicts whether an example was used to train a LoRA module for a sharing user involved in merging. To compare MIA risks, we evaluate the model's loss on each sharing user's training data, treating the data from selected users as members and that from non-selected users as non-members, and compute the AUC to measure the risk of MIA attacks. We report the average AUC across the 100 test users.

B.2 Training details

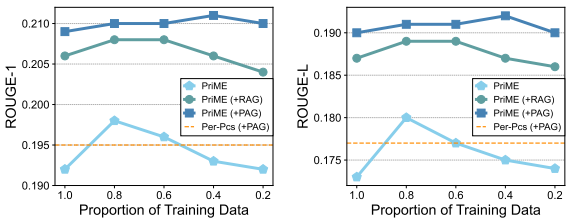
Hyperparameters. Table 7 summarizes the hyperparameters used for task adaptation, LoRA training for sharing users, Per-Pcs, and PriME. For LoRA training, we used the LoRA module hyperparameters reported in Tan et al. (2024a), but

Task	Task Adaptation			Sharer PEFT			Sharer Gate			Per-Pcs		PriME		
	batch	ep	lr	batch	ep	lr	batch	step	lr	top-k	batch	LoRA	budget	data %
LaMP-1	16	3	1e-4	8	1	1e-4	6	100	1e-5	1	16	5	50	0.8
LaMP-2	16	3	1e-4	6	3	2e-4	6	100	2e-5	3	16	3	30	0.6
LaMP-3	8	3	1e-4	4	2	1e-4	4	100	1e-5	1	6	5	50	1.0
LaMP-4	8	3	1e-4	8	3	1e-4	6	50	2e-5	1	16	3	30	0.8
LaMP-5	8	3	1e-4	8	2	1e-4	6	50	2e-5	1	10	3	50	1.0
LaMP-7	16	3	1e-4	8	1	1e-4	6	50	2e-5	2	16	3	30	1.0

Table 7: **Hyperparameters.** Hyperparameters for task adaptation, PEFT training, Per-Pcs, and PriME on LaMP.

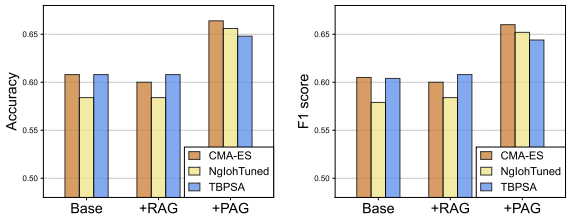


(a) LaMP-1

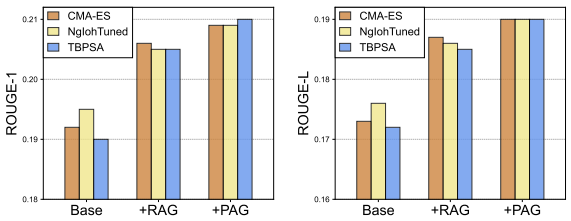


(b) LaMP-4

Figure 5: **Effects of subsampling data.** PriME outperforms Per-Pcs, which uses the full dataset, often with just 20% of the training data.



(a) LaMP-1



(b) LaMP-4

Figure 6: **Comparison of optimizers.** PriME achieves competitive results across a range of gradient-free optimization algorithms.

additionally experimented with training hyperparameters such as learning rate and batch size. The configurations achieving the best performance on the target metrics were selected.

We note that while we successfully reproduced most of the Per-Pcs results, our LaMP-3 results were noticeably lower despite extensive efforts. This suggests that Per-Pcs may be more sensitive to hyperparameter choices on certain tasks. Nevertheless, PriME still outperforms the originally reported Per-Pcs results.

Compute resources. We conducted our experiments mainly on NVIDIA A6000 GPUs, requiring one GPU to train a single model with LoRA.

C Additional Results

This section provides additional experimental results, analysis, and discussion.

