Don't Start Over: A Cost-Effective Framework for Migrating Personalized Prompts Between LLMs

Ziyi Zhao[1], Chongming Gao[1*], Yang Zhang[2], Haoyan Liu[1*], Weinan Gan[3], Huifeng Guo[3], Yong Liu[3], Fuli Feng[1],

[1]University of Science and Technology of China, Hefei, China
[2]National University of Singapore, Singapore
[3]Huawei Technologies Co., Ltd, Shenzhen, China
re2477036@mail.ustc.edu.cn, chongming.gao,zyang1580@gmail.com, liuhaoyan@ustc.edu.cn,
ganweinan1,huifeng.guo,liu.yong6@huawei.com, fulifeng93@gmail.com

Abstract

Personalization in Large Language Models (LLMs) often relies on user-specific soft prompts. However, these prompts become obsolete when the foundation model is upgraded, necessitating costly, full-scale retraining. To overcome this limitation, we propose the **P**rompt-level **U**ser **M**igration **A**dapter (**PUMA**), a lightweight framework to efficiently migrate personalized prompts across incompatible models. PUMA utilizes a parameter-efficient adapter to bridge the semantic gap, combined with a group-based user selection strategy to significantly reduce training costs. Experiments on three large-scale datasets show our method matches or even surpasses the performance of retraining from scratch, reducing computational cost by up to 98%. The framework demonstrates strong generalization across diverse model architectures and robustness in advanced scenarios like chained and aggregated migrations, offering a practical path for the sustainable evolution of personalized AI by decoupling user assets from the underlying models.

Code — https://github.com/Kimagure7/Dont-Start-Over

Introduction

The rapid advancement of large language models (LLMs) (Vaswani et al. 2017; Brown et al. 2020; Chowdhery et al. 2023; OpenAI et al. 2024) has fundamentally reshaped the landscape of NLP, demonstrating remarkable capabilities across a multitude of tasks. As these powerful models are increasingly integrated into real-world applications, the focus is shifting from general-purpose utility to deep personalization – a critical step to meet user expectations for experiences tailored to their unique needs and preferences (Zhang et al. 2025b; Chen et al. 2024; Salemi et al. 2024). This evolution is prominent in burgeoning domains such as personal assistants (Dong et al. 2023; Zhang et al. 2024), adaptive education (Wang et al. 2024), and recommendation systems (Gao et al. 2025a; Fan et al. 2025; Gao et al. 2025b; Cai et al. 2025; Zhang et al. 2025a; Shi et al. 2024).

Soft prompts (Lester, Al-Rfou, and Constant 2021) have emerged as a key technology for realizing this deep personalization. The core idea is to encode and carry each user's unique preferences and knowledge within a lightweight, efficient, and non-invasive dedicated vector – the *soft prompt*. This approach has given rise to a new application architecture: a "1+N" hybrid system (Tan et al. 2024; Li, Zhang, and Chen 2023). Such a system combines a single, powerful, general-purpose foundation model (the "1") with thousands or even millions of independent soft prompts (the "N"), where each prompt represents an individual user's personalized representation. This method allows for efficient, user-specific customization without altering the parameters of the LLM model.

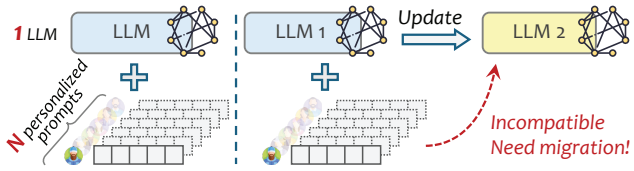


Figure 1: The "1+N" system (left) and an illustration of the migration of personalized prompts to a new LLM (right).

However, this approach introduces a critical vulnerability: the efficacy of soft prompts is inherently tied to the specific foundation model they were trained on. The lifecycle of these "1+N" systems inevitably involves replacing the foundation model, whether upgrading to a more powerful successor or adopting a smaller, more efficient variant to meet new deployment constraints. Any such replacement shatters the semantic alignment between the prompts and the model, rendering the entire corpus of personalized soft prompts obsolete. Consequently, the valuable personalization accumulated across the user base is invalidated, necessitating a complete and cost-prohibitive retraining of all prompts from scratch. This challenge motivates the central question of our work: *Can we migrate a large corpus of personalized soft prompts from a source model to a new target model with high fidelity, but at a fraction of the computational cost of full retraining?*

Migrating user-level personalization presents a distinct challenge compared to existing soft prompt transfer paradigms. Most research has focused on the task level, where the goal is to transfer a single, public prompt trained

*Corresponding authors.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

on one task (e.g., natural language inference) to accelerate fine-tuning on a different target task (e.g., text classification) (Vu et al. 2022; Asai et al. 2022). In this case, the knowledge is impersonal and designed for cross-task use. In contrast, our work focuses on migrating thousands of private, user-specific soft prompts, each tailored to an individual.

To tackle the challenge of personalized soft prompt migration, we decompose it into two coupled subproblems: *semantic incompatibility* – how to enable the target model to interpret prompts trained on a different source model – and *migration efficiency* – how to scale migration to support tens of thousands of users with minimal cost.

We propose **P**rompt-level **U**ser **M**igration **A**dapter (**PUMA**), a lightweight framework designed for both semantic alignment and scalable migration. It comprises two key components: (1) a parameter-efficient adapter trained end-to-end to bridge semantic gaps between models; and (2) a user selection strategy that groups users by their prompt embeddings and output variance to form a small, representative training subset, significantly reducing cost without sacrificing performance.

We evaluate our framework on the task of personalized recommendation, focusing on migration across models of varying sizes (e.g., LLaMA 1B $\rightarrow$ 3B) and even across model families (e.g., LLaMA $\rightarrow$ Qwen). Experiments on three large-scale datasets demonstrate that our method matches or surpasses the performance of costly from-scratch retraining, while reducing computation by up to 98%. These results highlight the scalability and efficiency of our approach for real-world personalization.

We further extend our method to two advanced migration paradigms: *chained migration*, where a soft prompt is sequentially migrated across multiple target LLMs, and *aggregated migration*, where soft prompts from multiple base models are fused into a single target model. The results show that leveraging multiple sources, especially in the aggregated setting, enhances knowledge integration and leads to improved post-migration performance, offering valuable insights for scalable user adaptation.

In summary, our main contributions are threefold:

- We are the first to identify and formalize the challenge of migrating user-level personalized soft prompts across foundation models.
- We propose *PUMA*, a novel, lightweight framework that uses an end-to-end trained adapter for effective migration, along with a group-based user sampling strategy to significantly enhance efficiency.
- Experiments on three large-scale datasets show our approach matches and often surpasses the performance of retraining from scratch, but with substantially lower computational cost.

Related Work

Personalization in Large Language Models

The demand for LLMs to cater to individual user needs and preferences has grown significantly (Liu et al. 2025). To meet this demand, researchers have explored various techniques, such as retrieval-augmented generation (Gao et al. 2024; Salemi et al. 2024), prompt engineering (Kang et al. 2023; Liu et al. 2023), and reinforcement learning (Jang et al. 2023). Among these, parameter-efficient fine-tuning (PEFT) methods have become a prominent approach for personalization (Lester, Al-Rfou, and Constant 2021; Hu et al. 2022). Tan et al. (2024) employed personalized PEFT modules to capture user preferences. Huang et al. (2024) proposed selective prompt tuning to achieve a personalized dialogue by adaptively selecting the appropriate soft prompts.

While effective, these PEFT personalization methods are coupled to their base models, requiring costly retraining upon model updates. To the best of our knowledge, our work is the first to investigate the migration of these user-level personalized parameters across different LLMs.

Coreset Selection

Our user selection strategy is an approach to coreset selection, a field whose primary objective is to identify a small representative data subset (a "coreset") to approximate training on a full dataset (Mirzasoleiman, Bilmes, and Leskovec 2020). Existing methodologies largely fall into two categories: score-based and optimization-driven (Albalak et al. 2024). Score-based methods rank data points using various metrics. For instance, some approaches use geometric properties like k-means clustering to select a diverse set (Sorscher et al. 2022), while others prioritize "informative" examples identified by high prediction uncertainty or training loss (Coleman et al. 2020; Zheng et al. 2023). Optimization-driven methods, in contrast, select a coreset by matching its training gradients to those of the full dataset, using first-order (Mirzasoleiman, Bilmes, and Leskovec 2020) or more refined second-order (Pooladzandi, Davini, and Mirzasoleiman 2022) information.

However, approaches requiring a per-sample evaluation pass, such as those based on gradients or uncertainty, are computationally infeasible at our scale. We therefore propose an efficient selection strategy that circumvents this specific bottleneck.

Knowledge Transfer in Soft Prompt

Research on the transferability of soft prompts has primarily focused on improving training efficiency and model generalization across different downstream tasks. For instance, Su et al. (2022) empirically studied the transferability of soft prompts across different downstream tasks and pre-trained language models (PLMs). Vu et al. (2022) utilized a retrieval-based strategy that measures task similarity to select the most relevant prompt as the initialization for a new task. Similarly, Asai et al. (2022) introduced an attention-based mechanism to mix multiple prompts for a new task.

Our work, however, addresses a different problem. These prior studies focus on transferring general, task-level knowledge — a "one-to-one" or "few-to-one" transfer problem. In contrast, we tackle the "N-to-N" challenge of migrating a large corpus of user-personalized prompts, preserving them as valuable assets across foundation model changes.

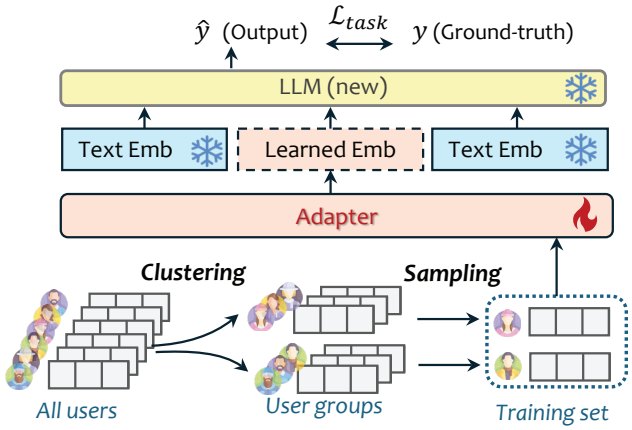


Figure 2: Illustration of PUMA, consisting of a group-based user selection strategy and a migration adapter. Users are first clustered via K-means on personalized embeddings, then sub-grouped by output variance, from which training users are sampled.

Methodology

We will first formalize the cross-model migration problem, then present our Adapter-based migration framework, and introduce the group-based user selection strategy designed to make this process efficient.

Problem Formulation

To clearly delineate the scope of our research, we first formalize the core concepts and objectives.