C.1 Sampling training data

In Section 4.3, we show that training data can be reduced by 40% while maintaining performance comparable to or better than Per-Pcs on LaMP-2. As shown in Figure 5, the amount of training data can be significantly reduced also on LaMP-1 and LaMP-4, often down to 20%, while still achieving better test performance than Per-Pcs.

C.2 Comparison of optimization algorithms

As illustrated in Figure 6, PriME consistently achieves competitive results across a range of gradient-free optimization algorithms on LaMP-1 and LaMP-4. Similar to LaMP-2, more general-purpose optimizers, such as CMA-ES and NgIohTuned, demonstrate effectiveness with PriME.

C.3 Prompt sensitivity

We conduct additional experiments to evaluate the sensitivity of prompt-based methods to (1) repeated

Source	Output
Per-Pcs	10 Living Room Decor Ideas That Will Make You Want To Throw A Party (PHOTOS)
$\alpha = 0.0$	10 Living Room Decor Ideas That Will Make Your Home The Perfect Place To Entertain (PHOTOS)
$\alpha = 2.0$	10 Ways to Make Your Living Room Look More Expensive
Top sim.	10 Living Room Decor Ideas That Will Make Your Home The Perfect Place To Entertain (PHOTOS)
Per-Pcs	10 Things You Didn't Know About Presidential Vacations
$\alpha = 0.0$	The Most Expensive Vacations In The World
$\alpha = 2.0$	10 Vacations You Can't Take
Top sim.	World Leaders' Vacations (PHOTOS)
Per-Pcs	The Middle Class Is Under Attack
$\alpha = 0.0$	The Middle Class Is Dying. Here's How to Save It
$\alpha = 2.0$	The Trickle-Down Economy Is A Failure
Top sim.	The Middle Class Is Dying. Here's How to Save It

Table 8: **Qualitative comparisons.** Examples comparing Per-Pcs, PriME with varying α, and the top similar user.

Task	Metric	RAG	PAG	
		$k = 1$	$k = 0$	$k = 1$
LaMP-2	Acc ↑	0.412 ± 0.014	0.439 ± 0.027	0.467 ± 0.020
	F1 ↑	0.336 ± 0.020	0.340 ± 0.019	0.379 ± 0.020
LaMP-4	R-1 ↑	0.202 ± 0.001	0.196 ± 0.002	0.207 ± 0.001
	R-L ↑	0.184 ± 0.001	0.177 ± 0.001	0.188 ± 0.000

Table 9: **Variance from repeated decoding.** Performance is relatively consistent across repeated decoding.

Task	Metric	Original		GPT	
		$k = 0$	$k = 1$	$k = 0$	$k = 1$
LaMP-2	Acc ↑	0.470	0.490	0.419	0.441
	F1 ↑	0.361	0.402	0.316	0.354
LaMP-4	R-1 ↑	0.194	0.207	0.194	0.188
	R-L ↑	0.176	0.206	0.175	0.187

Table 10: **PAG with different user profiles.** How user profiles are phrased in text can have a notable impact on PAG performance.

decoding and (2) varied input queries. For the repeated decoding experiment, we report the mean and standard error over three independent runs. For the varied input queries experiment, we rephrase the original user profiles using GPT-4o-mini and evaluate how performance changes compared to the original profiles, using the same random seed for decoding.

Table 9 summarizes the results of the repeated decoding experiment on LaMP-2 and LaMP-4. Overall, we observe low variance in performance across random seeds, suggesting that sampling has limited effect on the results. Table 10 presents results from the varied input queries experiment, where user profiles are rephrased using GPT-4o-mini. In LaMP-2 (multi-class classification), profiles typically mention users' most frequently labeled movie genres (e.g., action, comedy), while in LaMP-4 (generation), they describe general writing styles. On LaMP-4, performance with rephrased profiles is similar to the original, with a slight drop in the $k = 1$ case. In contrast, LaMP-2 shows a more noticeable decline. Our analysis reveals that GPT often rephrased class labels such as "sci-fi" to "science fiction", leading to predictions that do not fully match the expected label set. These findings suggest that prompt-based methods can be sensitive to how user profiles are phrased, particularly in classification tasks.