Personalization with Soft Prompts Consider a frozen, pre-trained source foundation model, M_s, and a set of users $\mathcal{U} = \{u_1, u_2, \ldots, u_{|\mathcal{U}|}\}$ and items $\mathcal{I} = \{i_1, i_2, \ldots, i_{|\mathcal{I}|}\}$. To capture individual user preferences, we learn a unique soft prompt, $p_u \in \mathbb{R}^{l \times d_s}$, for each user $u \in \mathcal{U}$. Here, l is the prompt length and d_s is the embedding dimensionality of the source model M_s.

The goal is to optimize the entire collection of user prompts, $\{p_u\}_{u \in \mathcal{U}}$, by minimizing a task-specific loss function. This process trains the prompts to elicit personalized predictions from the frozen model M_s. The optimization is formally expressed as:

$$\underset{\{p_u\}_{u \in U}}{\operatorname{argmin}} \sum_{(u,i,y) \in \mathcal{D}} \mathcal{L}_{\text{task}} \left(M_s \left(T(p_u, \phi(i)) \right), y \right), \quad (1)$$

where $\mathcal{D}$ is the training dataset containing user-item-outcome tuples (u, i, y), $\phi(i)$ is the textual representation of an item, and T is a template that structures the user prompt p_u and item text for the model. $\mathcal{L}_{\text{task}}$ measures the discrepancy between the model's prediction and the ground truth y (e.g., a rating).

Cross-Model Prompt Migration The primary challenge occurs when the source model M_s is replaced with a target model M_t, especially when M_t has a different architecture or embedding dimension ($d_t \neq d_s$), where d_t is the embedding dimensionality of the target model. This mismatch creates a semantic gap, making the original prompts $\{p_u\}$ incompatible.

To address this, our goal is to learn a lightweight migration function, Φ, parameterized by θ. This function maps each source prompt p_u to a functionally equivalent prompt p'_u for the target model:

$$p'_u = \Phi_\theta(p_u), \quad \text{where } p_u \in \mathbf{R}^{l \times d_s} \text{ and } p'_u \in \mathbf{R}^{l \times d_t}$$

The optimal parameters θ^* are learned by minimizing the task loss on the target model. During this optimization, both the target model M_t and the source prompts $\{p_u\}$ remain frozen, so only the migration function Φ is trained:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{(u,i,y) \in D} \mathcal{L}_{\text{task}} \left(M_t \left(T(\Phi_\theta(p_u), \phi(i)) \right), y \right) \quad (2)$$

Adapter Framework

We implement the migration function Φ as a lightweight adapter – a feed-forward network with residual connections and Layer Normalization. This architecture balances the representational capacity with low computational cost.

Crucially, this adapter is trained end-to-end by optimizing the task-specific loss in Eq. (2), ensuring that the learned transformation is functional and preserves the downstream utility of each personalized prompt.

Extension to Advanced Migration Topologies. The flexibility of PUMA's design naturally extends to more complex migration scenarios that mirror real-world system evolution. We investigate two such advanced topologies: *chained migration* and *aggregated migration*. Chained migration ($M_A \rightarrow M_B \rightarrow M_C$) handles sequential model changes. Aggregated migration fuses personalization from multiple source models into a single target ($[M_A, M_B] \rightarrow M_C$), a common need after system mergers or A/B testing. We achieve this by concatenating a user's source prompts (e.g., $[p_u^A; p_u^B]$) and mapping the composite vector to the target model, thereby synthesizing a richer user representation from diverse sources. We empirically validate both topologies in our experiments.

Efficient Migration via Group-based User Selection

To make the migration computationally tractable for large user populations, we train the adapter on a small data subset $\mathcal{D}' = \{(u, i, y) \in \mathcal{D} \mid u \in \mathcal{U}'\}$. The core challenge lies in constructing a compact user subset $\mathcal{U}'$ that is highly representative of the entire user population.

Our selection strategy pivots on a key insight: an ideal user subset must embody both the diversity of user preferences and the spectrum of their complexity. To capture preference diversity, we leverage the geometric structure of the source prompt embeddings. For complexity, we posit that a user's historical output variance serves as an effective and computationally cheap proxy. Users with low variance exhibit consistent, easily modeled preferences (e.g., always giving high ratings). In contrast, high-variance users display more nuanced tastes, posing a greater learning challenge. A

robust migration function must be trained on examples spanning this entire spectrum of complexity. We implement this strategy through a two-stage selection process:

Stage 1: Cluster for diversity. We apply K-means clustering to the source prompt $\{p_u\}_{u\in\mathcal{U}}$. This partitions users into k distinct clusters, ensuring that our selection captures a wide array of learned preference profiles.

Stage 2: Grouped sampling by variance. Within each cluster, we stratify users by output variance and then sample from these bins, using a normal distribution to weight the medium-variance groups more heavily.

Experiments

Our experiments answer four research questions (RQs): **RQ1**: How effective is PUMA at migrating user personalization compared to the costly baseline of full retraining? **RQ2**: How efficiently does our group-based user selection strategy reduce computational costs while preserving performance, relative to random sampling? **RQ3**: Does PUMA generalize across diverse model architectures and families? **RQ4**: How robust is PUMA when applied to advanced migration topologies, such as chained and aggregated migration?

Experimental Setup

Datasets. We utilize three large-scale datasets (see Table 1 for statistics). **Amazon (Movies & TV)** (Hou et al. 2024): A subset of the Amazon Product Review dataset[1] containing 1-to-5 star ratings, used for explicit rating prediction. **Yelp**[2]: A dataset of 1-to-5 star ratings for local businesses (e.g., restaurants), modeling real-world consumer preferences. **MIND** (Wu et al. 2020): A news recommendation dataset based on implicit feedback (clicks) to predict click-through rate (CTR).

Statistics	Amazon	MIND	Yelp
#Users	30,287	50,000	32,850
#Items	96,636	19,368	129,076
#Records	1,166,752	3,892,068	1,543,687
Records/User	38.52	77.84	46.99
Sparsity	99.96%	99.60%	99.96%
Positive Ratio	N/A	17.80%	N/A

Table 1: Statistics of the datasets used in our experiments.

Evaluation Metrics. We evaluate performance using standard metrics appropriate for each task. For the rating prediction tasks on Amazon and Yelp, we use root mean square error (RMSE) and mean absolute error (MAE). For the click-through rate (CTR) prediction task on MIND, we report the area under the ROC curve (AUC) and user-weighted AUC (uAUC), which is the average of AUC scores computed for each individual user.

[1]https://amazon-reviews-2023.github.io/

[2]https://business.yelp.com/data/resources/open-dataset/

Implementation Details. To address RQ1 and RQ2, we simulate a common model upgrade scenario, migrating personalized prompts from a Llama-2-1B-Instruct source model to a more powerful Llama-2-3B-Instruct target (Grattafiori et al. 2024). To assess generalization and robustness (RQ3 and RQ4), we conduct migrations across a diverse portfolio of five models: Llama3.2-3B-Instruct, Qwen2.5-3B-Instruct (Yang et al. 2024; Team 2024), Stablelm-2-1_6b-chat (Bellagente et al. 2024), Phi-3-mini-4k-instruct (Abdin et al. 2024), and Gemma-3-1b-it (Team et al. 2025).

We conducted all experiments on NVIDIA A100 GPUs using PyTorch 2.5. We leverage foundation models from the Hugging Face Hub, keeping their parameters frozen throughout all training phases. We first pre-train the user-specific soft prompts (length $l = 1$) on the source model for 15 epochs with a learning rate of 5×10^{-4}. Subsequently, the PUMA adapter is trained for 4 epochs using the FusedAdam optimizer with a learning rate set to 1×10^{-4} and a batch size set to 32.

We tailor the loss function to the specific task of each dataset. For the rating prediction tasks (Amazon and Yelp), we employ a hybrid loss objective. We extract the logits corresponding to the five discrete rating tokens ("1" through "5") from the LLM's output. A cross-entropy loss ($\mathcal{L}_{CE}$) is applied to these logits to treat rating prediction as a classification problem. Concurrently, we feed the same logits into a lightweight MLP head to regress a continuous rating value, which is supervised by a mean squared error loss ($\mathcal{L}_{MSE}$). The final objective is a weighted sum: $0.8\cdot\mathcal{L}_{MSE}+0.2\cdot\mathcal{L}_{CE}$ (Dong et al. 2025). For the binary CTR prediction task on the MIND dataset, we employ a standard binary cross-entropy loss ($\mathcal{L}_{BCE}$) computed on the logit of the "yes" token (Zhang et al. 2025a).

Compared Methods. We benchmark PUMA against two fundamental baselines to establish performance boundaries (RQ1). **Full Retraining** serves as the performance upper bound, where all user prompts are retrained from scratch on the target model (M_t). Conversely, **Random Initialization** provides a performance lower bound by using randomly initialized vectors. To ensure a fair comparison in rating prediction tasks, the task-specific MLP head is still trained.

For evaluating efficiency (RQ2), we compare our group-based selection strategy against a comprehensive suite of alternative sampling methods:

- **Simple Baselines:** We start with Random Sampling, which selects users uniformly at random. We also test simple single-heuristic methods like Variance Bucketing and Loss Bucketing, which stratify the user pool by either historical output variance or task loss on the source model before sampling.
- **Clustering-based Strategies:** These methods leverage the geometry of the prompt embeddings. K-means Stratified clusters users and samples proportionally from each cluster. A variant, K-means with PCA, first applies a dimensionality reduction step to the prompt embeddings prior to clustering.

Method	Trainable Params (M)	Amazon		MIND		Yelp	
		RMSE↓	MAE↓	AUC↑	uAUC↑	RMSE↓	MAE↓
Baselines							
Full Retraining	N/A	0.9414	0.6296	0.5778	0.5289	1.1994	0.9269
Source Model Performance	N/A	0.9438	0.6306	0.5742	0.5312	1.2005	0.9369
Random Initialization	N/A	1.2352	1.1168	0.4917	0.4883	1.6671	1.4981
PUMA	88.1	**0.9135**	**0.5701**	**0.6546**	**0.6552**	**1.1073**	**0.8493**

Table 2: Performance comparison of different methods for prompt migration. The best results for each metric are highlighted in bold. The arrows ↓ indicate that lower is better, while ↑ indicates that higher is better. PUMA was trained using data from the entire user population in this experiment.

Method	h/Epoch	Epochs	Total (h)	Speedup
Full Retraining	3.00	8	24.0	1x
PUMA (2k users)	0.16	3	**0.48**	**50x**

Table 3: Efficiency comparison on the Amazon dataset.

- **Hybrid and Feature-based Clustering Strategies:** We test hybrid methods like K-means + FPS, which applies Farthest Point Sampling within clusters for diversity, and K-Means + Loss Stratification, which subgroups clusters by source model loss. We also evaluate K-means (on FFN Activ.), a strategy that clusters users based on richer feature vectors instead of on soft prompts; these are extracted by feeding each soft prompt into the source model and concatenating the activations from its final three FFN layers (Su et al. 2022).

Performance Results

We now present the empirical evaluation of PUMA, systematically addressing its effectiveness, efficiency, generalization, and robustness.