C.4 Qualitative examples

To better illustrate qualitative differences among methods, we include in Table 8 several example outputs from the benchmark. Specifically, we compare outputs from Per-Pcs, PriME with $\alpha = 0.0$, PriME with $\alpha = 2.0$, and the output generated with the top similar user's module. As shown by these examples, PriME tends to produce outputs more similar to those of the most similar user and, with a greater similarity penalty, generates more (often substantially) rephrased outputs.

D Prompt Templates

For a fair comparison with the baseline results reported in Tan et al. (2024a), we adopt the same prompt templates and reproduce them below. Table 11 summarizes the prompt templates used to generate user profiles for each benchmark task, while Table 12 shows the prompt templates used for the actual personalization tasks. Note that the prompts in Table 12 are prepended with the user profile, user history, or both, depending on whether RAG, PAG, or both are used.

Task	Prompt Template
LAMP-1	Write a summary, in English, of the research interests and topics of a researcher who has published the following papers. Only generate the summary, no other text. User History: {USER HISTORY} Answer:
LAMP-2	Look at the following past movies this user has watched and determine the most popular tag they labeled. Answer in the following form: most popular tag: <tag>. User History: {USER HISTORY} Answer:
LAMP-3	Based on this user's past reviews, what are the most common scores they give for positive and negative reviews? Answer in the following form: most common positive score: <most common positive score>, most common negative score: <most common negative score>. User History: {USER HISTORY} Answer:
LAMP-4	Given this author's previous articles, try to describe a template for their headlines. I want to be able to accurately predict the headline given one of their articles. Be specific about their style and wording, don't tell me anything generic. User History: {USER HISTORY} Answer:
LAMP-5	Given this author's previous publications, try to describe a template for their titles. I want to be able to accurately predict the title of one of the papers from the abstract. Only generate the template description, nothing else. User History: {USER HISTORY} Answer:
LAMP-7	Given this person's previous tweets, try to describe a template for their tweets. I want to take a generic sentence and rephrase it to sound like one of their tweets, with the same style/punctuation/capitalization/-wording/tone/etc. as them. Only give me the template description, nothing else. User History: {USER HISTORY} Answer:

Table 11: **Profile generation prompts.** Prompt templates used for profile generation.

Task	Prompt Template
LAMP-1	### User Instruction: Identify the most relevant reference for the listed publication by the researcher. Select the reference paper that is most closely related to the researcher's work. Please respond with only the number that corresponds to the reference. Paper Title: {QUERY PAPER TITLE} Reference: [1] - {OPTION1} [2] - {OPTION2} Answer:
LAMP-2	### User Instruction: Which tag does this movie relate to among the following tags? Just answer with the tag name without further explanation. tags: [sci-fi, based on a book, comedy, action, twist ending, dystopia, dark comedy, classic, psychology, fantasy, romance, thought-provoking, social commentary, violence, true story] Description: {QUERY MOVIE DESCRIPTION} Tag:
LAMP-3	### User Instruction: What is the score of the following review on a scale of 1 to 5? just answer with 1, 2, 3, 4, or 5 without further explanation. Review: {QUERY REVIEW} Score:
LAMP-4	### User Instruction: Generate a headline for the following article. Article: {QUERY ARTICLE} Headline:
LAMP-5	### User Instruction: Generate a title for the following abstract of a paper. Abstract: {QUERY ABSTRACT} Title:
LAMP-7	### User Instruction: Paraphrase the following text into tweet without any explanation before or after it. Text: {QUERY TEXT} Tweet:

Table 12: **Personalization prompts.** Prompt templates used for personalization.