Effectiveness (RQ1) The results presented in Table 2 clearly establish PUMA's effectiveness. Across all three datasets, our framework not only successfully migrates user personalization but consistently outperforms the strong baseline of a full, from-scratch retraining. On the rating prediction tasks, PUMA achieves a lower RMSE on both Amazon (0.9135 vs. 0.9414) and Yelp (1.1073 vs. 1.1994). This superior performance is also evident on the MIND dataset, where PUMA significantly improves the uAUC from 0.5289 (full retraining) to 0.6552, demonstrating its efficacy across diverse task formulations.

Notably, PUMA's consistent outperformance suggests a fundamental advantage over retraining from scratch. We hypothesize that this stems from reframing the problem: instead of learning thousands of individual user representations in isolation, PUMA learns a single, supervised mapping function. This shared adapter appears to discover a more generalized and robust transformation into the target model's semantic space. The power of this generalized mapping is further explored in our analysis of advanced migration scenarios (RQ4).

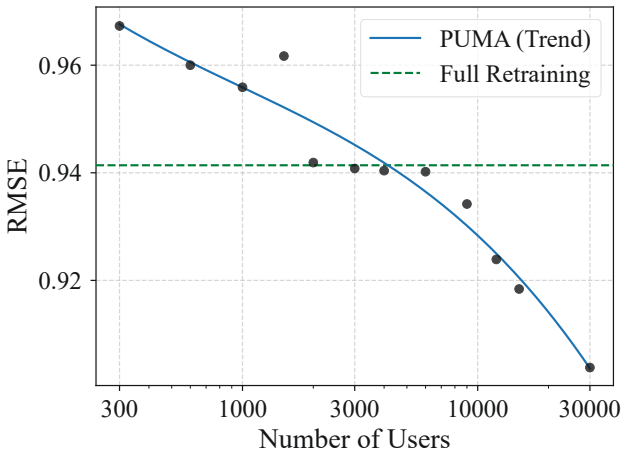


Figure 3: Performance of random user sampling on Amazon.

Efficiency (RQ2) Our evaluation of user selection strategies reveals the remarkable efficiency of the PUMA framework. As shown in Table 4, our "K-Means + Variance Stratification (PUMA)" approach is highly cost-effective. By operating under a strict computational budget (2,000 users for Amazon/Yelp, 1,500 for MIND), it not only surpasses random sampling at the same scale but consistently matches or exceeds the performance of baselines that use 3-4 times more data. For example, on the Amazon dataset, our method with 2,000 users (RMSE 0.9315) outperformed the "Random (6k)" baseline (RMSE 0.9320), despite using only one-third of the users. This efficiency is even more pronounced when compared to full retraining, where PUMA achieves a remarkable 50x speedup, as detailed in Table 3.

Figure 3 starkly illustrates the inefficiency of naive random sampling. The performance trend reveals that approximately 5,000 randomly selected users are required merely to match the RMSE of a full, from-scratch retraining. This substantial data requirement highlights the prohibitive computational cost of an unguided sampling approach and firmly establishes the need for our more sophisticated, group-based selection strategy to achieve high-fidelity migration efficiently.

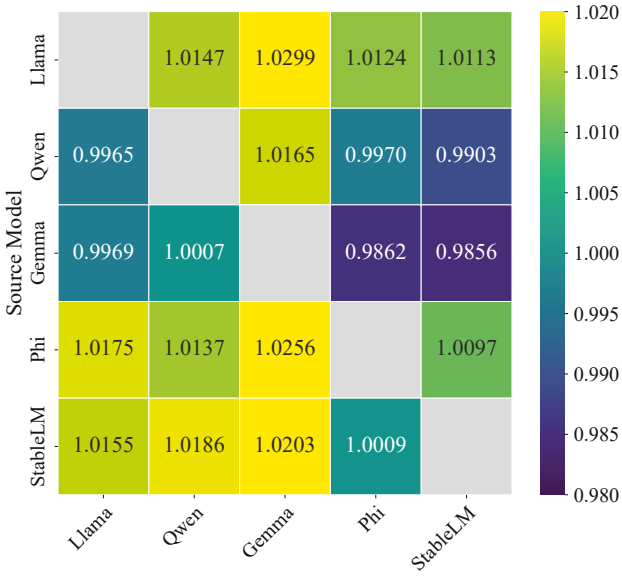
Method	Amazon		MIND		Yelp	
	RMSE ↓	MAE ↓	AUC ↑	uAUC ↑	RMSE ↓	MAE ↓
Baselines						
Random	0.9419	0.6155	0.5916	0.5861	1.1146	0.8627
Random (6k)	0.9320	0.6098	0.6585	0.6636	1.1128	0.8578
Single-Heuristic Selection						
Variance Bucketing	0.9508	0.6350	0.5934	0.5888	1.1171	0.8666
Loss Bucketing	0.9454	0.6209	0.5623	0.5845	1.1127	**0.8595**
K-Means Stratified	0.9546	**0.5885**	0.5830	0.5927	1.1152	0.8618
K-Means with PCA	0.9351	0.6005	0.5608	0.5652	1.1138	0.8659
Hybrid Selection Strategies (on Prompt Embeddings)						
K-Means + FPS	0.9355	0.6113	0.5990	0.5966	1.1147	0.8735
K-Means + Loss Stratification	0.9371	0.6065	0.5779	0.5884	1.1122	0.8634
K-Means + Variance Stratification (PUMA)	**0.9315**	0.5986	**0.6346**	**0.6344**	**1.1111**	0.8616
Ablation: Strategies using FFN Activations						
K-Means (on FFN Activ.)	0.9373	0.6096	0.5624	0.5697	1.1112	0.8690
K-Means (on FFN Activ.) + Loss Stratification	0.9467	0.6339	0.6152	0.6298	1.1115	0.8633
K-Means (on FFN Activ.) + Variance Stratification	0.9365	0.6043	0.5877	0.5936	1.1184	0.8616

Table 4: Performance comparison of user selection strategies. The best result is in **bold** and the second-best is underlined. Baselines are excluded from highlighting. ↓ indicates lower is better; ↑ indicates higher is better. *Note*: Except for Random (6K), which was trained on 6,000 users, all strategies were trained under a fixed budget of 2,000 users for Amazon/Yelp and 1,500 users for MIND.

Generalization across Architectures (RQ3) To evaluate PUMA's generalization, we migrated prompts across diverse model architectures and families. For these and the subsequent experiments in RQ4, we kept the training set fixed at 6,000 users. To uniformly assess the impact of model migration across different model families, we define the relative improvement, or gain, as the ratio of the RMSE of the fully retrained model to the RMSE of the migrated model.

The heatmap in Figure 4 confirms that PUMA generalizes remarkably well. Migrations between distinct model families consistently yield substantial performance gains, often matching or even exceeding those of a full retraining. As expected, migration efficacy is bounded by the quality of the source prompts. For instance, migrating from a weaker source model (e.g., Gemma-3) to a stronger target yields significant improvement, though it may not reach the performance of retraining from scratch on that target. Conversely, migrating from a high-performing source (e.g., Phi-3) can surpass it.

This outcome is logical, as the migration process transfers existing knowledge but does not create it. This observation motivates our final question: can the PUMA framework fuse knowledge from multiple sources to achieve even richer personalization? We explore this possibility in RQ4.



	Llama	Qwen	Gemma	Phi	StableLM
Pretrained RMSE	0.9414	0.9485	0.9606	0.9369	0.9365

Figure 4: Performance gain heatmap for different architectures. (Gain = full retrained RMSE / migrated RMSE)

Advanced Migration Scenarios (RQ4) In this section, we explore two advanced migration scenarios: chain migration and aggregated migration.

(1) Robustness in Chain Migration. To test our framework's robustness against cumulative error from successive model changes, we conducted a chain migration across five models: Llama3.2 → Qwen2.5 → Gemma-3 → StableLM-2 → Phi-3. At each step, the newly migrated prompts become the source for the next migration, without any retraining on the original user data.

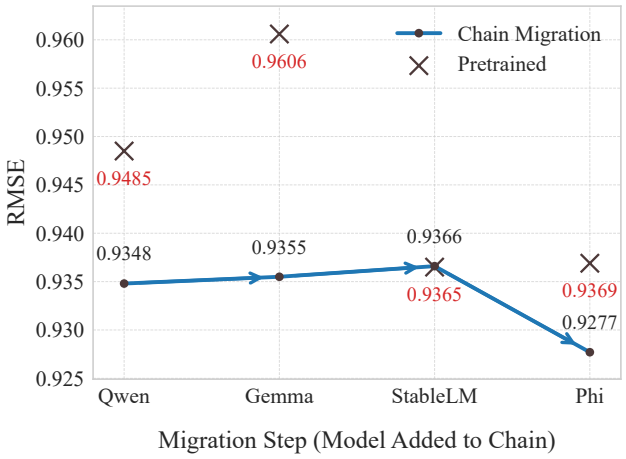


Figure 5: Stability in chain migration (started from Llama).

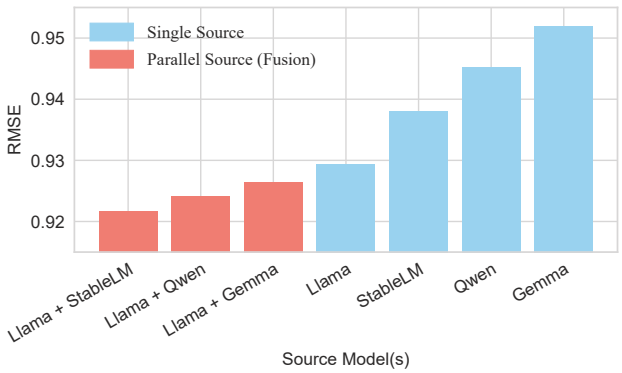


Figure 6: Performance of aggregated migration from one or two source models to Phi-3.

The results, shown in Figure 5, demonstrate remarkable stability. The performance (RMSE) remains consistently strong throughout the entire chain, starting at 0.9348 and ending at 0.9277. This result not only outperforms retraining from scratch at each individual stage but also validates PUMA's robust ability to preserve user personalization across multiple successive model updates.

We do observe a minor performance dip when migrating from the Gemma model. This is expected, as some information loss is likely when transitioning to a comparatively weaker model in the chain. Nevertheless, the subsequent migration step still achieves a final performance level superior to that of a direct migration from a pretrained Gemma to StableLM (as shown in Figure 4), underscoring the framework's robustness.

(2) Knowledge Fusion from Multiple Sources. We then investigated PUMA's capacity for knowledge fusion through an aggregated migration scenario. In this experiment, we combined prompts from two distinct source models and migrated them to a single target model, which we set as Phi-3.

As illustrated in Figure 6, the aggregated migration strategy consistently outperforms migrations from any single source model. For instance, by fusing prompts from "Llama + StableLM", the resulting model achieves an RMSE of 0.9217. This marks a substantial improvement over migrating solely from Llama (0.9293) or StableLM (0.9380). The combination of these two models yielded the best overall performance, likely because they were also the strongest individual performers in prior migration tests.

This consistent improvement highlights a principle of knowledge synergy: different foundation models capture complementary aspects of user preferences. By providing the adapter with a composite representation drawn from these diverse sources, the migration process becomes more effective. This finding reframes prompt migration from a maintenance task into a strategic opportunity to enhance personalization. It suggests a cumulative approach, where personalized assets are continuously enriched by integrating knowledge from multiple model ecosystems, evolving into dynamic, improvable profiles rather than static parameters tied to a single platform.

Conclusion & Discussion

In this paper, we address the critical challenge of migrating personalized soft prompts across foundation models by proposing PUMA, a lightweight adapter framework paired with an efficient user selection strategy. Our experiments on three large-scale datasets show that PUMA matches or surpasses the performance of full retraining while reducing computational costs by up to 98%. The framework demonstrates strong generalization across diverse model architectures, robustness in complex chained migrations, and a novel capability to enhance personalization by fusing knowledge from multiple source models. By decoupling personalized assets from the underlying models, PUMA provides a practical and sustainable solution for the long-term evolution of personalized AI systems.

Our work opens several promising avenues for future research. One direction is to enhance the user selection process. While our group-based strategy is highly efficient, a learning-based approach, such as using Reinforcement Learning (RL) to train a selection policy, could further push the performance ceiling by discovering more potent user combinations than static heuristics. Furthermore, our current framework focuses exclusively on migrating user-specific assets. We plan to extend PUMA to simultaneously migrate both user and item embeddings. This would enable a more comprehensive knowledge transfer, which is crucial in personalized systems where item representations are also learned and valuable. Finally, we aim to address more complex and realistic migration scenarios, such as handling cold-start personalization. In cases where the target dataset contains new users not present in the source system, the learned migration adapter could be leveraged to quickly initialize their prompts. This would bypass the need for extensive training from scratch and offer an efficient solution for onboarding new users to the target model.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62402470), the Fundamental Research Funds for the Central Universities of China (WK2100000053, PA2024GDSK0107), Anhui Provincial Natural Science Foundation (2408085QF189), the Postdoctoral Fellowship Program of CPSF (GZC20241643), and Anhui Postdoctoral Scientific Research Program Foundation (No.2025B1063). This research is supported by the advanced computing resources provided by the Supercomputing Center of the USTC.

References

Abdin, M.; Aneja, J.; Awadalla, H.; Awadallah, A.; Awan, A. A.; et al. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219.

Albalak, A.; Elazar, Y.; Xie, S. M.; Longpre, S.; Lambert, N.; Wang, X.; Muennighoff, N.; Hou, B.; Pan, L.; Jeong, H.; Raffel, C.; Chang, S.; Hashimoto, T.; and Wang, W. Y. 2024. A Survey on Data Selection for Language Models. *Transactions on Machine Learning Research*. Survey Certification.

Asai, A.; Salehi, M.; Peters, M.; and Hajishirzi, H. 2022. ATTEMPT: Parameter-Efficient Multi-task Tuning via Attentional Mixtures of Soft Prompts. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 6655–6672. Association for Computational Linguistics.

Bellagente, M.; Tow, J.; Mahan, D.; Phung, D.; Zhuravinskyi, M.; et al. 2024. Stable LM 2 1.6B Technical Report. arXiv:2402.17834.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; et al. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc.

Cai, S.; Gao, C.; Zhang, Y.; Shi, W.; Zhang, J.; Bao, K.; Wang, Q.; and Feng, F. 2025. K-order Ranking Preference Optimization for Large Language Models. In *Findings of the Association for Computational Linguistics: ACL 2025*, 4844–4859.

Chen, J.; Liu, Z.; Huang, X.; Wu, C.; Liu, Q.; Jiang, G.; Pu, Y.; Lei, Y.; Chen, X.; Wang, X.; Zheng, K.; Lian, D.; and Chen, E. 2024. When large language models meet personalization: perspectives of challenges and opportunities. *World Wide Web*, 27(4).

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; et al. 2023. PaLM: scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24(1).

Coleman, C.; Yeh, C.; Mussmann, S.; Mirzasoleiman, B.; Bailis, P.; Liang, P.; Leskovec, J.; and Zaharia, M. 2020. Selection via Proxy: Efficient Data Selection for Deep Learning. In *International Conference on Learning Representations*.

Dong, X. L.; Moon, S.; Xu, Y. E.; Malik, K.; and Yu, Z. 2023. Towards Next-Generation Intelligent Assistants Leveraging LLM Techniques. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, 5792–5793. Association for Computing Machinery.

Dong, Z.; Hu, L.; Chen, J.; Wang, Z.; and Wu, F. 2025. Comprehend Then Predict: Prompting Large Language Models for Recommendation with Semantic and Collaborative Data. *ACM Trans. Inf. Syst.* Just Accepted.

Fan, C.; Gao, C.; Shi, W.; Gong, Y.; Zhao, Z.; and Feng, F. 2025. Fine-grained List-wise Alignment for Generative Medication Recommendation. NeurIPS '25.

Gao, C.; Chen, R.; Yuan, S.; Huang, K.; Yu, Y.; and He, X. 2025a. SPRec: Self-Play to Debias LLM-based Recommendation. WWW '25, 5075–5084.

Gao, C.; Gao, M.; Fan, C.; Yuan, S.; Shi, W.; and He, X. 2025b. Process-Supervised LLM Recommenders via Flow-guided Tuning. SIGIR '25, 1934–1943. ISBN 9798400715921.

Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; and Wang, H. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Hou, Y.; Li, J.; He, Z.; Yan, A.; Chen, X.; and McAuley, J. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952*.

Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Huang, Q.; Liu, X.; Ko, T.; Wu, B.; Wang, W.; Zhang, Y.; and Tang, L. 2024. Selective Prompting Tuning for Personalized Conversations with LLMs. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 16212–16226. Association for Computational Linguistics.

Jang, J.; Kim, S.; Lin, B. Y.; Wang, Y.; Hessel, J.; Zettlemoyer, L.; Hajishirzi, H.; Choi, Y.; and Ammanabrolu, P. 2023. Personalized Soups: Personalized Large Language Model Alignment via Post-hoc Parameter Merging. arXiv:2310.11564.

Kang, W.-C.; Ni, J.; Mehta, N.; Sathiamoorthy, M.; Hong, L.; Chi, E.; and Cheng, D. Z. 2023. Do LLMs Understand User Preferences? Evaluating LLMs On User Rating Prediction. arXiv:2305.06474.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 3045–3059. Association for Computational Linguistics.

Li, L.; Zhang, Y.; and Chen, L. 2023. Personalized Prompt Learning for Explainable Recommendation. *ACM Trans. Inf. Syst.*, 41(4).

Liu, J.; Liu, C.; Zhou, P.; Lv, R.; Zhou, K.; and Zhang, Y. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. arXiv:2304.10149.

Liu, J.; Qiu, Z.; Li, Z.; Dai, Q.; Zhu, J.; Hu, M.; Yang, M.; and King, I. 2025. A Survey of Personalized Large Language Models: Progress and Future Directions. arXiv:2502.11528.

Mirzasoleiman, B.; Bilmes, J.; and Leskovec, J. 2020. Coresets for Data-efficient Training of Machine Learning Models. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 6950–6960.

OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; et al. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Pooladzandi, O.; Davini, D.; and Mirzasoleiman, B. 2022. Adaptive Second Order Coresets for Data-efficient Machine Learning. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 17848–17869.

Salemi, A.; Mysore, S.; Bendersky, M.; and Zamani, H. 2024. LaMP: When Large Language Models Meet Personalization. arXiv:2304.11406.

Shi, W.; He, X.; Zhang, Y.; Gao, C.; Li, X.; Zhang, J.; Wang, Q.; and Feng, F. 2024. Large Language Models are Learnable Planners for Long-Term Recommendation. SIGIR '24, 1893–1903. ISBN 9798400704314.

Sorscher, B.; Geirhos, R.; Shekhar, S.; Ganguli, S.; and Morcos, A. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 19523–19536. Curran Associates, Inc.

Su, Y.; Wang, X.; Qin, Y.; Chan, C.-M.; Lin, Y.; Wang, H.; Wen, K.; Liu, Z.; Li, P.; Li, J.; Hou, L.; Sun, M.; and Zhou, J. 2022. On Transferability of Prompt Tuning for Natural Language Processing. In Carpuat, M.; de Marneffe, M.-C.; and Meza Ruiz, I. V., eds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3949–3969. Association for Computational Linguistics.

Tan, Z.; Zeng, Q.; Tian, Y.; Liu, Z.; Yin, B.; and Jiang, M. 2024. Democratizing Large Language Models via Personalized Parameter-Efficient Fine-tuning. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 6476–6491. Association for Computational Linguistics.

Team, G.; Kamath, A.; Ferret, J.; Pathak, S.; Vieillard, N.; Merhej, R.; et al. 2025. Gemma 3 Technical Report. arXiv:2503.19786.

Team, Q. 2024. Qwen2.5: A Party of Foundation Models.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010. Curran Associates Inc.

Vu, T.; Lester, B.; Constant, N.; Al-Rfou', R.; and Cer, D. 2022. SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5039–5059. Association for Computational Linguistics.

Wang, S.; Xu, T.; Li, H.; Zhang, C.; Liang, J.; Tang, J.; Yu, P. S.; and Wen, Q. 2024. Large Language Models for Education: A Survey and Outlook. arXiv:2403.18105.

Wu, F.; Qiao, Y.; Chen, J.-H.; Wu, C.; Qi, T.; Lian, J.; Liu, D.; Xie, X.; Gao, J.; Wu, W.; and Zhou, M. 2020. MIND: A Large-scale Dataset for News Recommendation. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3597–3606. Association for Computational Linguistics.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; et al. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671*.

Zhang, K.; Kang, Y.; Zhao, F.; and Liu, X. 2024. LLM-based Medical Assistant Personalization with Short- and Long-Term Memory Coordination. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2386–2398. Association for Computational Linguistics.

Zhang, Y.; Feng, F.; Zhang, J.; Bao, K.; Wang, Q.; and He, X. 2025a. CoLLM: Integrating Collaborative Embeddings Into Large Language Models for Recommendation. *IEEE Trans. on Knowl. and Data Eng.*, 37(5): 2329–2340.

Zhang, Z.; Rossi, R. A.; Kveton, B.; Shao, Y.; Yang, D.; Zamani, H.; Dernoncourt, F.; Barrow, J.; Yu, T.; Kim, S.; Zhang, R.; Gu, J.; Derr, T.; Chen, H.; Wu, J.; Chen, X.; Wang, Z.; Mitra, S.; Lipka, N.; Ahmed, N.; and Wang, Y. 2025b. Personalization of Large Language Models: A Survey. arXiv:2411.00027.

Zheng, H.; Liu, R.; Lai, F.; and Prakash, A. 2023. Coverage-centric Coreset Selection for High Pruning Rates. In *International Conference on Learning Representations*.